2011-03-03

Utilizing Universal Probability of Expression Code (UPC) to Identify Disrupted Pathways in Cancer Samples

Michelle Rachel Withers
Brigham Young University - Provo

Utilizing Universal Probability of Expression Code (UPC)

to Identify Deregulated Pathways in Cancer Samples

Michelle R. Withers

A project submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Dr. W. Evan Johnson, Chair
Dr. E. Shannon Neeley
Dr. David Engler

Department of Statistics

Brigham Young University

April 2011

ABSTRACT

Utilizing Universal Probability of Expression Code (UPC)
to Identify Deregulated Pathways in Cancer Samples

Michelle R. Withers
Department of Statistics, BYU
Master of Science

Understanding the role of deregulated biological pathways in cancer samples has the potential to improve cancer treatment, making it more effective by selecting treatments that reverse the biological cause of the cancer. One of the challenges with pathway analysis is identifying a deregulated pathway in a given sample. This project develops the Universal Probability of Expression Code (UPC), a profile of a single deregulated biological pathway, and projects it into a cancer cell to determine if it is present. One of the benefits of this method is that rather than use information from a single over-expressed gene, it provides a profile of multiple genes, which has been shown by Sjoblom et al. (2006) and Wood et al. (2007) to be more effective. The UPC uses a novel normalization and summarization approach to characterize a deregulated pathway using only data from the array (Mixture model-based analysis of expression arrays, MMAX), making it applicable to all microarray platforms, unlike other methods. When compared to both Affymetrix's PMA calls (Hubbell, Liu, and Mei 2002) and Barcoding (Zilliox and Irizarry 2007), it performs comparably.

Keywords: oncogenic pathways, microarray normalization

ACKNOWLEDGMENTS

I would first like to thank Dr. Evan Johnson for all of his patience and assistance on this project. I would also like to thank Dr. Andrea Bild at the University of Utah for the data used in this project, Brandon Crowther for his initial work on MMAX, and all my peers who assisted by giving both verbal and written feedback. Additionally, I would like to thank the Office of Research and Creativity for partially funding this project with an ORCA grant, as well as the mentored research opportunity provided by the Department of Statistics. Lastly, I know I could not have done this without the encouragement and support of my wonderful husband, Drew.

CONTENTS

CHAPTER 1

INTRODUCTION

According to the American Cancer Society, an estimated 1,529,560 new cancer cases will be diagnosed in the United States in 2010. Cancer is a broad term to describe a widespread, complex, and diverse problem. It is commonly known that cancer is caused by harmful cells that proliferate throughout the body. What is less commonly known is that there is no one cause for all types of cancer. Within a specific cancer type, there are subtypes, and individual cases within each subtype may have different biological causes. Currently, some treatments respond better to certain subtypes because they have been designed to reverse the biological cause of the cancer. For example, in breast cancer, patients who have a subtype characterized by ER+ respond well to tamoxifen, but for patients with ER- the harmful side effects outweigh the benefit from the treatment. If the contributing biological factors in a specific cancer could be determined, the patient could initially receive the most effective treatment, increasing the chance of remission. The purpose of this paper is to describe a method that uses microarray data from a single deregulated oncogenic pathway to determine if it is present in a cancer sample. This method is one step in the process of achieving more effective, more personalized treatments.

Before describing the method, a biological background to pathways and microarrays must be established. First, biological pathways will be defined, followed by a description of what happens when the a pathway is deregulated. Next, oncogenic pathways and their role in cancer development will be addressed. One specific oncogenic pathway used to demonstrate this method, the RAS pathway, will then be addressed, followed by a description of microarrays and their application to cancer research. Finally, after the background has been established, a brief overview of the UPC method will be described.

Biological Pathways

A biological pathway is a series of actions among molecules that lead to a certain product or change in a cell, much like a car traveling from point A to point B, with a series of stoplights and turns in between (Figure 1.1). Biological pathways range from processes at the molecular level, such as degrading RNA and producing proteins, and those at the cellular level, like apoptosis (cell death), to processes that affect the whole body, such as regulating body temperature and hemostasis.

When a pathway is deregulated at least one step in the process is altered, causing a chain reaction that can have detrimental effects on the body. Rather than the pathway going from point A to point B, result C is observed (Figure 1.1). For example, hemoglobin is a protein responsible for binding to oxygen, allowing the air we breathe to enter the blood stream. However, if there is one mutation on one chromosome, one protein is misshapen and can cause sickle cell anemia. More generally, the cause of a deregulated pathway goes back to some error in the DNA. Any given deregulated pathway can cause a multitude of problems, including a different protein being produced or a gene being "turned on" when it should be "turned off". Whatever the specific changes caused by a deregulated pathway, the ramifications are harmful to the body.

Oncogenic Pathways

One of the critical kinds of pathways in cancer research are oncogenic pathways. Oncogenic pathways promote cell differentiation (reproduction). When an oncogenic pathway is deregulated, harmful cells remain alive or healthy cells do not divide. Many cases of cancer are caused by deregulated oncogenic pathways; cancerous cells do not experience apoptosis (cell death). Rather, they stay alive and continue to reproduce, causing the growth and spreading of cancer.

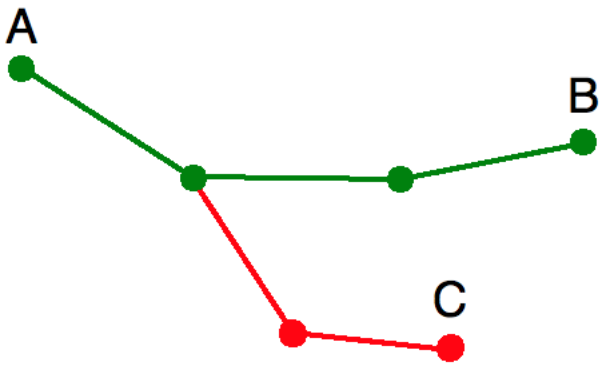

Figure 1.1: If a biological pathway is deregulated, it will go a different direction, from A to C, rather than rather than going on its intended route, from A to B. Deregulated biological pathways cause problems in the body, the most extreme cases being life threatening diseases including sickle cell anemia and cancer.

The RAS pathway is one example of an oncogenic pathway that is responsible for transmitting signals to the cell, among other things. A signal is sent from outside the cell, received by the RAS protein, and passed from protein to protein to regulate lipid metabolism, DNA synthesis, and cytoskeletal organization, which are processes necessary for cell differentiation. The amount of the RAS protein is carefully managed; however, if there is a single nucleotide mutation, the RAS protein could possibly signal to the cell to differentiate unchecked. This is a significant problem in cancerous cells. In fact, the deregulated RAS pathway is present in 20% of all tumors, 90% of pancreatic cancers, and 35% of lung cancers. Certain drugs have been shown effective in treating the RAS mutation, including the inhibitor of farnesyltransferase which blocks the maturation of RAS (Goodsell 1999).

Microarrays

One of the challenges in pathway analysis is quantifying the active pathways. Determining the specific gene mutations causing the deregulated pathway is difficult with current technology. However, the overall changes in gene expression caused by the deregulated pathway can be observed. For example, if a deregulated pathway causes a different protein to be

made, it would be characterized by an increase in the expression of that specific protein. Little is known about the specific genetic changes resulting from the deregulated pathways, but by comparing control cells to cells where a certain pathway has been over-expressed, the genes affected by the deregulated pathway can be determined. The genes' expression values are measurable using microarrays.

Microarrays are a genetic tool that calculate measurements on thousands of RNA strands simultaneously, indirectly allowing us to access gene expression values. RNA is purified from a tissue sample and bound to fluorescent dye. One-channel arrays use only one color of fluorescent dye while two-channel arrays use two dyes. The RNA strands are then placed onto a small chip, and the concentration of the dye, an indication of the gene expression, is measured. In a given cell, there will be genes that are active (expressed) and background genes that are inactive (unexpressed).

While microarrays are useful, they are not perfect. Microarrays only yield the relative expression levels and therefore cannot determine if a given the gene is expressed or unexpressed because it does not have a "control". Additionally, microarray data are noisy due to known and unknown biases, including the nucleotide composition of the gene and natural variation.

In order to use the microarray data in pathway analysis, normalization that removes known biases must be applied to the data. The MMAX normalization will be presented, justified, and applied to the deregulated RAS pathways and control samples. Once the data are normalized using MMAX, they are summarized to determine the probability that each gene is expressed. Then the steps to compute the UPC, the profile of the deregulated pathway, are outlined. This portion yields a probability that a single pathway is activated. The UPC method is described and illustrated using the RAS pathway and 51 lung cancer samples. Overall, 45 cancer samples were classified as having a deregulated RAS pathway, 2 were marginal, and 4 did not have a deregulated RAS pathway. The UPC method was applied to the RAS pathway as a demonstration but can be applied to any single oncogenic

pathway. Hopefully, the UPC method will contribute to the growth and development of targeted cancer treatment.

CHAPTER 2

LITERATURE REVIEW

Cancer pathway identification using microarrays requires two major tiers of analysis. This section will provide an overview of the literature that addresses the main components of the analysis: (1) one-channel microarray normalization and summarization and (2) pathway profiling and signature analysis.

2.1 Normalization and Summarization of One-Channel Microarrays

Error and bias in microarray data from the DNA composition of the probe and batch effects necessitate the normalization of the data before summarization and analysis. One of the first methods of one-channel normalization was global median normalization (Edwards 2003). This method is not considered rigorous enough compared to the newer methods, including quantile normalization (Bolstad et al. 2003). Another normalization method that performs well for tiling arrays in ChIP-chip experiments is the model-based approach presented by Johnson et al. (2006). Four major methods of microarray correction, normalization, and summarization exist: Microarray Analysis Suite 5.0 (Hubbell et al. 2002); Model Based Expression Index, MBEI (Li and Wong 2001); Robust Multichip Analysis, RMA (Irizarry et al. 2003); and Significance Analysis of Microarrays, SAM (Tusher, Tibshirani, and Chu 2001). Barcoding (Zilliox and Irizarry 2007) and a mixture model approach (Parmigiani, Garrett, Anbazhagan, and Gabrielson 2002) will also be presented.

Global Median Normalization

Global median normalization, mentioned by Edwards (2003), was one of the first approaches to normalize one-channel arrays. This method subtracts the background intensity from the

signal intensity,

$$i_{ps} = i^s_{ps} - i^b_{ps}, \tag{2.1}$$

where i^s_{ps} represents the signal intensity and i^b_{ps} is the background intensity. One of the disadvantages of this method is that it allows a negative intensity, preventing the necessary log transformation. The negative values are treated as missing, removing valuable information and introducing bias. Edwards (2003) presents a method that improves global normalization by accounting for negative and very small intensity values; however, global median normalization is still considered an inferior method.

Quantile Normalization

The main objective of quantile normalization, presented by Bolstad, Irizarry, Astrand, and Speed (2003), is to transform the values on each array so the distributions are identical. In order to perform this normalization, the following algorithm is used:

1. Given n arrays of length p, form X of dimension $p \times n$ where each expression array is a column;

2. Sort each column of X to give X_{sort};

3. Take the means across rows of X_{sort} and assign this mean to each element in the row to get X'_{sort};

4. Get $X_{normalized}$ by rearranging each column of X'_{sort} to have the same ordering as original X.

Bolstad et al. (2003) concedes that one of the major limitations of this method is that it may misrepresent the values in the tails. Generally, in practice, this has been problematic because genes that are highly expressed are usually of great interest. Another disadvantage to this method is that it does not account for the bias due to the nucleotide composition of the probe.

Model-Based Normalization

Johnson et al. (2006) present a model-based approach for normalizing tiling arrays. The MAT model for the log transformed expression value, $Y_m \sim N(X\theta_m, \sigma_m^2)$, is shown in Equation 2.2.

$$x_i\theta_m = \alpha_m n_{iT} + \sum_{j=1}^{25} \sum_{k\in\{A,C,G\}} \beta_{jkm} I_{ijk} + \sum_{l\in\{A,C,G,T\}} \gamma_{lm} n_{ik}^2, \tag{2.2}$$

where n_{ik} is the nucleotide k count in probe i, α_m is the baseline value based on the number of T's on the probe, I_{ijk} is an indicator function such that $I_{ijk} = 1$ if the nucleotide at position j is k in probe i, β_{jkm} is the effect of each nucleotide k (except T which is already modeled in α) at each position j, and γ_{lm} is the effect of nucleotide count squared. Unlike the methods previously mentioned, this model removes bias by accounting for the probe composition. While Johnson et al. (2006) presented the method only for tiling arrays, Kapur, Xing, Ouyang, and Wong (2007) extended this model to Exon arrays, showing its superiority to the Affymetrix GC bin background model.

Microarray Analysis Suite 5.0 (Mas5)

The algorithm used by Affymetrix computes the signal in the following algorithm,

$$signal = \text{Tukey Biweight}(\log(PM_j - CT_i)), \tag{2.3}$$

where CT_i is a function of the MM probes. The discriminant score, $R_i = \frac{PM_i - MM_i}{PM_i + MM_i}$, is then computed for each probe. Once these values are computed, the Wilcoxon Signed Rank Test is used to calculate the p-value for each probe pair. The p-value is then compared to pre-defined significance levels, described below:

- Present if p-value $< \alpha_1$
- Marginal if $\alpha_1 \leq p$-value $< \alpha_2$
- Absent if $\alpha_2 \leq p$-value.

The defaults for α_1 and α_2 are 0.04 and 0.06, respectively. This summarization is common, especially among biologists who rely on the commercial processing performed by Affymetrix. One difficulty with this method is deciding whether to include marginal genes as expressed or unexpressed (Hubbell et al. 2002).

Model Based Expression Index (MBEI)

Li and Wong (2001) take a model based approach, beginning with a non-linear baseline array normalization. Next, they assume PM-MM for each individual probe response follows the model,

$$y_{ij} = PM_{ij} - MM_{ij} = \theta_i \phi_j + \epsilon_{ij}, \tag{2.4}$$

where θ_i is the expression index in chip i, and j indicates the probe pair. In order for the model to be estimable, the constraint $\sum \phi_j^2 = J$, the number of probes, is imposed. The parameters are estimated by iterating between ϕ and θ. This model allows for a straightforward way to identify the outlier probes and arrays. Once the outliers are removed, the probes within each probeset are then summarized using the model,

$$\log_2 (y_{ij}) = \beta_j + \alpha_i + \epsilon_{ij}, \tag{2.5}$$

where α_i is the probe effect and β_j is the $\log_2$ transformed expression values. A separate model is fit for each probeset.

Robust Multi-array Average (RMA)

Irizarry et al. (2003) also mention a way to normalize one channel Affymetrix GeneChip® arrays. After extensive data exploration, they developed robust multi-array average (RMA), a model that corrects for the background noise, normalizes using quantile normalization (Bolstad et al. 2003), and fits the linear model to the normalized, $\log_2$ transformed intensities. The algorithm is outlined in more detail below.

1. Model each intensity (PM) using $PM_{ijn} = bg_{ijn} + s_{ijn}$, where bg_{ijn} is the background in the i^{th} array, and s_{ijn} is the intensity signal for the i^{th} array, j^{th} probe pair number, and n^{th} probe set. Assuming s_{ijn} is exponentially distributed and bg_{ijn} is normally distributed, calculate $B(PM_{ijn}) = E(s_{ijn}|PM_{ijn})$, and impose the restriction $B(PM_{ijn}) > 0$.

2. Normalize data using quantile normalization

3. Apply a $\log_2$ transformation to the now normalized data

4. Fit the $\log_2$ transformed, normalized, background-adjusted values using the additive model:

$$Y_{ijn} = \mu_{in} + \alpha_{jn} + \epsilon_{ijn} \tag{2.6}$$

   where Y is the transformed, background-corrected values, α_j is the probe affinity effect, μ_i is the log scale expression for the i^{th} array, and ϵ_{ijn} is the error. Note $\sum_j \alpha_j = 0$ and ϵ_{ijn} are independently and identically distributed with mean 0.

5. In order to minimize the effect of outlier probes, use median polish or another robust procedure to estimate parameters.

6. Use the estimates of μ_i as the log_2 measure of expression, referred to as the RMA.

RMA performance is comparable to MBEI performance but is generally considered better and is more widely used in practice.

Significance Analysis of Microarrays (SAM)

SAM operates under the assumption that each gene has a different amount of variation (Tusher et al. 2001). Each gene is assigned a score based on the relative values

$$d(i) = \frac{\bar{x}_I(i) - \bar{x}_U(i)}{s(i) + s_0}, \tag{2.7}$$

where $\bar{x}_I(i)$ and $\bar{x}_U(i)$ are the average levels for the i^{th} gene in states I and U, $s(i)$ is the standard deviation of repeated expression measurements, and s_0 is a small positive constant that minimizes the coefficient of variation. The $d(i)$ are then ranked based on their magnitude, and a threshold, $\Delta = 1.2$, is selected. The values are plotted on a normal plot, and lines are drawn Δ units above and below the *observed* = *expected* line. Those observations above and below the threshold lines are deemed significant. This method is superior to both pairwise fold change and fold change methods, and it compensates for the False discovery rate (FDR) using a permutation-based test.

Mixture Model Analysis

The framework established by Parmigiani, Garrett, Anbazhagan, and Gabrielson (2002) assumes the microarray data come from a three component mixture model (Figure 2.1). Genes that are under expressed and over expressed, the lowest and highest of the three components, are uniformly distributed, and genes that are normally expressed are normally distributed. Each component has its corresponding parameters which are estimated using Bayesian hierarchical analysis. This approach yields the probability that a given expression value is from a distribution. For a given gene, if the probability is largest for the under-expressed, it is assigned -1. Similarly, those genes from the normally-expressed and over-expressed genes are assigned 0 and 1, respectively. The microarray summarization presented in this paper has a useful application to signature analysis, which will be discussed later. One disadvantage to this method is that it does not account for the probe bias, both from the DNA makeup and from dead probes.

Barcoding

Zilliox and Irizarry (2007) introduce a novel method to determine if a gene is expressed or unexpressed. Using data from previous microarray experiments in the GEO database, information on the behavior of certain genes was obtained. More specifically, the distribution

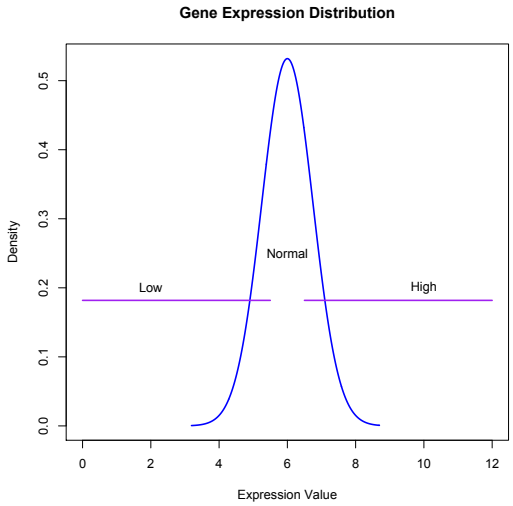

Figure 2.1: The three component mixture model used by Parmigiani et al. (2002) has three distributions, two uniform and one normal. For any given expression value, the probability it belongs to each distribution can be found.

of each gene was computed. For each gene, a cutoff value was selected based its distribution (Figure 2.2). If the probe expression was above the cutoff, it was labeled "on" and assigned a 1, and if it was below the cutoff, it was assigned a 0. For a given tissue, they compiled the individual codes to create a barcode. This has interesting applications to pathway analysis that will be mentioned later. Zilliox and Irizarry (2007) have extended their method to select the cutoff based on statistical methods rather than select an arbitrary value. However, their results have yet to be published.

Initially, it appears a binary classification would result in the loss of valuable information. However, Tuna and Niranjan (2009) justify the reduction of measurement precision to the binary level, showing that a binary reduction is essentially equivalent to using the expression levels. Their argument is supported by their evidence that information can be recovered using specific higher dimensional binary clustering algorithms. Other research groups have implemented the barcoding approach, supporting it as a powerful method for gene expression. Dudley, Tibshirani, Deshpande, and Butte (2009) even applied it across microarrays from different labs and tissues. However, barcoding requires the combination

of vast data sets, and while there are publicly available databases of data, it is a tedious process to compile the expression distributions for each gene. Hence, the method only works for a limited set of microarray platforms. Additionally, the growth of custom designed arrays limits the application of this method to many experiments.

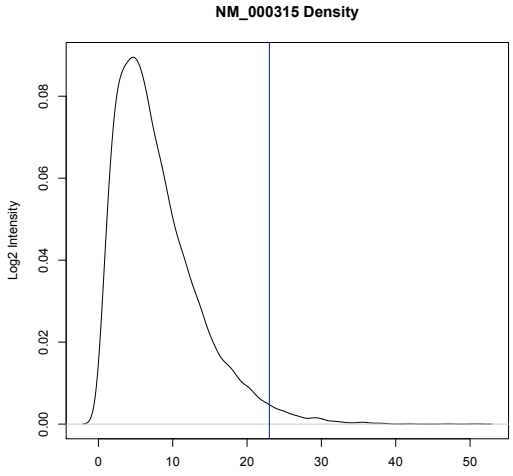

Figure 2.2: This density plot is an example of a probe distribution used to barcode microarray data (Zilliox and Irizarry 2007). Each gene on the array has its own distribution based on previous experiments from GEO. A cutoff value is selected based on the distribution of each gene, and those genes with values above the threshold are deemed "expressed". Expressed and unexpressed genes are assigned 1 and 0, respectively.

2.2 Pathway Signature Analysis

In order to quantify what is occurring in a cell, microarray data measures thousands of probes. Naturally, with so many contributing genes, an apt way to describe the activity in the cell is to use the measurements from multiple genes rather than one single summary statistic. This measure is called a gene signature of profile. Profiling has been shown to be more effective than simply looking at one gene (Sjoblom et al. 2006; Wood et al. 2007). While there are different methods of gene selection, profiling is a standard procedure in cancer research. In fact, gene profiles of cancer samples are commonly used both to identify new subtypes of cancer and to diagnose new cancer samples. Some of the first studies to use

this technology include Bittner et al. (2002) and Alizadeh et al. (2002). Since then, the use of profiles has become commonplace. More current studies using profiling include an overview of breast cancer profiling (Geyer and Reis-Filho 2009) and lung cancer profiling (Boutros et al. 2009). These studies rely on cancer samples to create profiles. While this helps identify similarities, the effects of the cancer are confounded with the biological change that caused the cancer. In order to separate these two effects, a few studies have used deregulated oncogenic pathway profiles and projected them into cancer samples to determine the cause of the cancer. The first landmark study to use oncogenic pathways was Ferrando et al. (2002). Two additional studies that used separate data from deregulated pathways, Bild et al. (2006) and Chang et al. (2009), are also summarized below. For both types of profiles, the adaptation of barcoding (Zilliox and Irizarry 2007) to pathway analysis and the marginal profiles presented by Parmigiani et al. (2002) are useful approaches. Both methods allow for the projection of a new cancer sample into a profile to determine the probability that a given sample is a certain subtype or contains a deregulated pathway.

Cancer Sample Profiling

Bittner et al. (2002) used three different clustering approaches to classify 31 cutaneous melanoma tumors, and Alizadeh et al. (2002) used hierarchical clustering to classify non-Hodgkin's lymphoma. Both groups found the microarray classification better than histological classification.

Recently, Geyer and Reis-Filho (2009) published a literature review of recent studies concerning breast cancer signature profiles. The paper summarizes the work of more than ten research groups, each who have developed a unique gene signature to determine the optimal prognosis. The signatures range from 21 to 186 genes. While these signatures may prove to be helpful, one major drawback is the lack of consistency between signatures. When subsets of the signatures were combined in a model, they did not perform any better than the individual signatures did on their own, perhaps because of a lack of sophisticated

statistical methods. Boutros et al. (2009) found a gene signature of only 6 genes to identify non-small-cell lung cancer using a permutation approach. They also justified the existence of many unique signatures to classify the same type of cancer.

Profiling, combined with clinical information, has been utilized in clinical situations successfully to improve diagnosis and prognosis. Winstead (2008) is one example of the successful implementation of this method. The model they developed uses both clinical and genetic markers from bronchoscopy to predict early stage lung cancer, making it much less invasive than the previous methods. Up to that point, there was great difficulty identifying early stage lung cancer.

Deregulated Oncogenic Pathway Profiling

One of the landmark studies in this field, Ferrando et al. (2002), took samples from patients with T-cell acute lymphoblastic leukemia. Using hierarchical clustering, they were able to identify a critical pathway in the development of a subtype of the cancer.

Using cancer cells to quantify cancer profiles is limited in its ability to identify what is biologically causing the cancer, thus hindering the development of targeted treatments. However, quantifying a known biological pathway and determining if it is contributing to a cancer sample provides a way to pinpoint a more effective treatment. Bild et al. (2006) take this approach. Certain pathways, including the RAS pathway, were amplified and put onto a microarray and compared to the control cells. The signature was calculated using principal components. Then, breast cancer samples were compared to the signature to see which had a specific pathway deregulated. Using hierarchical clustering, they also developed a profile that combined the signatures from known multiple pathways and had more success than the individual pathways. A more recent paper, Chang et al. (2009), focused on how to determine if multiple pathways are deregulated using Bayesian factor analysis. In this method, X is the matrix of the gene expression values $(n \times m)$; m is number of samples, n is number of genes. $X = AY + E$, where A is a sparsely defined matrix indicating which genes

are in the signature ($n \times k$, k is number of signatures) and the defining weights between gene signature pairs, and Y is a $k \times m$ matrix of of the scores of signatures across data set. E is the error. According to the paper, k is estimated "statistically", and the number of genes in a signature is allowed to vary. The algorithm iterates through the factor decomposition and a step that searches for other useful genes. This iteration yields a set of genes with estimated weights. Then the resulting weights are applied to new sample of expression values (this is the score): if the score is high, the activation level is high; if the score is low, the level is low.

Profile Projection

As mentioned before, Parmigiani et al. (2002) published a unique profile for multiple genes based on expression, using -1, 0, or 1 for low, normal, and high expression values, respectively. In order to find the most likely profile for a given gene, g, they found three quantities: $P(G = g|\text{low})$, $P(G = g|\text{normal})$, and $P(G = g|\text{high})$. These probabilities were used in calculating the marginal profiles. For example, if we were developing a three gene signature, the marginal profiles would be:

Profile ID	Gene 1	Gene 2	Gene 3
1	1	1	1
2	0	1	1
3	-1	1	1
4	1	0	1
5	0	0	1
6	-1	0	1
⋮	⋮	⋮	⋮
25	1	-1	-1
26	0	-1	-1
27	-1	-1	-1

Because the genes are assumed to be independent, the joint probability for a given profile is the product of the probability for each gene. From this, it is easy to find the profile with the highest probability. Additionally, the probability that a given cancer sample has a specific gene signature can be obtained.

The barcoding method developed by Zilliox and Irizarry (2007) can also be applied to signatures. As mentioned before, each probe is assigned a 0 or 1 based on its expression level. Then, a profile of 0s and 1s, the barcode, can be created from the selected genes.

CHAPTER 3

METHODS

Now that the literature has been summarized for this specific area of research, a method will now be presented to determine if a single deregulated pathway is active in a given cancer sample using a novel normalization method called Mixture Model Based Analysis of Expression Model (MMAX) and the pathway's genetic signature called the Universal Probability of expression Code (UPC). More specifically, Determining the probability that a pathway is expressed is a three step process:

1. First, the microarray data from the activated pathway are normalized and summarized. This will be accomplished using the Mixture Model Based Analysis of Expression Arrays (MMAX) normalization, which is presented and validated. This method will output the probability of expression for each gene.

2. Second, the UPC is calculated for an active pathway. The UPC will quantify the active pathway by selecting genes that are the most different between the activated samples and the control samples.

3. Third, the pathway is projected into a new cell to determine if it contributes to the cancer sample. In other words, the UPC from the active pathway will be compared to the cancer sample.

After each step is presented in more detail, the entire method will be demonstrated using the RAS pathway and 51 cancer samples.

3.1 Data Normalization and Summarization

Mixture Model Based Analysis of Expression Model (MMAX)

There are many sources of error in microarray data. Additionally, in a given sample, there will be many genes that are functioning at their normal level. These genes will contribute to the background noise, making it hard to determine which genes are expressed beyond the background value. The values obtained from the array for these expressed genes are the combined effects of the background and signal. The ultimate goal of MMAX is to remove the *background* noise, leaving only the quantity of interest: the *expressed* signal. With this in mind, MMAX begins with the assumption that the log of the expression values come from a two-component mixture model. The components represent the *background* and *expressed* (*background + signal*) distributions. More formally, we assume the log of the microarray data follow the following distribution:

$$Y = \Delta Y_E + (1 - \Delta) Y_B \tag{3.1}$$

where the $\Delta \sim Bernoulli(\pi)$, which implies $Pr(\Delta = 1) = \pi = P(Y_i = expressed)$. Additionally, each individual component follows the MAT distribution Equation 2.2: $Y_E \sim N(x\beta_1, \sigma_1^2)$ and similarly, $Y_B \sim N(x\beta_2, \sigma_2^2)$ (Johnson et al. 2006).

The standard approach for maximum likelihood estimation is to take the complete log likelihood and maximize the parameters, Θ, with respect to the complete data, T, y and Δ. However, Δ is unobserved, necessitating the use of the EM algorithm. This algorithm is commonly used in problems with unobserved data, Z, and problems with difficult maximum likelihood computations. Rather than impute unobserved data, it integrates over the missing data. More specifically, it maximizes over Q, the expected value of the of the complete data log likelihood. Mathematically, $Q(\Theta; \hat{\Theta}^{(j)}) = E[\ell_0(\Theta'; T)|\hat{\Theta}^{(j)}]$, where T is the combined data (observed and unobserved), $\hat{\Theta}^{j-1}$ is the most current estimate of Θ, and j indicates the iteration number.

Overall, the EM algorithm has two basic steps: Expectation (E) and Maximization (M). The two steps iterate until convergence. The algorithm, adapted from the general form in Hastie et al. (2001), is outlined below.

1. Select initial values $\hat{\Theta}^0$ for $\Theta = (\pi, \beta_1, \beta_2, \sigma_1^2, \sigma_2^2)$.

2. (Expectation) Calculate the expected value of the log likelihood function for the conditional distribution of the missing data, $Q(\Theta; \hat{\Theta}^{(j)}) = E(\ell_0(\Theta'; T)|Z, \hat{\Theta}^{(j)})$, using the most current estimate of the parameters, $\hat{\Theta}^{j-1}$. In order to find Q in this specific context, the first step is to find the likelihood. Because Δ is an indicator, we can rewrite Equation 3.1 in this equivalent way

$$f(y|\Theta) = f_1(y)^{\Delta} f_2(y)^{(1-\Delta)}.$$

Also, recall

$$f_1(y) = (2\pi\sigma_1^2)^{-1/2} exp\left(-\frac{1}{2\sigma_1^2}(y - x\beta_1)'(y - x\beta_1)\right),$$

and similarly,

$$f_2(y) = (2\pi\sigma_2^2)^{-1/2} exp\left(-\frac{1}{2\sigma_2^2}(y - x\beta_2)'(y - x\beta_2)\right).$$

From the properties of conditional probability, the joint distribution is given by $f(y, \Delta|\Theta) = f(y|\Delta, \Theta)f(\Delta|\Theta)$. Substituting in the Bernoulli density function for $f(\Delta)$, the likelihood becomes

$$L(y, \Delta|\Theta) \quad \propto \quad \prod_{i=1}^{n} f_1(y_i)^{\Delta_i} f_2(y_i)^{1-\Delta_i} \pi^{\Delta_i} (1-\pi)^{1-\Delta_i}.$$

Let Δ_a be a diagonal matrix so that the ii^{th} element is Δ_i, and let Δ_b be a diagonal matrix so that the ii^{th} element is $1 - \Delta_i$. Inserting $f_1(y)$ and $f_2(y)$, the likelihood can

be written as:

$$L(\Theta|y,\Delta) = \pi^{\mathrm{tr}(\Delta_a)}(1-\pi)^{\mathrm{tr}(\Delta_b)}\frac{\exp\left(\frac{-1}{2\sigma_1^2}(y_i - X\beta_1)'\Delta_a(y_i - X\beta_1)\right)}{(2\pi\sigma_1^2)^{\mathrm{tr}(\Delta_a)/2}} \times \frac{\exp\left(\frac{-1}{2\sigma_2^2}(y_i - X\beta_2)'\Delta_b(y_i - X\beta_2)\right)}{(2\pi\sigma_2^2)^{\mathrm{tr}(\Delta_b)/2}}.$$

The next step to find Q is to find the log likelihood:

$$\begin{aligned}\ell(\Theta|y) = {} & \mathrm{tr}(\Delta_a)\log(\pi) + \mathrm{tr}(\Delta_b)\log(1-\pi) + \\ & \left(\frac{-1}{2\sigma_1^2}(y_i - X\beta_1)'\Delta_a(y_i - X\beta_1)\right) + \\ & \left(\frac{-1}{2\sigma_2^2}(y_i - X\beta_2)'\Delta_b(y_i - X\beta_2)\right) - \\ & \mathrm{tr}(\Delta_a)\log(2\pi\sigma_1^2)/2 - \mathrm{tr}(\Delta_b)\log(2\pi\sigma_2^2)/2.\end{aligned}$$

Finally, the expected value of the log likelihood is computed, leaving:

$$\begin{aligned}Q = \mathrm{E}[\ell(\Theta; y, \Delta)|\hat{\Theta}^j] = {} & \mathrm{E}\left[\mathrm{tr}(\Delta_a)\log(\pi) + \mathrm{tr}(\Delta_b)\log(1-\pi) + \right. \\ & \left(\frac{-1}{2\sigma_1^2}(y_i - X\beta_1)'\Delta_a(y_i - X\beta_1)\right) + \\ & \left(\frac{-1}{2\sigma_2^2}(y_i - X\beta_2)'\Delta_b(y_i - X\beta_2)\right) - \\ & \left.\mathrm{tr}(\Delta_a)\log(2\pi\sigma_1^2)/2 - \mathrm{tr}(\Delta_b)\log(2\pi\sigma_2^2)/2\right] \\ = {} & \log(\pi)\mathrm{E}[\mathrm{tr}(\Delta_a)] + \log(1-\pi)\mathrm{E}[\mathrm{tr}(\Delta_b)] + \\ & \left(\frac{-1}{2\sigma_1^2}(y_i - X\beta_1)'\mathrm{E}[\Delta_a](y_i - X\beta_1)\right) + \\ & \left(\frac{-1}{2\sigma_2^2}(y_i - X\beta_2)'\mathrm{E}[\Delta_b](y_i - X\beta_2)\right) + \\ & \log(2\pi\sigma_1^2)\mathrm{E}[\mathrm{tr}(\Delta_a)]/2 + \log(2\pi\sigma_2^2)\mathrm{E}[\mathrm{tr}(\Delta_b)]/2.\end{aligned}$$

Since Δ_i is unobservable, $\mathrm{E}(\Delta_i) = \gamma_i$ is estimated using the following function:

$$\gamma_i = \frac{\hat{\pi}^{(j-1)} f_1^{(j-i)}(y_i)}{(1-\hat{\pi}^{(j-1)}) f_2^{(j-i)}(y_i) + \hat{\pi}^{(j-1)} f_1^{(j-i)}(y_i)},$$

and is updated at each iteration for updated values of the parameters. Note that this estimation would be difficult if the likelihood did not contain linear functions of Δ. γ_i is the probability that $\Delta_i = 1$, or the probability that y_i is from the *expressed* distribution given that the maximum likelihood estimates for the parameters are set at the most recent value in the algorithm.

3. (Maximization) In the maximization step, the maximum likelihood estimates for Θ are updated using the current value of γ_i. The maximum likelihood estimates for Θ are derived below. Recall:

$$\begin{aligned}\ell(\Theta|y) &= \mathrm{tr}(\Delta_a)\log(\pi) + \mathrm{tr}(\Delta_b)\log(1-\pi) + \\ &\left(\frac{-1}{2\sigma_1^2}(y_i - X\beta_1)'\Delta_a(y_i - X\beta_1)\right) + \\ &\left(\frac{-1}{2\sigma_2^2}(y_i - X\beta_2)'\Delta_b(y_i - X\beta_2)\right) - \\ &\mathrm{tr}(\Delta_a)\log(2\pi\sigma_1^2))/2 - \mathrm{tr}(\Delta_b)\log(2\pi\sigma_2^2)/2.\end{aligned}$$

The partial derivative is derived and maximized for each parameter. The maximum likelihood estimate for π is performed first:

$$\begin{aligned}\frac{\delta\ell}{\delta\pi} &= \frac{\mathrm{tr}(\Delta_a)}{\pi} - \frac{\mathrm{tr}(\Delta_b)}{1-\pi} \\ \frac{\delta\ell}{\delta\pi} &= \frac{\mathrm{tr}(\Delta_a)}{\pi} - \frac{n - \mathrm{tr}(\Delta_a)}{1-\pi} \\ 0 &= \frac{\mathrm{tr}(\Delta_a)}{\hat{\pi}} - \frac{n - \mathrm{tr}(\Delta_a)}{1-\hat{\pi}} \\ &\Rightarrow \quad \hat{\pi} = \frac{\sum \Delta_a}{n}.\end{aligned}$$

Next, the maximum likelihood estimate for β_1 is found:

$$\begin{aligned}
\frac{\delta \ell}{\delta \beta_1} &= \frac{\delta}{\delta \beta_1}(y - x\beta_1)'\Delta_a(y - x\beta_1) \\
&= \frac{\delta}{\delta \beta_1}\left(y'\Delta_a y - \beta_1' x'\Delta_a y - y'\Delta_a x\beta_1 + \beta_1' x'\Delta_a x\beta_1\right) \\
&= -2x'\Delta_a y + 2x'\Delta_a x\beta_1 \\
0 &= -2x'\Delta_a y + 2x'\Delta_a x\hat{\beta}_1 \\
&\Rightarrow \; x'\Delta_a y = x'\Delta_a x\hat{\beta}_1 \\
&\Rightarrow \; \hat{\beta}_1 = (x'\Delta_a x)^{-1}x'\Delta_a y.
\end{aligned}$$

Similarly, $\hat{\beta}_2 = (x'\Delta_b x)^{-1}x'\Delta_b y$. Next, the maximum likelihood estimate for σ_1^2 derived.

$$\begin{aligned}
\frac{\delta \ell}{\delta \sigma_1^2} &= \frac{(y - x\beta_1)'\Delta_a(y - x\beta_1)}{2\sigma_1^4} + \frac{-\mathrm{tr}(\Delta_a)}{\sigma_1^2} \\
0 &= \frac{(y - x\hat{\beta}_1)'\Delta_a(y - x\hat{\beta}_1)}{2\hat{\sigma}_1^4} = \frac{\mathrm{tr}(\Delta_a)}{\hat{\sigma}_1^2} \\
&\Rightarrow \; \hat{\sigma}_1^2 = \frac{(y - x\hat{\beta}_1)'\Delta_a(y - x\hat{\beta}_1)}{\mathrm{tr}(\Delta_a)},
\end{aligned}$$

and similarly, $\hat{\sigma}_2^2 = \frac{(y - x\hat{\beta}_2)'\Delta_b(y - x\hat{\beta}_2)}{\mathrm{tr}(\Delta_b)}$.

The expectation and maximization step are repeated until the parameter estimates converge. MMAX outputs probability of expression, γ_i, for each gene. This summarization of the data is critical for the next step of the analysis.

Model Validation

In order to test the performance of MMAX, it was compared to Barcoding (Zilliox and Irizarry 2007) and Afffymetrix's Mas5 (Hubbell et al. 2002) using a data set from Affymetrix where the true expression was known, making it comparable to a simulation study. When compared to other established methods, Affymetrix's PMA calls and Barcoding, MMAX performed similarly, if not better, at determining the probability of expression.

The data utilized for this comparison are from the Human Genome U133 array, one of two data sets in Affymetrix's Latin Square Experiment for Expression Algorithm Assessment.

As the title implies, this data set will allow us to assess the performance of the algorithm since the true expression values are known. According to their website, *affymetrix.com*, the data set has 3 replicates for 14 hybridizations of 42 spiked transcripts that range from 0.125 picoMolars (pM) to 512 pM. Only the values with a concentration of 0.125 pM to 32 pM will be utilized. In this situation, genes with a concentration above 32 pM did not provide additional information about the performance abilities of the three methods considered. We would expect to see a gene considered *expressed* once there is any concentration of RNA. However, it would be useful to know which genes have a higher concentration than others because it allows us a more accurate comparison between levels: if a gene has a concentration of 32 pM, it would be more similar to a gene with 24 pM concentration than a 0.125 pM concentration, even though technically both are *expressed*; we would consider a gene with a concentration of 32 pM to be more expressed than a gene with 0.125 pM. Retaining the quantitative expression about how much a gene is expressed is the ideal result.

All three methods were applied to the data set mentioned above to find the probability of expression for each. The probability of expression for a given gene using Barcoding is either 1 or 0 based on the findings of Zilliox and Irizarry (2007). For Mas5, the *mas5* package in R was utilized. The package outputs the call, either present, marginal, or absent, for each gene. Then, the probabilities were assigned: 1 to present, 0.5 to marginal, and 0 to absent. It is important to note that the Mas5 method is not typically extended to include the probability of expression. The third method, MMAX, outputs the probability of expression. These values are presented in Figure 3.1 and Table 3.1. Overall, MMAX performs better than Mas5 and Barcoding at detecting the increase in the concentration. While Barcoding is more accurate in the lowest values, it does not perform well in the higher concentrations. It is not able to detect the 16 pM and 4 pM concentration even though it detects the 8 pM concentration. Mas5 detects the presence of the concentration well, but is not sensitive to the increase: the probability of expression is 1 from 2 pM to 32 pM. In general, MMAX performs as well as, if not better, than both Barcoding and Mas5. Additionally, MMAX

preserves the quantitative information in the different concentrations, which is advantageous. Rather than lose information by reducing the gene expression to binary values, this method allows us to retain the information that a gene with a higher concentration is more highly expressed than a gene with a much lower concentration.

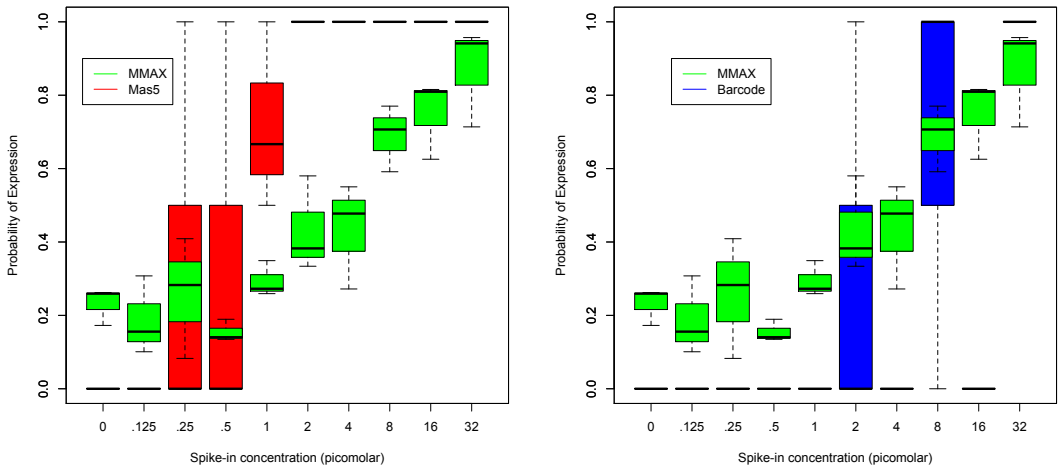

Figure 3.1: This figure shows the probability of expression for the three methods using the Affymetrix Latin Square data. MMAX performs as well, if not better than both methods. MMAX also retains the quantitative information regarding the level of expression for each gene.

Table 3.1: This table contains a summary of the probability of expression computed for the three methods. MMAX gradually increases to 1, which is the desired outcome. Barcoding performs poorly: at 4 pM and 16 pM it does not detect any gene expression. Mas5 does increase but not as steadily as MMAX, losing some of the information about the level of expression.

	Average Probability of Expression		
Concentration (pm)	MMAX	Barcode	PMA
0	0.23	0.00	0.00
0.125	0.19	0.00	0.00
0.25	0.26	0.00	0.33
0.5	0.15	0.00	0.33
1	0.29	0.00	0.72
2	0.43	0.33	1.00
4	0.43	**0.00**	1.00
8	0.69	0.67	1.00
16	0.75	**0.00**	1.00
32	0.87	1.00	1.00

Now that the method is shown to perform comparably to other methods and arguably better than Barcoding, we can proceed with the analysis. The next step is to use the probabilities that are output from MMAX to find a fingerprint or signature of a single active pathway.

3.2 Calculating the Universal Probability of Expression Code

Now that the probability of expression has been obtained for each gene, the next step is to characterize the active pathway by using a small subset (200) of the genes, which is a genetic

signature for a active pathway. We will call this characterization the Universal Probability of Expression Code (UPC). The UPC method is detailed below as well as the results for when it was applied to the active RAS pathway. Note that the justification for selecting 200 genes comes from Sjoblom et al. (2006) and Wood et al. (2007), where they show it is better to use a collection of genes rather than a few genes.

1. The gene probability of expression is calculated for both the active pathway (γ_{ij}) and control cells (δ_{ij}) using MMAX. Note that j indicates the sample and i indicates the gene.

2. The z-statistic is computed to see if the the control and active pathway are different. More formally, we are testing to see if $p_{control} = p_{activepathway}$, for each gene. The p-values are then ranked from lowest to highest and any p-value less than $1e-10$ is considered significant.

3. Those 200 genes with significant p-values that have the largest absolute difference, $a_i = |\bar{\delta}_{i.} - \bar{\gamma}_{i.}|$, are selected, where $\bar{\delta}_{i.}$ is the average of the probability of expression for the control samples and $\bar{\gamma}_{i.}$ is the average probability of expression for the active pathway samples.

4. After the genes are selected, the last step in computing the UPC is to take the average probability of expression, $\bar{\gamma}_{i.}$, for each selected gene. This results in a vector of 200 average probabilities, the UPC.

The justification for the third step comes from the central limit theorem: the average pathway, $\bar{\gamma_{ij}} \sim N(p_\gamma, \frac{p_\gamma(1-p_\gamma)}{n})$. Similarly, the average control is distributed, $\bar{\delta_{ij}} \sim N(p_\delta, \frac{p_\delta(1-p_\delta)}{n})$. We are testing the hypothesis:

$$H_0 : p_\gamma - p_\delta = 0. \tag{3.2}$$

It can be shown that the distribution of $p_\gamma - p_\delta \sim N(0, \frac{p_\gamma(1-p_\gamma)}{n_1} + \frac{p_\delta(1-p_\delta)}{n_2})$. And, because we are using an approximation of the variance, the $t_{n_1+n_2-2}$ distribution will be utilized. We will select the genes with the largest difference between the control and the pathway, which correspond to the smallest p-values.

The UPC method was applied to a published dataset (GEO accession number GSE3151) containing h133+ microarrays for 10 samples of active RAS pathway and 10 control cells (Bild et al. 2006). The analysis is shown, step by step:

1. *The gene probability of expression is calculated for both the active pathway (γ_{ij}) and control cells (δ_{ij})*: The heatmap (Figure 3.2) shows the genes from the active RAS and control samples; there are clear differences in the probability of expression for the two groups. For some genes there is a clear separation: when there is a high probability of expression (red) for control samples, there is a corresponding low expression (yellow) for the active RAS samples.

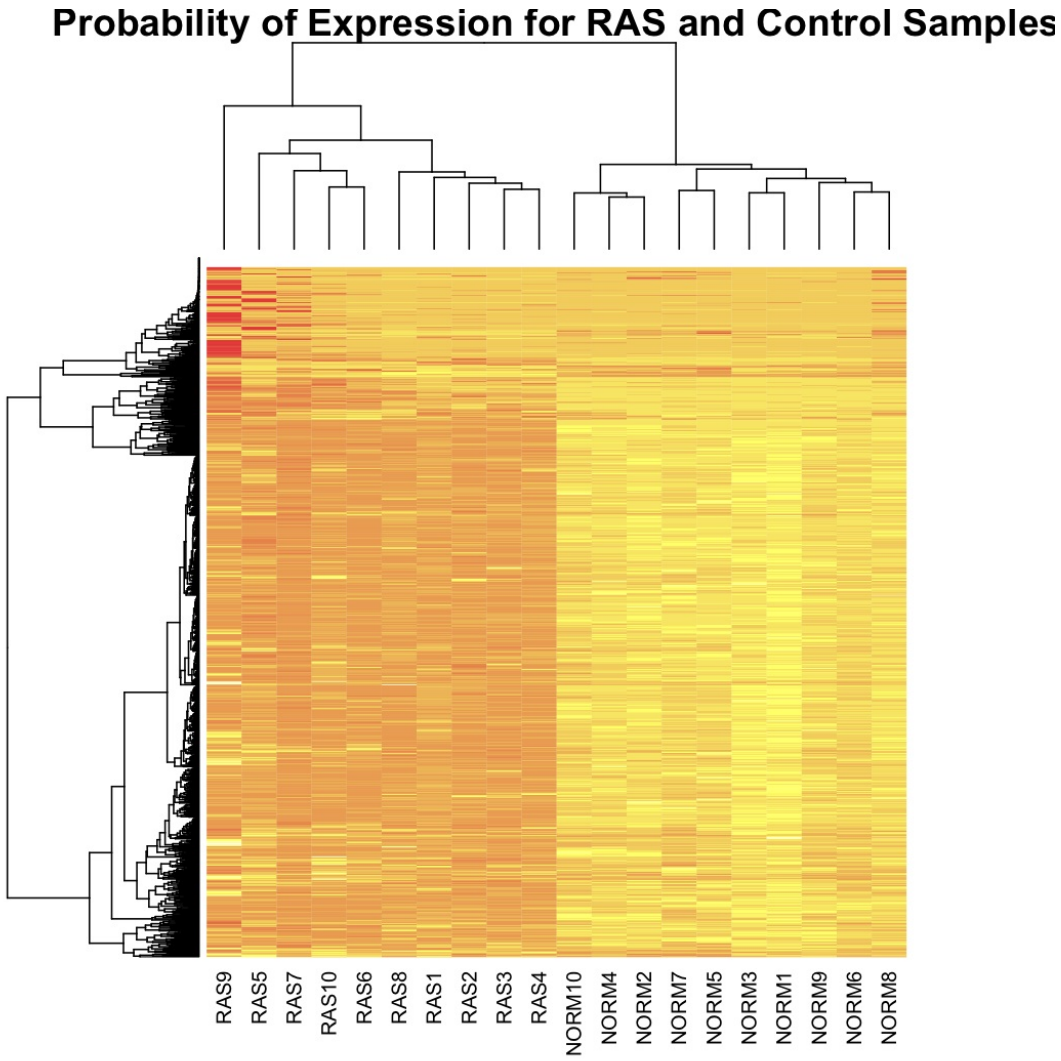

Figure 3.2: The probability of expression was computed for the genes in 10 RAS samples and 10 control samples. This heatmap shows the probability of expression for a random collection of 1000 genes. It is clear that some genes are vey different between the two groups, while other genes are more similar.

2. *The z-statistic is computed to see if the the control and active pathway are different. The p-values are then ranked from lowest to highest.*: The density of the significant absolute differences (p-value $< 10e-12$)is shown in Figure 3.3. The cutoff for the 200

significant genes with the largest average difference is approximately 0.8794, indicated by the vertical line.

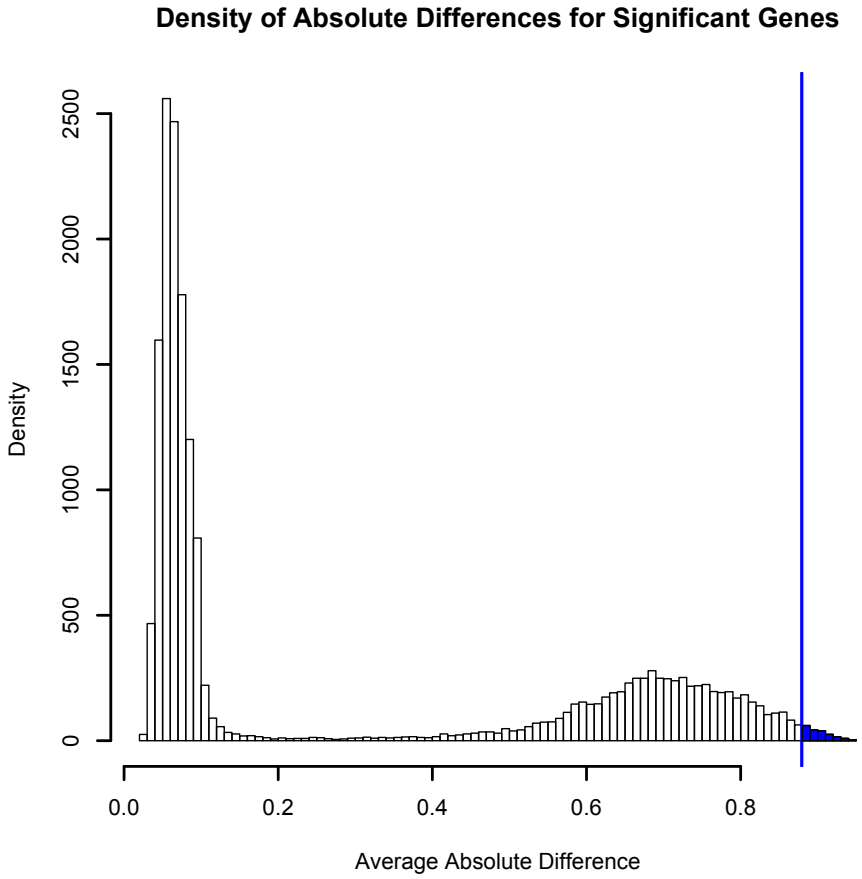

Figure 3.3: This plot contains the density of the absolute average differences for significant genes. We selected the 200 largest differences, the values where $a_i > 0.8794$.

3. *The genes corresponding to the 200 largest a_i are selected as the genes for the UPC*: A portion of the genes selected for the RAS UPC are shown in Figure 3.4. The differences between the control cells and RAS pathway are evident.

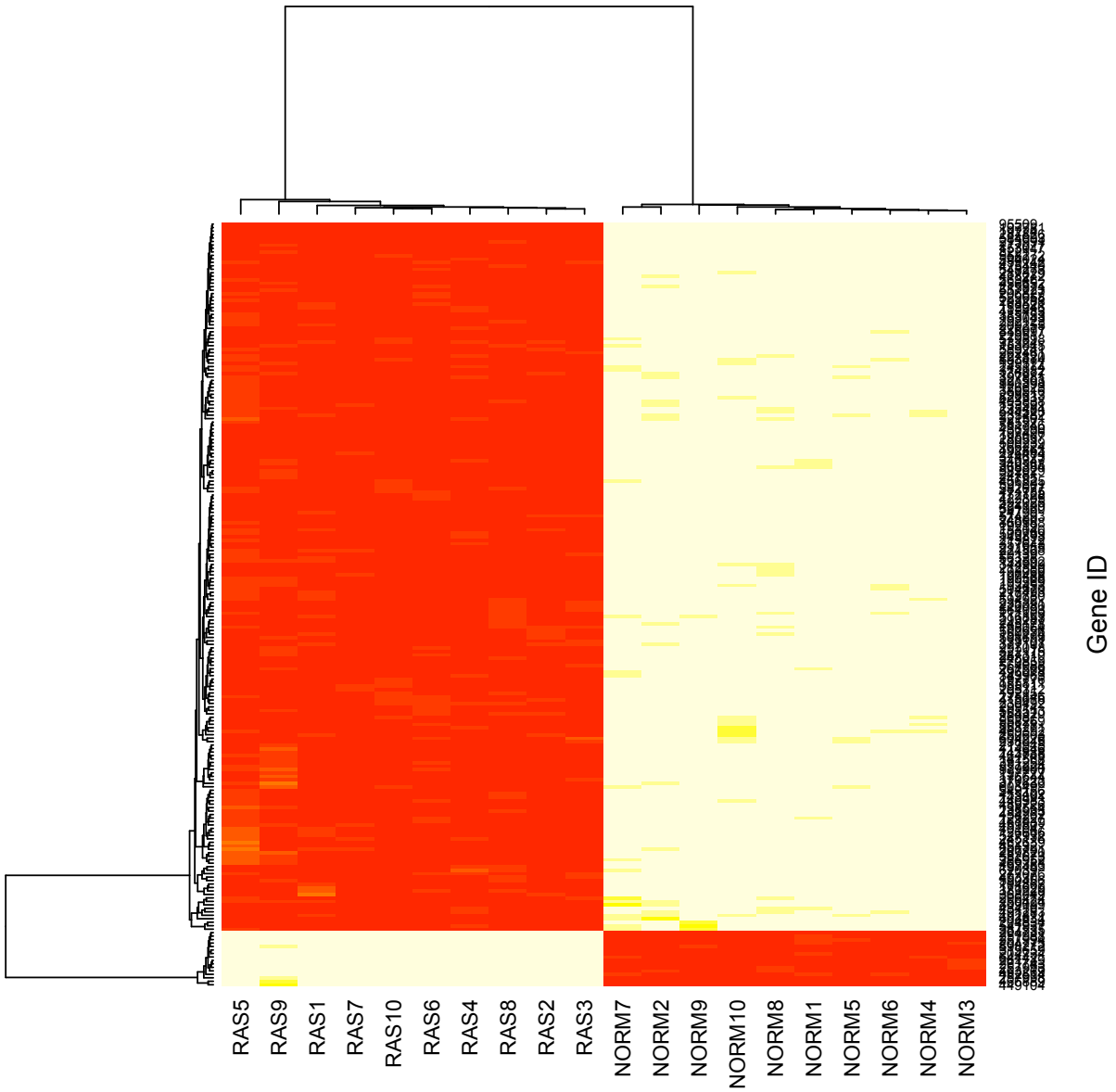

Figure 3.4: The genes selected for the RAS UPC are shown. There are clear differences in the control samples and the RAS samples.

4. *After the genes are selected, the last step in computing the UPC is to take the average probability of expression for* γ_{kj}: The UPC was calculated for the RAS pathway using the average of the 10 RAS samples. A portion of the UPC is shown in Table 3.2, and the complete UPC is contained in the Appendix A.

Table 3.2: A portion of the RAS UPC is shown below.

n	Gene	Probe ID	Probability of Expression
1	ZNF555	1553286_at#56_57	0.08
2	SRRM2	1554671_a_at#1098_635	0.12
3	SRRM2	1554671_a_at#358_217	0.09
4	SRRM2	1554671_a_at#732_245	0.08
5	ST6GAL2	1555123_at#362_635	0.08
6	RAP1A	1555339_at#219_599	0.05
7	RAP1A	1555339_at#835_173	0.10
8	RAP1A	1555340_x_at#218_599	0.06

We now have the UPC for the active RAS pathway: a fingerprint for the active RAS pathway. The process has been illustrated for the RAS pathway but can be applied to any other active pathway for which there are control and active samples.

To summarize, the UPC is a profile or fingerprint of an active pathway, containing 200 genes and the probability that that gene is expressed. All that remains is to project the UPC into cancer samples to see if it is present.

3.3 Measuring Pathway Projection

The previous section detailed the method of obtaining the profile of the active pathway; we now want to compare or project that active pathway to an individual cancer sample to see if they are similar or different, which will indicate wheter that the active pathway is present or not present, respectively. In this section we will illustrate a straightforward way to do this. Note it is not a very rigorous or developed method, but it illustrates the utility of the UPC. In fact, a very interesting extension to this project would be to develop and compare more

methodological approaches, particularly in the classification step. The steps for projecting the UPC into a given cancer sample are as follows:

1. Find probability of expression for the genes corresponding to the UPC using MMAX (Note, all of the genes will be used in the computation of the probabilities in MMAX, but only the 200 probabilities of the genes in the UPC are used to compute if the RAS is present).

2. Reduce the UPC and cancer probabilities to binary values.

3. Find the percent of concordant genes between the RAS UPC and the cancer sample. We will treat this as the probability that the pathway is present in a given cell.

4. Classify each sample in regards to the RAS UPC based on the probability computed above. The classification method was determined arbitrarily, based on two cutoffs and creating three classification groups: active, marginal, and inactive.

This method was applied to 51 lung cancer samples on h133+ arrays used by Bild et al. (2006) (GEO accession number GSE3141). The RNA was extracted from frozen tissue primary lung samples and was put onto expression arrays, one array for every sample. In this particular data set, there are samples from squamous cell lung carcinoma and adenocarcinoma. Adenocarcinoma is more likely to contain an activated RAS pathway, while squamous cell carcinoma is not likely to have an activated RAS pathway. We know the subtypes of the samples, which will give us some indication of the accuracy of our method: we expect to see more squamous cell carcinoma samples classified as having an active RAS pathway, and similarly, we expect more adenocarcinoma cells to be classified as having an inactive RAS pathway.

The probabilities of expression for the 51 lung cancer samples are displayed in Figure 3.5. There was a clear separation in these particular samples, hence an arbitrary classification scheme was selected: samples with a probability higher than 65% were classified as containing

an active RAS pathway (ON), samples with a probability lower than 45% had an inactive RAS pathway (OFF), and the samples whose probabilities fell in between are marginal (MAR).

The classification breakdown is contained in Table 3.3. According to this classification scheme, 45 samples contain an active RAS pathway, 4 did not, and 2 were marginal. A classification heatmap was also created (Figure 3.6). The heatmap uses the gene probabilities rather than the binary values, highlighting and confirming our results. We see that the samples classified as ON more closely resemble the RAS UPC, and those classified as OFF more closely resemble the control (NORM).

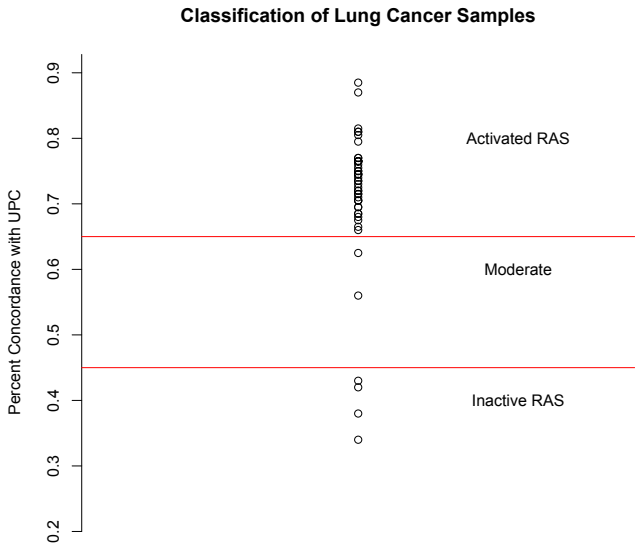

Figure 3.5: Cancer samples were classified based on their probability of concordant genes with the UPC. 45 samples had an active RAS pathway, 2 were marginal, and 4 had an inactive RAS pathway.

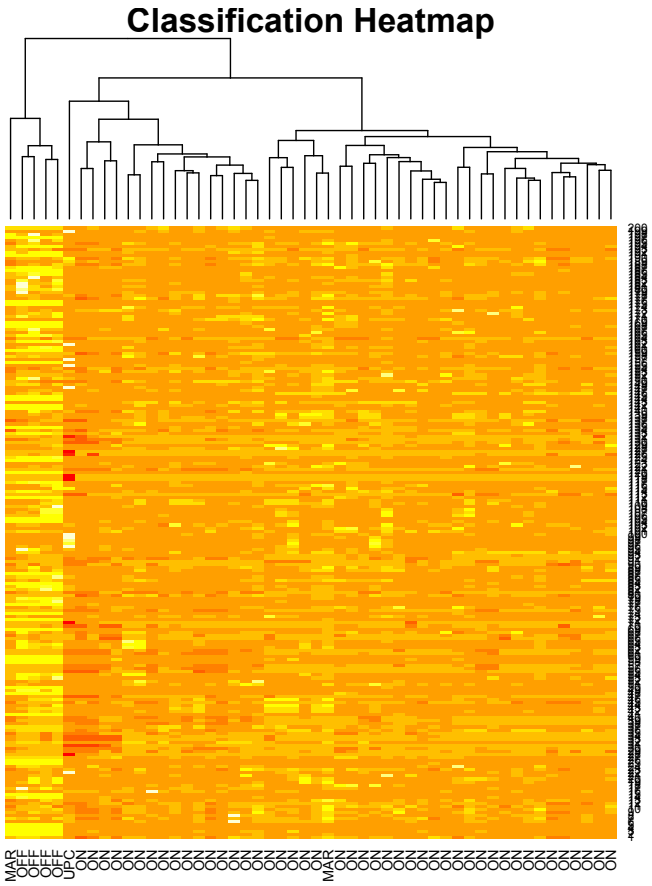

Figure 3.6: Cancer samples were classified based on their probability of concordant genes with the UPC. There is a clear separation in the samples that were classified as having an active RAS (ON) and those inactive samples (OFF).

Table 3.3: This table contains the classification of the 51 lung cancer samples. The majority of the samples have an active RAS pathway.

RAS status	Frequency
Active ($p > 0.65$)	45
Moderate ($0.45 < p < 0.65$)	2
Inactive ($p < 0.45$)	4

One difficultly with this portion of the analysis is that we do not know if our classification method is accurate since we do not know the truth about the cancer samples. We do, however, know the subtype of cancer for each sample, which provides some insight on how our method is performing. Recall that adenocarcinoma generally has an activated RAS pathway, and squamous cell carcinoma generally has an inactive RAS pathway. The classification was compared to the subtypes of the cancer (Table 3.4). Overall, both subtypes have a majority of samples with an active RAS pathway. This is what we expected for adenocarcinoma, but we were expecting a larger portion of adenocarcinoma cells to contain an inactive RAS. While we cannot make any quantitative statements about our method's performance, it does not seem to be able to distinguish the adenocarcinoma samples that contain an inactive RAS pathway.

Table 3.4: This table contains the classification of the 51 lung cancer samples compared to the known cancer subtypes. More squamous cell carcinoma samples were classified as active than we might expect. The classification, however, meets our expectation for adenoccarcinoma.

RAS status	Adenocarcinoma	Squamous Cell Carcinoma
Active ($p > 0.65$)	25	20
Moderate ($0.45 < p < 0.65$)	1	1
Inactive ($p < 0.45$)	1	3

According to Goodsell (1999), the active RAS pathway is present in 35% of lung cancers. Using our arbitrary classification scheme, 88% of lung cancer samples were determined to have an active RAS pathway. This result is not consistent with the literature. While this result was slightly disappointing, it is not surprising; since we are dealing with cancer, there are many things wrong in the cell, which would confound our results. More specifically, there could be another deregulated pathway that causes genes to behave differently. This would

cause the behavior of one single pathway to be confounded with the other pathways in the cell. With this in mind, the next logical development for this method would be to extend it to multiple pathways, unconfounding the results.

3.4 Conclusion

The goal of this project was to present a method that would quantify an active pathway then project it into a cancer sample to see if it is present. MMAX normalization outputs the probability a gene is expressed; and since it performed comparably to other existing methods, we then proceeded to use those probabilities from control and active pathway samples to determine the profile or UPC of the gene. Lastly, the UPC was projected into the cancer samples to see if the active RAS pathway was present. We determined that 45 of 51 samples had an active RAS pathway present. While the pathway projection was not as successful as we expected it to be, we did find that MMAX performs as well if not better than existing methods. Ultimately, this method has the potential to inform the treatment decisions made by researchers and doctors, increasing the recovery rate and decreasing recovery time.

BIBLIOGRAPHY

Alizadeh, A. A., Elsen, M. B., Davis, R. E., Ma, C., Lossos, I. S., Rosenwald, A., Boldrick, J. C., Sabet, H., Yu, T. T. X., Powell, J. I., Yang, L., Marti, G. E., Moore, T., Jr, J. H., Lu, L., Lewis, D. B., Tibshirani, R., Sherlock, G., Chan, W. C., Greiner, T. C., Welsenburger, D. D., Armitage, J. O., Warnke, R., Levy, R., Wilson, W., Grever, M. R., Byrd, J. C., Botstein, D., Brown, P. O., and Staudt, L. M. (2002), "Gene expression signatures define novel oncogenic pathways in T cell acute lymphoblastic leukemia," *Nature*, 403, 503–511.

Bild, A. H., Yao, G., Chang, J. T., Wang, Q., Potti, A., Chasse, D., Joshi, M.-B., Harpole, D., Lancaster, J. M., Berchuck, A., Jr, J. A. O., Marks, J. R., Dressman, H. K., West, M., and Nevins, J. R. (2006), "Oncogenic pathway signatures in human cancers as a guide to targeted therapies," *Nature*, 439, 353–357.

Bittner, M., Meltzer, P., Chen, Y., Jiang, Y., Seftor, E., Hendrix, M., Radmacher, M., Simons, R., Yakhini, Z., Ben-Dor, A., Sampas, N., Dougherty, E., Wang, E., Marincola, F., Gooden, C., Lueders, J., Glatfelter, A., Pollock, P., Carpten, J., Gillanders, E., Leja, D., Dietrich, K., Beaudry, C., Berens, M., Alberts, D., Sondak, V., Hayward, N., and Trent, J. (2002), "Molecular classification of cutaneous malignant melanoma by gene expression profiling," *Nature*, 406, 536–540.

Bolstad, B. M., Irizarry, R. A., Astrand, M., and Speed, T. P. (2003), "A comparison of normalization methods for high density oligonucleotide array data based on variance and bias," *Bioinformatics*, 19, 185–193.

Boutros, P. C., Lau, S. K., Liu, M. N., Shepherd, F. A., Der, S. D., Tsao, M.-S., Penn, L. Z., and Jurisica, I. (2009), "Prognostic gene signatures for non-small-cell lung cancer,"

Proceedings of the National Academy of Sciences, 106, 2824–2828.

Chang, J. T., Carvalho, C., Mori, S., Bild, A. H., Gatza, M. L., Wang, Q., Lucas, J. E., Potti, A., Febbo, P. G., West, M., and Nevins, J. R. (2009), "A Genomic Strategy to Elucidate Modules of Oncogenic Pathway Signaling Networks," *Molecular Cell*, 34, 104–114.

Dudley, J. T., Tibshirani, R., Deshpande, T., and Butte, A. J. (2009), "Disease signatures are robust across tissues and experiments," *Molecular Systems Biology*, 5, 307.

Edwards, D. (2003), "Non-linear normalization and background correction in one-channel cDNA microarray studies," *Bioinformatics*, 19, 825–833.

Ferrando, A. A., Neuberg, D. S., Stauntan, J., Loh, M. L., Huard, C., Raimondi, S. C., Behm, F. G., Pui, C.-H., Downing, J. R., Gilliland, D. G., Lander, E. S., Golub, T. R., and Look, A. T. (2002), "Gene expression signatures define novel oncogenic pathways in T cell acute lymphoblastic leukemia," *Cancer Cell*, 1, 75–87.

Geyer, F. C., and Reis-Filho, J. S. (2009), "Microarray-based Gene Expression Profiling as Clinical Tool for Breast Cancer Management: Are We There Yet?" *International Journal of Surgical Pathology*, 17, 285–292.

Goodsell, D. S. (1999), "The Molecular Perspective: The ras Oncogene," *The Oncologist*, 4, 263–264.

Hastie, T., Tibshirani, R., and Friedman, J. (2001), *The Elements of Statistical Learning Data Mining, Inference, and Prediction*, Springer: Canada, 236–243.

Hubbell, E., Liu, W.-M., and Mei, R. (2002), "Robust estimators for expression analysis," *Bioinformatics*, 18, 1585–1592.

Irizarry, R. A., Hobbs, B., Collin, F., Beazer-Barclay, Y. D., Antonellis, K. J., Scherf, U., and Speed, T. P. (2003), "Exploration, normalization, and summaries of high density oligonucleotide array probe level data," *Biostatistics*, 4, 249–264.

Johnson, W. E., Li, W., Meyer, C. A., Gottardo, R., Caroll, J. S., Brown, M., and Liu, X. S. (2006), "Model-based analysis of tiling-arrays for ChIp-chip," *Proceedings of the National Academy of Sciences*, 103, 1245712462.

Kapur, K., Xing, Y., Ouyang, Z., and Wong, W. H. (2007), "Exon arrays provide accurate assessments of gene expression," *Genome Biology*, 8, R82.

Li, C., and Wong, W. H. (2001), "Model-based analysis of oligonucleotide arrays: Expression index computation and outlier detection," *Proceedings of the National Academy of Sciences*, 98, 31–36.

Parmigiani, G., Garrett, E. S., Anbazhagan, R., and Gabrielson, E. (2002), "A statistical framework for expression-based molecular classification in cancer," *Journal of the Royal Statistical Society B*, 64, 717–736.

Sjoblom, T., Jones, S., Wood, L. D., Parsons, D. W., Lin, J., Barber, T. D., Mandelker, D., Leary, R. J., Ptak, J., Silliman, N., Szabo, S., Buckhaults, P., Farrell, C., Meeh, P., Markowitz, S. D., Willis, J., Dawson, D., Willson, J. K. V., Gadzar, A. F., adn Leo Wu, J. H., Liu, C., Parmigiani, G., Park, B. H., Bachman, J. E., Papadoulos, N., Vogelstein, B., Kinzler, K. W., and Velculescu, V. E. (2006), "The Consensus Coding Sequences of Human Breast and Colorectal Cancers," *Science*, 314, 268–274.

Tuna, S., and Niranjan, M. (2009), "Classification with binary gene expressions," *Journal of Biomedical Science and Engineering*, 2, 390–399.

Tusher, V. G., Tibshirani, R., and Chu, G. (2001), "Significance analysis of microarrays applied to the ionizing radiation response," *Bioinformatics*, 19, 185–193.

Winstead, E. R. (2008), "Lung Cancer Test Aims to Improve Early Detection," *National Cancer Institute*, 5, 4, 10.

Wood, L., Parsons, D. W., Jones, S., Lin, J., Sjoblom, T., Leary, R. J., Shen, D., Boca, S. M., adn Janine Ptak adn Natalie Siliman, T. B., Szabo, S., Dezso, Z., Ustyanksky,

V., Nikolskaya, T., Nikolsky, Y., Karchin, R., Wilson, P. A., Kaminker, J. S., Zhang, Z., Crowshaw, R., Willis, J., Dawson, D., Shipitsin, M., Wilson, J. K. V., Sukumar, S., Polyak, K., Park, B. H., Pethiyagoda, C. L., Pant, P. V. K., Ballinger, D. G., Sparks, A. B., Hartigan, J., Smith, D. R., adn Nickolas Papadopoulos, E. S., Buckhaults, P., Markowitz, S. D., Parmigiani, G., Kinzler, K. W., Velculescu, V. E., and Vogelstein, B. (2007), "The Genomic Landscapes of Human Breast and Colorectal Cancers," *Science*, 318, 1108–1113.

Zilliox, M., and Irizarry, R. A. (2007), "A gene expression barcode for microarray data," *Nature Methods*, 4, 911–913.

APPENDICES

APPENDIX A

UPC FOR DISRUPTED RAS PATHWAY

Table A.1: The RAS UPC is shown below.

n	Gene	Probe ID	Probability of Expression
1	ZNF555	1553286_at#56_57	0.08
2	SRRM2	1554671_a_at#1098_635	0.12
3	SRRM2	1554671_a_at#358_217	0.09
4	SRRM2	1554671_a_at#732_245	0.08
5	ST6GAL2	1555123_at#362_635	0.08
6	RAP1A	1555339_at#219_599	0.05
7	RAP1A	1555339_at#835_173	0.10
8	RAP1A	1555340_x_at#218_599	0.06
9	RAP1A	1555340_x_at#770_41	0.09
10	RAP1A	1555340_x_at#836_173	0.10
11	STH	1555752_at#593_699	0.06
12	LZTS2	1555881_s_at#1109_1101	0.10
13	C17orf64	1555985_at#479_133	0.08
14	FN1	1558199_at#592_721	0.11
15	TMTC3	1560017_at#1132_795	0.09
16	TMTC3	1560017_at#705_1025	0.08
17	LOC283856	1560707_at#165_723	0.09

18	LOC100287432	1561229_at#151_23	1.00
19	DNAH3	1563290_at#624_501	0.06
20	MARCKSL1	200644_at#638_779	0.10
21	CAT	201432_at#941_633	0.09
22	ALDH3A2	202053_s_at#262_1097	0.10
23	ALDH3A2	202054_s_at#902_257	0.09
24	MGP	202291_s_at#773_173	0.10
25	TNFSF10	202688_at#760_381	0.09
26	TNFSF10	202688_at#78_57	0.10
27	GSTA4	202967_at#214_667	0.08
28	BCL6	203140_at#1007_215	0.12
29	MMD	203414_at#241_775	0.09
30	SKP2	203625_x_at#1056_197	0.10
31	SKP2	203625_x_at#464_137	0.11
32	SKP2	203625_x_at#583_261	0.09
33	APOBEC3G	204205_at#339_215	0.10
34	GAS1	204457_s_at#1077_927	0.09
35	GAS1	204457_s_at#459_217	0.07
36	GAS1	204457_s_at#52_539	0.09
37	GAS1	204457_s_at#588_55	0.08
38	GAS1	204457_s_at#60_553	0.08
39	CROT	204573_at#351_723	0.09
40	MAPK4	204708_at#9_191	0.11
41	MDM4	205655_at#631_91	0.06
42	SLC7A4	205864_at#445_491	0.09
43	IL3RA	206148_at#326_949	0.10

44	CDSN	206193_s_at#458_459	0.10
45	OVOL1	206604_at#1103_197	0.12
46	DDX17	208151_x_at#1109_103	0.09
47	HIST1H2BI	208523_x_at#50_691	0.11
48	HIST1H2BE	208527_x_at#1143_519	0.09
49	HIST1H2BE	208527_x_at#842_639	0.06
50	HIST1H1E	208553_at#1156_521	0.11
51	HIST1H1E	208553_at#230_485	0.12
52	HIST1H1E	208553_at#506_411	0.08
53	H2BFS	208579_x_at#1142_519	0.09
54	H2BFS	208579_x_at#843_639	0.06
55	DDX17	208719_s_at#1063_133	0.10
56	DDX17	208719_s_at#1110_103	0.09
57	DDX17	208719_s_at#97_199	0.08
58	FOS	209189_at#473_1003	1.00
59	FOS	209189_at#837_781	1.00
60	SYT11	209198_s_at#738_569	0.06
61	PEG3	209242_at#366_993	0.11
62	MAF	209348_s_at#86_751	0.08
63	HMGN3	209377_s_at#636_691	0.11
64	SF3A2	209381_x_at#1097_1139	0.09
65	GATA3	209604_s_at#695_859	0.08
66	KLK2	209854_s_at#1151_433	0.10
67	HIST1H2BD	209911_x_at#1144_519	0.08
68	KCNH2	210036_s_at#1061_361	0.10
69	SLC12A5	210040_at#493_605	0.08

70	BMPR2	210214_s_at#321_293	0.08
71	BMPR2	210214_s_at#760_917	0.10
72	RUNX1	210365_at#743_11	0.09
73	MCAM	211042_x_at#936_461	0.09
74	RUNX1	211181_x_at#5_655	0.09
75	RUNX1	211182_x_at#3_655	0.09
76	HAP1	211222_s_at#1078_985	0.07
77	CASP1	211366_x_at#462_561	0.09
78	RUNX1	211620_x_at#2_655	0.11
79	TEX261	212084_at#874_575	0.09
80	MTUS1	212093_s_at#1011_1041	0.09
81	MTUS1	212093_s_at#256_1109	0.09
82	PHLDB1	212134_at#1015_721	0.12
83	LIMCH1	212327_at#143_747	0.09
84	CXCR7	212977_at#1107_931	0.11
85	CXCR7	212977_at#421_107	0.11
86	SOSTDC1	213456_at#234_731	0.08
87	HNRNPH1	213472_at#783_699	0.09
88	PPBP	214146_s_at#1114_575	1.00
89	PPBP	214146_s_at#417_837	1.00
90	PPBP	214146_s_at#570_11	1.00
91	PPBP	214146_s_at#94_391	0.99
92	RAP2A	214487_s_at#413_181	1.00
93	ZNF33B	215022_x_at#232_1033	0.10
94	RUNDC3B	215321_at#272_217	0.09
95	ZNF277	215887_at#340_299	0.08

96	PDGFA	216867_s_at#1081_339	0.12
97	PDGFA	216867_s_at#540_733	0.08
98	PDGFA	216867_s_at#811_983	0.08
99	RUNX1	217263_x_at#4_655	0.08
100	IGHG1	217369_at#1109_375	0.11
101	IGHG1	217369_at#118_177	0.09
102	TNS3	217853_at#597_315	0.08
103	HERC6	219352_at#938_265	0.08
104	HPSE	219403_s_at#998_293	1.00
105	ADAMTS5	219935_at#469_1049	0.09
106	ASB7	219996_at#486_773	0.10
107	IGKC	221651_x_at#246_385	0.07
108	IGKC	221671_x_at#245_385	0.06
109	WDR59	221981_s_at#131_977	0.10
110	HIST1H2BD	222067_x_at#1141_519	0.09
111	BBS2	223227_at#802_367	0.10
112	CCDC8	223495_at#646_777	0.09
113	PXMP4	224210_s_at#336_849	0.05
114	RNF17	224384_s_at#552_1013	0.11
115	CXXC5	224516_s_at#1030_257	0.10
116	IGKC	224795_x_at#244_385	0.06
117	MYLK	224823_at#318_689	0.09
118	APCDD1	225016_at#760_741	0.09
119	KIAA1370	225327_at#272_591	0.09
120	FBXO32	225803_at#183_1127	0.08
121	TP53INP1	225912_at#594_27	0.10

122	C17orf89	225966_at#357_419	0.08
123	TTC8	226120_at#543_109	0.11
124	HSPB6	226304_at#507_1065	0.11
125	LRIG3	226908_at#14_201	0.08
126	ZNF503	227195_at#7_721	0.11
127	EPHA4	227449_at#259_579	0.05
128	EPHA4	227449_at#676_639	0.09
129	NFYA	228433_at#185_269	0.09
130	SPHKAP	228509_at#144_683	0.08
131	LOC100292443	228526_at#885_709	0.11
132	LFNG	228762_at#838_777	0.08
133	C3orf38	229174_at#87_159	0.10
134	S1PR5	230464_at#50_639	0.06
135	C4orf22	231565_at#912_535	0.97
136	LOC203274	232034_at#978_507	0.11
137	MBNL2	232138_at#728_215	0.11
138	DNHD1	232240_at#4_451	0.10
139	DUSP27	232252_at#1069_1079	0.08
140	DIRAS1	232854_at#689_621	1.00
141	KIAA0182	232988_at#876_899	0.10
142	SLC4A9	233183_at#662_759	0.98
143	CDS2	233630_at#927_201	0.08
144	LAMA3	234608_at#4_893	1.00
145	STAG3L1	235263_at#876_773	0.09
146	GSTA4	235405_at#45_549	0.07
147	RNF144B	235549_at#1066_551	0.06

148	TMEM20	236219_at#633_737	0.05
149	NFATC4	236270_at#571_541	0.10
150	VGLL2	236352_at#254_159	0.11
151	DTWD1	236649_at#90_1083	0.09
152	OTX1	238839_at#589_301	0.09
153	ENAM	240586_at#505_285	0.07
154	CCNL1	241495_at#292_713	0.07
155	CCNL1	241495_at#456_109	0.07
156	CCNL1	241495_at#4_233	0.08
157	COPS7B	243628_at#102_781	0.06
158	XIST	243712_at#196_975	0.11
159	XIST	243712_at#325_945	0.09
160	SYNE2	243841_at#1130_285	0.08
161	LCP2	244556_at#842_93	0.10
162	IGLON5	244694_at#49_215	0.09
163	LOC100130502	244744_at#488_761	0.08
164	HBEGF	38037_at#985_73	1.00

APPENDIX B

DOCUMENTED CODE

B.1 Code for MMAX

EM Algorithm Implementation

```
#########################################################
######################### Introduction ##################
#########################################################
#This purpose of this code is to normalize data in a 2-comp mixture
#model fit for each GC group
 (MMAX without the model).
#This code assumes that both components are normally distributed,
#and utilizes the EM algorithm
#(pg.238 of Elements of Statistical Learning, Data Mining, Inference
#and Prediction by Hastie, etc.)
#to estimate the parameters.
#There are two steps. 1. Parameter estimation, and 2. Normalization.
#Each step will be performed for each GC group using a loop, as
#mentioned above.

#########################################################
####### Step 1: Parameter Estimation using EM Algorithm ########
#########################################################
```

```
#Note that this function will estimate parameters for 1 GC group.
#It is designed to repeat for each GC group,
#nested inside OneChannelNormalize.

EM.GC <- function(dataset,pihat) #need data and initial estimate
#for pihat.
The default for pihat is 0.5, and the support is [0,1].
{
results <- rep(NA,5)
#EM Algorithm
#Step 1: Initial Values
m1 <- summary(dataset)[2] #initial mean for 'bkgrnd' is the 25th %ile
m2 <- summary(dataset)[4] #initial mean for 'b + e' is the 75th %ile
s1 <- var(dataset)/2 #initial variance estimate for
'background' distribution is half of the sample variance
s2 <- var(dataset)/2 #initial variance estimate for
'background + expressed' distribution is half of the sample variance.
pi <- pihat #Initial value for pi, proportion of values from the
#'background + expressed' distribution.
pi.old <- 1

while(abs(pi-pi.old) > tol)
{
#Step 2: Expectation
gam <- pi*dnorm(dataset,m2,sqrt(s2))/( (1-pi)*
dnorm(dataset,m1,sqrt(s1)) + pi*dnorm(dataset,m2,sqrt(s2)) )
```

```
#Step 3: Maximization
m1 <- sum((1-gam)*dataset)/sum(1-gam)
m2 <- sum(gam*dataset)/sum(gam)
s1 <- sum((1-gam)*(dataset-m1)^2)/sum(1-gam)
s2 <- sum(gam*(dataset-m2)^2)/sum(1-gam)
pi.old <- pi
pi <- sum(gam/length(dataset))
}
results <- c(m1,m2,s1,s2,pi)  #Note, this outputs variance estimates,
not standard deviation estimation
names(results) <- c("Mu 1","Mu 2","Sigma^2 1","Sigma^2 2","Pi")
return(results)
}

#Testing EM.GC function
#Data is simulated from 2 separate distributions
y1 <- rnorm(50,15,1)
y2 <- rnorm(50,5,1)
#Set tolerance
tol <- 0.001
#Data is combined, and
EM.GC(c(y1,y2),.6) #Works, although with only 50 observations,
#the variances are much greater than expected.
#More data points to see if it gets better estimates:
y1 <- rnorm(5000,15,1)
y2 <- rnorm(5000,5,1)
#Set tolerance
```

```
tol <- 0.001
#Data is combined, and
EM.GC(c(y1,y2),.6) #Works, although with only 50 observations,
#the variances are much greater than expected.

#I'll want a function that reads in the data from the Microarray File.
#Need a sample file- It will need to return the values as well
#as the GCcount for the sequence. If I'm given a DNA sequence
find.gc <- function(seq)
{
a <- unlist(strsplit(seq,NULL))
count <- sum(a=='C'|a=='G')
return(count=count)
}
seq <- as.character(data[,cols[3]])
GC <- sapply(seq,gc)
for(i in 1:nobs)
{
GCCount[i,1] <- as.numeric(GC[[i]][1])
}

gcCount <- find.gc(seq)

OneChannelNormalize <- function(data,GCCount=NULL,useGC=FALSE,
minGC=5,pihat=.5,tol=.0001)
{
if(useGC==FALSE)
```

```
{
y <- data
params <- EM.GC(y,pihat,tol)
print(params)
if(params[1] < params[2])
{
mu <- params[1] #Smaller mean is the background mean
sigma2 <- params[3] #Sigma squared for background mean
}
else
{
mu <- params[2] #Smaller mean is the background mean
sigma2 <- params[4] #Sigma squared for background mean
}
normalize <- (y - mu)/sqrt(sigma2)
norm <- normalize
}
else
{
norm <- rep(NA,length=length(data))
##This establishes which groups will be the "GC groups"
GCCount <- as.numeric(GCCount) #Allows String Input
GCgroups <- NULL
for(i in sort(unique(GCCount)))
{
if(sum(GCCount==i) >= minGC)
{
```

```
GCgroups <- c(GCgroups,i)
}
}

##This groups GC count into the groups that have more than the min.
for(i in sort(unique(GCCount)))
{
if(sum(GCCount==i) < minGC )
{
GCCount[GCCount==i] <- GCgroups[order(abs(GCgroups-i))[1]]
#Puts into
#the closest group (groups down for ties)
}
}

##This is the actual normalization method
for (i in GCgroups)
{
y <- data[GCCount==i]
params <- EM.GC(y,pihat,tol)
#print(params)
if(params[1] < params[2])
{
mu <- params[1] #Smaller mean is the background mean
sigma2 <- params[3] #Sigma squared for background mean
}
else
```

```
{
mu <- params[2] #Smaller mean is the background mean
sigma2 <- params[4] #Sigma squared for background mean
}

  normalize <- (y - mu)/sqrt(sigma2)
norm[GCCount==i] <- normalize
}
}
return(norm)
}

#Testing Normalization Function:
y1 <- rnorm(50,15,1)
y2 <- rnorm(50,5,1)
gclist <- rep(c(1,2,3,4),25)
gclist3 <- c(rep(1,2),rep(c(2,3),48))
test1 <- OneChannelNormalize(c(y1,y2)) #Tests the function with no GC
test2 <- OneChannelNormalize(c(y1,y2),gclist,useGC=TRUE)
#Tests the function with GC, large GC groups
test3 <- OneChannelNormalize(c(y1,y2),gclist3,useGC=TRUE)
#Tests the function with GC, large GC groups

plot(density(test1))
plot(density(test2))
plot(density(test3))
plot(density(c(y1,y2)))
```

Model Validation

```
# Get the necessary packages
source("http://bioconductor.org/biocLite.R")
biocLite("affy")
library('affy')
setwd('/Users/michelle/Desktop/Research/Barcode/AffyLatin')

#Read in the data: pass in a cel file
data1 <- ReadAffy("12_13_02_U133A_Mer_Latin_Square_Expt1_R1.CEL",
"12_13_02_U133A_Mer_Latin_Square_Expt1_R2.CEL",
"12_13_02_U133A_Mer_Latin_Square_Expt1_R3.CEL")
#mas5 normalization: pass in the "ReadAffy" object
calls <- mas5calls(data1)
write.table(calls,'ATTEMPT.txt')
calls <- t(read.table('ATTEMPT.txt',header=T))
head(calls)
abar <- (calls=='M')*.5 + (calls == "P")*1
colnames(abar) <- c("A1","A2","A3")

################
setwd("Users/michelle/Desktop/Research/Barcode/AffyLatin")

obar <- read.table("completeresultsfull.txt")
rownames(obar) <- obar[,1]
obar <- as.matrix(obar[,-1])
colnames(obar) <- c("A1","A2","A3")
```

```
xbar1 <- read.table("bar1.txt")
xbar2 <- read.table("bar2.txt")
xbar3 <- read.table("bar3.txt")
ubar <- cbind(xbar1[,-1],xbar2[,-1],xbar3[,-1])
rownames(ubar) <- xbar1[,1]

obar[1:20,] #Our method
ubar[1:20,] #Barcoding

mobar <- apply(obar,1,mean)
mobar <- mobar[1:22215]

mubar <- apply(ubar,1,mean)
sum(mubar==1) ##Very strong agreement with expressed genes
sum(mubar >0 & mubar <1)
sum(mubar==0)

mabar <- apply(abar,1,mean)

sum(mubar) ##Includes fraction values
sum(mabar)
sum(mobar>.9419)
cutoff <- .9419

ourbar <- mobar #Round probabilities?
for(i in 1:length(mobar))
```

```
{
if(mobar[i]>cutoff)
{
ourbar[i] <- 1
}
else
{
ourbar[i] <- 0
}
}

sum(ourbar)

x <- cbind(mubar,ourbar)
j <- sum(x[,1]==1 & x[,2]==1)
j/1338 #27.5% agreement (stringent cutoff)

y <- cbind(ubar,mobar)

design <- rep(c(1:10),each=3)
design2 <- rep(c(1:10),each=9)
design3 <- rep(rep(c(1:10),each=3),3)

designvals <- c(0,.125,.25,.5,1,2,4,8,16,32)

rows <- c(which(rownames(ubar)=="203508_at"),
which(rownames(ubar)=="204563_at"),
```

```
which(rownames(ubar)=="204513_s_at"),
which(rownames(ubar)=="204205_at"),
which(rownames(ubar)=="204959_at"),
which(rownames(ubar)=="207655_s_at"),
which(rownames(ubar)=="204836_at"),
which(rownames(ubar)=="205291_at"),
which(rownames(ubar)=="209795_at"),
which(rownames(ubar)=="207777_s_at"),
which(rownames(ubar)=="204912_at"),
which(rownames(ubar)=="205569_at"),
which(rownames(ubar)=="207160_at"),
which(rownames(ubar)=="205692_s_at"),
which(rownames(ubar)=="212827_at"),
which(rownames(ubar)=="209606_at"),
which(rownames(ubar)=="205267_at"),
which(rownames(ubar)=="204417_at"),
which(rownames(ubar)=="205398_s_at"),
which(rownames(ubar)=="209734_at"),
which(rownames(ubar)=="209354_at"),
which(rownames(ubar)=="206060_s_at"),
which(rownames(ubar)=="205790_at"),
which(rownames(ubar)=="200665_s_at"),
which(rownames(ubar)=="207641_at"),
which(rownames(ubar)=="207540_s_at"),
which(rownames(ubar)=="204430_s_at"),
which(rownames(ubar)=="203471_s_at"),
which(rownames(ubar)=="204951_at"),
```

```
which(rownames(ubar)=="207968_s_at"))

#####################################################
### chunk number 4: MMAX boxplot
#####################################################
#Our barcode
boxplot(mobar[rows]~design,col="green",xaxt='n',ylab="Probability
of Expression",xlab="Spike-in concentration (picomolar)")
axis(1,c(1:10),c(0,".125",".25",".5",1,2,4,8,16,32))
title("MMAX")
#abline(-.1,1/9,lwd="3")

mmax <- matrix(mobar[rows],ncol=3,byrow=T)
apply(mmax,1,mean)
#####################################################
### chunk number 5: Barcodeboxplot
#####################################################
#Their barcode
boxplot(mubar[rows]~design,col="blue",xaxt='n',ylab="Probability
of Expression",
xlab="Spike-in concentration (picomolar)")
axis(1,c(1:10),c(0,".125",".25",".5",1,2,4,8,16,32))
title("Barcode")
#abline(-.1,1/9,lwd="3")

brcode <- matrix(mubar[rows],ncol=3,byrow=T)
apply(brcode,1,mean)
```

```
#####################################################
### chunk number 7: Affy Plot
#####################################################
boxplot(mabar[rows]~design,col="red",xaxt='n',ylab="Probability
of Expression",
xlab="Spike-in concentration (picomolar)")
axis(1,c(1:10),c(0,".125",".25",".5",1,2,4,8,16,32))
title("Mas5")
abline(-.1,1/9,lwd="3")

laffy <- matrix(mabar[rows],ncol=3,byrow=T)
apply(laffy,1,mean)

cbind(apply(mmax,1,mean),apply(brcode,1,mean),apply(laffy,1,mean))
#####################################################
### chunk number 7: Comparison Box Plot
#####################################################
boxplot(mabar[rows]~design,col="red",xaxt='n',ylab="Probability
of Expression",
xlab="Spike-in concentration (picomolar)")
axis(1,c(1:10),c(0,".125",".25",".5",1,2,4,8,16,32))
title("Comparison of Methods for Spike-In Concentration")
boxplot(mubar[rows]~design,col="blue",add=T,xaxt='n')
boxplot(mobar[rows]~design,col="green",xaxt='n',add=T)
legend(.3,.9,lty=1,col=c("green","blue","red","black"),c
("MMAX","Barcode","Affy","Truth"))
```

```
abline(-1,1/10,lwd="3")

#For just barcode and our method:
boxplot(mubar[rows]~design,col="blue",xaxt='n',ylab="Probability
of Expression",xlab="Spike-in concentration (picomolar)")
axis(1,c(1:10),c(0,".125",".25",".5",1,2,4,8,16,32))
title("Comparison of Methods for Spike-In Concentration")
boxplot(mobar[rows]~design,col="green",xaxt='n',add=T)
legend(1.5,.9,lty=1,col=c("green","blue"),c("MMAX","Barcode"))
```

B.2 Code for UPC calculation

```
setwd("/Users/michelle/Desktop/Project Data")
#####################################################
############ Step 1: Process and Normalize the data ######
#####################################################

#This step was performed using the MMAX program

#############################################################
##### Step 2: Calculating the UPC for the disrupted RAS pathway ######
#############################################################

#1. Calculate probability of expression for pathway & control
#This was done using the MMAX program.
#Heatmap of the probability of expression for RAS and control
#(Note: Only a random sample of 1000 genes were chosen)
set.seed(12345)
```

```
control.file <- NULL
ras.file <- NULL
bigdata <- matrix(NA,ncol=20,nrow=604258)
for(i in 1:10)
{
control.file[i] <- paste("0159_62",28+i,"_h133+_GFP-",i,"
_cel.norm.txt",sep="")
ras.file[i] <- paste("0159_67",42+i,"_h133+_RAS-",i,"
_cel.norm.txt",sep="")
bigdata[,i] <- read.table(control.file[i],comment.char="")[,3]
#Control 1-10
bigdata[,i+10] <- read.table(ras.file[i],comment.char="")[,3]
# RAS 11-20
}
step1 <- sample(1:604258,1000)
random.genes <- bigdata[step1,]
colnames(random.genes) <- c("NORM1","NORM2","NORM3","NORM4",
"NORM5","NORM6","NORM7","NORM8","NORM9","NORM10",
"RAS1","RAS2","RAS3","RAS4",
"RAS5","RAS6","RAS7","RAS8","RAS9","RAS10")
heatmap(random.genes,labRow=NA,ylab=NA,main="Probability of
Expression for RAS and Control Samples")

#2. Significant genes are selected
X <- matrix(c(rep(1,10),rep(0,20),rep(1,10)),ncol=2)
cont <- c(1,-1)
#Approach 1: Using p-values
```

```
denom <- matrix(NA,nrow(bigdata),1)
tstat <- matrix(NA,nrow(bigdata),1)
diff <- matrix(NA,nrow(bigdata),1)
for(i in 1:nrow(bigdata))
{
y <- bigdata[i,]
b <- solve(t(X)%*%X)%*%t(X)%*%y
diff[i,] <- (b[1,]-b[2,])
s2 <- t(y-X%*%b)%*%(y-X%*%b)/(ncol(bigdata)-2)
denom[i,] <- s2[1,1]*cont%*%solve(t(X)%*%X)%*%cont
tstat[i,] <- cont%*%b/sqrt(denom[i,1])
}

pval <- 2*(1-pt(abs(tstat),ncol(bigdata)-2))
sort(abs(diff),decreasing=T)[201]
significants <- bigdata[((pval<=0.00000000001 &
abs(diff) > 0.8793205)),]
dim(significants)
heatmap(significants)
#absdiff.pdf
hist(abs(diff[pval<=0.00000000001]),
main="Density of Absolute Differences
 for Significant Genes",ylab='Density',lwd=2,
 xlab="Average Absolute Difference",breaks=100)
abline(v=sort(abs(diff[pval<=0.00000000001]),
decreasing=T)[201],lwd=2,col="blue")
hist(sort(abs(diff[pval<=0.00000000001]),
```

```
decreasing=TRUE)[1:195],col="blue",add=T)

#3. Selecting the 200 genes with the largest absolute difference.
significants <- bigdata[((pval<=0.00000000001 &
abs(diff) > 0.8793205)),]
sig.n <- c(which(pval<=0.00000000001 &
abs(diff) > 0.8793205))
probes <- read.table(control.file[1],comment.char="")[,1]
results <- cbind(probes,diff,bigdata)
sig.probe <- results[sig.n,]
rownames(sig.probe) <- sig.probe[,1]
#upc1.pdf
heatmap(sig.probe[,-c(1:2)],labRow=NA,ylab=NA,
main="Genes Selected for UPC")

#4.  Heatmap of 'Average' pathway:
upc <- rowMeans(bigdata[sig.n,11:20])
NORM <- rowMeans(bigdata[sig.n,1:10])
#heatmap(cbind(upc,bigdata[sig.n,11:20],NORM,bigdata[sig.n,1:10])
,Colv=NA,Rowv=NA)
heatmap(cbind(upc,bigdata[sig.n,11:20],bigdata[sig.n,1:10]))
#Output the significant probes easily
probes[sig.n]
```

B.3 Code for pathway projection

```
TRAIN <- cancer.files <- c("0176_6642_h133+_98-691.norm.txt",
"0176_6602_h133+_98-711.norm.txt",
```

```
"0176_6621_h133+_98-771.norm.txt",
"0176_6639_h133+_98-1063.norm.txt",
"0176_6613_h133+_97-587.norm.txt",
"0176_6601_h133+_98-320.norm.txt",
"0176_6606_h133+_97-1026.norm.txt",
"0176_6607_h133+_98-933.norm.txt")

colnames(vals.train) <- c(rep("A",4),rep("S",4))
vals.train <- matrix(NA,nrow=200,ncol=8)
for(i in 1:8)
{
vals.train[,i] <- read.table(TRAIN[i],comment.char="")[sig.n,3]
}
heatmap(cbind(upc,vals.train))

# Projecting it into the cancer sample
TEST <- cancer.files <- c("0176_6608_h133+_96-475.norm.txt",
"0176_6610_h133+_99-671.norm.txt","0176_6611_h133+_98-683.norm.txt",
"0176_6612_h133+_97-403.norm.txt","0176_6598_h133+_10-00.norm.txt",
"0176_6625_h133+_00-011.norm.txt","0176_6632_h133+_00-315.norm.txt",
"0176_6614_h133+_98-543.norm.txt","0176_6616_h133+_99-692.norm.txt",
"0176_6617_h133+_98-657.norm.txt","0176_6618_h133+_99-440.norm.txt",
"0176_6619_h133+_99-728.norm.txt","0176_6620_h133+_98-1146.norm.txt",
"0176_6622_h133+_98-1216.norm.txt","0176_6623_h133+_98-1014.norm.txt",
"0176_6624_h133+_99-830.norm.txt","0176_6626_h133+_98-152.norm.txt",
"0176_6627_h133+_98-1293.norm.txt","0176_6628_h133+_98-1296.norm.txt",
"0176_6489_h133+_97-0949.norm.txt","0176_6629_h133+_98-375.norm.txt",
```

```
"0176_6491_h133+_98-0292.norm.txt","0176_6630_h133+_98-967.norm.txt",
"0176_6496_h133+_98-0679.norm.txt","0176_6631_h133+_99-1017.norm.txt",
"0176_6499_h133+_99-0077.norm.txt","0176_6500_h133+_99-0055.norm.txt",
"0176_6633_h133+_00-151.norm.txt","0176_6594_h133+_98-985.norm.txt",
"0176_6634_h133+_99-1067.norm.txt","0176_6595_h133+_98-821.norm.txt",
"0176_6635_h133+_99-301.norm.txt","0176_6596_h133+_98-853.norm.txt",
"0176_6636_h133+_99-137.norm.txt","0176_6597_h133+_99-927.norm.txt",
"0176_6640_h133+_98-343.norm.txt","0176_6599_h133+_98-506.norm.txt",
"0176_6641_h133+_98-186.norm.txt","0176_6600_h133+_99-1033.norm.txt",
"0176_6643_h133+_98-723.norm.txt","0176_6645_h133+_98-197.norm.txt",
"0176_6603_h133+_98-401.norm.txt","0176_6604_h133+_96-3.norm.txt")

vals.test <- matrix(NA,nrow=200,ncol=43)
for(i in 1:43)
{
vals.test[,i] <- read.table(TEST[i],comment.char="")[sig.n,3]
}

colnames(vals.test) <- c("S","A","A","S","S","S","S","A","S",
"A","A","S","A","A","A","S","A","S","A","S","S","S","A","A","A",
"S","A","S","A","S","A","S","S","A","S","A","S","A","A","A","A",
"S","A")

heatmap(cbind(vals.test,vals.train,upc,NORM))

cancer.upc <- matrix(NA,nrow=200,ncol=51)
cancer.upc <- cbind(vals.test,vals.train)
```

```
colnames(cancer.upc) <- c(colnames(vals.test),colnames(vals.train))
percent <- NULL
for(i in 1:ncol(cancer.upc))
{
percent[i] <- mean(round(cancer.upc[,i],0)==round(upc,0))
}
UPC <- as.matrix(upc)
colnames(UPC) <- ("UPC")
on <- cancer.upc[,which(percent > .65)]
colnames(on) <- rep("ON",ncol(on))
off <- cancer.upc[,which(percent < .45)]
colnames(off) <- rep("OFF",ncol(off))
marg <- cancer.upc[,which(percent > .45 & percent < .65)]
colnames(marg) <- rep("MAR",ncol(marg))
heatmap(cbind(UPC,on,marg,off),
main="Classification Heatmap",Rowv=NA)
heatmap(cbind(UPC,cancer.upc),
main="Classification Heatmap")

cbind(percent>.65,percent < .45,colnames(cancer.upc))
sum(percent>.65&colnames(cancer.upc)=="A") #25
sum(percent>.65&colnames(cancer.upc)=="S") #20
sum(percent<.45 &colnames(cancer.upc)=="A") #1
sum(percent<.45 &colnames(cancer.upc)=="S") #3

plot(rep(1,51),percent,ylab="Percent Concordance with UPC",xlab=""
,axes=F,ylim=c(.2,.9))
```

```
axis(side=2,at=(seq(.2,1.0,by=.1)),labels=seq(.2,1.0,by=.1))
abline(h=.45,col='red')
abline(h=.65,col='red')
title("Classification of Lung Cancer Samples")
text(1.25,.8,"Activated RAS")
text(1.25,.6,"Moderate")
text(1.25,.4,"Inactive RAS")
```