Applications of Statistics and Machine Learning to Genomic Clinical Group Classification

Jake Sauter

December 2018

Abstract

Over the past decade microarray technology has been emerging as a potentially unparalleled force for use clinical diagnosis from genetic sampling. Despite the large amount of possible power from quantifiable characteristics of different biological outcomes, many difficulties plague the entry of microarrays into the diagnostic process. Advances in non-parametric permutation based statistical analysis have combated many issues with conservative multiple comparison corrections, and have allowed for the selection of differentially expressed genes with the user selected false discovery rates. With the power to statistically determine genes of importance for discerning clinical groups, machine learning classification of new samples into the clinical groups is made much easier and produces more robust classifiers. The robustness of these classifiers has not previously been achievable when considering all possible genes as influencing the difference of the two groups. The subsequent chapters detail the biological mechanisms made of use by microarrays to determine gene expression level of individual patients, describe many statistical procedures for determining differentially expressed genes between clinical groups, and covers implementations of various machine learning techniques to classify new samples successfully using the statistically selected differentially expressed genes.

Acknowledgements

First and foremost I want to thank my advisor Mark Baker. Without him I may have never become as intimate with statistics as I have now become, and never seen the potential power behind prefacing complex classification structures with robust statistical feature selection methods. I appreciate all of his contributions of time, ideas and guiding me through all of the of a stumblings of an undergraduate finding his statistical way.

I would also like to thank SUNY Oswego for granting me the opportunity of undertaking this capstone, and providing wide potential domains allowed to be covered in such capstone projects. This freedom contributed to my success immensely as I was able to explore areas that I have never before, and investigate important project matters as they came up. Following important leads on seemingly powerful techniques lead to many project changing advances and inevitably to the overall success of the project.

Contents

Chapter 1

Biological Background

It is important to understand the nature of one's data in order to fully comprehend and accurately interpret the results of proper data analysis. Due to the nature of microarrays, we first must introduce the reader to fundamentals in molecular biology that are made of use by microarrays to measure precursors of different biological outcomes.

Microarrays and their associated study belong to the field of Genomics, being "the field that encompasses investigations into the structure and function of very large number of genes undertaken in a simultaneous fashion"[1]. To understand this field of study more we must understand what a gene is, and to understand what a gene is we must first start with the biological material of Deoxyribonucleic Acid (DNA). DNA is the genetic material that encodes for the structure, size, function, and quantity of different proteins that a cell needs to function, as well as other important biological information. The DNA material is composed of four bases, being Adenine (A), Cytosine (C) , Guanine (G), and Thymine (T). Out of the four bases listed, (A) and (T) are said to be complementary bases, and (C) and (G) are said to be complementary bases. By complementary bases, we mean that these bases when positioned close enough together will bind to each other, more formally these pairings occur due to the "Watson-Crick Rules".

We will build on this idea of complementary bases moving forward. First we note the standard form of DNA being the double-helixed, as can be seen in Fig. 1.

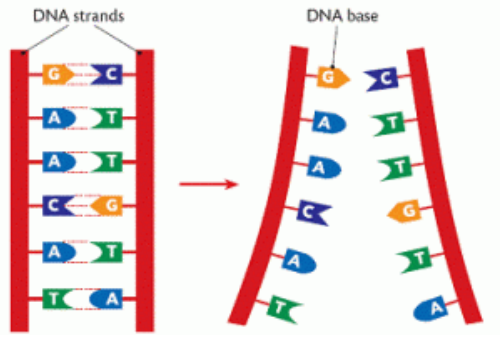

Figure 1: Double stranded DNA being decomposed into two complementary DNA strands

This form of DNA can also be said to be the double-stranded form of DNA because as seen in the diagram, DNA of this form can be separated into two chains or strands of the prior mentioned bases. We now stumble upon an important biological result used by microarrays, that of complementary single stranded DNA. We will call two single stranded DNA sequences complementary if each of their bases is complementary in every index of the two strands.

Now we may finally return to genes. Genes are simply particular sequences on these DNA strands. We can tell that an area of a DNA sequence is a gene because it surrounded on both ends by a start and stop codon. The codons that we are mentioning are the building blocks of genetically stored information. It was discovered that DNA encodes amino acids (the building blocks of protein) in triplets of DNA bases, and these triplets of DNA bases are grouped together and called codons. Specifics about codons are slightly beyond the scope needed for microarray applications, though interestingly a large amount of redundancy is built into this genetic code, being found mathematically that this coding method was highly optimal, taking similar steps taken for redundancy in computer network communications.

Now that we understand the originating form of genetic material, we can study the biological process of converting DNA into the needed biological end result of protein.

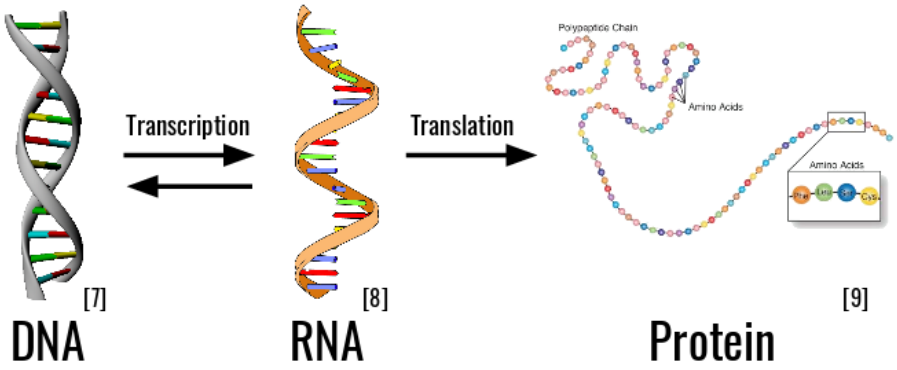

Figure 2: The Central Dogma of Molecular Biology flow chart, demonstrating genetic material originating as DNA, being transcribed into RNA, then being translated into the biological end result of protein composed of amino acid

This process of converting DNA into protein must go through a medium of Ribonucleic Acid (RNA), as DNA material must be kept safe from possible harm in the

nucleus of the cell. Thus RNA begins its life by copying a needed genetic sequence from DNA in the nucleus called the transcription process, later to permeate the nucleus membrane and to be translated into the chain of amino acids known as protein. In the future analysis, we will be interested in measuring how much RNA associated with each gene makes it out of the cell nucleus, which then affects the amount of influence a particular gene has on end protein.

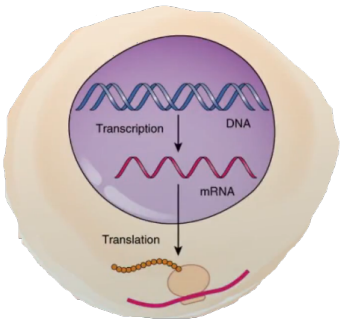

Figure 3: Visualization of the Central Dogma of Molecular Biology. This diagram shows that DNA is kept safe from disturbances in the cell nucleus, with RNA copying needed genetic sequences, permeating the nucleus membrane, then finally being translated into the needed protein.

We have now covered all necessary biological background to introduce the measurement device known as a microarray. As previously stated, microarrays aim to measure the expression level of genes to determine how much impact each gene is having on the end biological results.

Chapter 2

Microarrays

In this section we will discuss the core process implemented by microarrays to measure gene expression level, explore different microarray manufacturing techniques, highlight some of the challenges encountered in the use of microarrays for there analyses, and conclude on some points on the reliability and reproducibility of microarray

measurements.

2.1 Introduction

Now that we have a thorough enough understanding of the biological mechanisms we are probing, we can investigate how microarrays can be used to study gene expression levels of collected tissue samples. Microarrays are chips composed of many cells. Each of these cells can be used to measure the expression level of a single gene though a probe. This probe can be thought of as a well, filled with anchored complementary DNA to the gene the particular well is measuring for.

Before introducing a sample to a microarray for measurement of gene expression level, we first must preprocess this sample. The RNA in the sample must be reverse transcribed back into single stranded DNA, as to be able to bind to the complementary DNA in the probes. Once the RNA is extracted and reverse transcribed, a fluorescent marker is binded the single stranded DNA in the sample, and the sample can finally be introduced to the chip.

When this sample is deposited onto the microarray, the reverse transcribed RNA associated with each gene will hybridize (pair to) the complementary DNA in the probe associated with that gene. We can then scan the microarray for the fluorescence level of each probe, which will indicate the amount of RNA that made it out of the cell nucleus associated with that particular gene. We now reference back to the introduction section, when we defined this level to be the expression level of that gene, and stated that this level will indicate how much impact the gene will have on the end biological result of the protein.

Thus we can see once we have established our biological background, the microarray measurement process is not complicated in it own right with knowledge of the background processes needed. In this section we have described the general concept of a microarray, though there have been many different implementations from various competing companies in attempts to make more accurate and reliable measurements of these expression levels.

2.2 Manufacturing Techniques

Two major manufacturing techniques exist to produce microarrays, and each contains its own strengths and weaknesses. In this section we will outline the deposition

technique and the in situ synthesis technique, though there have also been some exceptional fabrication techniques implemented by a few companies.

The first fabrication technique that we will describe is that of deposition techniques. For this fabrication technique, the complementary DNA for each gene is prepared away from the chip, then deposited in each probe using thin pins dipped in the genetic material. These prepared samples can be short sequences long enough to discern a gene, to as long as entire genes. In order to have enough genetic material to measure using the microarray, amplification techniques are introduced to replicate DNA.

The choice of cloning technique is also possible in the deposition method, as ESTs (expressed sequence tags) and PCSs (polymerase chain reactions) are both feasible options. ESTs are cheap and single pass sequences of the entire clone libraries, and result in partial sequences of the desired genes, though are long enough to still uniquely identify gene specific fragments. PCRs produce better clones than ESTs but are more expensive. These better clones are achieved through post amplification purification.

The second major microarray manufacturing technique is in situ synthesis. A hint about how in situ synthesis works can be found by defining the latin "in situ", meaning "on site" or "locally". This matches well with the manufacturing technique, as opposed to deposition techniques where the cDNA was prepared away from the chip, the sequence for every probe is constructed locally on the chip. Three main manufacturing techniques can be used for in situ synthesis, being the photolithographic, ink jet, and electrosynthesis techniques. The photolithograph technique uses a photolithographic mask for each bases, with the mask allowing the current base being deposited on the chip to only be deposited on the probes that currently require that base in the sequence being built. As for the ink jet method, technology very similar to ink-jet printers is used, though as opposed to having black, cyan, magenta and yellow, the four cartridges are the four DNA bases being (A), (C), (G), (T). Finally for the electrosynthesis approach, electrons in each probe are turned on when the needed nucleotide base solution is currently on the chip as to attract the needed base close enough to bind to the previous base deposited on the probe. As we can see many of these techniques are ingenious and all share the commonality of constructing the needed sequence one base at a time while on the microarray. We will soon mention the many strengths of these techniques, though first mention that although these

technique allow for precise sequences to be constructed for each probe, the probability of error in the sequence increases with its length so sequences tend to be shorter.

Major strengths are seen in these in situ manufacturing techniques. One major strength of this technique is allowing for the detection of multiple splice variants of the genes, where a splice variant is an instance of a singular base difference occurring in the genetic sequence being studied. A very clear strength when compared to deposition techniques is that we will know nearly exactly the sequence each probe is measuring the expression level of. This level of confidence in the sequence cannot be ensured in deposition techniques as in many cases the function of the sequence to a spot is unknown due to inability to backtrack from ESTs to the original gene.

We will now note the key differences between the deposition and in situ synthesis microarray manufacturing techniques. Deposition can allow for longer sequences, though sometimes the sequences of each probe are unknown. More variability is introduced into the system in deposition, though with the correct experimental design the results are easier to analyze. As for in situ synthesis, shorter sequences are used, though the sequence at each probe is known in detail. More reliable data can be produced through in situ synthesis, though this data could be more difficult to analyze. Now that we are familiar with the strengths of many different types of microarrays, in a scientific work such as this we must also note the challenges and pitfalls of microarrays.

2.3 Challenges

Microarrays so far have seemed to develop into a very powerful tool to measure the difference in distinct biological outcomes though gene expression levels. With only this previous information it would seem preposterous that microarrays have yet to be introduced as a diagnostic tool in the standard clinical setting. We will now introduce the challenges that microarrays face, and must overcome to be introduced as such a tool.

The measurement process of the gene expression levels, though theoretically not a very complex process, has many individual steps that can go wrong. At each step in the preparation of the sample, manufacturing of the microarray, and during the actual measurement procedure, noise can be introduced that will obscure the final results. We can assume with such a following these procedures will become more

refined over time, though this is not the only issue. Indeed we can agree that the techniques themselves are complex and can be a challenging process to perfect, though it is not only the technology that is at fault in these measurements and analysis. Some challenges plague their introduction to diagnosis due to user error. It has been shown that normalization of the microarray data is not always performed in the same fashion from lab to lab. Experimental design has also proven to be a troubling topic as reviews of previous experiments have shown pitfalls of the experimental design that have nearly rendered the results useless.

These previous challenges are not special to microarray analysis though. Scientific procedures will always be complex and need refining, and laboratory staff could always benefit from more training. There are some challenges that are by nature inherent solely to microarray analysis. Due to the magnitude of the studies done with microarrays, thousands of sequences are measured on a single chip in hopes of finding the few differences between one group of individuals and another having different clinical conditions. The task of finding a few genes that indicate this difference can be likened to finding a needle in a haystack, and issues arise when attempting to perform classical statistical analysis on these measurements. These analysis issues come from the small number of samples in respect to the number of variables ; usually thousands of genes measured from tens of patients leads to many statistical challenges.

The last issue particular to microarrays that we will review might have been raised by the reader. If we are interested in the end biological impact of specific genes, and these genes are affecting the final protein structure, and to become this protein structure the RNA must be translated into the protein, aren't we missing out on the effect of an entire process on the expression of the gene? This is a valid question to raise though this translation process is out of the scope of possibility to assess with microarrays. It can then be said that if supporters of microarray analysis believe that its results are valid, this analysis makes the assumption that the translation process introduces a small enough effect that the expression levels we are measuring are a good approximate estimate. We will later see for ourselves how successful the analysis can be even without monitoring this step in the process.

2.4 Reliability and Reproducibility

A large issue with previous microarray studies is that of the reproducibility of these studies. Each of the previous mentioned manufacturing techniques has their own weaknesses, as well as slightly different results concerning accuracy, sensitivity, specificity and robustness. These differences can lead to issues in reproducibility of results from one platform to another and questions the integrity of the platforms. As we have mentioned, these issues become much more important as microarrays are proposed as critical diagnostic tools, that must produce reliable and reproducible results time and time again.

A fabrication issue mentioned in the prior section that pertains to the reliability and reproducibility of in situ synthesis platforms is that the nucleotide probes synthesized on the chip are not 100 percent accurate due to base skipping. In context of deposition techniques, it was not mentioned, but a large problem can be incorrect cDNA intermixed on the probes, leading to skewed expression levels. These issues must be assessed to have a lesser impact if results are successful, as microarray analysis is based on the assumption that probes produce specific signals under the single (currently rather lenient) hybridization process. This hybridization process has been shown to not be bulletproof as widespread cross-hybridization of transcripts on microarrays has been reported.

Accuracy and sensitivity of these measurements are also very important metrics that must be robust to lead to the success of microarray diagnosis. To determine the sensitivity of microarrays, studies have shown that the concentration range of the sample must be strictly equal to or larger than the manufacturer's suggestion, and concentrations any lower than this threshold can cause non-meaningful measurements. Unfortunately a large number of experiments (slightly under half) have been though to not use large enough transcript concentration samples. The validation of this accuracy is not a simple process. In order to validate the metric, one must know the true concentrations for a large number of transcripts in the sample. This is no feasible as spike-in studies must be done to assure these concentrations, and limits to the number of genes in these studies is under 100, making validation of accuracy of about all 10,000 genes currently an immensely expensive and time consuming task that is not performed.

In good news for in situ synthesis techniques, the relationship between probe

length and detection limits has been evaluated by many papers. A drop-off in sensitivity to specificity was found that enables the optimal probe length choice to be determined. This is such good news to in situ synthesis, as the probability of an error in the probe sequences can be limited to only that which is necessary.

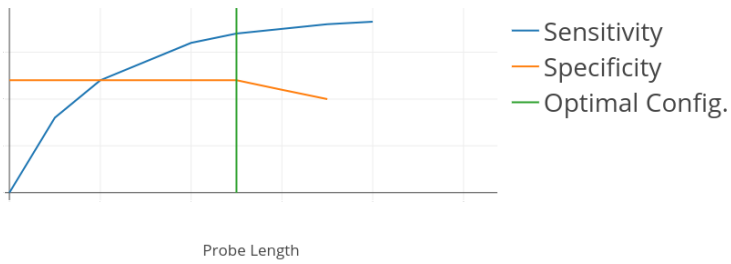

Figure 4: Sensitivity/Specificity behavior of probe length to detection limit

In regards to reproducibility of microarray results, we must investigate cross-platform consistency. Achieving the desired level of cross-platform consistency was be a great achievement for microarrays as it would allow for less replication experiments to need to be done, and allow researchers to more quickly build a universal gene expression database. Due to the importance of this task, cross platform consistency has become one of the top required characteristics of most platforms for reliability. This is not the best method for the validation of microarray technology, as all platforms may be consistently incorrect, though its introduction to importance in manufacturing has led to more inter-manufacturer consistency. Some of these inconsistencies in microarray measurements come from the poor understanding of the relationship between probe sequences, target concentrations and probe intensities. Some probes may require more energy to bind then others, meaning less target sequences would bind than other targets in the solution even if the same quantity is present. Lastly one large issue again does not come from microarrays, but from the biological factors being measured ; more than 50 percent of human genes are alternatively spliced, meaning probes must be intelligently designed to bind to all of the variants of a desired target, a more than simple task that is most likely not done the same way between manufacturers.

Hopefully this section was informative on the downfalls of microarrays, though has not discouraged the reader from believing in the power of microarrays in the

diagnostic process. The FDA follows the belief of the power of this technology, and in September of 2006 spawned the Microarray Quality Control Project (MAQC) with initial goals to provide quality control tools to the microarray community, develop guidelines for microarray data analysis, establish sound quality control metrics and thresholds, and finally to evaluate the advantages and disadvantages of various data analysis methods. The goals of this project update periodically and with the power the community and the FDA microarray technology is continuing to advance towards use as a diagnostic tool.

Chapter 3

Data

The data that was used for the analysis for this exploration of methods originated from Golub (1999)[2]. This data contained 72 total samples of microarray measurements for 7000 genes from two clinical groups. 47 samples were from the Acute L Leukemia (ALL) group and 25 samples were from the Acute M Leukemia (AML) group. The goal of our analysis is to statistically select the differentially expressed genes between these two groups, and to used these features to successfully classify a portion of the data that we have held out from the statistical selection and training procedures.

Chapter 4

Multiple Comparisons

With a solid background in the biological processes that microarrays exploit, how microarrays measure gene expression levels and the strengths and weaknesses of microarrays, we are now prepared to begin exploring possible analysis techniques for the data produced from this technology. In this section we will describe that statistical challenges involved in analyzing microarray data, and introduce successful

techniques that have been implemented prior, as well as a technique developed just for microarrays.

4.1 Statistical Background

In the situation we are in, given the microarray measurements from the two studied clinical groups, our first goal is to select the differentially expressed genes between these two groups. In order to do this, we can use a standard statistical hypothesis test for each each, considering the values of the particular gene over both groups. A hypothesis test aims to produce a p-value, being the probability of rejection the null hypothesis (in this case that there is no difference between the expression level of the gene over the two groups) given random chance of sampling error. We know to reject this null hypothesis when our test statistic (being some function of the gene expression values in each of the groups) is above some critical value of that test statistic, given the user defined acceptable probability of rejecting the null hypothesis when it is true.

The specific problem of microarray analysis covered in this section is in choosing our acceptable probability of rejecting the null hypothesis when it is true. This is a problem in microarray analysis as when we are performing the large amount of simultaneous hypothesis tests that we are, we must look at this probability globally between all hypothesis tests, and cannot just concern ourselves with this probability only for one test.

4.2 Common Corrections

In order to correct the significance level for each single hypothesis test to control for the global significance level of all of the simultaneous tests combined (also called the Family Wise Error Rate), some common corrections will be shown. The first of these common corrections is the Bonferroni correction. This is the simplest correction and simply divides the global significance level by the number of hypothesis tests. Denoting g is the global (or experimental) significance level and s is the significance level of a single hypothesis test this can be phrased as

$\alpha_s = \frac{\alpha_g}{R}$ for R genes

Another correction in the same class as the Bonferroni correction has been derived

very intuitively and is called the Sidak correction for multiple comparisons. The derivation for this correction is as follows

$P(\text{correct}) = (1-p)$
$P(\text{globally correct}) = (1-p) * (1-p) * ... * (1-p) = (1-p)^R$ for R genes
$P(\text{wrong somewhere}) = 1 - P(\text{globally correct}) = 1 - (1-p)^R$
$\alpha_g = 1 - (1-\alpha_s)$

Now in solving this previous equation for the significance level for each independent test we arrive at the formula

$\alpha_s = 1 - (1-\alpha_g)^{\frac{1}{R}}$

Unfortunately both of these corrections produce very small significance levels for a reasonable number of genes R common in microarray studies. These small significance levels cause for the independent tests to be too conservative, and thus fails to select genes to be differentially expressed that have a high likelihood of being so.

4.3 Step-Wise Corrections

The Holm's step-wise group of methods was developed for the case of much too conservative tests, and works in the following way. First, the global significance level α_g is chosen. Next the genes are ordered increasingly by their individual p-values found from their test statistics. Then the p-values of each gene are compared with a threshold that depends on the position of the gene in the list of ordered values. This threshold is described by $\frac{\alpha_g}{R}$ for the first gene, $\frac{\alpha_g}{R-1}$ for the second gene, and so on. We then define the integer k to be the largest gene index i such that $p_i < \frac{\alpha_g}{R-i+1}$, and the null hypothesis is rejected (thus the gene is selected as differentially expressed) for $i = 1, 2, \cdots, k$. Another method resides in this family of Holm's step-wise method and is called the False Discovery Rate (FDR) method. The initial steps of this method are similar as to described before, ordering the genes in order of decreasing p-value, though the p-values are then compared to the threshold of $\frac{1}{R}\alpha_g$ for the first gene, $\frac{2}{R}\alpha_g$ for the second gene, and so on. In this method we let the integer k be the largest gene index i such that $p_i < \frac{i}{R}\alpha_g$.

4.4 Permutation Corrections

The Holm's step-wise methods mentioned do allow for less conservative test's than the Bonferroni and Sidak correction, though miss a key point that must be encompassed in the study of genetics. These prior mentioned tests make the assumption that the expression levels of all genes are independent, and this is simply not a valid assumption as the expression level of some genes have a direct impact on the expression levels of others. This information can be taken into account when studying permutations of the data, in which the group labels are randomly assigned to each gene.

This can be seen more clearly in a concrete example. In the study we are concerned with we have measurements for approximately 7000 genes from each patient. These patients are separated by their cancer type into two groups, which can be thought of as each patient having either the group label "ALL" or the group label "AML". By permuting these group labels randomly between patients and performing the group level statistical analysis we will later discuss, we will be able to calculate the distribution of the test statistic while taking the dependency of the data into account. In the the Westfall and Young (W-Y) step-down correction, the p-value for gene i is then calculated to the the proportion of times the value of the calculated test statistic for the real labels was less than or equal to the value of the test statistic calculated calculated for a random permutation.

$$p - \text{value for gene } i = \frac{\text{number of permutations for which } t_i^{(b)} \geq t_i}{\text{total number of permutations}}$$

where $t_i^{(b)}$ are the calculated t-values from gene i and permutation b

4.5 SAM

Later we will discuss a statistical technique that was developed specifically for selecting differentially expressed genes called Significance Analysis of Microarrays [1]. For now we will only go over how this method corrects for simultaneous hypothesis tests. Similarly to the previous method, this method uses permutations to take the dependency structure of the data into account, though does so slightly differently.

SAM does attempt to estimate the distribution of the test statistic over permutations of the data, but then uses this to measure for the amount of genes selected as differentially expressed in each permutation. With this information SAM attempts to

control what is called the false discovery rate (FDR), being the proportion of genes out of the selected differentially expressed genes from the original data that were most likely not differentially expressed. To further explain how this works, we first must briefly talk about how SAM decides a gene is differentially expressed. SAM computes a test-statistic very similar to the one used in the t-test, a common test that we will discuss in a later section. SAM then compares the test statistic with the expected value of the test statistic, being the mean test statistic from all random permutations of the data for that particular gene. If the actual test statistic is greater than a specified distance away from the expected test statistic (what we will later call delta), then SAM selects that gene to be differentially expressed. This process is repeated for all permutations of the data, and the amount of differentially expressed genes for each permutation are saved and ordered. The median of these values is then divided by the amount of genes called differentially expressed from the original non permuted data to calculated the FDR associated with a given delta. SAM provides an output table of a range of delta values associated with their calculated FDR in order to allow for the experimenter to control for the desired FDR by controlling the delta parameter.

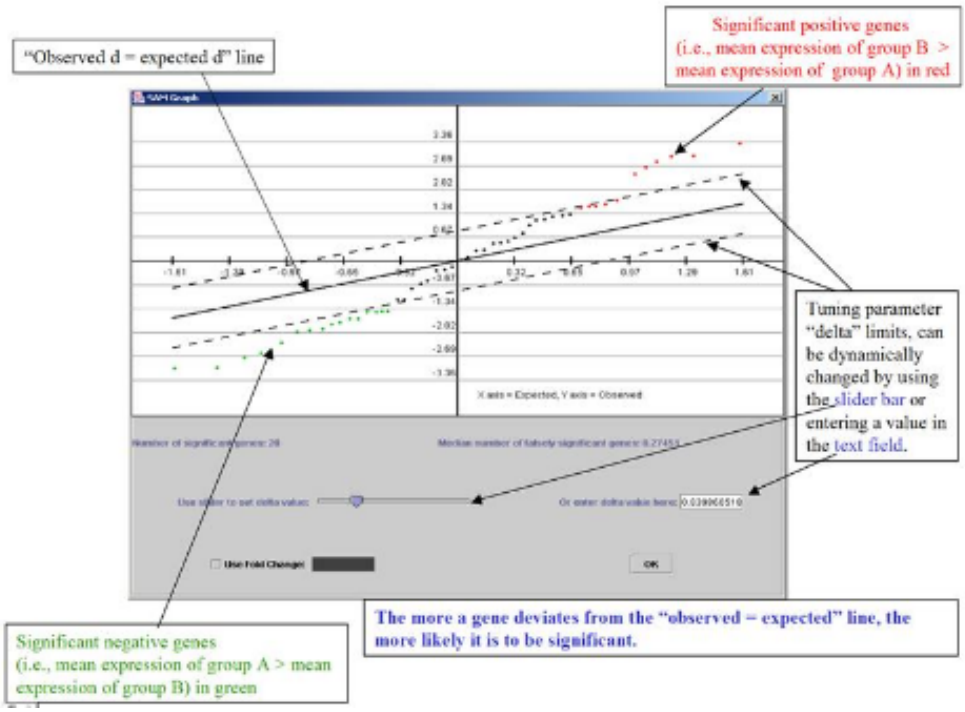

Figure 5: Plot of one iteration of the permutation procedure used by SAM to determine the FDR.

Chapter 5

Principal Components Analysis

Reiterating the problem we are concerned with, we are given gene expression values for approximately 7000 genes from patients in two clinical group. Our goal is to train a machine learning classifier to correctly predict the true clinical group of a never before seen sample.

5.1 Motivation

This is a difficult problem for standard statistical analysis and machine learning techniques as the number of variables (genes) is much larger than the number of samples (patients). In order to perform analysis on this data, we must somehow develop a representation of the samples in a much smaller dimensional space that still captures most of the variance of the data and the relation of data points to others. This family of techniques is called dimensionality reduction. We will specifically study the form of dimensionality reduction called Principal Components Analysis, in which the input is all of the samples and variables and the output is a smaller dimensional representation of each samples. The new variables that this method develops are linear combinations of the variables provided and are called principal components. Principle components are named in order of how much variance of the data they capture ; the first principal component captures the most variance, the second principal component captures the second most variance of the data, and so on.

5.2 Methods

We will now discuss how these principal components are formulated. This method begins by computing the co-variance matrix of the data. This co-variance matrix has a row and a column corresponding to each gene, so an entry in the co-variance matrix is the co-variance between the gene represented by the column and the gene represented by the row. This can be matrix can be visualized as

$$A = \begin{bmatrix} cov(x_1, x_1) & cov(x_1, x_2) & \dots & cov(x_1, x_n) \\ cov(x_2, x_1) & cov(x_2, x_2) & \dots & cov(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ cov(x_n, x_1) & x_{d2} & \dots & cov(x_n, x_n) \end{bmatrix}$$

Now we can formally define the principal components to be the eigenvectors of this matrix, and we know the ordering of the principal components by sorting these eigenvectors by their associated eigenvalue.

The common representation for the equation of an eigenvector is

$$Az_1 = \lambda_1 z_1.$$

We can see in this equation the associated eigenvalue of the eigenvector 1 is z_1, so if 1 is the first principal component, then z_1 must be the largest eigenvalue associated with all eigenvectors.

5.3 Results

Common descriptive results of the PCA process is a bar graph, with each bar representing the principal component and the height of each bar representing how much variance is described by this principal component. This is an informative result as it shows how important each principal component is and could convey valuable information to the data interpreter, such as the tradeoff of having a one more higher dimensional output space versus how much more variance of the data is captured by this dimension.

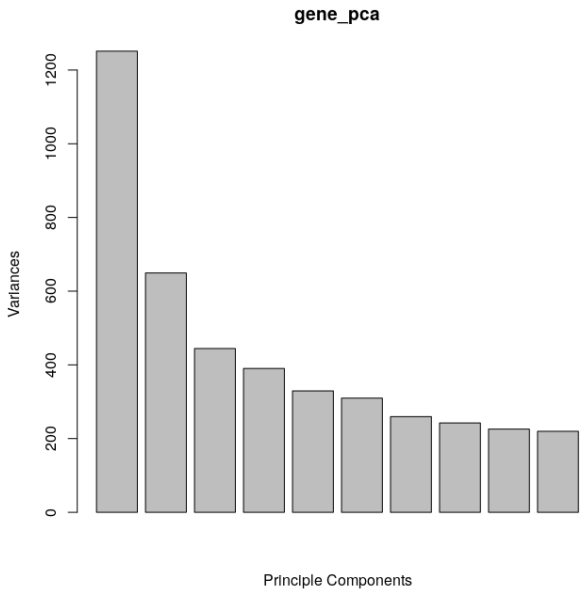

Figure 6: Bar graph of variance of data described by principal components, in descending order of principal component number.

Also to intuitively determine how separable the two classes were when considering all genes, the first three principal components of each patient were graphed. In the context of machine learning, we wish to extrapolate from our training set to be able to perform and test on our testing set. With this is mind, only the training set was used to compose the principal components, then the first three principal components of patients in the testing set were graphed. In Fig. 6 the red samples are from the ALL group and the blue samples are from the AML group. We see that there is some separability between the two groups, though the aren't separated completely allowing for any sort of decision boundary to be drawn between them.

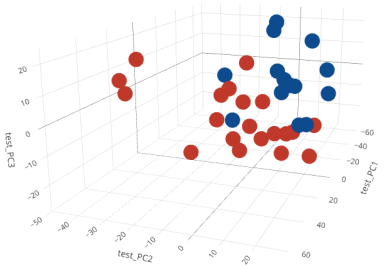

Figure 7: 3-Dimensional plot of Principal Components 1,2 and 3 of samples from the data set.

Chapter 6

Clustering

In the previous section we saw that we can reduce the dimensionality of the data set while maintaining a lot of the important variance described in the data. The final figure that was shown in this section was a visualization of the first three principal components of the patients that belong to two different clinical groups, and we concluded that intuitively they appear to be fairly separated. We can solidify this intuition by forming clusters of data points, that is grouping data points that are close together in the space that we have shown. The proportion of each of the clinical groups in each cluster can more formally tell us how separable these classes are in their current state using the first three principal components.

Though we are not the only ones to see the usefulness of cluster analysis in analyzing gene sequence expression data. This technique has in fact been one of the most frequently used techniques in this sort of analysis, though has been misused as interpretations of these clusters must be done very carefully.

Clustering is appropriate when there is no a priori knowledge about the data as well, as it can provide the user with groups of common samples. The specifics of what makes these samples common is an interesting further question in the analysis and can provide important clinical information.

6.1 Distances

Distance is usually thought of in terms of the standard euclidean distance, being the distance seen in the Pythagorean theorem. This however is only one type of distance, and other definitions of distance can be used to convey different characteristics of the difference between two points in an n-dimensional space.

$$d_E(x,y) = \sqrt{(x_1-y_1)^2 + (x_2-y_2)^2 + \cdots + (x_n-y_n)^2}$$
$$= \sum_{i=1}^{n}(x_i-y_i)^2$$

There are many other distances that can be used in this situation as long as they meet the standard criteria for a distance. The criteria that a distance must

meet include being symmetric, such as the distance from point a to b should be the same as the distance from point b to a ; the distance must always be a real number greater than or equal to 0 ; the distance must meet the triangle inequality, that is the distance between two points a and b should be shorter than or equal to the sum of the distances from a to a third point c and b to this same third point. This last rule that distances must meet can more easily be expressed as distances must be the shortest path between two points.

The euclidean distance is commonly used as it is both intuitive and well known, making results easy to interpret. Another common distance metric used is the Manhattan Distance. This distance can easily be thought of (and is named after) the traversal of city blocks from one point to another in a blocked city design. In this coordinate system only movements along axis directions are allowed.

$$d_M(x,y) = |x_1 - y_1| + |x_2 - y_2| + \cdots + |x_n - y_n|$$
$$= \sum_{i=1}^{n} |x_i - y_i|$$

The last distance that we will review in this section can be effective when comparing gene expression levels between experiments. This distance is the Pearson Correlation distance, and is proportional to the co-variance of the coordinates of the two points. This distance is commonly used for testing the reliability of equipment or experimental conditions

$$d_R(x,y) = 1 - r_{xy} \text{ where}$$
$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i = \overline{x})^2)}\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}}$$

Though this distance can be very powerful in this situation, it can be a very bad metric if a gene is incorrectly measured. Since this is the case, a variation called the jackknife correlation is sometimes used as it aims to solve the sensitivity issue by leaving out one dimension on each iteration, then selecting the distance to be the minimum of the calculated jackknife distances. Again this measure is not foolproof, though it is an improvement to the classical Pearson Correlation distance.

6.2 Clustering Algorithms

Many different clustering algorithms exists, and along with different algorithms some methods have differing forms of output as well, though the most common form of output is a set of clusters of points in the original input space. It is important to note that the clustering procedure is not necessarily deterministic, as some clustering algorithms can be applied differently to the same data, and some clustering algorithms begin with a random choice of clusters. That being said membership of a pattern to a cluster should be further examined before importance is given to the membership. In the same line of reasoning the fact that two patterns belong to the same cluster does not necessarily mean that there is any form of similarity between them without further observation. More cautionary notes on clustering is that absolutely anything can be clustered, and that given enough patterns, clusters will always form. Lastly one must be sure of the distance metric they are using during this clustering process, as in most cases the cluster assignments are highly dependent on this metric, and changing this metric will drastically affect the results of the cluster analysis. It may seems that we have shaped the interpretation of clustering analysis to be a daunting task, though we are simply encouraging the inspection of the results before any further conclusions should be drawn from the analysis.

Clustering is applied frequently in the real of genetics and more warnings of cluster analysis come with this specific problem type. Given the number of genes usually analyzed, there is no scientific value in observing some genes behaving in a similar way ; this scientific value should instead come from what can be said about the genes that fall in the same cluster, and what can be done with said genes.

6.2.1 K-Means

One of the simplest, fastest and most widely used clustering algorithms is the K-Means clustering algorithm. The name of this clustering algorithms comes from the user defined number of clusters in the output of the algorithm, usually defined as K.

The K-Means clustering algorithms works in the following way. First, K points are randomly assigned to be the center of the output clusters. Then, the distance from every point in the data set to every cluster center is calculated ; these distances are then used to assign each point to the cluster of the closest cluster center. The mean of each cluster is then calculated and assigned to be the new center of the cluster.

The distances from every data point to each cluster center are calculated again and this process is repeated until now data points move from one cluster to another.

A common note about this clustering method is that care should be taken in centroid initialization so that a cluster is not initialized far away from all points, leaving an empty cluster. To fend against this, a common practice is to initialize centroids as K points chosen randomly from the existing patterns, ensuring that the starting cluster centers are in an area populated by data ad each cluster will have at least one pattern.

6.2.2 PAM

Another popular clustering algorithm is Partitioning Around Medoids (PAM). PAM clustering has the same result as K-Means, being clusters of data points in the original input space. PAM starts with computing a dissimilarity matrix from the original data with the chosen distance metric. This dissimilarity matrix has a row and a column to represent each data point of the data, so an entry is this matrix represents the distance between the data point represented by its row and the data point represented by its column. Once this dissimilarity matrix is computed, an algorithm very similar to K-Means is implemented to form the clusters. The only difference between PAM and K-Means is that each cluster center must be a data point present in the data set (the median of each cluster is chosen to represent the new cluster as opposed to its mean) ; this is why all distances are computed first, as we will be able to lookup the distance from each data point to the center of each cluster as the center of each cluster is itself another data point in the dataset. This method inherits a robustness in respect to outliers as the median statistic is robust to outliers ; this is particularly important in the common situation in which many elements do not have a clear membership to any specific cluster.

6.2.3 Hierarchical Clustering

Hierarchical clustering is the first clustering algorithm that we are encountering with a slightly different output form than the propior clustering algorithms that we have encountered. The aim of hierarchical clustering is not to form completely distinct unrelated clusters, but to form a complete tree with individual data points as leaves as the root as the convergent point of all branches. Each branch in the resulting

dendrogram can be seen as a cluster, with the largest cluster being the whole data set and the smallest cluster being each independent data point.

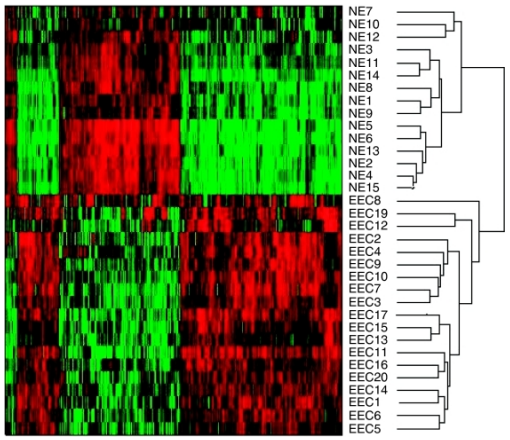

Figure 8: Depiction of Hierarchical clustering result.

This clustering method is a popular clustering method for gene expression levels as this analysis aims to tell more about the realationary structure of all genes. Hierarchical clustering can provide information similar to K-Means clustering but for varying cluster sizes. This is not a great direct comparison as there are differences in how the clustering is performed, though is a good intuitive lense to grasp the power of hierarchical clustering.

Hierarchical clustering can be performed in two ways, being bottom-up (agglomerative) method and a top-down (divisive) method. We will first describe the agglomerative process. First each data point is assigned to a cluster of its own, then the distance from each cluster to every other cluster is calculated. Next clusters are merged with their single most similar cluster. These distance calculation and merging steps are then repeated until only one cluster remains. We will now describe the divisive method. First we consider the whole set of data points to be clustered, then any one of the large number of non-hierarchical clustering algorithms is used to divide each cluster into two clusters ; for instance the K-Means algorithm with K set to two. This splitting step is then repeated until the resulting clusters only contain one patter each.

6.2.4 Biclustering

Biclustering is a very interesting type of clustering that we will only cover briefly. In microarray data, one can observe that the activities of genes are not independent of each other, thus it is important to study groups of genes and not single genes. In the previous algorithms that we have looked at, it was assumed that related genes should have similar expression profiles across all samples, though this assumption does not hold in all experiments. Biclustering was proposed to overcome these limitations by simultaneously clustering both rows and columns of a given matrix of data. This method helps to discover local patterns that cannot be identified by standard one-way clustering algorithms. The applications of Biclustering to microarray data can be seen by clustering simultaneously by expression level and by experiment, which will catch clusters of similar genes under varying experimental conditions.

6.3 Confidence in Cluster Assignments

As we mentioned in the introduction to this section, due to the non-deterministic nature of many clustering algorithms, it should not be taken for granted that a data point will always belong to the same cluster under different conditions or runs of a clustering algorithm. In order to develop a measure of how confident we are that a specific data sample belongs to a particular cluster, a bootstrapping approach can be implemented. In this approach, a goodness of fit metric is based on how many repeats of the same clustering algorithm applied to the data results with the data point in this same cluster. This method essentially clusters the data many times and the confidence of a data point belonging to a cluster is inversely proportional to the amount of times it is in a different cluster.

6.4 Assessing Goodness of Clusters

We will also briefly review one way, and the most frequently used way, of assessing the goodness of fit of clusters. This metric is very simple though can seem less intuitive with the introduction of summation notation. We define the intracluster total sum of squares distance to be the total sum of squared distances from all of the data points in a cluster to the center of the cluster. We also define the intercluster total sum of squares distance between two clusters to be the total sum of squared distances from

all data points in one of the clusters to the mean of the other cluster. Now in order to assess the goodness of fit of clusters the ratio of intercluster total sum of squared distances to intracluster total sum of squared distances can be examined. If this ratio is greater than 1, it is indicating that the clusters are further apart than their radii, and thus the clusters are well defined and not overlapping. The larger this ratio, the tighter and further apart the clusters, thus more desirable clusters.

A cheaper measure can be used to assess cluster fit. In the previous metric the intracluster total sum of squares distance can be substituted with the diameter of the smallest sphere including all members of the cluster. Also in the previous metric the intercluster total sum of squares distance can be substituted with the distance between these said spheres of cluster containment. This metric is rarely seen though as it is a much more sensitive measure.

6.5 Results

A few of the methods discussed were implemented and applied to the data of this work's interest. First the k-means clustering algorithm was applied. The standard k-means implementation is available in R, although it is not possible to change the distance metric to be used. In order to compare the results of the Euclidean distance and Manhattan distance, a custom implementation of k-means was made by this work.

Figure 9: Results shown from k-means clustering implementation

In the interest of the robustness of PAM, or otherwise known as k-medoids clus-

tering, this method was also applied. The implementation of this method was also available in R, and allowed for the choice of distance metric.

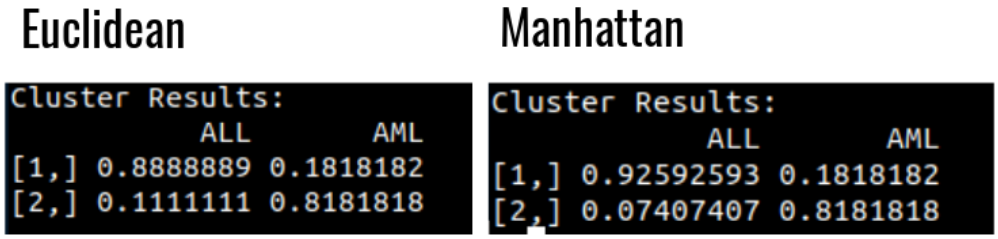

Figure 10: Results shown from k-medoids clustering implementation

We see from both of these clusterings that the classes appear to be fairly separable, with the exception of a few cases.

Chapter 7

Selecting Differentially Expressed Genes

So we have seen in the previous section that when we clustered the first three principal components of the approximately 7000 genes from the two clinical groups that there was a fair amount of separation between the two groups, but we aim to widen this gap. A part of this problem is that the principal components are composed of linear combinations of all approximately 7000 genes, though not nearly all of the genes are responsible for the difference between the two clinical groups. In order to select the genes that can be used to discern the two groups (or the genes that we will call differentially expressed), we can implement statistical tests between the gene expression levels of the two groups to determine which genes have statistically different expression levels between the two groups while taking into account the magnitude of the difference of the mean expression levels in each group and the total variance of the gene in each of the groups.

7.1 Fold Change & Unusual Ratio

As we discussed, statistical techniques can be extremely useful in this setting, though a commonly implemented technique for preprocessing does not rely on statistics at all. The Fold Change Ratio technique is simple and intuitive, mainly serving the purpose of removing genes with very similar expression levels between the two groups, thus removing genes that are highly unlikely differentially expressed. This method is called the fold change ratio as it examines the fold change of the mean expression level of the gene from one of the groups to another. More simply this metric determines how many times larger this mean expression level is from one group to the next. A common procedure used when filtering genes before a later statistical test is determining the fold change ration of all genes between the experimental groups, and discarding all genes with a fold change ratio less than 1.5 from being considered by future tests.

A still not completely sound by at least statistical based approach is the unusual ratio method. Instead of using an arbitrary user-selected fold change threshold to select differentially expressed genes, the cut-off threshold is automatically adjusted to the standard statistical 2 away from the mean. This should not be used as a method to select differentially expressed genes as no matter how many genes are up or down regulated, this method will always select the genes that are affected the most. This method does begin to introduce us to the use of statistics in selecting differentially expressed genes though does not impress.

7.2 Hypothesis Testing

A key note that must be made about microarray measurements of gene expression levels is that signal-to-noise ratios for genes with low expression levels tend to be lower than their counterparts with higher expression levels. The signal to noise ratios that we are discussing is the difference in gene expression level between the two groups over the amount of variance seen in the gene expression level over the two groups, and thus can be a highly informative measure for determining if genes are differentially expressed. Instead of forming our threshold on the change in expression level, we can more intelligently form a threshold on the ratio of this change in expression level relative to the variance of the gene.

This concept of signal to noise ratios has been a fundamental topic in statistics, and was very popular in the conception of William Sealy Gosset's statistical test

known as the Student's T-Test. This t-test was derived to allow statistically sound inference on normal data with small sample sizes, and can aid us greatly in selecting differentially expressed genes. We have discussed the test statistic for this test earlier, being the signal to noise ratio of difference of the mean of the gene expression level of the the two groups, divided by its variability in the two groups. This may be made clearer by the following equation:

$$T = \frac{signal}{noise} = \frac{\text{difference between group means}}{\text{variability of groups}} = \frac{|\overline{x}_1 - \overline{x}_2|}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

Previously we have talked about selecting the differentially expressed genes from some arbitrary threshold, though the t-test prescribes a threshold given a significance level. This significance level is based on the probability of rejecting the null hypothesis given it was true ; in our case the significance level can be seen as the probability of not selecting a gene as differentially expressed given that the gene is differentially expressed. In the prior section of multiple comparisons we have seen various corrections for this significance level in order to make all simultaneous hypothesis tests less conservative, allowing genes that are actually differentially expressed to be selected.

7.3 SAM

Though there are many possible significance level corrections for the t-test, and computing the critical value from the test-statistic can be permutation-based and take the dependency structure of the data into account, there is still a better method to statistically select differentially expressed genes. The Significance Analysis of Microarrays (SAM) method was developed by a Stanford lab specifically for this purpose. The test-statistic used by SAM is very similar to that used by the t-test. The only difference that can be seen in these statistics is the positive constant in the denominator of the test statistic used by SAM. This positive constant, called the exchangeability factor by the authors, serves an important role in the method. As we have discussed the variance for genes with low expression levels is higher than that of genes with higher expression levels. This can cause the test statistic to be much larger in situations of these low expression levels and high variance, and thus make the variance of the test statistic highly dependent on the expression level of the gene. Since SAM does not make any distributional assumptions about the gene, we want permutations of this

test statistic to accurately capture the characteristics of the test-statistic. In order to make these test statistics comparable and to dissociate the variance of this test statistic and the gene expression level, the addition of this small positive exchangeability factor in the denominator is crucial. This exhability factor is chosen to minimize the variance of this test statistic, and thus makes our nonparametric estimation of the distribution of the test statistic more plausible. This test statistic is depicted below.

$$d(i) = \frac{\overline{x}_1(i) - \overline{x}_2(i)}{s(i) + s_0}$$

As mentioned permutations of the data are used to determine the distribution of the test statistic, making this a valid test if the normality assumption does not hold. We have previously discussed how SAM then uses variations of the delta parameter (the distance between the expected statistic and the observed test statistic) to show the user how the false discovery rate varies with this parameter. This method is much more easily interpretable as it provides the estimated number of non-differentially expressed genes for a given delta, as well as the probability of a given selected gene being non-differentially expressed, a much clearer metric of dependability in each gene.

7.4 Moderated T-Statistic

Another possible approach to selecting differentially expressed genes is to make the assumption that the variance of all genes come from the same distribution, then to use a hierarchical model to determine if a gene is differentially expressed. If the assumption of this method is valid, then the estimation of one pooled gene variance should be much more accurate and reliable than that of each independent model made by the previous two tests. Thus the test-statistic for this method would use the same variance for each test, with the only difference in tests over different genes accounting for the difference in their mean expression levels between the two groups.

$$t = \frac{\hat{B}_{gj}}{\hat{\sigma}/\sqrt{n}}$$

In order to calculate this variance, an empirical Bayes method is used, being a procedure in which the prior distribution is estimate from the data. This family of methods is used as an approach to setting the hyper-parameters of known distribu-

tions to best fit the data. In our case we are interested in finding the best estimation of assuming that the data is normal. Here we are calling our model hierarchical because we are estimating at two levels. First we are building a linear model for each gene. The prior distribution for the common **g** between all models is then estimated using an empirical Bayes approach. We can see that in terms of the context of the general linear model the contrast of interest, being the different in the mean expression level of gene g between the two groups is called $\hat{B}_g$, the standard deviation of gene g is called s_g, and the variance of gene g is called v_g.

$$t = \frac{\hat{B}_{gj}}{s_g/\sqrt{v_{gj}}}$$

Through the course of applying this method in "Linear Models and Empirical Bayes Methods for Assessing Differential Expression in Microarray Experiments", Gordon K. Smyth derives the estimated sample variance of gene g to be

$$\tilde{s}_g^2 = \frac{d_0 s_0^2 - d_g s_g^2}{d_0 + d_g}$$

where s_0^2 is the mean of the estimated prior of σ^2 with d_0 degrees of freedom. This newly derived variance from the empirical model is then substituted into the standard t-statistic to produce a more robust model under the assumptions of the technique.

$$\tilde{t} = \frac{\hat{B}_{gj}}{\tilde{s}_g/\sqrt{v_{gj}}}$$

7.5 Discussion

We have seen that the moderated t statistic approach can product more robust results than other methods, given that the assumption of equal variance of all genes is valid. This assumption however is not clear cut and is not accepted by all, so the use of this method should be done so with the discretion of the experimenter. As we have mentioned in the section on SAM, the authors of the method found that the variance of genes was dependent on gene expression level, and that it might even be completely gene-dependent. If this is true, then the assumption of the moderated t statistic is not valid and neither are its results.

Chapter 8

Machine Learning

Machine learning refers to the the set of topics dealing with the creation and evaluation of algorithms that facilitate pattern recognition, classification, and prediction, based on models derived from existing data Book. In this section we will review different classes of machine learning, methods of assessing how well machine learning algorithms are performing, and finally review machine learning techniques and their application to our problem of interest.

8.1 Introduction

The term machine learning applies a wide set of problems and solutions. We will now discern these different types of learning and problem types and define them more clearly. First we will review Supervised Learning. In supervised learning, objects are classified using a set of features. The true classes of examples are known in advance and the goal is to build a model that can extrapolate from the labelled examples to successfully classify new samples. This method can be applied to the continuous case of time-series data as well to predict the next value in a time series given the values in the previous time series. Observing the problem of our interest in this work, it is easy to see that we have a discrete supervised learning problem on our hands. We have two different classes of patient type with measurements of each of the patients gene expression levels. Our task is to extrapolate from a subset of these patients and to successfully classify the cancer classification of patients that we have not use to build our prediction model.

Another type of machine learning technique is Unsupervised Learning. In unsupervised learning the data provided is unlabelled and the goal is to discover similarities between objects. This may sound very similar to a previous section of this work and that is because it is! Clustering is a form of unsupervised learning, and as we say can provide insight into the relations of data points in the dataset, with further examination allowing more information on these similarities. Another type of learning very similar to this is called Semi-Supervised learning. In this problem type the labels of some samples are known. These known labels then can be applied after clustering is

performed, being used to label the clusters that the known samples end up in. This method of machine learning makes the assumption that similar samples are closer together in feature space than others.

We now review the different problem types, or goals, that different machine learning problems can have. These problem types can be separated into three main classes, being class prediction, class comparison, and class discovery. The goal of class prediction is in line with the powers of the method of supervised learning. In a class prediction problem the classes are known in advance and labelled data is provided to extrapolate from. The data provided in class comparison problem types is of the same form as the data provided for the class prediction problem type, however the goal is slightly different. As opposed to using this data to build a model to attempt to classify new samples, we are interested in directly studying the provided data to determine what differentiates the different classes. This problem type can be seen to be classical discriminant analysis. Unlike in the previous problem types, the classes of the samples provided in a class discovery problem are not known in advance. The goal of this problem type is to identify the samples that share certain features. As we see here the labels of the data are not know, and thus unsupervised machine learning techniques, such as clustering, are applicable, and will meet the goal of the problem type by identifying samples that are similar.

We see that some problems types are more similar than others. Class prediction problems usually require a feature selection step that determines useful and non useful feature of the data for classification. It seems then that class prediction problems encapsulate class comparison tasks as well, but this would be a mistake to say as class comparison tasks attempt to determine all features that are different between the two classes, while feature selection only attempts to determine the features that are needed to successfully classify new samples into either class. A useful example for this is in the case of trying to classify new tractors as either John Deere tractors of Kubota tractors. The only features needed to discern these two tractor types is their color, though if we were in the market to buy a tractor and comparing the two brands we would want to know everything that makes these two brands of tractors different, not just simply the color Book. The class discovery task on the other hand is much different than both of these problem types, as the classes of the given samples are not known and the objective of the analysis is very different. We are noting the different of these problem types so heavily as it is very important to know your problem type

in machine learning before attempting to apply any techniques, to ensure that you are applying the right methods to the data and asking the right questions.

8.2 Context of Supervised Learning

For the rest of this paper we will only be reviewing supervised learning techniques as this is the apprirate family of techniques for the problem we are interested in and we have mentioned methods to assess the other problem types already. In this section we will layout the common context and jargon of supervised machine learning approaches.

In supervised learning approaches we assume that we wish to classify a collection of $i = 1, \cdots, n$ objects into k predefined classes. In our case, we are wishing to classify our n patients that we have held out for testing into one of the two clinical cancer groups. Then a classifier C(x) may be viewed as K discriminant functions $g_c(x)$, such that the object with feature vector x will be assigned to the class c for which $g_c(x)$ is maximized over all class labels c in $\{1, \cdots, K\}$. This classifier, being the set of discriminant functions, can be seen as partitioning the feature space into K disjoint subsets, one for each class. Two main approaches exists to identify these discriminant functions $g_c(x)$. The first approach is to assume knowledge about the underlying class conditional probability density functions (PDFs) and assign $g_c(x) = f(p(x \mid y = c))$ where f is a monotonically increasing function such as ln. Intuitively this probability based classifier will classify the object with feature vector x to its most probable class. In order to estimate these probability of class membership functions both parametric and non-parametric techniques can be used. As opposed to this technique where knowledge of the class conditional PDFs is assumed, the class boundaries can be directly estimated without explicitly calculating the the PDFs at all.

8.3 Error Estimation and Validation

In order to assess how well our classifiers are doing we first discuss some common metrics and how they can be applied. A commonly used tool to analyze how well a classifier is performing is the confusion matrix, being a matrix with rows corresponding to the true classes of the samples and columns corresponding to the predicted classes of the samples. Each entry of the confusion matrix contains the amount of samples that belong to the true class represented by the row and the predicted class

by the classifier represented by the column. The rows of this matrix are commonly standardized to more easily show how well each sample class is being handled by the classifier.

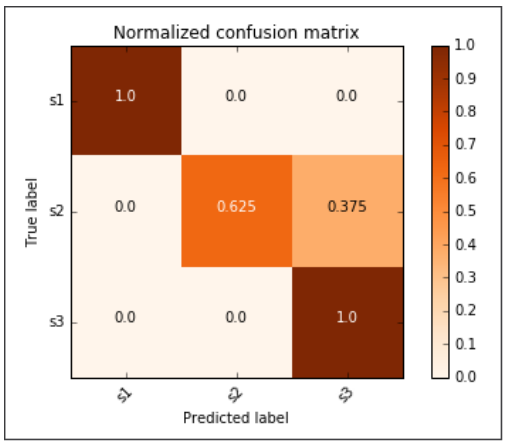

Figure 11: Depiction of confusion matrix for the 3 class case.

We have discussed the confusion matrix though have not gone into great detail about how it is composed. A key factor is that we wish to use this matrix to determine how well the classifier can predict new samples from extrapolating from given samples. This can best be facilitated by splitting up our given data into a training set and a testing set. The training set will be extrapolated from and be used for building the classifier. Now in order to assess how effectively our classifier predicts the true class of new samples, we can use the testing set that we have held out from training. Since we know the true class of every sample in this testing set, we can compare the predicted class from the classifier of the sample versus the known true class of the sample in order to build this confusion matrix.

The method of holding data we are given is very effective to determine how well our classifier will perform on new data, though is not optimal to build the most robust classifier. We wish to use all of the data that we can in order to build the best classifier possible, but we also wish to effectively measure how well this classifier is going to perform on new samples. A method known as n-fold cross validation was developed just for this situation. In n-fold cross validation the data is split into n disjoint subsets, referred to as folds. The classifier C(x) is then trained and tested n-times, on each of these train-test runs C(x) is trained with n-1 folds and tested on the fold was held out for the current run. Performance metrics such as specificity (the

percentage of cases belonging to class properly predicted as class c) and specificity (the percentage of classes not belonging to class c that were properly predicted not as class c). These metrics are then averaged over all of the runs to provide information on how well the classifier would extrapolate when trained on all of the training data.

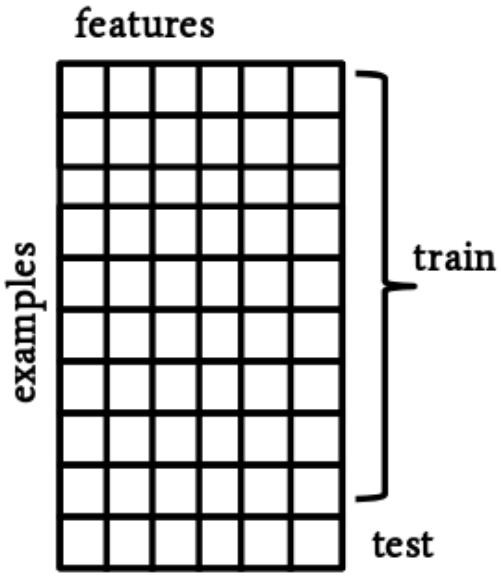

Figure 12: Depiction of k fold cross validation

In some situations the data may even be too scarce for n-fold cross validation, and then leave-one-out (LOO) cross validation must be used. This method is simply n-fold cross validation in which the number of folds n is equal to the number of data points. The metrics obtained from LOO cross validation may have a low bias, but a very large variance as some samples may be very helpful in extrapolation, and when left out the classifier performs significantly worse.

We will now speak more on the dynamics of the training and testing sets. It is common, and highly recommended that there is no bias to the selection of these sets. In order to accurately represent all samples in the training and testing sets, they are usually chosen at random from the existing data.The training set is usually a great deal larger than the testing, so the testing set being even sampled from the data is highly important to obtain representative performance metrics. That being said is it best to evenly represent each class in the training and testing sets, though this does not always happen as in the real world some classes have been more frequently

sampled before the data even gets to the analyst. When the number of samples is very different between the K classes, the dataset is said to be unbalanced. In this case, it is best ensure that the training set is balanced as many classifiers will favor the richer data set and skew predictions.

8.4 Feature Selection

A commonly made mistake in the field of machine learning is to attempt to build a model with too many features and not enough examples, In general the number of features should be much smaller than the number of examples in order for proper extrapolation. This is a very large problem in context of microarray analysis as gene expression levels for thousands of genes are recorded, though usually only for tens of patients. Commonly in order to reduce the number of features in the data set, attempts are made to only select the features that are helpful in discerning the defined classes in a process called feature selection. Two methods of feature selection include wrapper methods and filter methods. For wrapper methods, the classifier is trained on different subsets of features to determine the best subset available and the impact of losing the other features. For filter methods, tests for mutual information using correlation coefficients of statistical significance tests are used to ensure not intercollinearity between features. In context of the problem of our interest, we are using statistical analysis to determine the differentially expressed genes ; in context of machine learning it can be said that we are performing feature selection to determine features that aid in discerning the two given classes of labels.

8.5 Classifiers

Now that we understand the context of which we are working, we can begin to review different implementations of classifiers.

8.5.1 Quadratic and Linear Discriminants

Quadratic discriminants is a standard classification approach applicable to continuous features. This method assume that for each class c, the feature vector x follows a multivariate normal distribution with mean c and variance c. Using the multivariate normal probability density function and replacing the true class mean and co-variance

matrices with simple derived estimates (m_c and σ^2 respectively), the discriminant function can be computed as

$$g_c(x) = -(x - m_c)\hat{\sigma}_c^{-1}(x - m_c)^T - log(|\hat{\sigma}_c|)$$
where
$m_c = \frac{1}{n_c}\sum_{i=1}^{n_c} x_i$, $\hat{\sigma}_c = \frac{1}{n_c}(x_i - m_c)(x_i - m_c)^T$

The discriminant functions $g_c(x)$ are monotonically related to the densities $p(x \mid y = c)$, yielding higher values for larger densities. The values of the discriminant functions will differ from one class to another only on the basis of the estimations of the class mean and co-variance matrix. After the formation of these discriminant functions $g_c(x)$ for each class c, a new object can be classified by selecting the class for which the discriminant function is the largest. This classification approach produces nonlinear (quadratic) class boundaries and this is called the quadratic discriminant or Gaussian classifier.

As opposed the assumptions of the quadratic classifier, one could assume that the co-variance matrices c for $c = 1, \cdots, K$ are all the same. In this case a single pooled co-variance matrix is used, being especially useful when the number of samples in each class is too low to produce a reliable estimate. The resulting classifier uses hyper-planes as class boundaries, endowing the name normal-based linear discriminant.

To further cope with situations where the number of features is comparable to the number of samples, a further simplification of the co-variance matrix can be used by setting all off-diagonal elements to be zero. This method then neglects the co-variations between features and has been found to outperform other types of classifiers of this variety on microarray analyses. This is interesting and counter-intuitive as we know that the expression level of some genes is directly dependent on the expression levels of others, and we would expect this to be valuable information during the classification procedure.

8.5.2 KNN

The K-Nearest Neighbor (KNN) classifier is a simple and intuitive classifier. This classifier can be seen as a non-parametric method of density estimation, and therefore makes no assumptions about the underlying distribution of the data besides the continuity of feature values. As opposed to most other classifiers, there is actually

no training involved in the KNN approach. When a new sample must be classified, the distance between the new sample and all other samples in the training set are calculated. After these distances are calculated, the samples in the training set are then ordered from closest to furthest from this new sample. The parameter k is then used by only retaining the k nearest neighbors of the new sample. The number of data points of each class in the k remaining data points is the calculated (n_c for each class c), and the class that is most highly represented (the class with the largest value of n_c) is then chosen to be the class of the new object. Thus the discriminant function of the KNN classifier can be written as $g_c(x) = n_c$. We note that since all of the computation occurs in the classification phase, and that since the training data set must be retained for the classifier to function, that this classifier is not very computationally or memory efficient.

As opposed to some other learning methods KNN has what is called a hyper-parameter. A hyper-parameter is a parameter that determines how the machine learning classifier learns. In the case of KNN, the hyper-parameter is the parameter k. As k increases, the classifier becomes more robust as more data points are used to determine the true class of the new sample. This may be the case, but we should certainly not choose K to be the size of the data set, as then the predicted class will always be the most represented class in the training set. In order to optimize this hyper-parameter, we must further separate the testing set into a testing set and a validation set. The testing set will be used to test how well the classifier performs with each value of the hyper-parameter, and can be used to choose the optimal value of the hyper-parameter. Once this optimal hyper parameter value has been decided, the performance of the classifier can then be determined from its performance on the validation set, as the validation set was neither used to train the classifier, or to choose the optimal hyper-parameter for the classifier.

8.5.3 Decision Trees

As opposed to the previous classification method, the Decision Tree method does actually build a model for classification. A Decision Tree is a structure constructed by an iterative construction of individual features that are most salient at each node. At each of these levels, the input space X is repeatedly split into descendant subsets, starting with X itself. Usually decision trees are constructed top-down, beginning

at the root not and successively partitioning the feature space. This partitioning involves determining if the current node is terminal or not, if not then selecting the splitting feature and value at the current node, and if assigning the class label to the terminal node that minimizes the error rate.

The most common implementation of decision trees is the binary implementation, in which the data coming into the node is split into two seperate sets by the node by a single feature and value. Though this implementation is not the most optimal implementation possible, the interpretability of the decision rules leading to each classification lead it to be the most common implementation. The creation of these decision trees is very computationally expensive, as finding the threshold value that creates the best split at each node requires testing manny different thresholds for a single feature. Through the creation of this model is computationally expensive, once the model has been trained the training set no longer has to be retained and classification requires very few computations.

8.5.4 Artificial Neural Networks

Artificial Neural networks (ANNs) get their name from their biological counterparts. They are an interconnect network of simple processing units, usually referred to as neurons, nodes or units. Each of these nodes belong to a layer, in which every node from one layer is connected to every node of the next. In the simplest and most common approach the network proprograte forward through the layers, beginning with the input layer and ending with the output layer. The number of nodes in each of these layers is determined by the dimensionality of the input and output space. In our case each input node would take on the value of the gene expression level of one gene, and thus there would be an input node for every gene that we wish to use to classify the patient. If we decide to perform principal components analysis before classification, each node would take on the value of a principal component and there would be as many nodes as principal components that we wished to use for classification. As for the number of output nodes in our case, we wish to classify each sample into either the ALL class or the AML class. We will use two output nodes, the values of which we will use as the discriminant functions to determine the predicted class of the sample.

More about these processing units. Each node in the input layer usually does not

perform any processing, by simply takes on the value of the single feature it represents for the current sample. We now now that there can be many layers between the input and output layers, with these in between layers called the hidden layers as they have no connections outside of the network. Each of the nodes in these layers sum all of their inputs, apply the sigmoid function to this summed value, and then propagate this value forward to all nodes that it is connected to. As values are propagated forward through the network, they are multiplied by the weight or strength of the connection from the originating node to the terminating node of the signal. A bias term is also added during this propagation process.

$$\sigma(z) = \frac{1}{1 + exp(z)}$$

$$g_k(x) = \sigma[\sum_j \alpha_{j,k}\sigma(\sum_i x_i w_{i,j} + b_j^h) + b_k^0]$$

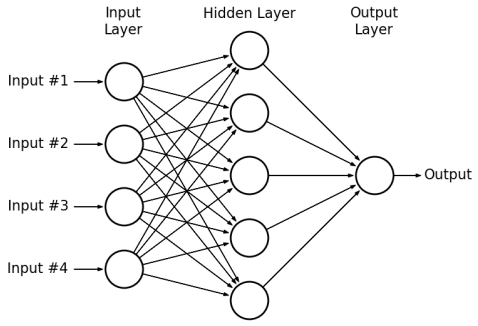

Figure 13: Depiction of Artificial Neural Network

Now that we understand the structure and processing of the neural network, we can discuss how it can be trained to produce informative discriminant functions. In order to assess how well the neural network has captured the data, we first determine its cost, being the total sum of differences squared of desired outputs for each sample and the provided outputs for each sample. In our case we wish the node representing the discriminant function of the true class to have a value of 1, and the node representing the discriminant function of the incorrect class to be 0. Once we have calculated the total cost of the network, and since the network is really just a differentiable function, we can compute the partial derivative of each weight with respect to the output of the network. We can then change each weight of the network based

on its contribution to the error times the total cost.

There are many different types of neural networks though in this work we just outline the simplest and most intuitive that could be applied to our current classification problem.

8.5.5 Logistic Regression & SVM

Logistic regression is one of the most popular and widely used classification algorithms and can be applied in situations in which the desired prediction is a discrete class, such as the problem this work is interested in.In this section we will introduce logistic regression and show how support vector machines (SVMs) can be seen as a addition or modification to logistic regression.

Logistic regression gets its name from the use of the sigmoid, or alternately titled, logistic function. The hypothesis function of linear regression, $h_\theta(x)$ where θ is the vector of parameters fit to the model using the training data is usually described as $h_\theta(x) = \theta^T x$. A slight modification of this hypothesis function can produce the hypothesis function for logistic regression, by letting the hypothesis function be $h_\theta(x) = g(\theta^T x)$ where $g(x)$ is the sigmoid function defined as

$$g(x) = \frac{1}{1 + e^{-z}}$$

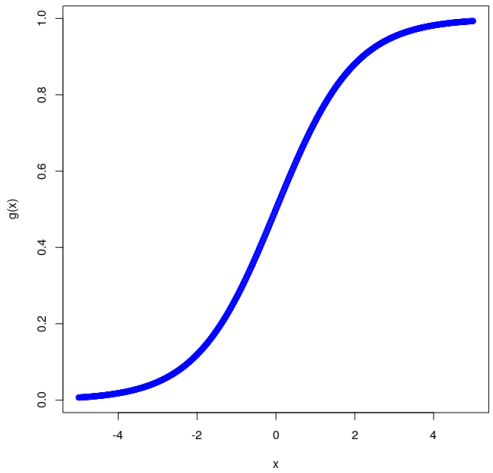

Figure 14: The logistic function

The interpretation of the output of this hypothesis function $h_\theta(x)$ is the probability of the sample with feature vector x belonging to the positive class ($y = 1$), defined by $h_\theta(x) = p(y = 1 \mid x; 0)$.

We first define a decision boundary to be a hyper-plane in the feature space in which data points on one side of the hyper-plane belong to one class, and data points on the other side of the hyper-plane belong to the other class. This decision boundary in the case of logistic regression can be intuitively derived by assigning the positive class ($y = 1$) if $h_\theta(x) > 0.5$, and the negative class ($y = 0$) if $h_\theta(x) \leq 0.5$. Since $g(x) = 0.5$ at $x = 0$, these conditions could be rewritten as $y = 1$ when $\theta^T(x) > 0$ and $y = 0$ when $\theta^T(x) \leq 0$.

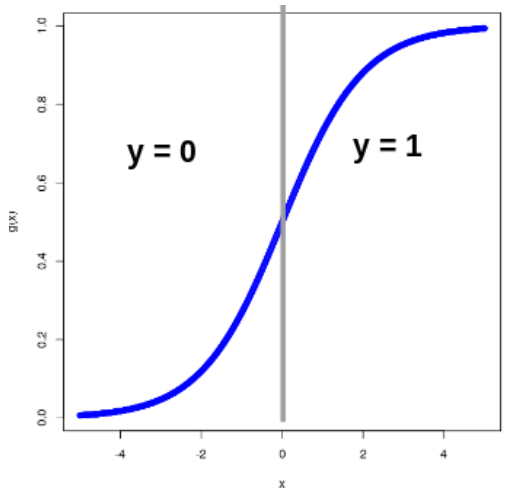

Figure 15: Depiction of decisions of classes made from logistic function

Simple implementations of logistic regression implement a linear decision boundary in the feature space. In terms of the parameter vector and hypothesis function this simply means that the parameter coefficients for each given feature are linear. Since in logistic regression we do not limit the form of this feature vector, we may also introduce nonlinear terms to form a more complex nonlinear decision boundary.

So far logistic regression has seemed to be somewhat intuitive, though we have not seen how to actually go about calculating the parameter vector θ to make the decisions that we have formulated. Similarly to in the ANN section, a cost function of the model is implemented. In simple linear regression this cost function was defined as

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{2}(h_\theta(x^{(i)}) - y^{(i)})^2$$

Though this is not ideal for us as this is a non-convex (non-bowl shaped) cost function with respect to the logistic hypothesis function. Before we discuss why this

cost function being non-convex is not good, we must dive further into why we need this cost function. Given the differential cost function, much like we saw in the ANN section we can find the partial derivative of each parameter with respect to this cost function and update the parameters accordingly to decrease the cost function. However this method will not work if the gradient of the cost function at any point does not lead to the global minima of the function. We can see that this will only happen when the cost function is convex, thus prompting us to define a convex cost function involving the hypothesis function defined for logistic regression. To adapt the cost function we saw for linear regression to logistic regression, we will first split the cost function into two more digestible cases.

$$Cost(h_\theta(x), y) = \begin{cases} -log(h_\theta(x)); y = 1 \\ -log(1 - h_\theta(x)); y = 0 \end{cases}$$

Graphically these function can be seen as

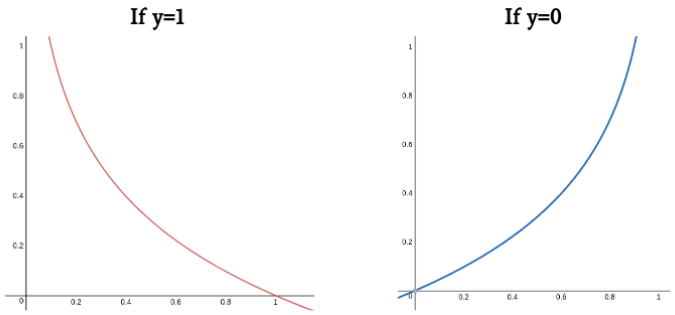

Figure 16: Graphic depictions of logistic regression piecewise cost function

Since the value of y can only take on the value of 1 for a positive case and 0 for a negative case, this cost function can be written in a single form

$$Cost(h_\theta(x), y) = -ylog(h_\theta(x)) - (1 - y)log(1 - h_\theta(x))$$

We should note here that this cost function can be derived from maximum likelihood estimated, though intuitively it also has many desirable properties for a cost function to have. This cost function is convex for the logistic hypothesis function, tends towards infinity as predicted classes approach the incorrect class, and is 0 when the predicted class is the correct class. We can better see this gradient descent update

rule in the following figure

$$\text{Repeat}\{$$
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$
$$\}$$

Figure 17: Gradient descent for selection of logistic parameter values

where α is the learning rate, being how far in the direction of the gradient we will step (or how much each parameter is updated by) at each iteration. If α is too large, we may overstep the minimum and if α is too small, convergence may take too long or never happen. Before moving onto support vector machines we note that logistic regression can be used to predict multi-class problems by developing a model for each class, then using each of these models as discriminant functions.

We will now discuss Support Vector Machines (SVMs). SVMs can sometimes provide a cleaner and more powerful way of learning complex nonlinear function compared to logistic regression and neural networks. As we mentioned before, in reviewing SVMs we can actually see them as a modification of logistic regression, in which we estimate the cost function in a simpler way and obtain a larger classification margin. If we view the logistic regression in light of producing a large margin classifier, if $y = 1$ we want $h_\theta \approx 1$, implying $\theta^T x >> 0$, and if $y = 0$ we want $\theta^T x << 0$. In modifying the cost functions, we will aim to produce this large margin classifier.

$$cost(h_\theta(x))_1 = m(1 - x) \text{ for } 0 < x < 1, 0 \text{ for } x \geq 1$$
$$cost(h_\theta(x))_0 = m(1 + x) \text{ for } -1 < x < 0, 0 \text{ for } x \leq -1$$

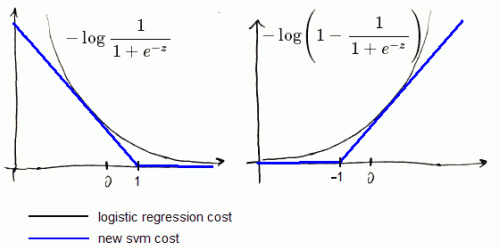

Figure 18: Depiction of SVM's cost function being an estimate of the logistic regression cost function

These cost functions add a margin to classification, as the cost is not 0 just when the example is classified correctly, but when it is classified correctly with a large margin (a margin of 1). This characteristic is what makes SVMs a large margin classifier.

Before we discuss the final cost function of SVMs we must first introduce the concept of regularization. Regularization refers to the process of scaling features to avoid overfitting, which can be facilitated by adding a term to the cost function called the regularization term. This regularization term provides incentive for lower feature scalings as the magnitude of the features will directly influence the cost function. Often the regularization terms is of the form $\lambda \sum_{i=1}^{m} \theta_j^2$ where λ is referred to as the regularization parameter. Now we can see how the cost function changes from in logistic regression to the cost function used by SVMs.

Usual modifications to this cost function include parameterizing the regularization coefficient from $A + \lambda B$ to $CA + B$, and all $1/m$ terms are dropped for simplification. Gradient descent can then be applied to this cost function to fit the parameters of the model θ, and be used to classify new samples. SVMs much like logistic regression do not output probabilities, but decisions.

Logistic Regression cost function:

$$J(\theta) = \frac{1}{m}[\sum_{i=1}^{m} y^{(i)}(-log h_\theta(x^{(i)})) + (1 - y^{(i)})(-log(1 - h_\theta(x^{(i)})))] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

$\Downarrow$

SVM cost function:

$$J(\theta) = \frac{1}{m}[\sum_{i=1}^{m} y^{(i)} cost_1(\theta^T x^{(i)}) + (1 - y^{(i)}) cost_0(\theta^T x^{(i)})] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

$\Downarrow$

Commonly Seen SVM cost function:

$$J(\theta) = C[\sum_{i=1}^{m} y^{(i)} cost_1(\theta^T x^{(i)}) + (1 - y^{(i)}) cost_0(\theta^T x^{(i)})] + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$

Figure 19: Derivation of SVM cost function from logistic regression cost function

8.6 Results

A few of the machine learning methods discussed in this section were applied to both the raw data that was provided, being all the genes, then to just the genes that were selected to be differentially expressed by SAM, controlling for an FDR of .05.

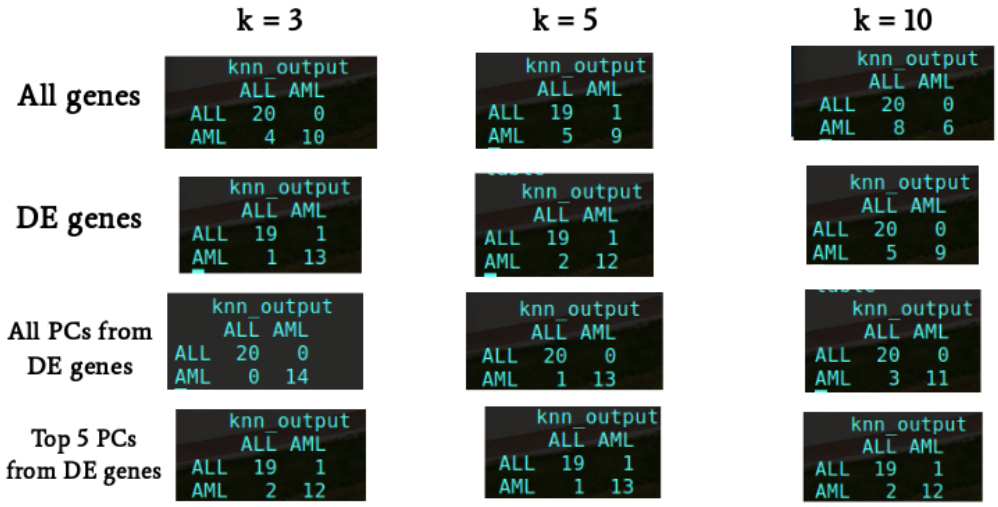

Figure 20: Results from various KNN implementations

We see from the results of K-Nearest Neighbors that the classifier performs drastically better after statistical differentially expressed gene selection.

A binary decision tree, logistic regression, and an SVM were all also applied to the top principal components of the differentially expressed genes and provided similar classification results.

Figure 21: Results from various classifier implementations

Chapter 9

Conclusion

We have seen over the course of this paper the importance of statistical analysis in selecting differentially expressed genes before further machine learning class prediction approaches. The excellent classification performance of the simple KNN classifier shows the power of this statistical analysis in these applications.

Over the course of this capstone project I was able to delve into many different topics and gain deep understandings in these fields. This capstone project has also motivated me to pursue graduate education in similar fields of work, and to build towards a career in Computational Genomics, or some field in the intersection of Biology, Medicine, Statistics and Computer Science.

References

[1] Draghici Sorin. Statistics and Data Analysis for Microarrays: Using R and Bioconductor. Chapman and Hall, 2012.

[2] 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, E. S. Lander Science 15 Oct 1999 : 531-537

[3] G. Smyth et al. Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. Statistical Applications in Genetics and Molecular Biology, 3(1):1027, 2004.

[4] "Introduction to Multi-Level Modeling." YouTube, Duke, 6 Feb. 2017, www.youtube.com/watch?v=m4fx_mzlBQI.

[5] "Machine Learning." Coursera, StandfordUniversity, 2018, www.coursera.org/learn/machine-learning.

Figure 1

[6] Klimas, Lisa. "Gene Expression and the D816V Mutation." Mast Attack, www.mastattack.org, 9 Aug. 2017, www.mastattack.org/2014/06/gene-expression-and-the-d816v-mutation

Figure 2

[7] Vong, Andrew. "3D DNA Strand Rendering in MAYA." av_designs, 27 Mar. 2015, www.andrewvong.com/3d-dna-strand-rendering-in-maya/.

[8] "Ribonucleic Acid (RNA)." RNA Extraction, Quinnipiac University, qu.edu.iq/bt/wp-content/uploads/2015/12/RNA-extraction.pdf.

[9] charity-stanton. "What Organic Molecule Is DNA?" SlideServe, 12 Nov. 2014, www.slideserve.com/charity-stanton/what-organic-molecule-is-dna.

Figure 3

[10] Catherine. "Role of MRNA in Protein Synthesis." Study.com, Study.com, 2018, study.com/academy/lesson/role-of-mrna-in-protein-synthesis.html.

Figure 5

[11] Cui, Yan. "Data File: Stanford_Large.Txt." Microarray Data Analysis II, compbio.uthsc.edu/microarray/lecture2.htm.]

Figure 8

[1] Draghici Sorin. Statistics and Data Analysis for Microarrays: Using R and Bioconductor. Chapman and Hall, 2012.

Figure 11

[12] unutbu. "How Can I Make My Confusion Matrix Plot Only 1 Decimal, in Python?" Stack Overflow, 2016, stackoverflow.com/questions/40264763/how-can-i-make-my-confusion-matrix-plot-only-1-decimal-in-python.

Figure 13

[13] Sharma, Aditya. "Understanding Activation Functions in Deep Learning." Learn OpenCV, Learnopencv, 30 Oct. 2017,www.learnopencv.com/understanding-activation-functions-in-deep-learning/.

Figure 16, 17, 18, 19

[5] "Machine Learning." Coursera, StandfordUniversity, 2018, www.coursera.org/learn/machine-learning.