Low-level Analysis of High-density Oligonucleotide Array Data: Background, Normalization and Summarization

by

Benjamin Milo Bolstad

M.Sc. (University of Waikato) 1998

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Biostatistics

in the

GRADUATE DIVISION
of the
UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:
Professor Terence P. Speed, Chair
Assistant Professor Sandrine Dudoit
Professor John Ngai

Spring 2004

The dissertation of Benjamin Milo Bolstad is approved:

Chair Date

Date

Date

University of California, Berkeley

Spring 2004

Low-level Analysis of High-density Oligonucleotide Array Data: Background, Normalization and Summarization

Copyright 2004
by
Benjamin Milo Bolstad

Abstract

Low-level Analysis of High-density Oligonucleotide Array Data: Background, Normalization and Summarization

by

Benjamin Milo Bolstad

Doctor of Philosophy in Biostatistics

University of California, Berkeley

Professor Terence P. Speed, Chair

Microarray experiments are currently widely applied in many areas of biomedical research. The Affymetrix GeneChip® system is a commercial high-density oligonucleotide microarray platform which measures gene expression using hundreds of thousands of 25-mer oligonucleotide probes. This dissertation addresses how probe intensity data from GeneChips® are processed to produce gene expression values and shows how better pre-processing leads to gene expression measures, that after further analysis, yield biologically meaningful conclusions. An ideal expression measure is one which is both precise and accurate.

A three-stage procedure for producing an expression measure is proposed. For each of the three stages, background correction, normalization and summarization, numerous methods are developed and assessed using spike-in datasets. Bias and variance criteria are used to compare the different methods of producing expression values. The methods are also judged by how well they correctly identify the differential genes. The background method has a significant effect on the bias, which is reduced, and the variability, which is usually increased. Non-linear normalization methods are found to reduce the non-biological variability between multiple arrays without introducing any significant bias. Robust multi-chip linear models are found to fit the data well and provide the recommended summarization method.

The summarization methodology is extended to produce test statistics for determining differential genes. These test statistics perform favorably at correctly detecting differential genes when

compared with alternative methods based on expression values. Finally, using case study data, no statistical benefit is found for using arrays hybridized with mRNA from a pool rather than from a single biological source.

Professor Terence P. Speed
Dissertation Committee Chair

To Judy

To My Parents

Contents

List of Figures

List of Tables

Acknowledgments

First, I'd like to acknowledge Terry Speed, whom has served has the adviser for the research described in this dissertation. He supported me through the many trials and tribulations of finishing this degree and put up with me delaying my qualifying exam for much longer than I should have. For this I am thankful and I look forward to possible future collaborations.

Francois Collin, an RMA collaborator, fellow student and all around nice guy. I look forward to reading your dissertation one day soon.

Rafael Irizarry, the other main RMA collaborator, has been a pleasure to work with. Particularly, the many discussions about software and helping to get me involved in Bioconductor.

This dissertation received a decidingly thorough proof reading from Julia Brettschneider for which I am very grateful. I also thank Julia for using and giving feedback on much of the software implementing the routines discussed in within. Also for putting up with my disparaging comments about the quality of software support at the SCF.

I owe a debt of gratitude to Sandrine Dudoit and Jean Yee Hwa Yang for recruiting me to work on SMA without which I may not have got involved in microarray data analysis. Thanks for giving me a future career.

I also wish to thank Yu Chaun Tai, Karen Vranizan and Yun Zhou for all using my R packages and other assorted software. Perhaps my propensity for rapidly answering email encouraged more questions than I desired, but it was a pleasure answering each and every one. A thank you to Nusrat Rabbee who proofread portions of this dissertation.

Mark Vawter, Prabhakara Choudary, Simon Evans, Jun Li, Hiroaki Tomita, Fan Meng and many others from the Pritzker Consortium were early and heavy users of the probe-level modeling methodology. I appreciate the feedback and many questions I received which helped the formulation of nicer descriptions about the methodology.

I would like to thank my parents Bill and Sylvie Bolstad for supporting my education throughout my life. As much as I tried to fight the call of the genes I ended up as a statistician. Perhaps one day the methods within this dissertation will help identify the statistic gene. Also a big thank you to my sister Rachel for putting up with arguments about the philosophy of science, the meaning of

life and many other completely inane topics.

Finally, I must express my most heartfelt gratitude to Judy Pang Bolstad. Without functioning as both the primary editor and a source of emotional support this dissertation might never have been finished. Thanks for putting up with many nights where I looked like I was glued to the computer and seemed like I was completely ignoring you, which I wasn't. I look forward to many years of post-PhD joy in our life together. Words alone can not truly convey how I feel about you.

Chapter 1

Introduction

This chapter introduces microarrays and gene expression. Section 1.2 describes the Affymetrix GeneChip® technology. Section 1.3 explains the topic of low-level analysis and provides an outline of this dissertation. In Section 1.4, two diagnostic plots that are used throughout the dissertation are described.

1.1 Introduction

Today, microarrays are becoming widely used in many areas of biomedical research. A microarray is a device designed to simultaneously measure the expression levels of many thousands of genes in a particular tissue or cell type. There are numerous different microarray technologies, including the cDNA arrays developed at Stanford (DeRisi et al., 1996), (Brown and Botstein, 1999) and the high-density oligonucleotide arrays produced by Affymetrix (Lockhart et al., 1996). This dissertation focuses on the analysis of data from the Affymetrix technology.

The high-density oligonucleotide array system produced by Affymetrix is known as the Affymetrix GeneChip® . In this disseration, we generally refer to GeneChips as arrays or chips. Discussions of how high-density oligonucleotide arrays can be used to measure gene expression are provided in Lockhart et al. (1996) and Lockhart and Winzeler (2000). Some of the earliest published studies where gene expression was monitored using Affymetrix GeneChip® arrays are Golub et al. (1999) and Winzeler et al. (1999).

Microarrays can be applied to the problems of gene discovery, the diagnosis of diseases, pharmacogenomics and toxicogenomics among others. Gene discovery is the process of finding genes that are differentially expressed between tissues from different conditions. When given expression profiles for a diseased and non-diseased tissues, a new sample can be diagnosed by measuring its expression profile and comparing it with the reference profiles. For more on diagnosis using arrays see the review article by Simon (2003). Pharmacogenomics is the process of discovering how a therapeutic response from a drug affects the expression profile of a patient (Regalado, 1999). More specifically, pharmacogenomics seeks to answer such questions as: why does a drug work better in some patients and not others? why is a drug toxic for some people? More about pharmacogenomics and microarrays is provided in Chin and Kong (2002) and Chicurel and Dalma-Weiszhausz (2002). Toxicogenomics is the study of how exposure to toxicants affects the genetic profiles of the exposed tissues: see Nuwaysir et al. (1999).

1.1.1 DNA and Gene Expression

The genetic material that contains the instructions for most organisms is known as *deoxyribonucleic acid* (DNA). DNA is composed of nucleotides, with each nucleotide itself consisting of three components: a base, a sugar and a phosphate. The nucleotides are joined together in long chains. The backbone of these chains consist of the sugar and phosphates, while individual bases hang off each sugar. There are four different bases: *adenine*, *cytosine*, *guanine* and *thymine*. These are usually known by the letters A, C, G, and T respectively. A DNA molecule consists of two complementary polynucleotide chains held together using hydrogen bonding. In particular, the bases A and T bind together, as do C and G. In this manner, we say that A is the *complement* of T, and C is the complement of G. The two sugar-phosphate strands form a double helix structure. DNA strands are typically millions of nucleotides in length. Each strand has a polarity such that a 5'-hydroxyl group begins the first nucleotide in the strand and a 3'-hydroxyl group ends the last nucleotide in the strand. Since the two strands are complementary one will run 5' to 3' and the other 3' to 5'.

RNA, *ribonucleic acid*, differs from DNA in several ways. Specifically, the sugar is ribose rather than deoxyribose and the base *uracil*, U, takes the place of thymine. U is also complementary to A. Unlike DNA, most RNA molecules are single-stranded and only 75-5000 nucleotides in length. Cells contain several types of RNA: messenger RNA (mRNA), transfer RNA (tRNA) and ribosomal RNA (rRNA).

A *gene* is a sequence of DNA that codes for a *protein*. The protein in turn controls a physical trait of the cell, for example eye or hair color. A strand of DNA contains many different genes. Proteins are sequences of twenty different types of *amino acids*. Each amino acid is encoded by a sequence of three bases. These three base groups are called *codons*. Although there are 64 possible triplets, there is some redundancy because several different codons code the same amino acid. There are 3 codons which do not encode to any amino acid. Instead, they serve as “stop” signals.

The process of synthesizing proteins from DNA occurs in two stages: *transcription* and *translation*. These are processed are collectively known as the *central dogma of molecular biology*. The first stage, transcription, is the transfer of information from the double-stranded DNA molecule to the single-stranded mRNA. An enzyme called *RNA polymerase* moves along one strand of DNA from the 5’ to the 3’ direction encoding the complementary sequence as an RNA strand. The strand of DNA from which the RNA is encoded is called the *antisense* strand and the other strand is called the *sense* strand. The mRNA is complementary to the antisense strand, and except that base T is changed to U is otherwise identical to the sense strand. Transcription begins at regions of the sequence known as *promoter* sites and ends at regions known as *terminator* sites.

The second stage, translation, is the process of translating the mRNA into a protein. This occurs using tRNA and rRNA. The codon, AUG, marks the location where translation should start. The tRNA molecules attach amino acids to the chain as an rRNA molecule moves along the mRNA. The process continues until one of the stopping codons is reached. At this point, the protein is complete and can serve its purpose in the cell.

More details on the biology of translation and transcription can be found in Gonick and Wheelis (1991) and Berg et al. (2002). Collectively, this process of converting a DNA sequence to a protein is called *gene expression*. For a particlar organism the DNA content of most cells is the same. In other words, the DNA for every gene is present in all the cells of that organism. However, the amount of mRNA and the proteins to which the mRNA are translated to varies between cells and also varies within a cell under different conditions. For example, consider two cell types: A and B. If we suppose that in cell A, genes 1 and 2 are transcribed into mRNA and then translated into proteins. but that in cell B, only gene 2 is transcribed into mRNA and translated into mRNA, but at a higher rate than in cell A. Then, we would say that gene 1 is expressed in cell A, but not in cell B and that Gene 2 was expressed at a higher rate in cell B than in cell A.

By studying which genes are expressed and which are not, in different kinds of cells or under

different experimental or environmental conditions, we can learn about how these genes affect the function of the cells. Traditionally, gene expression studies were done one gene at a time using technologies such as RT-PCR and Northern blots. The more recent development of microarray technologies allows the simultaneous measurement of the expression level of thousands of genes.

Since the focus of a gene expression study is in the function of genes, the interest should be in the function of the proteins. However, DNA microarrays focus on measuring the level of mRNA rather than differences in the levels of proteins. The assumption being made is that most mRNA gets translated into a protein. Dealing with protiens is much more complex than dealing with mRNA. There are methods for more directly monitoring protein expression, such as Western blots, 2d gels, and protein microarrays. However, this dissertation focuses only on DNA microarrays.

1.2 Affymetrix GeneChip® Technology

This section introduces the terminology used to describe GeneChips, how they are constructed and the workflow for generating raw data. General overviews of the technology are provided by Lipshutz et al. (1999) and Warrington et al. (2000). More detailed information about sample preparation, hybridization, scanning and basic analysis can be found in the Affymetrix Microarray Suite Users Guide (Affymetrix, 2001a) and the GeneChip® Expression Analysis Technical Manual (Affymetrix, 2003).

1.2.1 Some Basic Definitions

In order to produce a GeneChip array it is imperative that the sequence of the target organism is known. However, this does not present a particular difficulty because a number of organisms have now been completely sequenced and others are currently being sequenced. When given a known sequence, a number of 25-mer sequences complementary to the sequence for target genes are chosen. These sequences are known as *probes*. Typically 11 to 20 probes interrogate a given gene. This collection of probes is called a *probeset* and there are between 12,000 to 22,000 probesets on an array. Figure 1.1 illustrates the relationship between probes and probesets. Affymetrix uses a number of procedures to select which 25-mer sequences should be used for each gene. In particular, potential probes are examined for specificity, potential for cross hybridization and predicted bind-

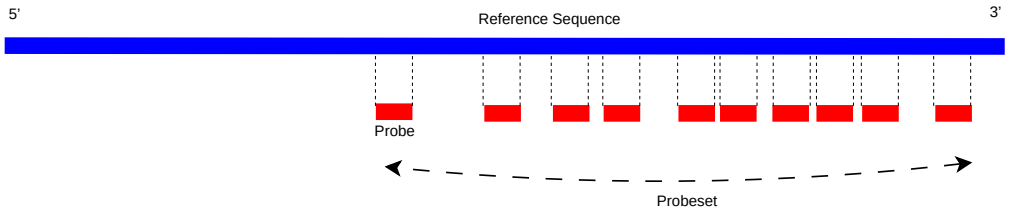

Figure 1.1: Multiple probes interrogating the sequence for a particular gene make up probesets.

Figure 1.2: Pefect Match and Mismatch Probes.

ing properties. Cross-hybridization occurs when a single stranded DNA sequence binds to a probe sequence which is not completely complementary. To match the properties of the sample amplification procedure, probes are 3' biased. That means that probes are chosen closer to the 3' end of the sequence. However, the probes are typically spaced widely along the sequence. More details about probe selection are described in Mei et al. (2003). Sometimes there is more than one probeset that interrogate the same gene, but each uses a different part of the sequence.

On a GeneChip there are two types of probes. A probe that is exactly complementary to the sequence of interest is called a *Perfect Match* (PM). A probe that is complementary to the sequence of interest except at the central base, which for 25-mers is the 13th base, is known as the *Mismatch* (MM). Examples of PM and MM probes are given in Figure 1.2. In theory, the MM probes can be used to quantify and remove non-specific hybridization. A PM and its corresponding MM probe are referred to as a *probe pair*.

1.2.2 Chip Manufacturing Process

Affymetrix GeneChips® are fabricated using a photolithographic procedure. By using a series of masks, 25-mer oligonucleotide probes are synthesised onto a wafer in such a manner that a large number of different sequences can be produced in parallel in a small number of steps (Fodor et al., 1991), (Fodor et al., 1993), (Pease et al., 1994). Figure 1.3 shows how this procedure is carried out. First, a 5 square inch quartz wafer is bathed in silane to produce a matrix of

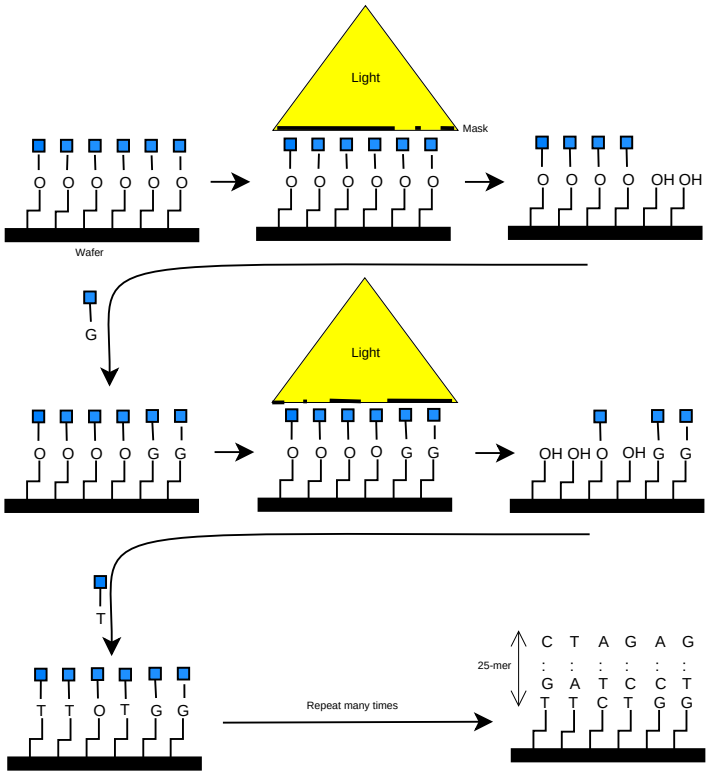

Figure 1.3: The Affymetrix GeneChip® is constructed using a photolithographic process. A series of masks are used to deprotect different locations and base by base oligonucleotides of length 25 are built in parallel.

covalently linked molecules attached to the surface. The density of these molecules determines the packing density of the probes. Photo-sensitive capped linkers are then attached to the silane matrix. A mask is then introduced with openings at predetermined locations. When UV light is shone through the mask, the exposed linkers become deprotected and available for binding. Once the desired locations are deprotected, a solution of a deoxynucleotide of the desired base (either A, T, C or G) with a photosensitive protection group is washed over the surface. At the unprotected locations, the nucleotide attaches to the surface or the end of the oligonucleotide. At the next stage, a different mask is placed over the wafer and another set of locations are deprotected. A solution of a different base is then flushed over the surface and binds at the exposed locations. The procedure is repeated until all probe locations reach 25 nucleotides in length. At each probe location there are millions of copies of the same oligonucleotide. It is important to note that each probe location is called a *probe cell*. Because there are only 4 possible bases for each location at each iteration a maximum number of $4N$ masks are needed to produce oligonucleotides of length N.

Earlier chips had all the probes for each probeset located contiguously on the array. However, if there are any spatial defects, this will create problems for entire probesets. Newer chips have the probes for each probeset spread out across the array to avoid these problems. A PM and MM probe pair are always adjacent on the array.

Once the wafer has been fully synthesized, it is deprotected and then diced into pieces, each an individual array. A single wafer can produce from 49-400 arrays depending on the feature size and number of probes per array. The resulting individual arrays are packaged into cartridges and are ready for use. Current typical arrays have between 500,000 (HGU95Av2) and 1,300,000 (HGU133 plus 2.0) probe locations. The current feature sizes are 11μm for HGU133 plus 2.0 arrays, 18μm for HGU133A arrays and 20μm for HGU95Av2 arrays.

There are currently 26 mass produced arrays or array sets commercially available from Affymetrix. These include arrays for the organisms Humans, Mice, Rat, Arabidopsis, Drosphilla, Yeast, Zebrafish, Canine and E.coli among others. It is also possible to purchase custom arrays with user desired sequences on the array.

1.2.3 Sample Preparation and Hybridization

Figure 1.4 highlights the sample preparation process for eukaroytes. The process begins with total RNA (or poly-A mRNA) isolated from the source tissue or cell line. The total RNA is then reverse transcribed to produce double-stranded cDNA using a series of reagents. A cleaning procedure is then carried out on the cDNA. Next biotin labeled cRNA is produced from the cDNA. After another cleaning procedure, the biotin labeled cRNA is fragmented. The fragments are typically 25-200 bases in length.

A number of controls are also used for Affymetrix arrays. Control Oligo B2 hybridizes to locations on the edges and corners of the array. BioB, bioC, bioD and cre are *E. coli* genes that are added at specified concentrations to check how well the hybridization, washing and staining procedures have performed. An additional five genes from *B. subtilis*, dap, thr, trp, phe, lys, are also used as controls.

The fragmented biotin labeled cRNA along with the controls, are mixed to form a hybridization cocktail. The array cartridge is then filled with the mixture and placed in a hybridization oven for 16 hours.

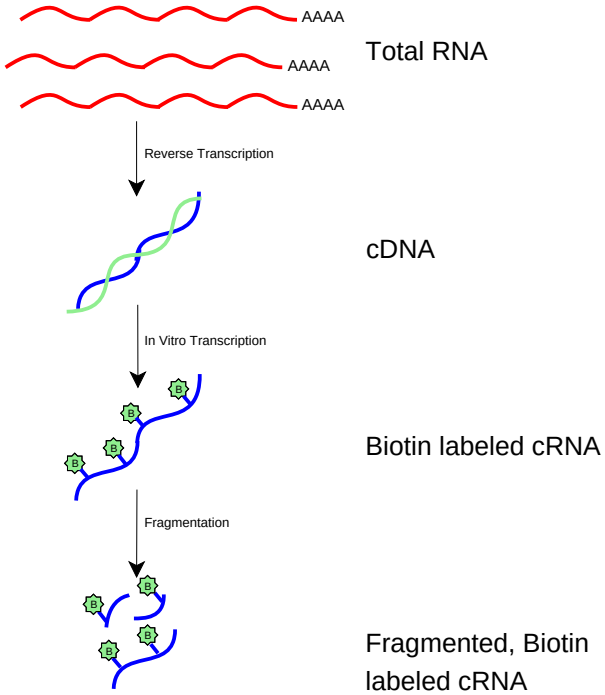

Figure 1.4: Typical target sample preparation for Eukaroytic organisms.

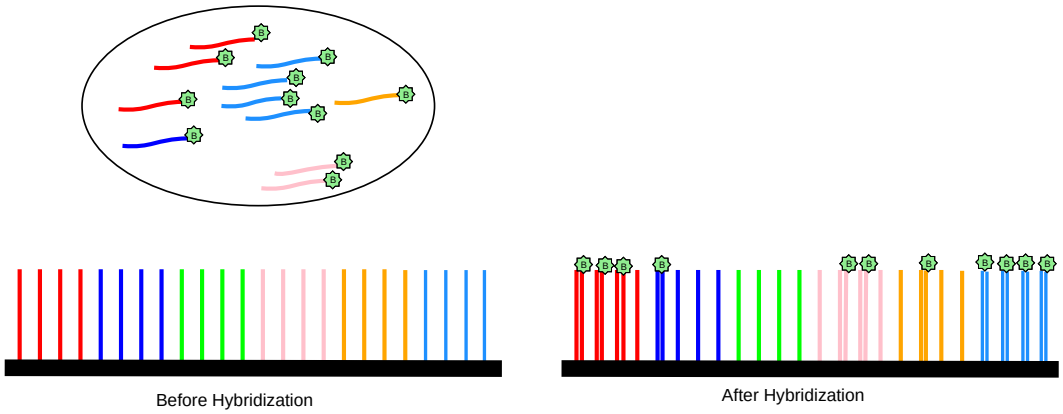

Figure 1.5: During the hybridization process cRNA binds to the array.

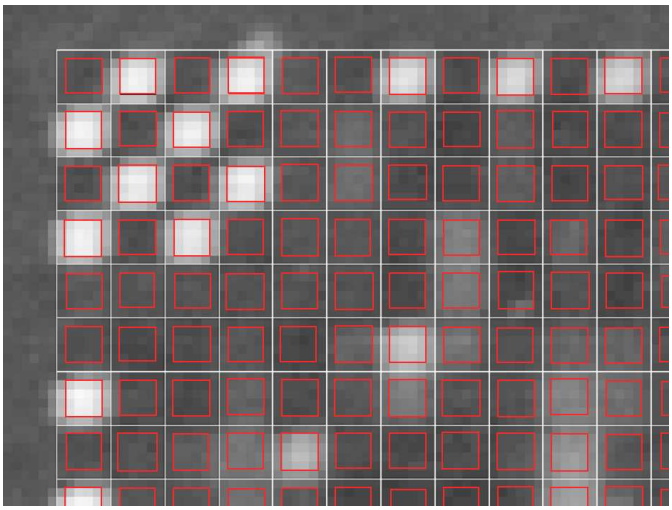

Figure 1.6: A section of a scanned image for chip. Image quantification takes place by gridding.

Figure 1.5 shows what happens during the hybridization procedure. Utilizing the complementary binding properties of DNA, the fragmented cRNA and controls bind to the oligonucleotides on the array. If there is more cRNA for a particular gene in the hybridization cocktail, then after hybridization there should be more material attached to the probes corresponding to that gene.

After hybridization, non-hybridized cRNA is removed from the cartridge and the array is placed in a fluidics station. Then, a series of washing and staining steps are applied to the array. The fluorescent staining agent streptavidin-phycoerythrin (SAPE) binds with the biotin labeling on the cRNA.

1.2.4 Scanning and Image Quantification

After the washing and staining process the array is removed from the fluidics station and placed in a scanner. Laser light is shone onto the array and excites the fluorescent staining agent. At locations where more cRNA hybridized a brighter signal should be emitted. The amount of signal emitted is recorded as a value in 16 bits, and by examining the entire chip an image is produced. The Affymetrix software stores this image in the *DAT* file.

Figure 1.6 shows a portion of an image for an array. The checker board pattern and bright spots on the edges correspond to control oligo B2 probes. These are used to superimpose and align a grid upon the image. Once the gridding has taken place, the border pixels are ignored and the internal pixels of each grid square are used to compute a probe intensity. In particular, the 75th percentile of

the intensities for these pixels gives the probe intensity for each probe cell. These probe intensity values are written into the *CEL* file. All of the analyses in this dissertation begin with data read from the CEL file.

1.2.5 Computing Expression Measures for High-density Oligonucleotide Data

The next step after reducing the image data to a CEL file is to process the data to produce expression values. Specifically, the PM and MM probe intensities for each probeset must be combined together to produce a summary value. Originally, this was done using the average difference (AvDiff) algorithm (Affymetrix, 1999). After a slight background adjustment, the intensity values for each MM were subtracted from the corresponding PM intensity, and for each probeset the average of these differences was taken. AvDiff had several drawbacks, for example it was noisy for low intensities and gave negative values. To remedy these problems Affymetrix proposed a new algorithm, now commonly referred to as MAS 5.0 (Affymetrix, 2001a). This no longer yielded negative values and made attempts to be more robust. However, as will be discussed in later chapters, it was still not completely satisfactory.

Numerous alternative methods of computing expression measures have been proposed. The most popular of these include the Model Based Expression Index (MBEI) (Li and Wong, 2001a) and the Robust Multi-chip Average (RMA) (Irizarry et al., 2003a), (Irizarry et al., 2003b). In addition, a framework for comparing expression measures has been given by affycomp (Cope et al., 2004). A major portion of this dissertation focuses on methods for producing expression measures.

1.3 Dissertation Outline

Low-level analysis of high-density oligonucleotide arrays involves the manipulation and modelling of probe intensity data. The goal of low-level analysis is to produce more biologically meaningful expression values. Ideally, expression values should be both precise (low variance) and accurate (low bias). A primary goal is to determine which genes are differentially expressed between treatment conditions. Other topics in low-level analysis include determining whether a gene is being expressed in a given tissue (prescence/absence) and also array quality assessment diagnostics. The focus of this dissertation will be on computing expression values. Another motivation for low-level

analysis is that it is possible that information becomes lost when moving from probe-level data to expression meaures.

A low-level analysis does not typically attempt to directly answer a question of biological interest, for example topics such as determining gene function, cell cycle studies and pathway analysis. Instead these are usually addressed by high-level analysis. A low-level analysis of the data should provide better expression measures which can be used in higher level analyses.

The following six chapters of this dissertation address topics in low-level analysis. The first four chapters examine procedures for constructing gene expression measures. Chapter 2 covers the topic of background correction which is an adjustment made on a chip by chip basis to signal intensities. Chapter 3 investigates procedures for normalization, which aims to reduce variability of non-biological origin between multiple arrays. Summarization, which is the process of combining the multiple probe intensities for each probeset to a single gene expression value, is studied in Chapter 4 and a three-stage framework for computing expression values is presented in Chapter 5. Each of these chapters considers numerous different methods of pre-processing array data and producing expression measures. To assess the effect that each method has on the resultant expression measures, series of spike-in datasets will be used. A spike-in dataset has RNA for particular probesets spiked-in at known concentrations. This gives a “truth” by which each method can be judged.

Next, Chapter 6 considers methods for detecting differential expression. Specifically, methods based on probe-level models are compared with methodologies based on gene expression measures. Since finding differentially expressed genes is often of primary interest in a microarray experiment, it is important to have a method which can correctly identify differential genes, without incorrectly identifying non-differential genes.

Finally, Chapter 7 is a case study on the effect that pooling has on gene expression estimates. It has been suggested that pooling reduces biological variability. Using 36 mice arrays, some that are hybridized using pooled mRNA and others that are hybridized from a single source, we investigate whether pooling is effective.

1.4 Two Important Diagnostic Plots

1.4.1 *MA* plots

The *MA*-plot has become a widely used tool in microarray analysis. It has been applied as a part of a number of normalization procedures. The papers by Dudoit et al. (2002), Yang et al. (2002) and Bolstad et al. (2003) provide more details about normalization. *MA*-plots are typically used to compare two color channels, two arrays or two groups of arrays. The vertical axis is the difference between the logarithms of the signals(the log ratio) and the horizontal axis is the average of the logarithms of the signals. The *M* stands for *minus* and the *A* for *add* (Smyth et al., 2003). Conveniently, *MA* is also mnemonic for *microarray*.

Some microarray researchers, such as Quackenbush (2002) and Cui et al. (2003), have referred to these as Ratio-Intensity (RI) plots. In wider statistical and medical literature, plots of differences against means are commonly known as Bland-Altman plots (Altman and Bland, 1983), (Bland and Altman, 1986). The concept of testing for agreement between two samples using means and differences traces as far back as Pitman (1939). A discussion of such plots along with the effect of different data transformations is given in Hawkins (2002). Another similar plot is the Tukey mean-difference plot, which plots difference versus average of the quantiles (Chambers et al., 1983). However, this dissertation does not use quantiles and exclusively uses the term *MA*-plot in preference to the alternatives.

The *MA*-plots in this dissertation have been constructed in the following manner: Let X_{ij} be intensity i on array j. To compare the two arrays j and k, the M and A values are computed by $M_i = \log_2(X_{ij}) - \log_2(X_{ik})$ and $A_i = \frac{1}{2}(\log_2(X_{ij}) + \log_2(X_{ik}))$. Since we work with logged expression values, these are typically just the difference and average respectively. The base 2 logarithm is used for convenience so that a unit change in M represents 2 fold-change in expression and a unit change in A represents a doubling of brightness. Because the probe-intensities are measured using a 16 bit image, the maximum possible value of A is 16.

Figure 1.7 shows how two arrays can be compared. On the left, we compare the two arrays by plotting intensities from one against the intensities on the other. The figure on the right compares the two arrays using an *MA*-plot. While there is a visible difference between the two arrays from the first plot, the non-linear trend is not apparent. Since there are more lower probe intensities than

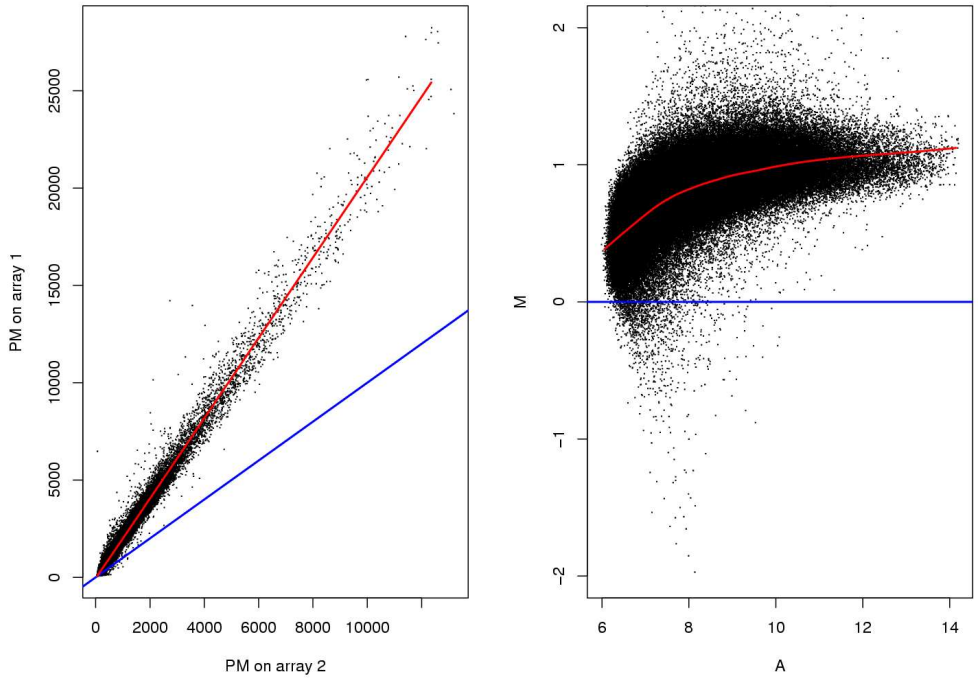

Figure 1.7: Comparing PM intensities from two replicate arrays By either plotting one array against the other or by using an *MA*-plot.

higher intensities, the log transformation allows us to more easily assess the behavior across all intensities. In the first plot, there is a tendency to view the trend based upon the higher intensities which are spread over a greater range, but it is hard to discern trends in the low intensity point cloud. The *MA*-plot makes the relationship between the arrays much easier to visually assess.

In this dissertation, we use *MA*-plots to compare the performance of various methods of computing expression measures. For non-differential probesets, it is ideal for the *MA*-plot to be tight around $M = 0$ across all intensities. A lowess curve (Cleveland, 1979) fitted to an *MA*-plot shows whether the M values are centered around 0 at each intensity value. The spread of the point cloud around the lowess curve allows us to measure the variability.

1.4.2 Receiver Operating Characteristic curves

Receiver Operating Characteristic (ROC) curves were introduced in the 1950s as a tool for deciding whether radio signals were noise or both noise and signal (Peterson et al., 1954). More recently they have gained widespread use in medical decision-making, for more see Lusted (1971), Swets (1988), Begg (1991) and Campbell (1994).

An ROC curve is a method by which we can assess the performance of a test. On the vertical axis we plot the rate of true positives, i.e. correct rejection of null hypotheses when they are false. On the horizontal axis we plot the rate of false positives, i.e. incorrectly rejecting true null hypotheses. An ideal test would give 100% true positives without any false positives.

Figure 1.8 shows ROC curves for three tests. A desirable test identifies more true positives for any level of false positives. Therefore, when examining ROC curves we judge the test with the highest curve as being the best method. In Figure 1.8 both Test2 and Test3 would be judged better tests than Test1. Sometimes the ROC curves cross, as Test2 and Test3 do in the figure. In this case, Test3 is better until the 20% false positive rate. After this point, Test2 is better. One method of comparing tests is to use the area under the curve (AUC). Tests with higher AUC are judged as being better.

In this dissertation, ROC curves are used to compare methods of detecting differentially expressed genes. In situations where it is known which probesets are differential and which are not, ROC curves are created by thresholding the test statistic at various levels and counting the number of differential genes correctly identified (the true positives) and the number of non-differential genes

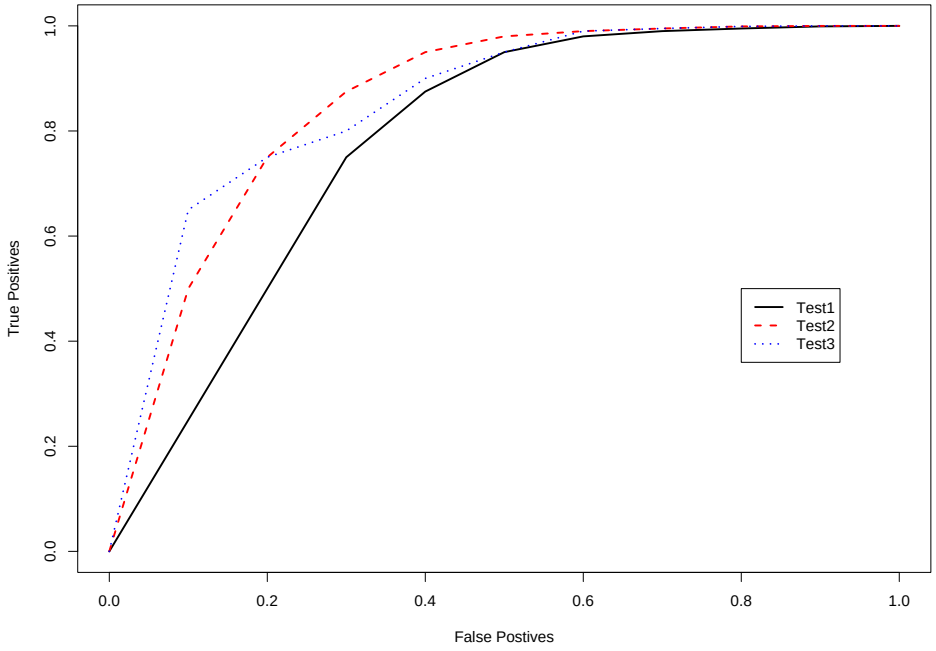

Figure 1.8: An ROC curve comparing three tests.

incorrectly identified as being differential (the false positives). Methods which correctly select more differential genes and fewer non-differential genes are judged to be better.

Chapter 2

Background Correction and Signal Adjustment

This chapter discusses background correction methods. Section 2.1 provides a clear definition of what constitutes a background correction/signal adjustment method. Section 2.2 describes some of the proposed background and signal adjustment methods. Section 2.3 compares the methods and Section 2.4 discusses the results of the comparison.

2.1 Introduction

The term *background correction*, also referred to as signal adjustment, describes a wide variety of methods. More specifically, a background correction method should perform some or all of the following:

1. Corrects for background noise and processing effects.
2. Adjusts for cross hybridization which is the binding of non-specific DNA (i.e. non-complementary binding) to the array.
3. Adjusts expression estimates so that they fall on the proper scale, or are linearly related to concentration.

It is important to note that this definition is somewhat broader than is often used in the wider community. Many times only methods dealing with the first problem have been referred to as background correction methods.

Unlike other array systems, such as cDNA microarrays, where pixels surrounding a spot can be used to compute the background adjustment, the probe intensities themselves must be used to determine any adjustment for Affymetrix Genechips. This is because probe locations are very densely spaced on the array.

2.2 Background Correction / Signal Adjustment Methods

2.2.1 RMA Convolution Model

The RMA convolution model background correction method is motivated by looking at the distribution of probe intensities. Figure 2.1 shows the probe intensity distribution for a group of typical arrays. We model the observed intensity as the sum of a signal and a background component. In particular, our model is that we observe $S = X + Y$, where X is signal and Y is background. Assume that X is distributed $\exp(\alpha)$ and that Y is distributed $\mathrm{N}\left(\mu, \sigma^2\right)$, with X and Y independent. Furthermore, assume that $Y \geq 0$ to avoid producing negative values. Thus, Y is normally distributed with truncation at 0. This model is motivated by the observed probe densities in Figure 2.1. Under this model the background corrected probe intensities will be given by $E(X|S = s)$. A formula for this quantity is derived below.

We define $\Phi(z)$ and $\phi(z)$ as the standard normal distribution function and density function respectively. More specifically

$$\Phi(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}w^2\right) dw$$

and

$$\phi(z) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}z^2\right).$$

Remembering that we observe only $S = X + Y$, under the conditions of this model, the density of the joint distribution of X and Y is given by

$$f_{X,Y}(x,y) = \alpha \exp(-\alpha x) \frac{1}{\sigma} \phi\left(\frac{y-\mu}{\sigma}\right) \text{ when } y > 0,\ x > 0$$

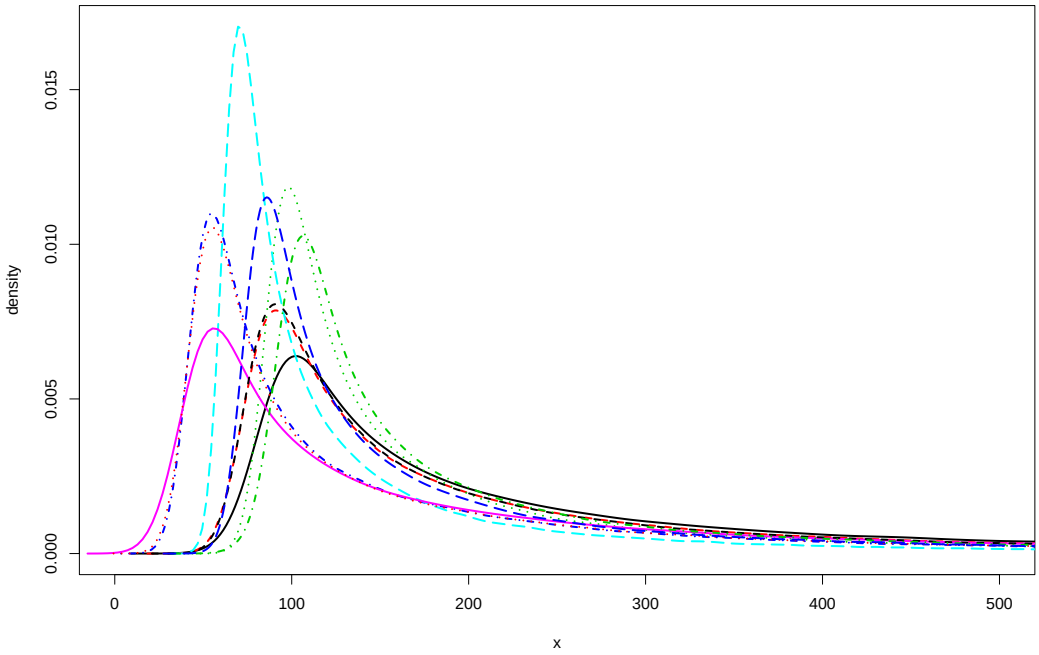

Figure 2.1: Smoothed histograms of the probe intensities for a number of arrays from the HGU95A spike-in dataset.

Then, we get the joint distribution of X and S from

$$f_{X,S}(x,s) = f_{X,Y}(x,s-x)|J|$$

where J is the Jacobian of the transformation. Now, $|J| = 1$ and so the joint distribution of X and S is

$$f_{X,S}(x,s) = \alpha \exp(-\alpha x)\frac{1}{\sigma}\phi\left(\frac{s-x-\mu}{\sigma}\right) = \alpha \exp(-\alpha x)\frac{1}{\sigma}\phi\left(\frac{x-s+\mu}{\sigma}\right)$$

The conditional distribution of X given S is

$$f_{X|S}(x|s) = \frac{f_{X,S}(x,s)}{\int_0^s f_{X,S}(x,s)\,dx}$$

where the denominator (the marginal pdf of S) is

$$\int_0^s \alpha \exp(-\alpha x)\frac{1}{\sigma}\phi\left(\frac{x-s+\mu}{\sigma}\right)dx$$

Let $w = \frac{x-s+\mu}{\sigma}$ so that $\sigma dw = dx$ and $x = \sigma w + s - \mu$. Making the substitution, the integral becomes

$$\begin{aligned}
&\int_{\frac{-s+\mu}{\sigma}}^{\frac{\mu}{\sigma}} \alpha \exp(-\alpha(\sigma w + s - \mu))\phi(w)\,dw \\
&= \alpha \exp(-\alpha(s-\mu)) \int_{\frac{-s+\mu}{\sigma}}^{\frac{\mu}{\sigma}} \exp(-\alpha\sigma w)\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}w^2\right)dw
\end{aligned}$$

Now, we consider the integral on the right hand side

$$\begin{aligned}
&\int_{\frac{-s+\mu}{\sigma}}^{\frac{\mu}{\sigma}} \frac{1}{\sqrt{2\pi}}\exp(-\alpha\sigma w)\exp(-\frac{1}{2}w^2)dw \\
&= \int_{\frac{-s+\mu}{\sigma}}^{\frac{\mu}{\sigma}} \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left(w^2 + 2\alpha\sigma w\right)\right)dw \\
&= \exp(\frac{1}{2}\alpha^2\sigma^2)\int_{\frac{-s+\mu}{\sigma}}^{\frac{\mu}{\sigma}} \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left(w^2 + 2\alpha\sigma w + \alpha^2\sigma^2\right)\right)dw \\
&= \exp(\frac{1}{2}\alpha^2\sigma^2)\int_{\frac{-s+\mu}{\sigma}}^{\frac{\mu}{\sigma}} \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}(w+\sigma\alpha)^2\right)dw
\end{aligned}$$

Let $z = w + \sigma\alpha$ and then the integral becomes

$$\int_{\frac{-s+\mu}{\sigma}+\alpha\sigma}^{\frac{\mu}{\sigma}+\alpha\sigma} \frac{1}{\sqrt{2\pi}}\exp(-\frac{1}{2}z^2)dz = \Phi\left(\frac{s-\mu-\alpha\sigma^2}{\sigma}\right) + \Phi\left(\frac{\mu+\alpha\sigma^2}{\sigma}\right) - 1$$

and the denominator is

$$\alpha \exp\left(\frac{1}{2}\alpha^2\sigma^2 - \alpha(s-\mu)\right)\left[\Phi\left(\frac{s-\mu-\alpha\sigma^2}{\sigma}\right) + \Phi\left(\frac{\mu+\alpha\sigma^2}{\sigma}\right) - 1\right]$$

thus,

$$
\begin{aligned}
f_{X|S}(x|s) &= \frac{f_{X,S}(x,s)}{\int_0^s f_{X,S}(x,s)\,dx} \\
&= \frac{\alpha \exp(-\alpha x)\,\frac{1}{\sigma}\phi\left(\frac{x-s+\mu}{\sigma}\right)}{\alpha \exp(\frac{1}{2}\alpha^2\sigma^2 - \alpha(s-\mu))\left[\Phi\left(\frac{s-\mu-\alpha\sigma^2}{\sigma}\right)+\Phi\left(\frac{\mu+\alpha\sigma^2}{\sigma}\right)-1\right]} \\
&= \frac{\exp\left(-\alpha x+\alpha(s-\mu)-\frac{1}{2}\alpha^2\sigma^2\right)\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{1}{2\sigma^2}(x-s+\mu)^2\right)}{\left[\Phi\left(\frac{s-\mu-\alpha\sigma^2}{\sigma}\right)+\Phi\left(\frac{\mu+\alpha\sigma^2}{\sigma}\right)-1\right]} \\
&= \frac{\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{1}{2\sigma^2}\left(x^2-2x(s-\mu)+(s-\mu)^2+2\sigma^2\alpha x-2\sigma^2\alpha(s-\mu)+\alpha^2\sigma^4\right)\right)}{\left[\Phi\left(\frac{s-\mu-\alpha\sigma^2}{\sigma}\right)+\Phi\left(\frac{\mu+\alpha\sigma^2}{\sigma}\right)-1\right]} \\
&= \frac{\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{1}{2\sigma^2}\left(x^2-2x(s-\mu-\alpha\sigma^2)+(s-\mu)^2-2(s-\mu)\sigma^2\alpha+\alpha^2\sigma^4\right)\right)}{\left[\Phi\left(\frac{s-\mu-\alpha\sigma^2}{\sigma}\right)+\Phi\left(\frac{\mu+\alpha\sigma^2}{\sigma}\right)-1\right]} \\
&= \frac{\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{1}{2\sigma^2}\left(x-\left(s-\mu-\alpha\sigma^2\right)\right)^2\right)}{\left[\Phi\left(\frac{s-\mu-\alpha\sigma^2}{\sigma}\right)+\Phi\left(\frac{\mu+\alpha\sigma^2}{\sigma}\right)-1\right]}
\end{aligned}
$$

Let $a = s-\mu-\sigma^2\alpha$ and $b=\sigma$

Therefore, the conditional distribution of x given S is

$$f(x|s) = \frac{\frac{1}{b}\phi\left(\frac{x-a}{b}\right)}{\left[\Phi\left(\frac{a}{b}\right)+\Phi\left(\frac{s-a}{b}\right)-1\right]}$$

and so

$$E(x|s) = \frac{1}{\Phi\left(\frac{a}{b}\right)+\Phi\left(\frac{s-a}{b}\right)-1}\int_0^s \frac{x}{b}\phi\left(\frac{x-a}{b}\right)dx$$

Let $z=\frac{x-a}{b}$ so $dz=\frac{dx}{b}$. Thus

$$
\begin{aligned}
\int_0^s \frac{x}{b}\phi\left(\frac{x-a}{b}\right)dx &= \int_{-a/b}^{\frac{s-a}{b}}(bz+a)\,\phi(z)dz \\
&= a\int_{-a/b}^{\frac{s-a}{b}}\phi(z)dz + b\int_{-a/b}^{\frac{s-a}{b}} z\phi(z)dz \\
&= a\left[\Phi\left(\frac{s-a}{b}\right)+\Phi\left(\frac{a}{b}\right)-1\right]+b\left[\phi\left(\frac{a}{b}\right)-\phi\left(\frac{s-a}{b}\right)\right]
\end{aligned}
$$

and so

$$E(X|S=s) = a + b\frac{\phi\left(\frac{a}{b}\right)-\phi\left(\frac{s-a}{b}\right)}{\Phi\left(\frac{a}{b}\right)+\Phi\left(\frac{s-a}{b}\right)-1}$$

In most Affymetrix micorarray applications $\phi\left(\frac{s-a}{b}\right)$ is neglible and $\Phi\left(\frac{s-a}{b}\right)$ is close to one. So in practice, it is only necessary to compute the first term in the numerator and the first term in the denominator to make the adjustment.

It is somewhat troublesome to estimate the parameters μ, σ and α. Some approaches are either painfully slow (the EM algorithm) or numerically unstable (Newton methods). An ad-hoc approach is used to estimate the parameters. First, a non-parametric density estimate of the observed probe intensities on an array is taken, the mode of which is used as the estimate of μ. Then, the variabilty of the lower tail about μ is used for σ and an exponential is fitted to the right tail to estimate α.

In this thesis, we have elected to only adjust PM probe intensities because we focus on expression measures which use only PM probes, but in principle we could adjust MM probe intensities using this method, either separately or together with the PM probe intensities.

2.2.2 Methods Proposed by Affymetrix

There are two separate adjustment steps that have been proposed by Affymetrix (2002). For our analysis, they are considered both separately and in the sequence in which they are used in the MAS 5.0 software (Affymetrix, 2001a), which is the location specific correction followed by the ideal mismatch adjustment. It should be noted that we created our own implementations of these methods based upon the available documentation and there may be some slight differences from the Affymetrix software.

Location Specific Correction

The goal of this step is to remove overall background noise. Each array is divided into a set of regions, then a background value for that is grid estimated. Then each probe intensity is adjusted based upon a weighted average of each of the background values. The weights are dependent on the distance from the centroid of each of the grids. In particular, the weights are

$$w_k(x,y) = \frac{1}{d_k^2(x,y) + \text{smooth}}$$

where $d_k(x,y)$ is the euclidean distance from location x,y to the centroid of region k. The default value for smooth is 100. Special care is taken to avoid negative values or other numerical problems

Let $P_{x,y}$ be the probe intensity at the location (x,y)
Divide array into K rectangular regions (default $K = 16$)
for $k = 1$ to K **do**
For the lowest 2% of probe intensities in grid k compute mean (call this background for region k) b_k and standard deviation (call this the noise for region k) n_k.
end for
for all probes on the array **do**
Compute $B(x,y)$ and $N(x,y)$
$P_{x,y} = \max\left(P_{x,y} - B\left(x,y\right), N_f * N\left(x,y\right)\right)$ (default $N_f = 0.5$)
end for

Table 2.1: Affymetrix Location Dependent Background.

for low intensity regions. Table 2.1 describes the algorithm in more detail. $B(x,y)$ is the weighted average of the b_k at location (x,y) and $N(x,y)$ is the weighted average of the n_k at location (x,y).

Ideal Mismatch

Originally, the suggested purpose of the MM probes was that they could be used to adjust the PM probes Affymetrix (1999) by subtracting the intensity of the MM probe from the intensity of the corresponding PM probe. However, this becomes problematic because, in a typical array, as many as 30% of MM probes have intensities higher than their corresponding PM probe (Naef et al., 2001). Thus, when raw MM intensities are subtracted from the PM intensities it is possible to compute negative expression values. Another drawback is that the negative values preclude the use of logarithms which have proved useful in many microarray data situations.

To remedy the negative impact of using raw MM values, Affymetrix introduced the *Ideal Mismatch* (*IM*) (Affymetrix, 2001b), which was guaranteed by design to be positive. The goal of this method is to use MM when it is physically possible (i.e. smaller than the corresponding PM intensity) and something smaller than the PM in other cases. First, a quantity Affymetrix refers to as the *specific background* (*SB*) is calculated for each probeset. This is computed by taking a robust average of the log ratios of PM to MM for each probe pair. If i is the probe and k is the probeset then for the probe

pair indexed by i and k the ideal mismatch IM is given by

$$IM_i^{(k)} = \begin{cases} MM_i^{(k)} & \text{when } MM_i^{(k)} < PM_i^{(k)} \\ \frac{PM_i^{(k)}}{2^{SB_k}} & \text{when } MM_i^{(k)} \geq PM_i^{(k)} \text{ and } SB_k > \tau_c \\ \frac{PM_i^{(k)}}{2^{\tau_c/(1+(\tau_c - SB_k)/\tau_s)}} & \text{when } MM_i^{(k)} ij \geq PM_i^{(k)} \text{ and } SB_i \leq \tau_c \end{cases} \tag{2.1}$$

where τ_c and τ_s are tuning constants referred to as the contrast τ (default value 0.03) and the scaling τ (default value 10) respectively. The adjusted PM intensity is given by subtracting the corresponding IM.

2.2.3 Correcting Low Intensity Signals: LESN

The RMA convolution model background method applies the largest relative adjustments to the smallest intensities and leaves the order of probe intensities invariant. Extending these principles, we devise methods where the lowest probe intensities are given the largest relative adjustment (toward 0) and the order is preserved. We call this signal adjustment approach LESN (Low End Signal is Noise).

Shifting

The simplest of these adjustments is to shift all intensities so that the minimum intensity on the chip becomes some predefined value. Ideally, we would like the minimum intensity on a chip to be zero, but since this becomes troublesome when logarithms are taken, we instead set this value to some small but non-zero value p_0.

Let p_{min} be the minimum probe value on the array. Then for a probe P_i the background adjusted value P_i' is given by

$$P_i' = P_i - (p_{min} - p_0).$$

Stretching

Rather than merely shifting the entire distribution of probe intensities downward, we can stretch out the lower tail to the minimum value. By using this method, low intensity probes are adjusted more

Name	Weight Function
Uniform (shifting)	1
Linear	$\frac{p_i - p_{max}}{p_{min} - p_{max}}$
Exponential decay	$\exp(-\frac{p_i - p_{min}}{\theta})$
Half Gaussian	$\exp\left(-\frac{(p_i - p_{min})^2}{\theta^2}\right)$
None	0

Table 2.2: Weighting functions for the LESN signal adjustment.

drastically than higher intensity probes. Let $w(P)$ be a background weighting function such that it is decreasing, takes on values in $[0,1]$ and has its maximum of 1 at p_{min} and minimum at p_{max}. The background correction is given by

$$P_i' = P_i - w(P_i)(p_{min} - p_0).$$

Some example background weighting functions are shown in Table 2.2. In this dissertation, we concentrate on the exponential decay and half gaussian weighting methods.

In Which Scale Should the LESN Correction Be Made?

To achieve more desirable effects (greater adjustment to low intensity probes), we apply the exponential decay and half gaussian weighting systems on the log scale. In other words we work with $\log_2$ probe intensities. The shifting method is carried out on original scale data.

Picking Parameters for the LESN Methods

Through experimentation, it has been found that $\theta = 4$ is a good choice for both the exponential decay and the half gaussian. These θ values were chosen by examining the Affymetrix HGU95A spike-in data (Appendix A.1), but similar results have been observed with other datasets. For p_0, we arbitrarily select 0.25 or in the $\log_2$ scale -2, which is small enough to give a wide range of expression values yet does not create numerical problems. As shown in Section 2.3, we also examine the results of using different values of θ.

2.2.4 Standard Curve Adjustment

As we will later see, it is typical to observe a non-linear relationship between computed expression value and concentration of mRNA when the truth is really known. This same relationship is observed over a number of datasets. Specifically, there is a small positive slope in the lower concentrations, an increase in slope in the mid-range of concentrations and then a leveling off to a lower slope at the highest concentrations where chemical saturation is likely. Such curves are indicated in Figure 2.2 for the Affymetrix HGU95A spike-in dataset. Since this common shape is observed, it makes sense to try to linearize it. This adjustment is illustrated in Figure 2.2. Unfortunately, there are not "truth" (concentration) values for each probeset. We need an estimate of a parameter that can be used as a proxy for concentration.

We use a joint model on PM and MM intensities to find such a parameter. For each probeset, the following model is fitted:

$$\log_2\left(PM_i^{(k)}\right) = \alpha_i + \varepsilon_i^{(k)}$$
$$\log_2\left(MM_i^{(k)}\right) = \alpha_i + \gamma + \varepsilon_i^{'(k)}$$

where α_i is a probe effect and γ is a parameter for the difference in overall levels between the *PM* and *MM* probes. Our estimates $\hat{\gamma}$ relate well with concentration, as can be seen in Figure 2.3. By estimating concentration using $\hat{\gamma}$, we can apply an adjustment to each probeset. Note that $\hat{\gamma}$ could also be used for thresholding Presence/Absence calls. Other models where a difference between the level of PM and MM was computed could also be used for this purpose, for example, the (robust) average of the difference between each PM and its corresponding MM (in the log-scale).

The standard curve refers to the mapping between concentration and an adjustment. In this chapter we established the adjustment based upon known spike-in concentrations as shown in Figure 2.2. However, in general without knowledge of truth about a number of probesets across a wide range of known concentrations it might be difficult to establish such a curve.

2.3 Comparing Background/Signal Adjustment Methods

This section compares the different background/signal adjustment methods by assessing their impacts on computed expression estimates. In particular, accurate (low bias) and precise (low variance)

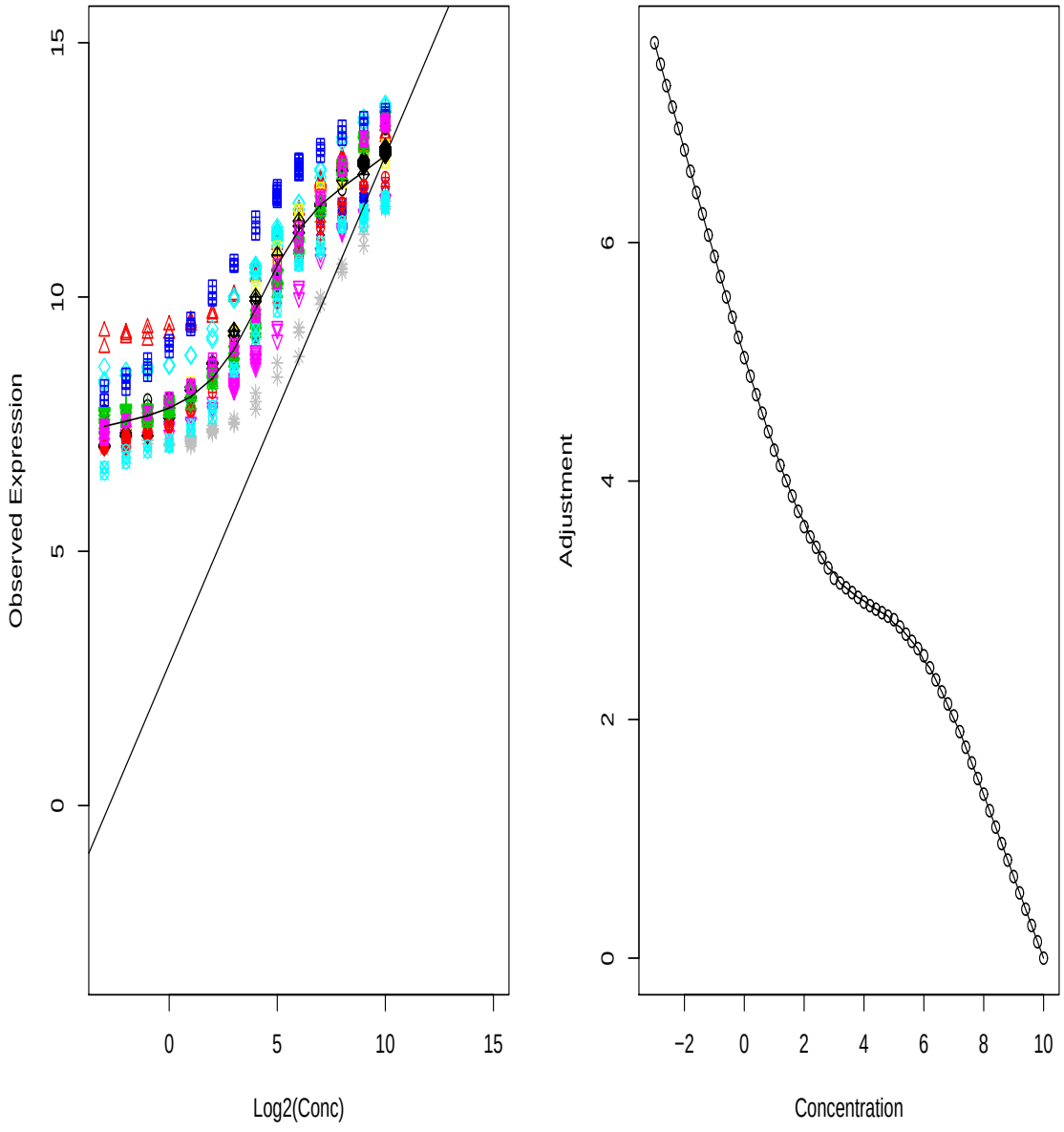

Figure 2.2: A concentration dependent pattern in expression values and a desire for linear relationship between expression value and concentration yields a concentration dependent adjustment. On the left expression values versus concentration values for spike-in probesets from the HGU95A dataset are plotted. A different symbol is used for each probeset. On the right is the adjustment required to linearize the expression value concentration relationship.

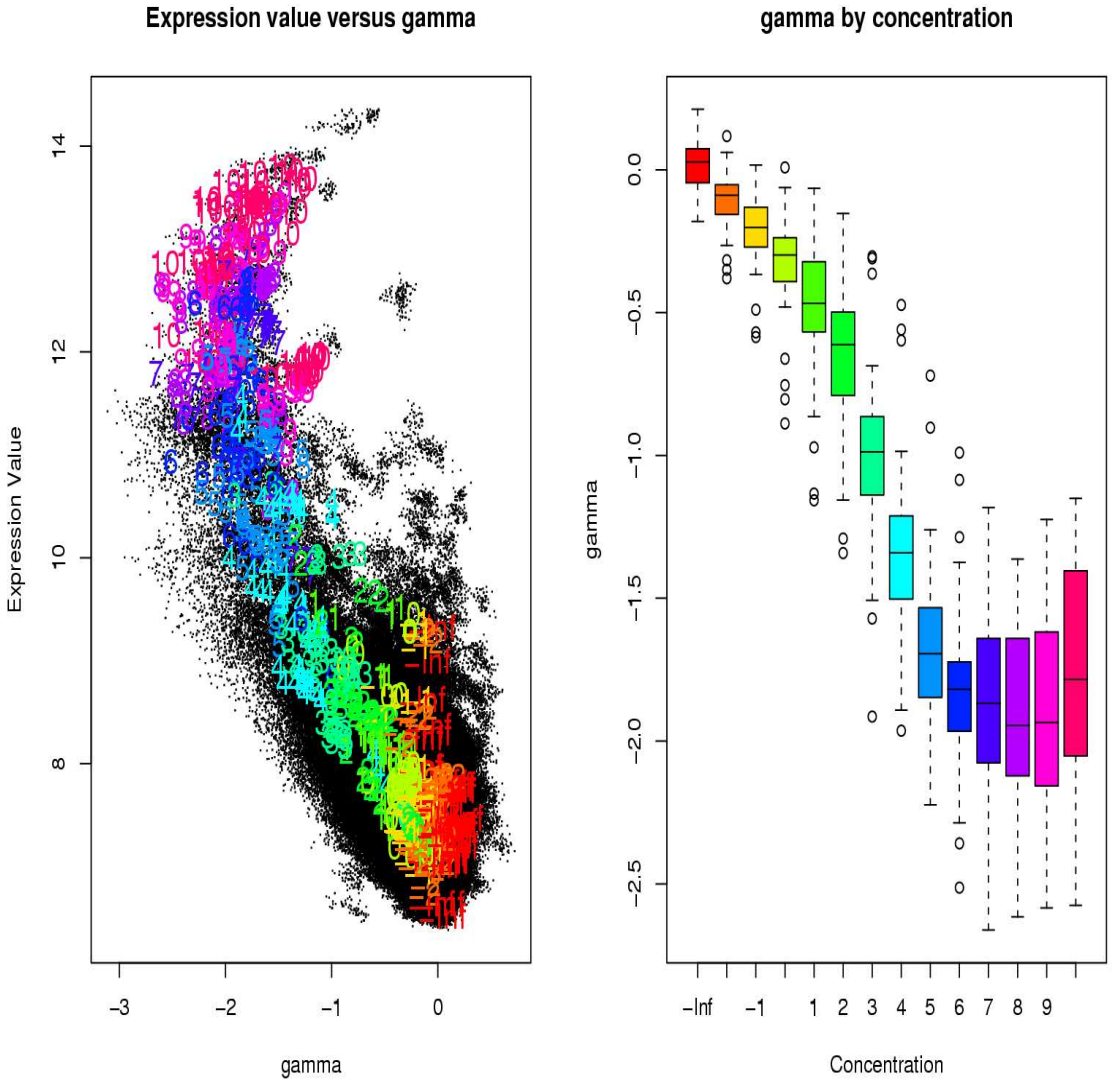

Figure 2.3: The estimates $\hat{\gamma}$ relate very well with concentration, particularly in the low concentrations. On the left the spike-in probesets are labeled by a number representing the true concentration. Non-spikeins are represented by black points.

expression estimates are desirable. We computed expression measures in the standard Robust Multi-chip Average (RMA) framework for the normalization and summarization steps. In other words, we used quantile normalization and the median polish summarization. These steps are explained in more detail in later chapters. In each case, we background corrected with the chosen method and then computed our expression measure in the standard way: probe-level quantile normalization, followed by median polish summarization. Note that this means our expression values will be in the $\log_2$ scale. In this section, we used the Affymetrix HG_U95A spike-in dataset, described in Appendix A.1, to assess and compare background methods. A spike-in dataset was useful for this purpose because it gave us a known "truth" to measure against.

2.3.1 Comparing Computed Expression Values with Known Concentration

The first comparison was to relate computed expression values with the known spike-in concentrations. In particular, plots of the observed expression, which is in the $\log_2$ scale, against the $\log_2$ of the known spike-in concentration were considered. Basically, a desirable observation would be a linear relationship between expression value and concentration. In addition, a slope near 1 would be ideal.

Such plots for the two cases: no background correction and when adjusted using the standard curve adjustment, are shown in Figure 2.4. Both plots showed a leveling off at higher intensities, most likely due to chemical saturation on the arrays. In other words, there was no material left on the array for additional cRNA of that particular type to bind to. The lower end showed a more linear curve after the standard curve correction.

The concentrations were divided into three groups: low, middle and high. Low was classified as $\log_2$ concentration less than or equal to 2, middle as between 2 and 8 and high between 8 to 10. Slope estimates were then computed for the three groups and overall slopes for each method were also calculated. These are shown in Table 2.3. The standard curve adjustment had reasonable slopes in the low and middle ranges. However it had a slightly reduced slope at the high end. The LESN corrections each had good overall slope, but this was due to overcorrecting in the mid-ranges.

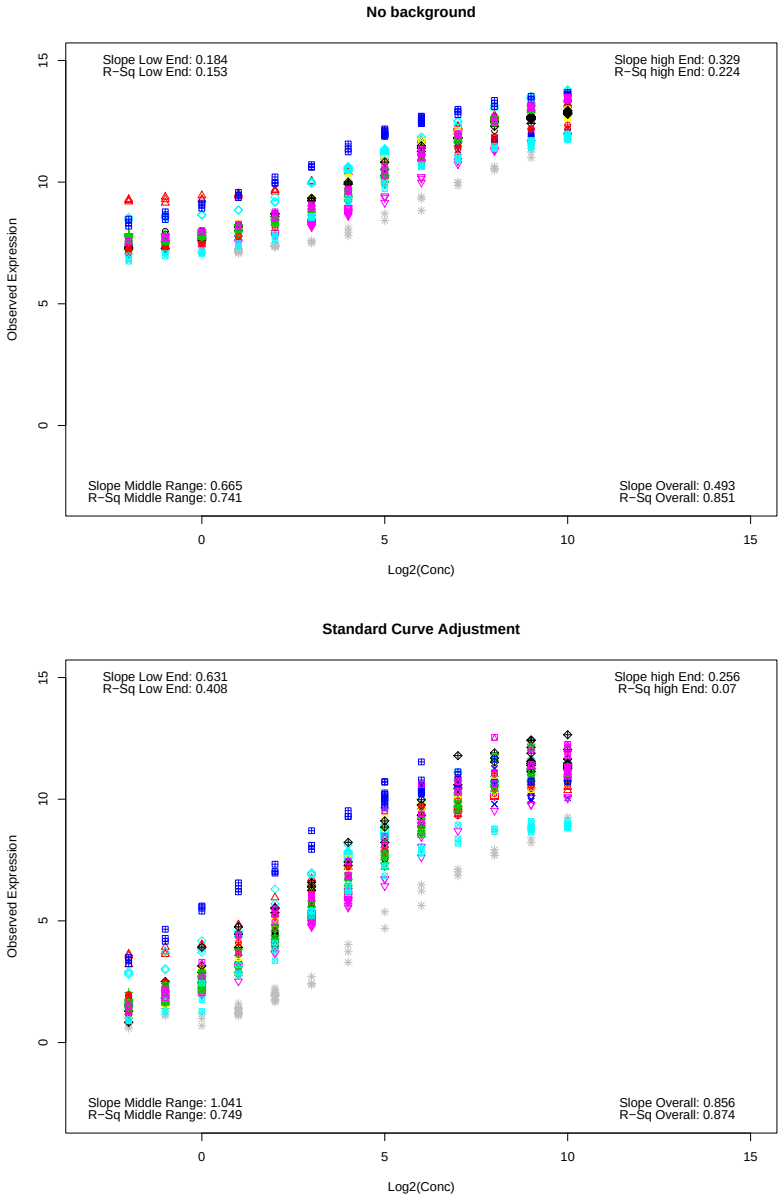

Figure 2.4: Plot of observed expression versus spike-in concentration on the log-scale, with each spike-in probeset represented using a different symbol. The curvilinear relationship indicated by the no background case is typical of corrections which ignore the MM information. The more linear relationship observed in the standard curve adjustment is more typical of cases which make use of MM information.

Method	Overall	Low	Middle	High
No Background	0.49	0.18	0.67	0.33
Convolution	0.63	0.38	0.78	0.33
MAS 5.0	0.59	0.32	0.75	0.33
Ideal Mismatch	0.69	0.52	0.82	0.30
MAS 5.0/IdealMismatch	0.70	0.56	0.82	0.29
LESN (Shifting)	0.56	0.27	0.73	0.33
LESN (Exp 3.5)	0.88	0.42	1.15	0.43
LESN (Exp 4.0)	0.88	0.41	1.16	0.45
LESN (Exp 4.5)	0.87	0.39	1.15	0.45
LESN (Normal 3.5)	1.06	0.39	1.46	0.58
LESN (Normal 4.0)	1.01	0.35	1.39	0.62
LESN (Normal 4.5)	0.96	0.31	1.32	0.63
Standard Curve Adjustment	0.86	0.63	1.04	0.26

Table 2.3: Slope estimates for the regression of observed expression on spike-in concentration. A higher slope is more desirable, with a slope of 1 the ideal. Low, Middle and High correspond to different levels of concentration.

2.3.2 Comparing Computed Fold-change with Expected Fold-change

In comparative experiments, accurate estimates of fold-change are important. We averaged data across spike-in concentration replicates, leaving us with 14 different spike-in concentration profile groups. Next, we looked at all pairwise comparisons between each of these 14 groups, giving us 91 total pairwise comparisons for each probeset. Our aim was to compare the observed fold-change with that given by the spike-in concentrations.

We plotted observed fold change versus expected fold change. These are shown for no background correction and the standard curve adjustment in Figure 2.5. A 45-degree line has been added for reference. We saw that when no background correction was applied, we were far from the line, but adding a correction brought us much closer to a 1-1 correspondence between observed fold-change and the truth.

Slope estimates for the regression of observed fold-change on the expected fold-change values for each of the background methods are compared are in Table 2.4. The LESN corrections have gave the highest slope and the standard curve also performed well. The lowest slopes were when no background correction was applied.

Small fold-changes are often of particular interest. Restricting ourselves to $\log_2$ fold-changes in the range (-2,2) we again examined slopes as shown in Table 2.5. A further restriction was that we

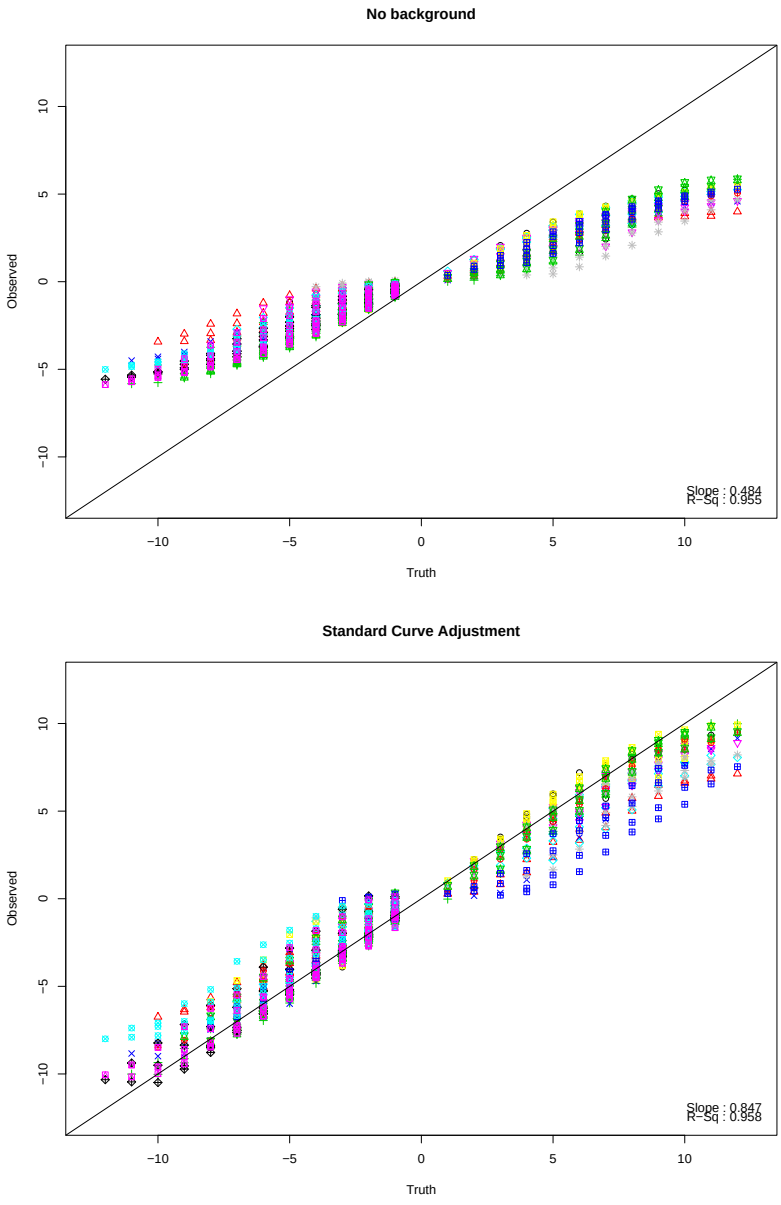

Figure 2.5: Plots of observed versus expected fold-change for expression measures computed using no background and when using the standard curve adjustment. The 45 degree line is indicated.

Method	Slope	R^2
No Background	0.48	0.96
Convolution	0.62	0.97
MAS 5.0	0.58	0.97
Ideal Mismatch	0.68	0.97
MAS 5.0/IdealMismatch	0.69	0.97
LESN (Shifting)	0.55	0.97
LESN (Exp 3.5)	0.87	0.96
LESN (Exp 4.0)	0.86	0.96
LESN (Exp 4.5)	0.86	0.96
LESN (Normal 3.5)	1.04	0.95
LESN (Normal 4.0)	0.99	0.95
LESN (Normal 4.5)	0.94	0.95
Standard Curve Adjustment	0.85	0.96

Table 2.4: Slope (and R^2) estimates for the regression of observed $\log_2$ fold-change on expected $\log_2$ fold-change. Higher slopes are more desirable with a slope of 1 the ideal.

looked only at low fold-change where the concentration was also low. In this case, a concentration of 4 pM (picomoles) or less was considered to be low and concentrations of 8 pM or above high. Slope estimates for these comparisons are also in this table.

2.3.3 Composite *MA*-plots

An *MA*-plot is a useful tool for comparing expression values in two groups. Let E_{ij} be the $\log_2$ expression value for probeset i on array j. Then, we define $M_{ijk} = E_{ij} - E_{ik}$ and $A_{ijk} = \frac{1}{2}(E_{ij} + E_{ik})$. In this analysis, we averaged across spike-in concentration replicates before computing M and A values. We referred to these plots as composite *MA*-plots because M and A values from all 91 different pairwise comparisons were placed onto the same set of axes. For these *MA*-plots, we annotated the spike-in comparisons with the appropriate $\log_2$ fold-change.

Figure 2.6 shows composite *MA*-plots for the cases of no background, convolution, MAS5/Ideal Mismatch and the Standard Curve adjustment. It was desirable for the non-differential probesets to be centered around 0 with low even variability. The MAS5/Ideal Mismatch correction was extremely noisy at the low end while the other three methods compared in these plots were less noisy. The no background and convolution *MA*-plots showed that the fold-change estimates for the spike-ins were quite attenuated for these methods. The MAS5/Ideal mismatch and Standard Curve Adjustment more accurately estimated the true fold changes.

Method	Overall	Low Conc.	High Conc.
No Background	0.42	0.21	0.50
Convolution	0.56	0.44	0.54
MAS 5.0	0.52	0.37	0.53
Ideal Mismatch	0.63	0.63	0.52
MAS 5.0/IdealMismatch	0.63	0.65	0.52
LESN (Shifting)	0.49	0.32	0.52
LESN (Exp 3.5)	0.76	0.50	0.77
LESN (Exp 4.0)	0.75	0.48	0.78
LESN (Exp 4.5)	0.75	0.46	0.79
LESN (Normal 3.5)	0.88	0.44	1.02
LESN (Normal 4.0)	0.85	0.40	1.02
LESN (Normal 4.5)	0.80	0.36	0.99
Standard Curve Adjustment	0.75	0.72	0.61

Table 2.5: Slope estimates for fold-change restricted to low fold change comparisons $|\log_2(FC)| \leq 2$. Higher slopes are better with a slope near 1 desirable.

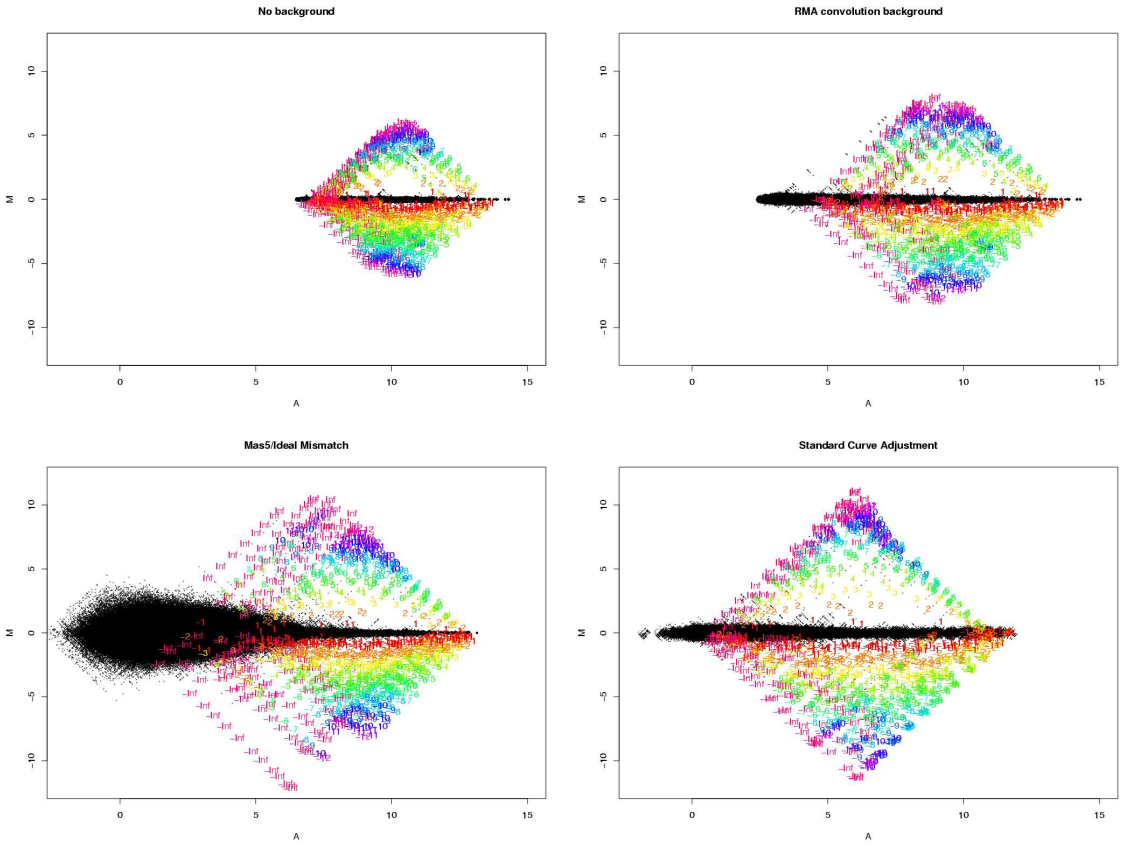

Figure 2.6: Composite *MA*-plots for: no background, convolution, MAS5/Ideal Mismatch and the Standard Curve Adjustment. Low variability of the non-differential probesets is desirable. Additionally we want the estimated fold-changes to accurately reflect the truth. The spike-in probesets are labeled by concentration. The non-differential probesets are plotted as points.

Method	Overall	Low	Middle	High
No Background	0.05	0.04	0.05	0.06
Convolution	0.12	0.13	0.13	0.11
MAS 5.0	0.11	0.14	0.11	0.09
Ideal Mismatch	0.30	0.33	0.48	0.20
MAS 5.0/IdealMismatch	0.41	0.86	0.56	0.20
LESN (Shifting)	0.09	0.11	0.09	0.08
LESN (Exp 3.5)	0.12	0.12	0.12	0.13
LESN (Exp 4.0)	0.12	0.11	0.12	0.13
LESN (Exp 4.5)	0.11	0.11	0.11	0.12
LESN (Normal 3.5)	0.08	0.06	0.08	0.11
LESN (Normal 4.0)	0.07	0.06	0.07	0.10
LESN (Normal 4.5)	0.07	0.06	0.07	0.09
Standard Curve Adjustment	0.21	0.20	0.21	0.21

Table 2.6: IQR range of M for non differential probesets. Low, middle and high refer to the the lowest third, middle third and highest third of A values. Lower values are better.

We examined the variability of the non-differential probesets in Table 2.6. Lower IQR values were better. We saw that the corrections making use of the ideal mismatch were particularly noisy in the low and middle ranges. Applying no background correction led to the least variability. Interestingly, the LESN corrections were slightly more variable in the higher range than in the lower and mid-ranges (although still well below the variability of the Ideal Mismatch corrections). The IQR was fairly stable across concentrations for the Standard Curve Adjustment, however it was among the more variable methods.

2.3.4 Detecting Differential Expression: ROC Curves

When analyzing high density oligonucleotide array data, it is important to identify genes that are differential without incorrectly picking non-differential genes. Ideally, it is desirable to have many true positives (call a truly differential probeset as changed) and few false positives (calling a non differential probeset as changed). Observed fold-change was used for choosing differential genes. The probesets with the most extreme fold-changes were selected as the differential probesets, and ROC curves (see Section 1.4.2) were used to compare the sensitivity and specificity of each of the different methods.

Figure 2.7 shows the ROC curves for each of the methods. The ideal curve would go to 1 on the y axis immediately (i.e., at $x = 0$). The two methods utlizing the ideal mismatch performed poorly since the respective curves were well below the other curves.

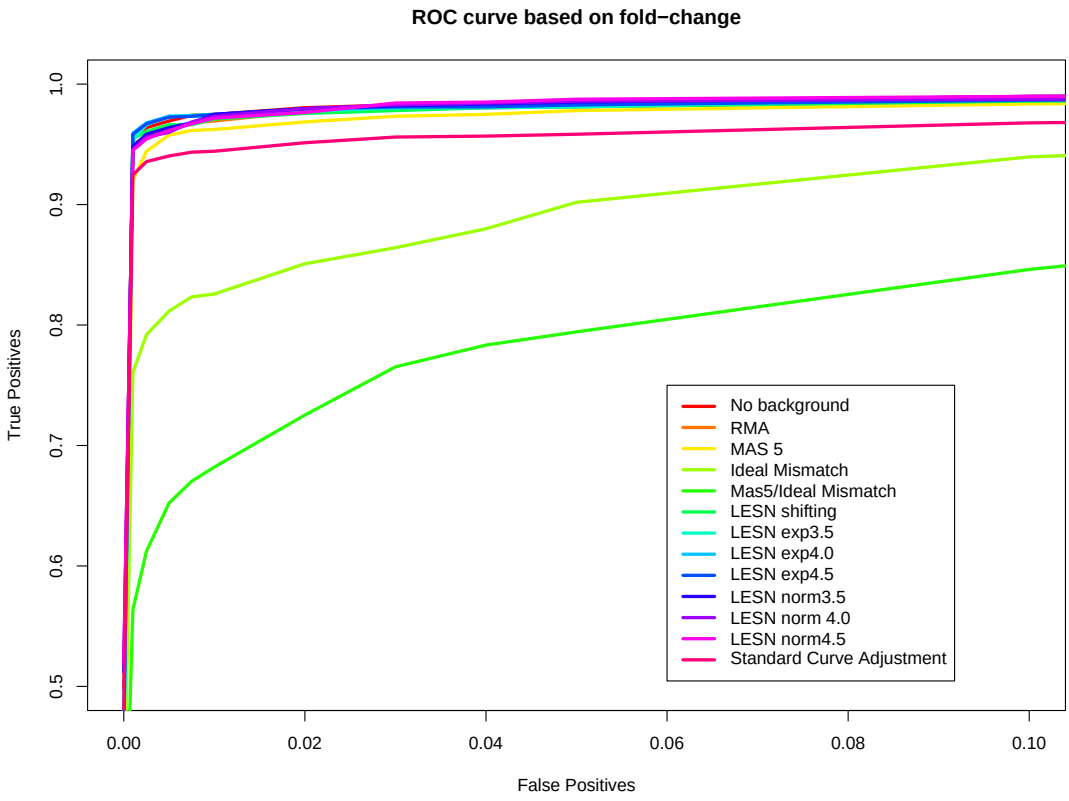

Figure 2.7: ROC curves based on fold-change. The ideal curve would reach 1.0 on the vertical (all true positives identified) when at 0 on the horizontal (no false positives). Higher curves are better, the two methods using the ideal mismatch do particularly poorly.

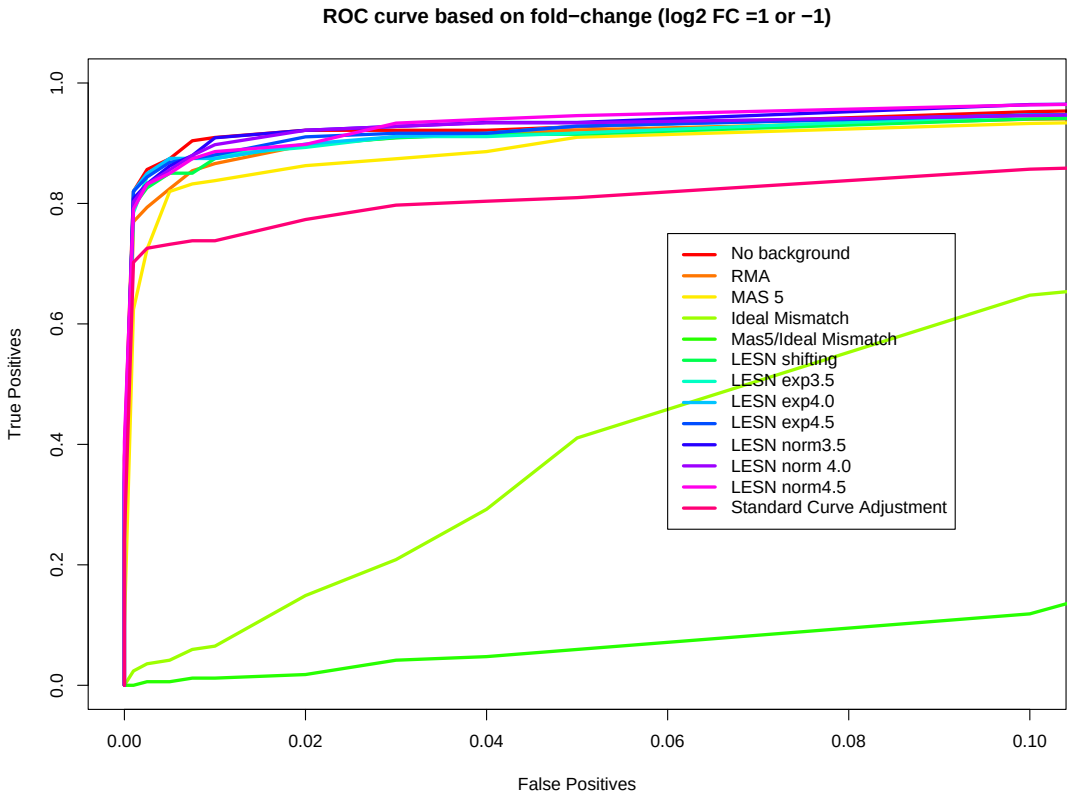

Figure 2.8: ROC curves based on fold-change where the only differences should be log FC = 1 (or -1). The ideal curve would reach 1.0 on the vertical (all true positives identified) when at 0 on the horizontal (no false positives). Higher curves are better.

Since low fold-change differential expression is often of particular interest, and usually much harder to detect than larger fold-change expression differences, we further restricted ourselves to $\log_2$ fold-changes equal to 1 (these are the smallest fold-changes in this dataset). An ROC curve for such fold-changes is shown in Figure 2.8. Compared to their respective curves for all FC, and the other methods in this comparison, the two methods using the ideal mismatch were extremely poor. The curves for the other methods were lower than the respective curves in Figure 2.7, but still each detected over 75% of the true positives (when we allowed 0.05% false positives).

Method	Linear	Accurate FC		Detect DE	
		Overall	Low	Overall	Low
No Background	No	No	No	Yes	Yes
Convolution	No	No	No	Yes	Yes
MAS 5.0	No	No	No	Yes	Yes
Ideal Mismatch	Yes	Yes	Yes	No	No
MAS 5.0/IdealMismatch	Yes	Yes	Yes	No	No
LESN (Shifting)	No	No	No	Yes	Yes
LESN (Exp 3.5)	No	Yes	No	Yes	Yes
LESN (Exp 4.0)	No	Yes	No	Yes	Yes
LESN (Exp 4.5)	No	Yes	No	Yes	Yes
LESN (Normal 3.5)	No	Yes	No	Yes	Yes
LESN (Normal 4.0)	No	Yes	No	Yes	Yes
LESN (Normal 4.5)	No	Yes	No	Yes	Yes
Standard Curve Adjustment	Yes	Yes	Yes	Yes	Yes

Table 2.7: Summary of the comparison of background adjustment methods. The standard curve adjustment performed well in all comparisons.

2.4 Discussion

In this chapter, we compared 13 different background adjustment methods which, when combined with quantile normalization and the median polish summarization, produced expression measures. Using the spike-in concentrations for this dataset, we were able to compare these expression measures and make judgements on the effect of each of the different background methods.

Our comparisons were either assessments of accuracy (bias), of precision (variance) or power to detect differentially expressed probesets. In particular, the comparisons showed how well the computed expression measures reflect the "truth," as well as how well they let us detect differential expression. We were also interested in how well our methods did in detecting differential expression and estimating fold-change with low intensity genes. A linear relationship between spike-in concentration and expression value was also desirable.

Table 2.7 summarizes the results of the comparisons carried out in this chapter. The methods compared can be grouped into four main categories. First, there were methods that did well in detecting differential genes, but performed poorly at estimating fold change, such as the Convolution method. The second set of methods performed well at estimating fold-change and had a linear relationship between concentration. However, they did poorly in regard to detecting truly differential genes. The Ideal Mismatch fell in this category. Most of the LESN corrections fell into the third group, which consisted of methods which did well at detecting differential expression as well as at accurately

estimating fold-change for higher fold-changes. However, these did more poorly at estimating fold change for lower intensity genes because the relationship between concentration and expression was not linear. Finally, the Standard Curve Adjustment fell into its own category, having performed well when judged under all the criteria.

There still remains one drawback with the Standard Curve Adjustment method which is that it appears to be difficult to generalize. In particular, establishing an adjustment curve is difficult without known spike-in concentrations. One possible solution is to produce tissue and array type specific standard curve and normalization vectors. This issue will not be explored further in this thesis and will remain an area for future study.

Another recently proposed background correction method, GCRMA (Wu et al., 2003), has been observed to have all the desirable properties explored in this chapter. This method is based upon sequence information, such as GC content, for each probe and stochastic models for binding affinities (Wu and Irizarry, 2004). While not considered in the comparisons in this chapter, or further in this dissertation, the GCRMA method shows great promise as a future method of background adjustment.

Chapter 3

Normalization

This chapter considers the important topic of normalization for single channel microarrays. Section 3.1 explains what normalization is and why it is used, Section 3.2 discusses the various algorithms that have been devised for normalization and Section 3.3 describes how normalization algorithms may be applied to high-density oligonucleotide data. In Section 3.4, the proposed normalization methods are compared, and in Section 3.5 the results of these comparisons are discussed.

3.1 Introduction

Normalization is the process of removing unwanted non-biological variation that might exist between chips in a microarray experiment. It has long been recognized that variability can exist between arrays, some of biological interest and other of non-biological interest. These two types of variation are classified as either interesting or obscuring by Hartemink et al. (2001). It is this obscuring variation that we seek to remove when normalizing arrays. Sources of obscuring variation can include scanner setting differences, the quantities of mRNA hybridized as well as many other factors. Hartemink et al. (2001) discusses these possible sources in more detail.

Numerous papers proposing normalization methods have recently been published. Comparisons of normalization methods for high-density oligonucleotide arrays are considered in both Bolstad et al. (2003) and Schadt et al. (2001). A technical report by Ballman et al. (2003) compares quantile and cyclic loess with an adaptation of the loess method called *fastlo* on the basis of probe-level

variability.

Probe-level normalization for high-density oligonucleotide arrays has been investigated in Bolstad et al. (2003), which compares the bias and variability of expression measures computed using different normalization methods. This paper defines two classes of normalization methods: complete data methods and baseline methods. Complete data methods use information from across all arrays to produce the normalization. The baseline methods select one array to represent the typical array, and then all of the other arrays are normalized to that array. It has been discovered that complete data methods are preferable to methods choosing a baseline array.

An important consideration when applying a normalization method to data from a typical comparative microarray experiment, is how many genes are expected to change between conditions and how these changes will occur. Two important assumptions were suggested by Zien et al. (2001). Specifically most normalization methods require that either the number of genes changing in expression between conditions be small or that an equivalent number of genes increase and decrease in expression. When neither of these are true then normalization should be applied only on arrays within each treatment condition group. However, in many cases, experiments are not properly randomized and there is confounding of conditions with sources of non-biological variation. For this reason, all of the arrays in a particular experiment are typically normalized together as a single group.

In this chapter, a further study of complete and baseline methods, focusing on the bias and variance of fold-change estimates will be conducted. To separate the potential confounding effect of normalization and background methods, the normalization methods are used without background or signal adjustment. In addition, a baseline method can also be made into a complete data method by creating a composite pseudo array and then normalizing to that chip. These extensions are referred to as composite methods.

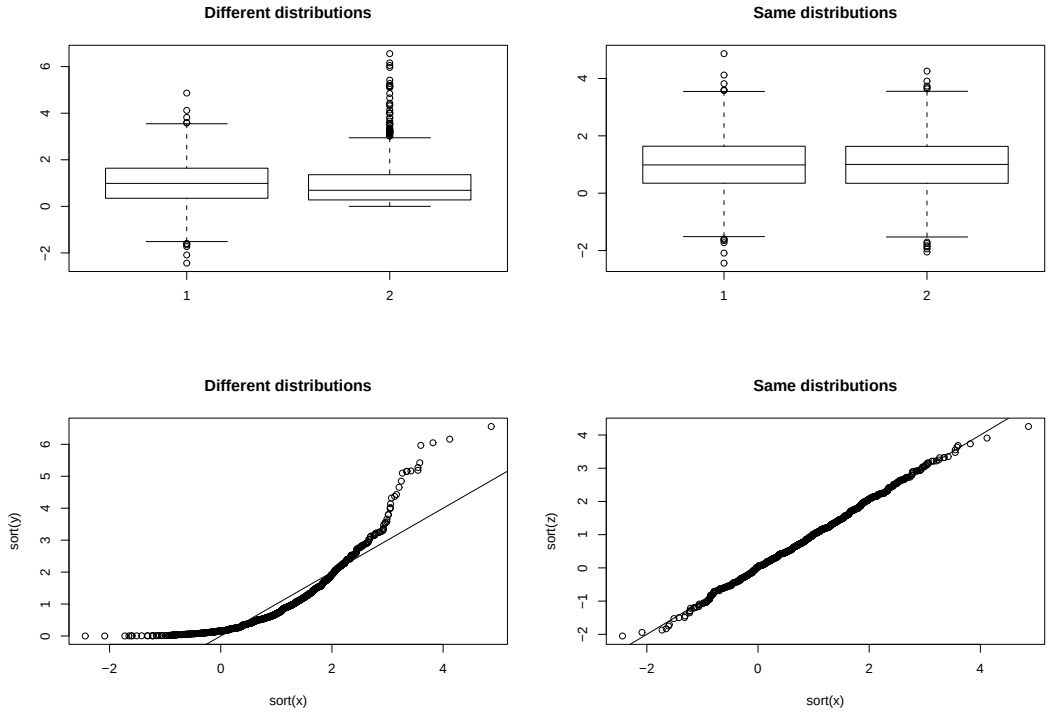

Figure 3.1: Quantile-Quantile plot motivates the quantile normalization algorithm.

3.2 Normalization methods

3.2.1 Complete Data Methods

Quantile Normalization

The goal of quantile normalization, as discussed in Bolstad et al. (2003), is to give the same empirical distribution of intensities to each array. A quantile-quantile plot will have a straight diagonal line, with slope 1 and intercept 0, if two data vectors have the same distribution, as illustrated in Figure 3.1. Thus, if the quantiles of two data vectors are plotted against each other and each of these points are then projected onto the 45-degree diagonal line, we have a transformation that gives the same distribution to both data vectors. This transformation is shown in Figure 3.2.

In n dimensions, a quantile-quantile plot where all data vectors have the same distribution would have the points lying on the line inscribed by the vector $\left(\frac{1}{\sqrt{n}}, \ldots, \frac{1}{\sqrt{n}}\right)$. This extenstion to n dimensions motivates the quantile normalization algorithm described in Table 3.1.

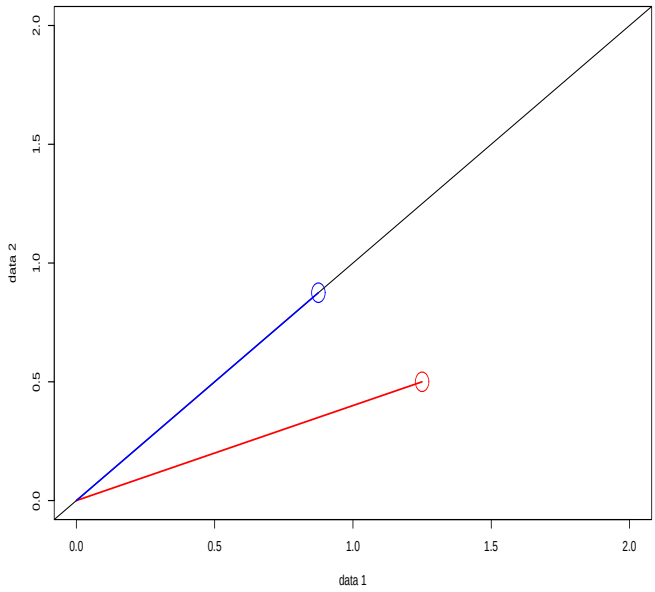

Figure 3.2: The quantile normalization adjustment in 2 dimensions.

1:	Given n arrays of length p, form X of dimension $p \times n$ where each 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 quantile equalized X'_{sort}.
4:	Get $X_{\text{normalized}}$ by rearranging each column of X'_{sort} to have the same ordering as original X.

Table 3.1: Quantile Normalization Algorithm.

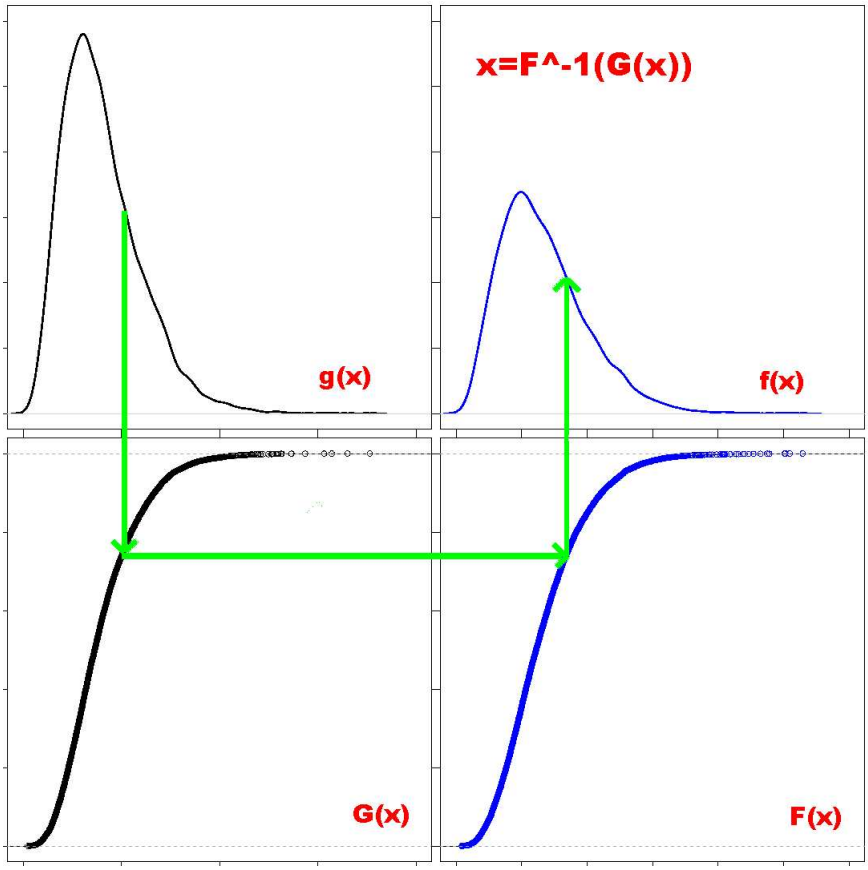

Figure 3.3: The quantile normalization method transforms the distribution of intensities from one distribution to another.

The quantile normalization method is a specific case of the transformation $x'_i = F^{-1}(G(x_i))$, where we estimate G by the empirical distribution of each array and F using the empirical distribution of the averaged sample quantiles. This transformation is illustrated in Figure 3.3. Extensions of the method could be implemented where F^{-1} and G are more smoothly estimated. However, we find the current method to perform satisfactorily in practice.

Traditionally, the mean has been used at step 3 in the algorithm described in Table 3.1, but this step could be modified to a more general procedure:

3: Apply function f_n to each row of X_{sort}, where f_n is a function of n data items, and assign this value to each element in the row to get X'_{sort}.

Some possible options for f_n are the mean, median, geometric average or an order statistic.

The quantile normalization method discussed here is not unique. There have been other proposed normalizations based upon quantiles, including Workman et al. (2002), where splines are fit to subsets of quantiles to estimate the normalizing relation. Another non-parametric method of giving each array the same distribution is discussed in Sidorov et al. (2002). There is also a quantile normalization method discussed in Amaratunga and Cabrera (2001).

Cyclic Loess

The cyclic loess method is a generalization of the global loess method, which is described in Yang et al. (2002), where cy5 and cy3 channel intensities are normalized on cDNA microarrays by using *MA* plots. When dealing with single channel array data, it is pairs of arrays that are normalized to each other. The cyclic loess method normalizes intensities for a set of arrays by working in a pairwise manner. With only two arrays, the algorithm is identical to that in Yang et al. (2002). With more than two arrays, only part of the adjustment is made. In this case, the procedure cycles through all pairwise combinations of arrays, repeating the entire process until convergence. One drawback is that this procedure requires $O\left(n^2\right)$ loess normalizations. Usually only one or two complete cycles through the data are required. The cyclic loess algorithm is outlined in Table 3.2. The indicies i and j in this algorithm index the arrays while the index k is used to represent probe or probesets. Convergence is measured by how much additional adjustment has occurred on that iteration through the dataset. To improve the runtime of the algorithm a typical implementation will use a subset of the data to fit the loess normalization curves.

Contrast

This method is described in Astrand (2003) and is also presented in Bolstad et al. (2003). Basically, it is another generalization of the methods described in Yang et al. (2002). In brief, the data is transformed to a set of contrasts, a non-linear normalization is performed and a reverse transformation occurs. The algorithm is shown in Table 3.3 and more complete implementation details are provided in Astrand (2003). It requires only $O(n)$ loess normalizations which is considerably fewer than with the cyclic loess method. As with the cyclic loess method, a subset of the data is used to fit the loess curves to considerably speed up the runtimes. One way that the subset may be chosen is to use a rank invariant set of probes, see Schadt et al. (2001) for a method to select such a set.

Let X be a $p \times n$ matrix with columns representing arrays and rows probes or probesets.
log transform the data $X \leftarrow \log_2 X$
repeat
  for $i = 1$ to $n-1$ **do**
    for $j = i+1$ to n **do**
      for $k = 1$ to p **do**
        Compute $M_k = x_{ki} - x_{kj}$ and $A_k = \frac{1}{2}\left(x_{ki} + x_{kj}\right)$
      end for
      fit a loess curve for M on A. Call this $\hat{f}$.
      for $k = 1$ to p **do**
        $\hat{M}_k = \hat{f}(A_K)$
        set $a_k = \frac{M_k - \hat{M}_k}{n}$
        $x_{ki} = x_{ki} + a_k$ and $x_{kj} = x_{ki} - a_k$
      end for
    end for
  end for
until convergence or reached maximum number of iterations
Take the anti-log $X = 2^X$

Table 3.2: Cyclic Loess Algorithm.

3.2.2 Baseline Methods

Scaling/Linear Method

In this method, which was proposed by Affymetrix and used both in versions 4.0 and 5.0 of their software (Affymetrix, 1999), (Affymetrix, 2001a), a baseline array is chosen and all the other arrays are scaled to have the same mean intensity as this array. This is equivalent to selecting a baseline array and then fitting a linear regression without intercept term between each array and the chosen array. Then, the fitted regression line is used as the normalizing relation. This method is outlined in Table 3.4. One proposed modification, is to remove the highest and lowest intensities when computing the mean, that is we use a trimmed mean. Affymetrix removes the highest and lowest 2% of the data. Affymetrix has proposed using scaling normalization after the computation of expression values, but in this thesis we also use it at the probe-level.

log transform the data $X = \log X$.

$Z = \log(X)M'$ where M is an orthonormal matrix.

for $i = 2$ to n **do**

fit a loess curve of the ith column of Z on the 1st column of Z. Call this curve $\hat{y}_{i-1}$.

end for

Define a mapping $[x, \hat{y}_1, \ldots, \hat{y}_{n-1}] \mapsto [x, 0, \ldots, 0]$.

Normalization is given by $\exp([x, \hat{y}_1, \ldots, \hat{y}_{n-1}]M) \mapsto \exp([x, 0, \ldots, 0]M)$.

Table 3.3: Contrast Normalization Algorithm.

Pick a column of X to serve as baseline array, say column j.

Compute (trimmed) mean of column j. Call this $\tilde{X}_j$.

for $i = 1$ to n, $i \neq j$ **do**

Compute (trimmed) mean of column i. Call this $\tilde{X}_i$.

Compute $\beta_i = \frac{\tilde{X}_j}{\tilde{X}_i}$.

Multiply elements of column i by β_i.

end for

Table 3.4: Scaling Normalization Algorithm.

Non-linear Method

Rather than using a linear normalizing relation, as in the scaling method, a non-linear relationship between each array and the baseline array can be used. Such methods have been proposed by Schadt et al. (2001) and are currently used in the dChip software (Li and Wong, 2001a). An outline of the procedure is given in Table 3.5.

Pick a column of X to serve as baseline array, say column j.

for $i = 1$ to n, $i \neq j$ **do**

Fit a smooth non-linear relationship mapping column i to the baseline j. Call this $\hat{f}_i$

Normalized values for column j are given by $\hat{f}_i(X_j)$

end for

Table 3.5: Non-linear Normalization Algorithm.

Numerous non-linear relationships have been used for this normalization method including cross-validated splines (Schadt et al., 2001), running median lines (Li and Wong, 2001b) and loess smoothers (Bolstad et al., 2003).

There are several ways in which the baseline array can be selected, or created. In this dissertation we have considered four different methods. The first method is to select the array with the median total intensity as the baseline. Similarly, a second method is to select the array with the median median intensity as the baseline array. With both of these options it is still possible that a troublesome array will be chosen as the baseline array. Instead a baseline array can be constucted using all the data, this is explained in the following subsection.

3.2.3 Composite Methods

A baseline method can be transformed into a complete data method by creating a composite array based upon data from all arrays. This is called a composite method. One method for constructing a composite chip is to take probe-wise means or medians. The composite chip is then used as the baseline array and the selected baseline method is used in the normal fashion. Because many problems arise when a single chip is chosen as a baseline, the composite approach should be preferred if a baseline method is used. Both the probe-wise mean and median chips have been used as baselines for the non-linear method in this chapter.

3.3 Probe-level, Probeset-level and Expression-level Normalization

There are three levels at which normalization can occur: probe-level, probeset-level and after computing expression. The topic of probe-level normalization is considered extensively in Bolstad et al. (2003). At this level, it is raw probe intensites, possibly after a background correction, that are normalized.

Probeset-level normalization occurs when all the probes in a probeset are normalized together as a group. For instance, we could compute the mean (or median) value of a probeset, normalize these summaries and then adjust individual probes based on the adjustment to the summary. In this dissertation, we focus only on adapting the quantile normalization method to operate in this manner. This adapted algorithm is described in Section 3.3.1.

Normalization can also take place after expression values have been computed. This is how the scaling normalization is carried out by Affymetrix (2001a). In our analysis, we carry out expression

Given n arrays of length p, form X_{all} of dimension $p \times n$ where each array is a column and each row corresponds to a row.
Let $J = 1, \ldots, N_A$ index the arrays
Let $n = 1, \ldots, N_P$ index the number of probesets
Let $i = 1, \ldots, I_n$ index the probes in probeset n.
Let $\mathbf{r}_n$ be a vector containing the indicies of the rows of X_{all} for probeset n.
Set $X_{\text{all}} \leftarrow \log_2 X_{\text{all}}$
for $n = 1$ to N_P **do**
  Choose the rows of X_{all} indexed by $\mathbf{r}_n$ to form an I_n by N_A matrix. Call this X_n
  To each column of X_n apply a summarization function f_s yielding a summary vector $\mathbf{S}_n$ of length N_A
  Subtract $\mathbf{S}_n$ from each row of X_n
  Set the nth row of X_{reduced} as $\mathbf{S}_n$.
end for
Quantile Normalize X_{reduced} using the algorithm in Table 3.1
for $n = 1$ to N_p **do**
  To all rows of X_n add nth row of X_{reduced}
  Copy the rows of X_n into X_{all} using the indicies in $\mathbf{r}_n$.
end for
Set $X_{\text{all}} \leftarrow 2^{X_{\text{all}}}$

Table 3.6: Probeset Quantile Normalization Algorithm.

level normalization using one of the standard methods after computing the expression measures for each array without probe or probeset-level normalization.

3.3.1 Probeset-level Quantile Normalization

The goal of a probeset normalization method is to preserve any parallelism in probe pattern across arrays. In particular, we normalize so that each probe in the probeset is adjusted in a similar manner. The Probeset Quantile Normalization algorithm is outlined in Table 3.6. For this analysis, we use either the mean or the median as our summary function f_s. In addition, we quantile normalize X_{reduced} on either the log scale or natural scale.

3.4 Comparing Normalization Methods

In order to compare normalization methods we used the GeneLogic AML spike-in dataset described in Appendix A.3 . A boxplot of the raw PM probe-intensities, by array, for this dataset is shown in Figure 3.4. Common background RNA was used on all the arrays and so we wanted the arrays to give us similar expression values. The clear differences in expression values between arrays shown in this plot indicated that normalization was required for this dataset. We assessed the performance of the different normalization methods by comparing computed expression measures. In this chapter, the expression measures were computed using the median polish summarization after normalization. Because of possible confounding between the different background methods and each normalization, no background adjustment was used. In the case of expression-level normalization, the summarization took place on unnormalized data. Chapter 5 examines the interaction between background adjustment and normalization methods.

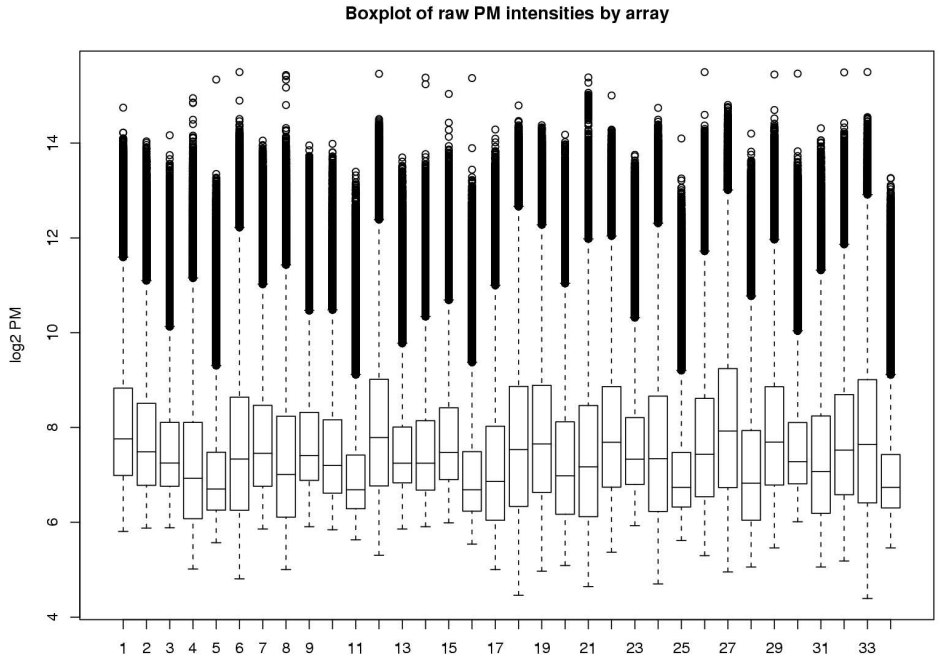

Figure 3.4: A boxplot of raw $\log_2$ PM intensities across arrays in Genelogic Spike-in dataset shows need for normalization.

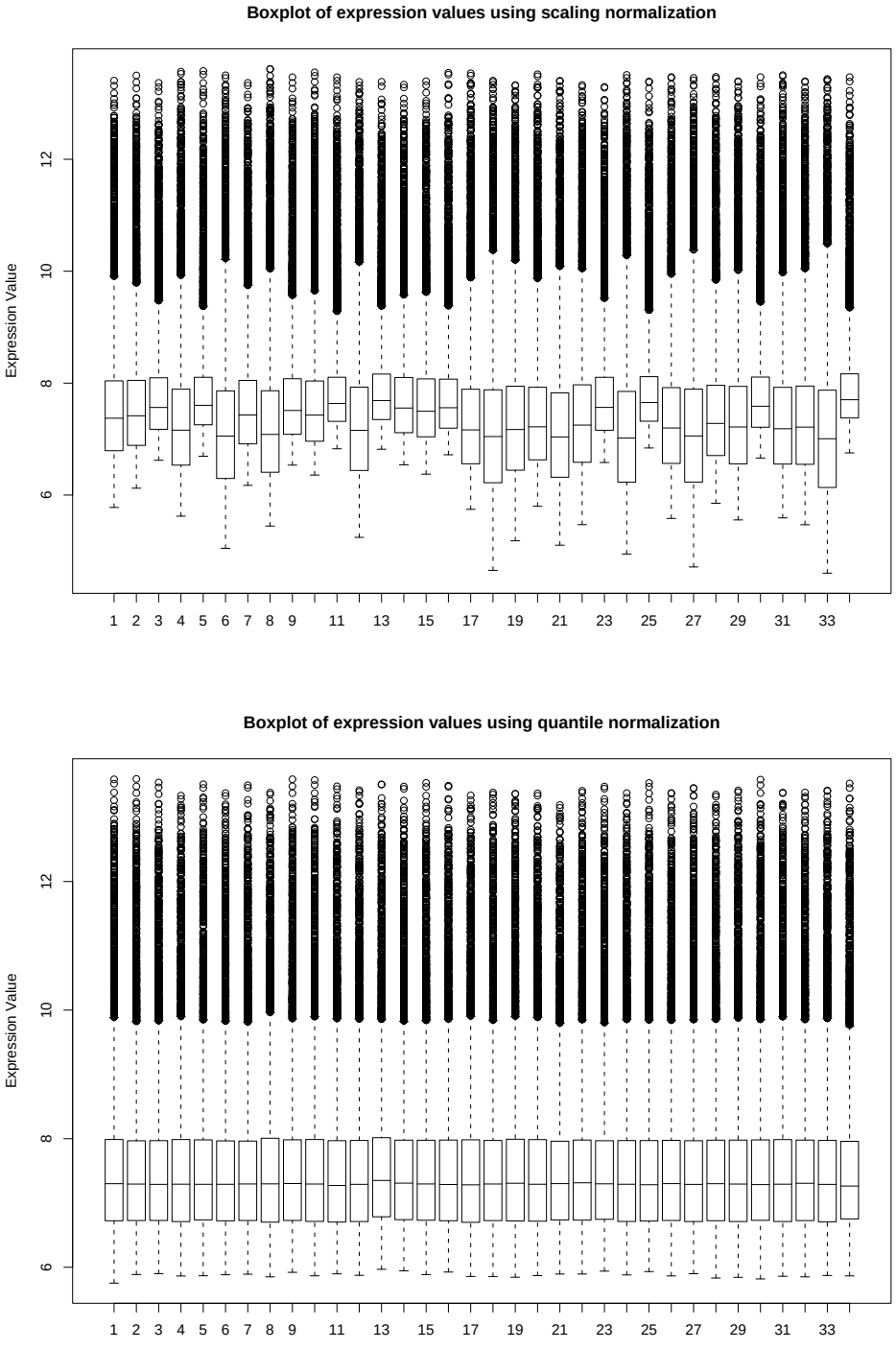

Figure 3.5: Boxplots of expression across arrays in Genelogic Spike-in dataset when using probe-level scaling normalization and when using probe-level quantile normalization.

Method	Probe level	Expression Level
None	0.315	
Quantile	0.097	0.106
Quantile (median)	0.096	
Quantile ($\log_2$)	0.097	0.105
Quantile ($\log_2$, median)	0.096	
Quantile (probeset)	0.160	
Quantile (probeset, median)	0.108	
Quantile (probeset, $\log_2$)	0.106	
Quantile (probeset, $\log_2$, median)	0.107	
Cyclic Loess	0.086	0.096
Contrast	0.101	0.105
Scaling	0.258	0.191
Nonlinear (median total)	0.104	0.114
Nonlinear (median median)	0.104	0.114
Nonlinear (pseduo-mean)	0.010	0.108
Nonlinear (pseudo-median)	0.098	0.103

Table 3.7: IQR of fold-change estimates for non-differential probesets. Smaller IQR are more desirable.

Figure 3.5 shows expression values by array after normalization. We saw more similarity across arrays than in Figure 3.4, but it was clear that the scaling normalization was insufficient. The distribution of expression values was much closer to identical when the quantile normalization algorithm was used. Similar boxplots for the other non-linear normalization methods showed much more striking resemblence between arrays than the scaling method did.

3.4.1 Assessing Variance and Bias of Non-diffential Probesets

In the GeneLogic AML dataset, there were 66 possible pairwise comparisons that could have been made between different spike-in concentration groups. For each of these pairwise comparisons, we computed fold-change estimates by taking the average of the expression measures over the replicate arrays in the two concentration groups. The difference between the two averages gave the $\log_2$ fold-change. Table 3.7 shows the IQR of the fold-change estimates for non-spikein probesets across all possible comparisons. The non-spikein probesets were expected to be non-differential. Since lower variability was better, we looked for methods with lower IQR.

In general, all of the normalization methods reduced the variability when compared to unnormalized data. The Cyclic Loess method provided the least variable non-differential probesets, and the various probe-level Quantile methods also performed well. The scaling normalization reduced

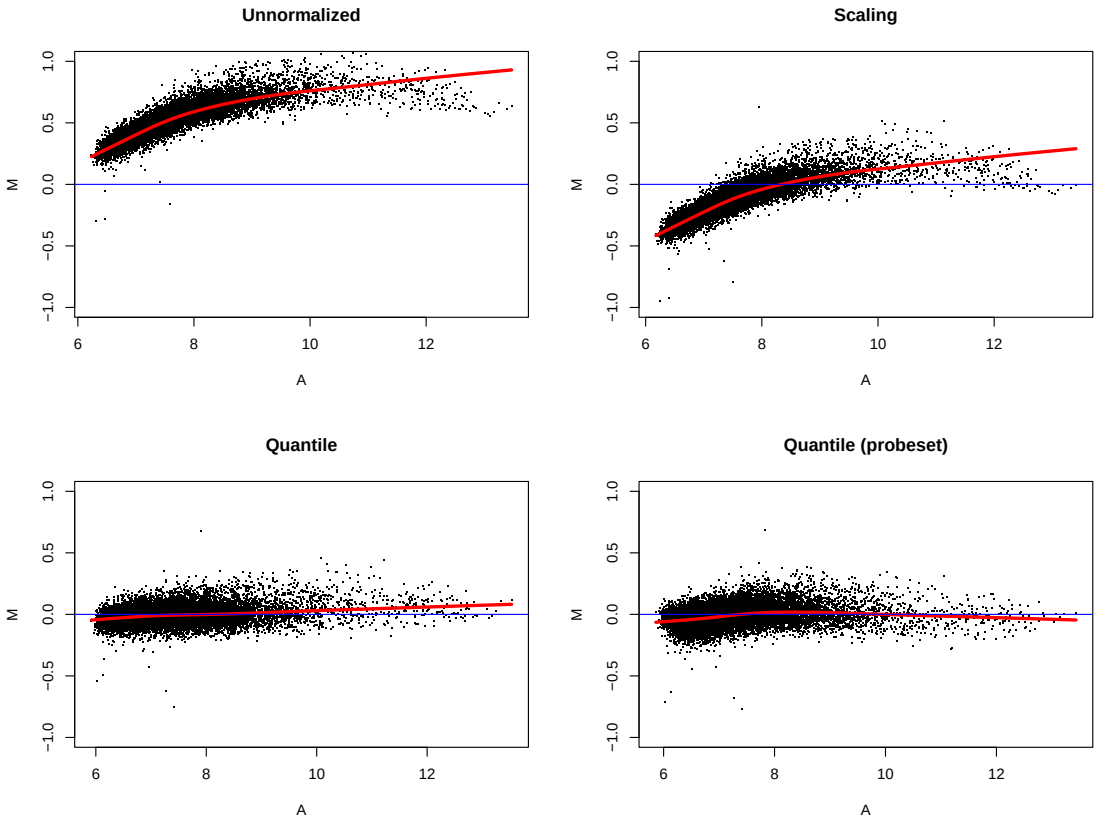

Figure 3.6: *MA*-plot comparing non-differential probesets from two groups of three arrays each. The probe-level scaling normalization centers the distribution of the M's around 0 but does not remove the non-linear trend. Probe-level quantile and Probeset quantile normalization gave plots that were closer to the ideal.

this variability only marginally. Another interesting observation was that variability among these fold-change estimates was lower for the probe-level normalization than that for the corresponding normalization carried out at the expression-level (except in the case of the scaling normalization). This difference suggested that probe-level normalization should be favored for reducing variability over an expression-level normalization.

An ideal *MA*-plot comparing two groups of arrays on non-differential probesets would be centered around $M = 0$ and the point cloud would be evenly distributed about $M = 0$ across the range of intensities. *MA*-plots for four methods are shown in Figure 3.6. This provided a good illustration of one of the drawbacks of the scaling normalization. Since scaling does not remove non-linear differences between arrays the lowess curve observed for the unnormalized data is just translated

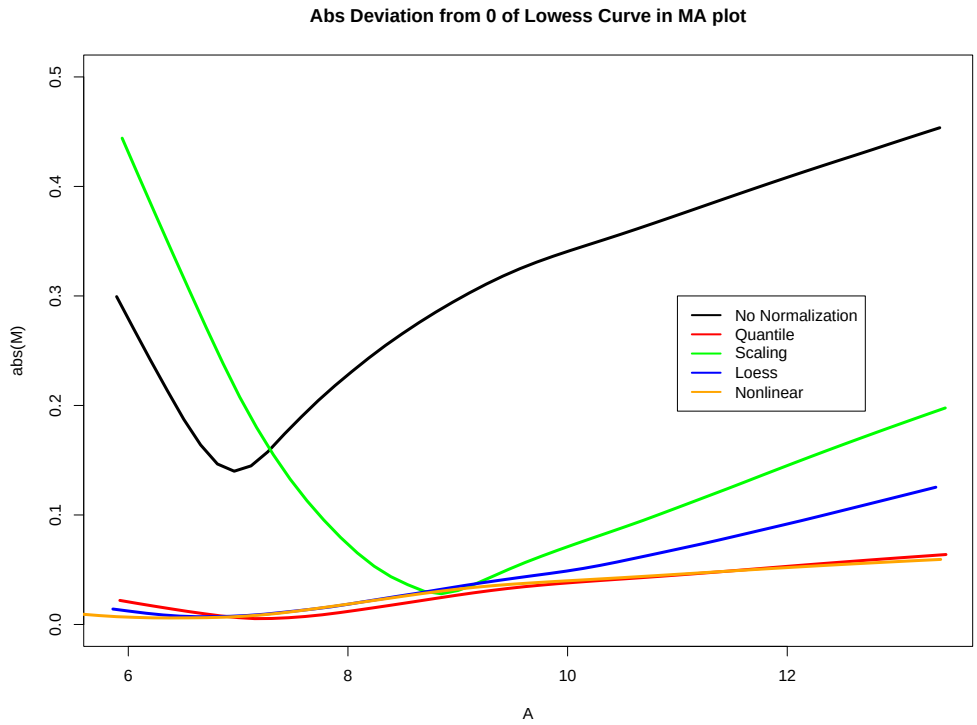

Figure 3.7: Absolute deviation of M curve from x-axis for all pairwise M vs A plots. Small even deviations are best.

to be centered around $M = 0$. The MA-plots for the probe-level Quantile and probeset Quantile normalizations were closer to the ideal for such plots.

The location of the lowess curve in the MA-plot provided a method of judging how biased or otherwise each of the methods were. Specifically, we looked at the absolute deviation of the lowess curve from the $M = 0$ axis. Figure 3.7 shows the average absolute deviation of the lowess curve for all pairwise MA-plots against average A. We saw that scaling did poorly, particularly in the lower intensity range where the deviations were even larger than for unnormalized data. Since there were fewer high intensity points, the lowess curves at the higher intensities were more variable and this accounted for the increasing trend we saw for all of the methods.

Table 3.8 gives the median absolute difference of the lowess curve to $M = 0$ for all of the methods. In this table, the expression-level normalizations had smaller values and thus, had lowess curves that were tighter around $M = 0$, as compared to the corresponding probe-level normalization. As shown in Figure 3.7 the scaling normalization performed poorly.

Method	Probe level	Expression Level
None	0.328	
Quantile	0.035	0.009
Quantile (median)	0.036	
Quantile ($\log_2$)	0.035	0.009
Quantile ($\log_2$, median)	0.036	
Quantile (probeset)	0.035	
Quantile (probeset, median)	0.026	
Quantile (probeset, $\log_2$)	0.017	
Quantile (probeset, $\log_2$, median)	0.025	
Cyclic Loess	0.044	0.017
Contrast	0.066	0.007
Scaling	0.121	0.169
Nonlinear (median total)	0.036	0.010
Nonlinear (median median)	0.036	0.010
Nonlinear (pseduo-mean)	0.034	0.008
Nonlinear (pseudo-median)	0.035	0.010

Table 3.8: Median absolute difference between *M* curve and x-axis. Smaller values are better.

3.4.2 Assessing Bias and Variability of Differential Probesets

In the GeneLogic AML dataset, there were 11 probesets that had been spiked-in at various pre-selected concentrations. We used these known concentrations, as a "truth," to assess the effect that each of the normalization methods had on estimates of fold-change.

After averaging across spike-in replicates, fold-change estimates were computed for all pairwise comparisons. Table 3.9 contains slope and R^2 estimates for the regression of observed fold-change against expected fold-change. In general, higher R^2 values are better as are higher slopes. All the normalization methods improved the R^2 over unnormalized expression values. The slopes reflected how the normalization affects the raw probe intensities. A greater range of expression values on an array resulted in a higher slope, which was the case with the non-linear method. The baseline array chosen for the non-linear method, using the both median total and median median intensity, had a larger spread of intensities than most arrays and thus it had higher slopes. The non-linear method applied using the pseudo-mean and pseudo-median baselines behaved similarly to the other normalization methods. There was no reason to believe that there should have been any bias in unnormalized data, thus we also compared each slope to the slope for unnormalized data. The slope for non-linear method was the furthest from the unnormalized data, while the Quantile method was the closest. One interesting observation is that all of the methods seemed to have lower slopes than the unnormalized case, indicating a slight attentuation of the FC estimates. From this analysis, it

Method	Probe level	Expression Level
None	0.591 (0.863)	
Quantile	0.588 (0.876)	0.586 (0.877)
Quantile (median)	0.579 (0.877)	
Quantile ($\log_2$)	0.580 (0.876)	0.578 (0.877)
Quantile ($\log_2$, median)	0.579 (0.877)	
Quantile (probeset)	0.585 (0.874)	
Quantile (probeset, median)	0.588 (0.876)	
Quantile (probeset, $\log_2$)	0.579 (0.877)	
Quantile (probeset, $\log_2$, median)	0.578 (0.877)	
Cyclic Loess	0.580 (0.872)	0.580 (0.873)
Contrast	0.582 (0.880)	0.579 (0.877)
Scaling	0.585 (0.875)	0.586 (0.880)
Nonlinear (median total)	0.617 (0.877)	0.616 (0.877)
Nonlinear (median median)	0.619 (0.877)	0.616 (0.877)
Nonlinear (pseduo-mean)	0.583 (0.876)	0.586 (0.877)
Nonlinear (pseudo-median)	0.579 (0.877)	0.577 (0.877)

Table 3.9: Comparing normalization methods using slope and R^2 estimates in parentheses for observed FC against expected FC.

was not clear that the quantile method or any of the other complete data methods had adverse effects on the bias of the observed fold-change estimates. In other words, there was not sufficent evidence to conclude that normalization was greatly decreasing or increasing the FC estimates.

3.4.3 Impact of Normalization on the Ability to Detect Differential Expression

By threshholding on observed fold-change, it is possible to declare a probeset as differential or not. For each of the possible comparisons, we counted how many differential probesets were identified before we found the first non-differential probeset. Over all 66 pairwise comparisons the average number of true differential probesets detected is shown in Table 3.10 for each of the possible methods. By normalizing, we were able to increase the number of truly differential probesets detected before reaching the first non-differential probeset. Expression-level normalization resulted in a slightly lower number of truly differential probesets being detected compared to the corresponding probe-level normalization. Apart from scaling, the normalization methods were fairly comparable.

It is important that we are also able to accurately detect low fold-change genes. We restricted ourselves to look only at those spike-ins which had absolute $\log_2$ fold-changes less than or equal to 1 (an expression difference of two-fold or less). The metric used to compare methods was the total area under the ROC curve. The areas that were observed are displayed in Table 3.11. The conclusions

Method	Probe level	Expression Level
None	7.74	
Quantile	8.30	8.26
Quantile (median)	8.27	
Quantile ($\log_2$)	8.29	8.24
Quantile ($\log_2$, median)	8.27	
Quantile (probeset)	7.96	
Quantile (probeset, median)	8.11	
Quantile (probeset, $\log_2$)	8.27	
Quantile (probeset, $\log_2$, median)	8.15	
Cyclic Loess	8.39	8.36
Contrast	8.31	8.30
Scaling	7.82	8.04
Nonlinear (median total)	8.27	8.21
Nonlinear (median median)	8.27	8.21
Nonlinear (pseduo-mean)	8.27	8.26
Nonlinear (pseudo-median)	8.33	8.33

Table 3.10: Average number of true positives identified when there are 0 false positives. There are a total of 11 differential spike-in probesets.

Method	Probe level	Expression Level
None	77.0	
Quantile	93.2	92.6
Quantile (median)	92.8	
Quantile ($\log_2$)	93.3	92.2
Quantile ($\log_2$, median)	92.8	
Quantile (probeset)	88.7	
Quantile (probeset, median)	92.0	
Quantile (probeset, $\log_2$)	93.0	
Quantile (probeset, $\log_2$, median)	92.1	
Cyclic Loess	93.0	91.8
Contrast	92.4	92.3
Scaling	83.0	86.3
Nonlinear (median total)	93.6	91.9
Nonlinear (median median)	93.6	91.9
Nonlinear (pseduo-mean)	92.9	92.0
Nonlinear (pseudo-median)	93.3	92.2

Table 3.11: Percentage of total area under ROC curve when looking for differential probesets with absolute $\log_2$ FC less or equal to 1. Higher areas are better.

from this table were very similar to those of the previous comparison with the expression-level normalization resulting in slightly smaller area and thus a lower number of differential probesets then the corresponding probe-level normalization. All the non-linear normalization methods behaved comparably.

3.4.4 Speed of complete data methods

Because loess normalizations are typically slow, the quantile normalization method is considerably faster than the other two complete data methods. While this is true in theory, it was important to examine this difference to see if it matters in practice. Some timing simulations using the R function `system.time()` were made to empirically compare the method. The simulations were run using R-1.8.1 with version 1.3.27 of the BioConductor *affy* package on a Fedora Core release 1 Linux operating system. The machine was configured with an Athlon XP 2500+ processor and 1 GB of RAM. For this simulation, only the 201800 PM probes for each HGU95A array were considered. Both the cyclic loess and constrast methods established normalizing relations using a subset of 5000 randomly chosen PM probes.

Method	**5**	**10**	**25**	**50**
Quantile	1.2	2.3	5.9	12.4
Cyclic Loess	31.4	138.7	793.5	3144.8
Contrast	77.6	173.9	462.7	963.0

Table 3.12: Runtimes in seconds to normalize different numbers of arrays using complete data methods.

Table 3.12 demonstrates how much faster quantile normalization is than either of the other two methods. We observed that for anything over a smaller number of arrays, the speed gained by using the quantile normalization was impressive.

3.5 Discussion

We have considered a number of normalization methods and the impact that they had on computed expression values. There were several differences in the analysis completed in this dissertation and that in Bolstad et al. (2003). Specifically, we removed any possible confounding with the background step by not making such any such adjustment. Probeset- and expression-level normaliza-

tions were considered and the focus was on FC estimates and the detection of differential expression. However, the results were similar to the conclusions found in Bolstad et al. (2003). Complete data methods were better than baseline methods in both variance and bias measures. All of the normalization methods successfully reduced variability of gene expression measures in comparison to unnormalized data. However, scaling normalization did not remove non-linear differences between arrays. Since these are typically observed in microarray experiments a normalization which deals with this effectively should be preferred. It appeared that expression-level normalization seemed to slightly reduce the ability to detect differential genes when compared with the corresponding probe-level normalization. An examination of running times found that the Quantile normalization was extremely quick relative to the other complete data methods. Since Quantile normalization performed satisfactorily in the comparisons made in this chapter, it should remain the preferred normalization for expression summarization.

Chapter 4

Summarization

This chapter considers the process of summarization, which is the final step in the production of a gene expression measure. In Section 4.1, we introduce summarization and discuss the approach that will be taken in the rest of the chapter. Section 4.2 introduces a number of summarization methods, and Section 4.3 compares the performance of the methods on spike-in data. Finally, Section 4.4 discusses the results of our comparison.

4.1 Introduction

Typically, Affymetrix GeneChip microarrays have hundreds of thousands of probes. These probes are grouped together into probesets. Within a probeset each probe interrogates a different part of the sequence for a particular gene. Summarization is the process of combining the multiple probe intensities for each probeset to produce an expression value. Averages (Affymetrix, 1999), robust averages (Affymetrix, 2001a) and multi-chip models are among the summarization procedures that have been used. The first multichip method was the Model Based Expression Index (MBEI) (Li and Wong, 2001a). This was a multiplicative model with additive errors. To incorporate robustness, into their method they had an algorithm for removing outlier probes, arrays and individual intensities as part of the model fitting procedure. An alternative multi-chip approach is to fit an additive multi-chip model with additive errors on the $\log_2$ scale. This has the advantage of stabilizing the variability across intensities. The RMA method (Irizarry et al., 2003a), (Irizarry et al., 2003b), takes this

approach. In this chapter, our examination of multi-chip models will also deal with additive models on $\log_2$ transformed data.

4.2 Methods

In this section, all expression measures are expressed in the $\log_2$ scale. When describing the summarization methods the following notation will be used:

β represents an expression value on the log scale

y represents a natural scale probe intensity (which may have previously been preprocessed using a background and normalization)

N_P is the number of probesets on the chip and N_A is the number of arrays

I_n is the number of probes (or probe-pairs) in probeset n

the superscript n represents the probeset with $n = 1, ..., N_P$

the subscript j represents the particular array with $j = 1, \ldots, N_A$

the subscript i represents the probe in the probeset with $i = 1, \ldots, I_n$

4.2.1 Single-chip Summarization

These methods use only probe information on an individual array to compute expression summaries for that array. The expression values for each array are computed in isolation from information in other arrays.

Average

In this method, the probe intensities within each probeset are averaged to produce an expression measure. In particular, the estimated expression value for probeset n on array j is given by

$$\hat{\beta}_j^{(n)} = \log_2 \left(\frac{\sum_{i=1}^{I_n} y_{ij}^{(n)}}{I_n} \right).$$

An alternative approach is to use the geometric mean, which is equivalent to taking the mean of the $\log_2$ probe intensities. In this case, the expression measure would be

$$\hat{\beta}_j^{(n)} = \frac{\sum_{i=1}^{I_n} \log_2\left(y_{ij}^{(n)}\right)}{I_n}.$$

The standard error for this expression summary would be given by

$$\text{SE}\left(\hat{\beta}_j^{(n)}\right) = \frac{\hat{S}_j^{(n)}}{\sqrt{I_n}}$$

where $\hat{S}_j^{(n)} = \sqrt{\frac{\sum_{i=1}^{I_n}\left(\log_2\left(y_{ij}^{(n)}\right)-\hat{\beta}_j^{(n)}\right)^2}{I_n-1}}$.

Median

Rather than using the mean, which might be affected by outliers, another option is to use the median, which is less affected by outliers. The expression summary values would be computed as

$$\hat{\beta}_j^{(n)} = \log_2\left(\text{Median}\left(y_{1j}^{(n)},\ldots,y_{I_nj}^{(n)}\right)\right)$$

or alternatively, as

$$\hat{\beta}_j^{(n)} = \text{Median}\left(\log_2\left(y_{1j}^{(n)}\right),\ldots,\log_2\left(y_{I_nj}^{(n)}\right)\right).$$

These expression summaries are identical when there are an odd number of probes, but differ slightly for probesets where there are an even number of probes.

Robust Average

Another method of summarization is to compute a robust average. Initially, Affymetrix using their AvDiff measure (Affymetrix, 1999) proposed first removing the smallest and largest probes and then taking the average of the remaining probes. In our framework, this expression summary would be

$$\hat{\beta}_j^{(n)} = \log_2\left(\frac{\sum_{i=2}^{I_n-1} y_{[i]j}^{(n)}}{I_n-2}\right)$$

where $y_{[i]j}^{(n)}$ is the i'th order statistic. However, this is a somewhat arbitrary cut-off and possibly insufficiently robust. Thus, we do not consider it further.

Another approach suggested by Affymetrix was to use a 1-step Tukey biweight, taken on the $\log_2$ scale, to give an expression summary. In particular, the procedure proposed in Affymetrix (2002) and Hubbell et al. (2002) is to compute

$$u_{ij}^{(n)} = \frac{\log_2\left(y_{ij}^{(n)}\right) - M}{cS + \varepsilon}$$

where M is the median of the $\log_2\left(y_{ij}^{(n)}\right)$ and S is the median of the absolute deviation from M i.e., the MAD. Here $c = 5$ is a tuning constant and $\varepsilon = 0.0001$ is chosen to avoid the problem of division by 0. Weights are defined by the bisquare function

$$w(u) = \begin{cases} 0 & \text{when } |u| > 1 \\ (1-u)^2 & \text{when } |u| \leq 1 \end{cases}$$

Finally, the expression measure is given by

$$\hat{\beta}_j^{(n)} = \frac{\sum_{i=1}^{I_n} w\left(u_{ij}^{(n)}\right) \log_2\left(y_{ij}^{(n)}\right)}{\sum_{i=1}^{I_n} w\left(u_{ij}^{(n)}\right)}.$$

This is the summarization step used in the MAS 5.0 software (Affymetrix, 2001a). The standard error for this 1-step Tukey biweight estimate is given by

$$\mathrm{SE}\left(\hat{\beta}_j^{(n)}\right) = \frac{\sqrt{\sum_{i=1,|u(n)_{ij}|<1}^{I_n} \left(\log_2\left(y_{ij}^{(n)}\right) - \hat{\beta}_j^{(n)}\right)^2 \left(1 - u_{ij}^{(n)^2}\right)^4}}{\left|\sum_{1,|u_{ij}^{(n)}|<1}^{I_n} \left(1 - u_{ij}^{(n)^2}\right)\left(1 - 5u_{ij}^{(n)^2}\right)\right|}.$$

The 1-step Tukey biweight is not the only method that can be used to compute a robust average. This method falls into a larger class of methods referred to as M-estimators (Huber, 1981). An M-estimator of location is defined by

$$\min_\theta \sum_{i=1}^{N} \rho\left(x_i - \theta\right) \tag{4.1}$$

where ρ is a suitable function. Reasonable properties for ρ include symmetry $\rho(x) = \rho(-x)$, a minimum at $\rho(0) = 0$, positive $\rho(x) \geq 0\ \forall x$ and increasing as the absolute value of x increases, i.e. $\rho(x_i) \geq \rho(x_j)$ if $|x_i| > |x_j|$.

Equation 4.1 leads to solving

$$\sum_{i=1}^{N} \psi\left(x_i - \theta\right) = 0$$

where ψ is the derivative of ρ. Note that for robustness, ψ should be bounded. This estimator is not scale invariant, but it may be made so by rescaling. Thus, the estimator becomes the solution of

$$\min_\theta \sum_{i=1}^{N} \rho\left(\frac{x_i-\theta}{s}\right)$$

Furthermore, there is a need to estimate s, where s is a scale estimate. One approach is to estimate both s and θ using a system of equations. The approach that we use is to estimate s using the median absolute deviation (MAD) which provides a robust estimate of scale. The above equation leads to

$$\sum_{i=1}^{N} \psi\left(\frac{x_i-\theta}{s}\right) = 0.$$

Now define $r_i = \frac{x_i-\theta}{s}$ and a weight function $w(r_i) = \frac{\psi(r_i)}{r_i}$. Then the previous equation can be rewritten as

$$\sum_{i=1}^{N} w(r_i)\, r_i = 0$$

which is the same as the set of equations that would be obtained if we were solving the iteratively reweighted least squares problem

$$\min \sum_{i}^{N} w\left(r_i^{(k-1)}\right) {r_i^{(k)}}^2$$

where the superscript (k) represents the iteration number. Since there are typically only 11-20 PM probes in each probeset, it is computationally feasible for us to use fully iterated M-estimators. Table 4.1 summarizes the functions that we use for our M-estimators. Many of the methods require the choice of a tuning constant. These constants are chosen such that the methods have 95% asymptotic efficiency when applied to the standard normal. The tuning constant for each method is shown in Table 4.2.

To better understand each of these functions, we have plotted the ρ, ψ and weight functions in Figures 4.1, 4.2 and 4.3 respectively. For comparative purposes, we have also included plots of the functions that we use for the standard mean (or linear regression) as labeled by L2. The ρ functions are all somewhat less than the x^2 of the L2 method, with several leveling out at a constant distance from the center. Five of the methods, Cauchy, Geman-McClure, Welsch, Tukey and Andrews, have redescending ψ functions. In other words, the distance between the value of $\psi(x)$ and 0 increases as $|x|$ increases, reaches a maximum distance and then begins to decrease again as $|x|$ continues to increase. In the case of the Tukey and Andrews methods, the distance returns completely to 0. This redecending property can cause the methods to converge to non-unique values depending on the

Name	$\rho(x)$	$\psi(x)$	$w(x)$
Huber	$\begin{cases}\text{if } \lvert x\rvert \le k \\ \text{if } \lvert x\rvert > k\end{cases}$	$\begin{cases}x^2/2 \\ k(\lvert x\rvert - k/2)\end{cases}$	$\begin{cases}x \\ k\,\mathrm{sgn}(x)\end{cases}$	$\begin{cases}1 \\ \frac{k}{\lvert x\rvert}\end{cases}$
"Fair"	$c^2\left(\frac{\lvert x\rvert}{c} - \log\left(1+\frac{\lvert x\rvert}{c}\right)\right)$	$\frac{x}{1+\frac{\lvert x\rvert}{c}}$	$\frac{1}{1+\frac{\lvert x\rvert}{c}}$
Cauchy	$\frac{c^2}{2}\log\left(1+(x/c)^2\right)$	$\frac{x}{1+(x/c)^2}$	$\frac{1}{1+(x/c)^2}$
Geman-McClure	$\frac{x^2/2}{1+x^2}$	$\frac{x}{\left(1+x^2\right)^2}$	$\frac{1}{\left(1+x^2\right)^2}$
Welsch	$\frac{c^2}{2}\left(1-\exp\left(-\left(\frac{x}{c}\right)^2\right)\right)$	$x\exp\left(-(x/c)^2\right)$	$\exp\left(-(x/c)^2\right)$
Tukey	$\begin{cases}\text{if } \lvert x\rvert \le c \\ \text{if } \lvert x\rvert > c\end{cases}$	$\begin{cases}\frac{c^2}{6}\left(1-\left(1-(x/c)^2\right)^3\right) \\ \frac{c^2}{6}\end{cases}$	$\begin{cases}x\left(1-(x/c)^2\right)^2 \\ 0\end{cases}$	$\begin{cases}\left(1-(x/c)^2\right)^2 \\ 0\end{cases}$
Andrews	$\begin{cases}\text{if } \lvert x\rvert \le k\pi \\ \text{if } \lvert x\rvert > k\pi\end{cases}$	$\begin{cases}k^2(1-\cos(x/k)) \\ 2k^2\end{cases}$	$\begin{cases}k\sin(x/k) \\ 0\end{cases}$	$\begin{cases}\frac{\sin(x/k)}{x/k} \\ 0\end{cases}$

Table 4.1: ρ, ψ and weight functions for some common M-estimators.

Name	**Tuning Constant**
Huber	1.345
"fair"	1.3998
Cauchy	2.3849
Welsch	2.9846
Tukey	4.6851
Andrews	1.339

Table 4.2: Default tuning constants (k or c) for M-estimation ρ, ψ and weight functions.

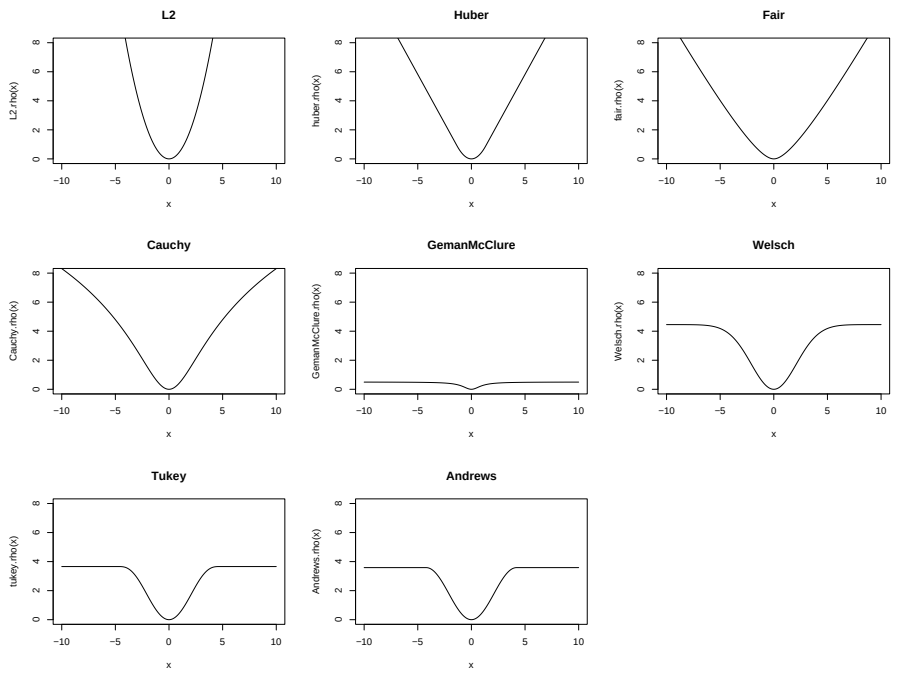

Figure 4.1: The ρ functions for some common M-estimators.

starting place. Finally, an examination of the weight function plots shows that all the methods give lower weights to outliers. The Andrews and Tukey methods are very similar and give zero weight to the most extreme outliers.

For the purposes of applying M-estimators to high density oligonucleotide array data, x_i should be replaced in the above discussion by $\log_2\left(y_{ij}^{(n)}\right)$ and thus the expression summary is given by $\hat{\beta}_j^{(n)} = \hat{\theta}$. We can also compute standard errors for this expression estimate. In particular, the asymptotic variance of an M-estimator is given by

$$\frac{\int \psi^2 dF}{\left[\int \psi' dF\right]^2}$$

where F is the distribution of the standardized resisuals, see Huber (1981) and Hampel et al. (1986) for derivations. The estimated standard error for our estimators is thus given by

$$\mathrm{SE}\left(\hat{\beta}_j^{(n)}\right) = \frac{1}{\sqrt{I_n}} \sqrt{\frac{\sum_{i=1}^{I_n} \psi\left(\frac{\log_2\left(y_{ij}^{(n)}\right) - \hat{\beta}_j^{(n)}}{s}\right)^2 / I_n}{\left(\sum_{i=1}^{I_n} \psi'\left(\frac{\log_2\left(y_{ij}^{(n)}\right) - \beta_j^{(n)}}{s}\right) / I_n\right)^2}}.$$

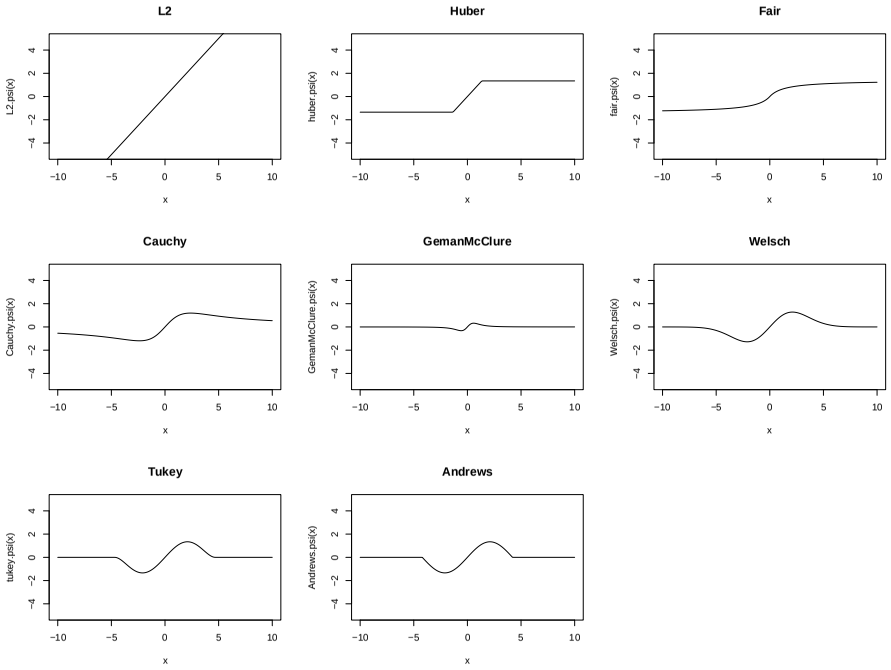

Figure 4.2: The ψ functions for some common M-estimators.

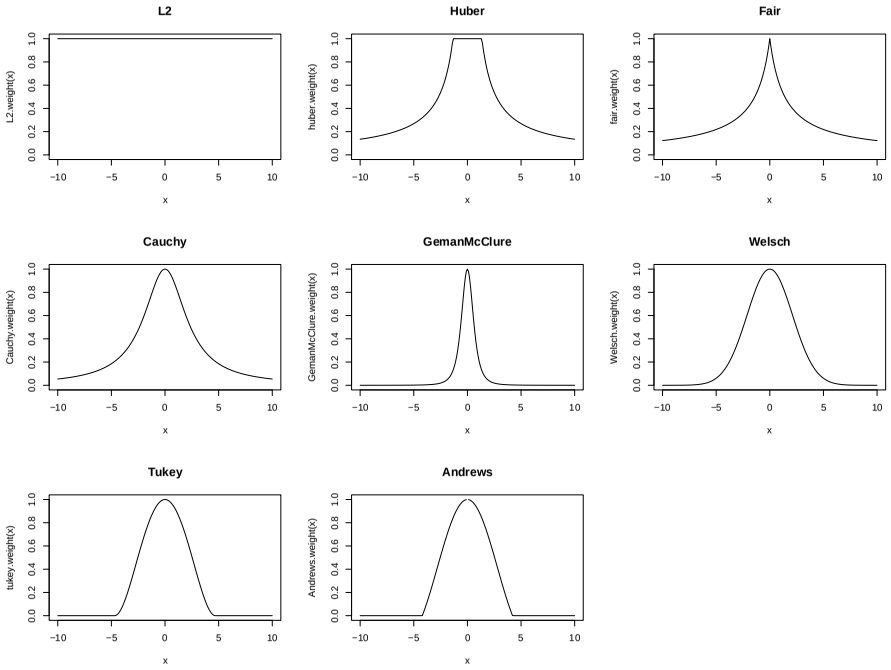

Figure 4.3: The weight functions for some common M-estimators.

k'th largest PM

A very simple approach to expression value is to simply take the k'th largest PM probe intensity in a probeset as the probeset expression summary. In particular, if $y^{(n)}_{[1]j},\ldots,y^{(n)}_{[I_n]j}$ are the order statistics for the probe intensities from probeset n on the j'th chip, then the expression value is given by

$$\hat{\beta}^{(n)}_j = \log_2\left(y^{(n)}_{[I_n-(k-1)]j}\right) \text{ where } 1 \leq k \leq I_n.$$

It has been suggested that taking $k = 2$ is a desirable value. On the other hand, taking the largest probe value ($k = 1$) would create more susceptibility to problems with saturation.

4.2.2 Multi-chip Linear Models

Multi-chip models are motivated by examining probe response patterns across arrays. Figure 4.4 shows the $\log_2$-scale probe response pattern for two probesets: one a spike-in probeset and the other a randomly selected non-differential probeset, each across 42 arrays. The parallel behavior in probe response across arrays and the relationship between concentration and expression level on each array motivates multi-chip models with probe and chip response parameters. It is commonly observed that the variability between different probes is larger than the variability of a single probe across mulitple arrays. Since there are often probes on individual arrays that behave discordantly due to non-biological causes, it is advantageous to fit the model robustly.

Linear Model

Ignoring robustness, the simplest model that can be used is that for each probeset $n = 1,\ldots,N_P$ we fit

$$\log_2\left(y^{(n)}_{ij}\right) = \beta^{(n)}_j + \alpha^{(n)}_i + \varepsilon^{(n)}_{ij}$$

where $\alpha^{(n)}_i$ is a probe effect and $\varepsilon^{(n)}_{ij}$ are independently and identically distributed errors. Note that we constrain $\sum_{i=1}^{I_N}\alpha^{(n)}_i = 0$ to make the model identifiable. This model is fit using standard linear regression techniques. The estimated $\hat{\beta}^{(n)}_j$ from the regression are the $\log_2$ expression values. Similarly, the standard error estimate is given by

$$\text{SE}\left(\hat{\beta}^{(n)}_j\right) = \frac{1}{\sqrt{I_n}}\sqrt{\frac{\sum_{i=1}^{I_n}\sum_{j=1}^{N_a}\left(\log_2\left(y^{(n)}_{ij}\right) - \hat{\alpha}^{(n)}_i - \hat{\beta}^{(n)}_j\right)^2}{I_nN_a - (N_a + I_n - 1)}}.$$

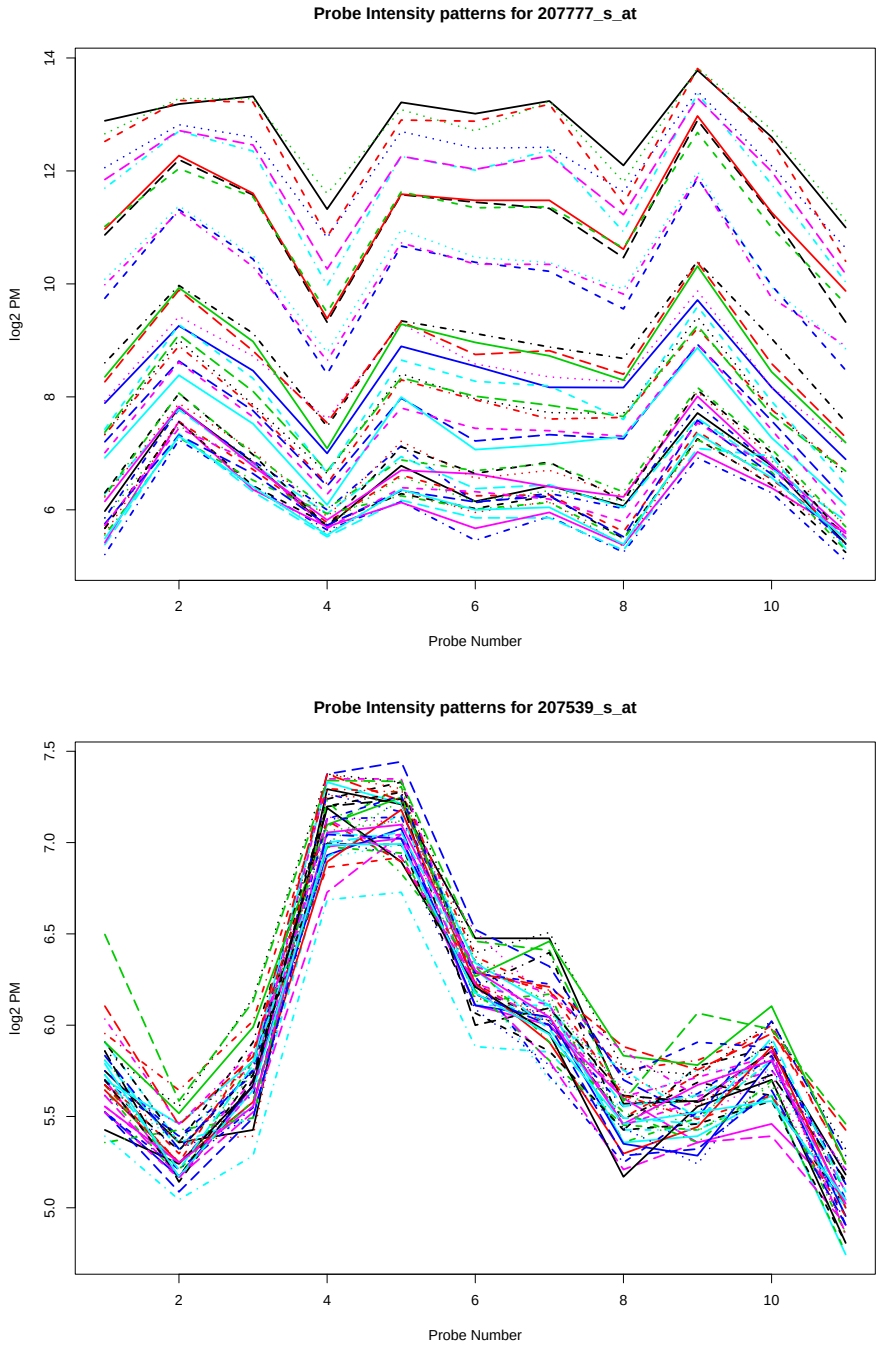

Figure 4.4: Probe response patterns for two probesets over 42 arrays. The probeset 207777_s_at was spiked-in at varying concentrations across the arrays. The probeset 207539_s_at was a randomly chosen non-differential probeset. The vertical scale differs.

However, one drawback is that the standard error estimates will be equal across arrays.

Median Polish

The median polish algorithm (Tukey, 1977) is a method for fitting the following model

$$\log_2\left(y_{ij}^{(n)}\right) = \mu^{(n)} + \theta_j^{(n)} + \alpha_i^{(n)} + \varepsilon_{ij}^{(n)}$$

with constraints $\text{median}(\theta_j) = \text{median}(\alpha_i) = 0$ and $\text{median}_i(\varepsilon_{ij}) = \text{median}_j(\varepsilon_{ij}) = 0$. The $\log_2$ expression values are given by $\hat{\beta}_j^{(n)} = \hat{\mu}^{(n)} + \hat{\theta}_j^{(n)}$. For our purposes the algorithm proceeds as follows: first a matrix is formed for each probeset *n* such that the probes are in rows and the arrays are in columns. This matrix is augmented with row and column effects giving a matrix of the form

$$\begin{array}{cccc} e_{11} & \dots & e_{1N_A} & a_1 \\ \vdots & & \vdots & \vdots \\ e_{I_n1} & \dots & e_{I_nN_A} & a_{I_n} \\ b_1 & \dots & b_{N_A} & m \end{array}$$

where initially $e_{ij} = y_{ij}^{(n)}$, and $a_i = b_j = m = 0$. Next, each row is swept by taking the median across columns (ignoring the last column of row effects) subtracting it from each element in that row and adding it to the final column (the $a_1, \dots, a_{I_n}, m$) . Then the columns are swept in a similar manner by taking medians across rows, subtracting from each element in those rows and then adding to the bottom row (the $b_1, \dots, b_{N_A}, m$). The procedure continues, iterating row sweeps followed by column sweeps, until the changes become small or zero. At the conclusion of this procedure $\hat{\mu} = m$, $\hat{\theta}_j = b_j$ and $\hat{\alpha}_i = a_i$. The e_{ij} elements will be the values of the residuals. This procedure may converge to different parameter estimates depending on whether rows or columns are swept first. In the analysis in this dissertation, rows are always swept first.

One drawback to the median polish procedure is that it does not naturally provide standard error estimates. Another is that we are restricted to balanced row-column effect models.

Robust Linear Model

This method again fits the model

$$\log_2\left(y_{ij}^{(n)}\right) = \beta_j^{(n)} + \alpha_i^{(n)} + \varepsilon_{ij}^{(n)}$$

where $\alpha_i^{(n)}$ is a probe effect and $\varepsilon_{ij}^{(n)}$ are assumed to be independently and identically distributed errors, with the constraint $\sum_{i=1}^{I_N} \alpha_i^{(n)} = 0$ which makes the model identifiable. Rather than using standard regression techniques, we use the M-estimation techniques as described previously.

Consider the general model, $Y = Xb + \varepsilon$ with X as the design matrix, Y the vector of dependent observations, b as the parameter vector and ε as the vector of errors. The fitting procedure used is iteratively re-weighted least squares. Let W be a diagonal matrix of weights such that $w_{ii} = w\left(\frac{y_i - \mathbf{x}_i\hat{b}}{\hat{s}}\right)$ for one of the weight functions defined in Table 4.1. $\hat{s}$ is an estimate of scale for which we use the median absolute deviation of the residuals, specifically $\hat{s} = \text{median}|\hat{\varepsilon}|/0.6745$. Then, the updated parameter estimates are given by

$$\hat{b} = \left(X^T W X\right)^{-1} X^T W Y$$

We continue iterating until convergence. Convergence can occur on different parameter estimates depending on the starting point, if we use the weights from a method with redescending ψ function. One good strategy is to use a fully iterated Huber M-estimator followed by a few steps using the redescending ψ function.

Huber (1981) gives three forms of asymptotic estimators for the variance-covariance matrix of parameter estimates $\hat{b}$.

$$\kappa^2 \frac{\sum \psi^2/(n-p)}{(\sum \psi'/n)^2} (X^T X)^{-1} \tag{4.2}$$

$$\kappa \frac{\sum \psi^2/(n-p)}{\sum \psi'/n} V^{-1} \tag{4.3}$$

$$\frac{1}{\kappa} \frac{\sum \psi^2}{n-p} V^{-1} \left(X^T X\right) V^{-1} \tag{4.4}$$

where

$$\kappa = 1 + \frac{p}{n} \frac{\text{Var}(\psi')}{E\,\psi'} \tag{4.5}$$

$$V = X^T \Psi' X$$

and Ψ' is a diagonal matrix of ψ' values.

In our case, the first of these estimates give the same standard error for $\hat{\beta}_j$ across arrays which may not prove useful for differentiating between arrays. The second and third forms provide standard errors that differ. Huber (1981) warns against estimating the variance-covariance matrix, as one would for a weighted least squares regression as it is "non-robust in general."

Many useful quantities for quality assessment of oligonucleotide arrays can be derived as by-products from this robust linear model procedure (Collin et al., 2003).

A drawback to the robust linear model approach as taken to summarization is that it can become very slow as the number of parameters in the model increases. Specifically, there are operations in the iteratively reweighted least squares that are $O\left(p^3\right)$, where p is the number of regression parameters. Often, it makes sense to fit a single parameter for each treatment group rather than each array. Such an approach will be discussed in Chapter 6.

4.3 Results

In this chapter, we make use of the Affymetrix U133A spike-in dataset described in Appendix A.2. The key features of this data are that 42 probesets have been spiked-in at variable concentrations against a background common cRNA. Our comparisons were limited to a representative subset of the methods we proposed. Specifically, for the robust linear model approach we used three different weighting schemes, Huber, Tukey Biweight and Geman-McClure, each progressively more robust. So as to avoid potential confounding each summarization method was applied to data that had not previously been preprocessed.

4.3.1 Assessing the Impact of Summarization Methods on Expression Values and Fold-change Estimates

The first comparison made was to examine the slopes of a regression line fitted to a plot of observed expression versus log spike-in concentration for the spike-in probesets. Table 4.3 contains the estimated slopes for each of the methods. Lower concentrations were those below 4 pM, the middle concentrations were those between 4 pM and 128 pM, and the higher concentrations were those above 128 pM. There did not seem to be a method that was the best, although second largest PM which had lowest slope across all intensities was clearly the worst of the methods.

To examine how fold-change estimates were affected by summarization, we considered the IQR of non-differential probesets and the slope of the regression of observed log FC against that expected from the spike-in concentrations. These figures are in Table 4.4. The robust multi-chip model

Method	All		Lower		Middle		Upper	
	Slope	R^2	Slope	R^2	Slope	R^2	Slope	R^2
Average Log	0.51	0.89	0.23	0.49	0.66	0.85	0.71	0.67
Log Average	0.49	0.88	0.20	0.32	0.66	0.86	0.69	0.74
Median Log	0.54	0.90	0.26	0.45	0.68	0.85	0.73	0.68
Tukey Biweight	0.54	0.90	0.25	0.46	0.68	0.86	0.74	0.69
Second Largest PM	0.49	0.86	0.20	0.21	0.66	0.83	0.63	0.73
Median Polish	0.52	0.89	0.23	0.46	0.69	0.85	0.69	0.64
Linear model	0.51	0.89	0.23	0.49	0.66	0.85	0.71	0.67
Robust Linear Model (Huber)	0.52	0.90	0.24	0.48	0.68	0.86	0.70	0.70
Robust Linear Model (Biweight)	0.53	0.89	0.24	0.42	0.69	0.86	0.72	0.71
Robust Linear Model (Geman-McClure)	0.53	0.88	0.22	0.36	0.71	0.86	0.70	0.68

Table 4.3: Slope (and R^2) for spike-in probesets. The ideal would be a slope near 1 that is even across intensities.

Method	IQR	Slope
Average Log	0.207	0.52
Log Average	0.214	0.52
Median Log	0.236	0.55
Tukey Biweight	0.225	0.55
Second Largest PM	0.244	0.51
Median Polish	0.205	0.54
Linear model	0.207	0.52
Robust Linear Model (Huber)	0.205	0.53
Robust Linear Model (Biweight)	0.205	0.54
Robust Linear Model (Geman-McClure)	0.210	0.54

Table 4.4: Assessing impact of summarization on FC estimates. IQR of fold-change estimates for non-differential probesets. Slope estimates are for the regression of observed fold-change against expected fold-change for spike-in probesets.

Method	0% FP	5% FP	AUC
Average Log	0.165	0.943	0.923
Log Average	0.127	0.933	0.905
Median Log	0.128	0.924	0.898
Tukey Biweight	0.133	0.932	0.907
Second Largest PM	0.094	0.885	0.849
Median Polish	0.146	0.945	0.923
Linear model	0.165	0.943	0.923
Robust Linear Model (Huber)	0.156	0.946	0.926
Robust Linear Model (Biweight)	0.141	0.945	0.925
Robust Linear Model (Geman-McClure)	0.137	0.934	0.914

Table 4.5: Assessing impact of summarization step on detecting differential expression using ROC curve quantities.

Method	Median	IQR	Range 80%
Average Log	-0.07	1.17	2.26
Log Average	-0.32	1.24	2.37
Median Log	0.00	1.05	2.28
Tukey Biweight	0.01	1.06	2.24
Second Largest PM	-0.96	1.42	2.35
Median Polish	0.00	0.14	0.32
Linear model	0.00	0.16	0.31
Robust Linear Model (Huber)	0.00	0.15	0.30
Robust Linear Model (Biweight)	0.00	0.15	0.30
Robust Linear Model (Geman-McClure)	0.00	0.12	0.34

Table 4.6: Summary statistics on residuals of non-differential probesets from the summarization methods.

methods had smaller IQR for the non-differential probesets. For the spike-in probesets, the slopes were very similar across methods. The second largest PM performed the worst in both comparisons.

ROC curves were used to compare the ability of the different methods to detect true differential expression. Table 4.5 shows three ROC curve quantities for each method. In particular, the true positive rate when the false positive rate was 0%, true positive rate at 5% false positives and the area under the curve up to 5% false positives. For all three quantities, higher values were better. The linear model and log average methods seemed to do the best at 0% false positives, the robust linear model was better at both 5% false positives and for AUC. Most methods did well in this comparison except the second largest PM method which was clearly the worst since it had significantly smaller values for each of the three quantities.

Finally, we considered the residuals for each of summary methods. A summary method that more closely modelled the observed data and thus had smaller residuals was better. Table 4.6 gives the

median residual, the IQR of the residuals and the range of the central 80% of the data. The most important observation was that the multi-chip models greatly reduced the variability of the residuals, because the probe-effect captured a great deal of the variability in the data.

4.3.2 Using Probe-level Models to Detect Outlier Probes and Arrays at the Probeset-level

As observed in Collin et al. (2003), the weights and residuals generated by the model fitting procedure allowed the construction of quality assessment diagnostics. Collin et al. (2003) focused on assessment at the array-level. That is, the focus was on identifying poorly performing chips. The focus here is on assessments at the probeset-level.

for each probeset k = 1 to N_p **do**
For each row count over j the number of $w_{ij}^{(k)} < W_\tau$ call these values C_{i+}.
Set $E_r = \sum_{i=1}^{I} C_{i+}/I$
For each column count over i the number of $w_{ij}^{(k)} < W_\tau$ call these values C_{+j}.
Set $E_c = \sum_{j=1}^{J} C_{+j}/J$
Compute $\chi_r^2 = \sum_{i=1}^{I} \frac{(C_{i+}-E_r)^2}{E_r}$
Compute $\chi_c^2 = \sum_{i=1}^{I} \frac{\left(C_{+j}-E_c\right)^2}{E_c}$
if $\chi_r^2 > \chi_{\text{crit1}}^2$ **then**
Check $\forall i$ if $C_{i+} > C_\tau J$ then probe i is an outlier for probeset k
end if
if $\chi_c^2 > \chi_{\text{crit2}}^2$ **then**
Check $\forall j$ if $C_{+j} > C_\tau I$ then array j is an outlier for probeset k
end if
end for

Table 4.7: A procedure for identifying outlier probes across arrays and outlier arrays across probes.

We constructed a procedure to identify probes and arrays that were the outliers. Specifically, we used the observed weights to screen probesets for both poorly performing probes (across a number of chips) and poorly performing arrays (as judged across a number of probes). The procedure is outlined in Table 4.7. The process works by first defining extreme outliers as any probe where the weight is smaller than some value W_τ. A useful choice for this is the value of the weight function

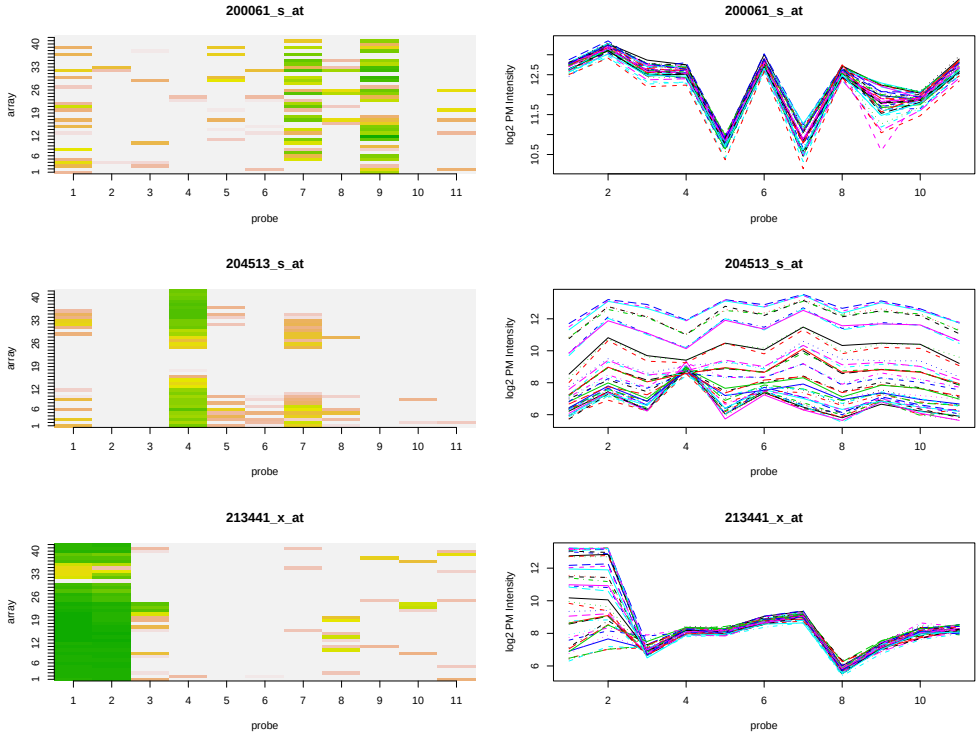

Figure 4.5: Three Probesets determined to have probe outliers. The first has a noisy probe. The second is a spike-in probeset, with probe 4 that does not seem to differentially hybridize except at very high concentrations. The third is a non-differential probeset with some probes (1,2) that seem to be cross hybridizing with a spike-in transcript.

at -3 or $+3$ from standard normal data. In the case of Huber's weight function, $W_\tau < 0.4483$. Then a check is made for whether there are differences in the number of severe outliers across rows and then across columns. with $\chi^2_{\text{crit1}} = \chi^2_{I-1}(1-0.001)$ and $\chi^2_{\text{crit2}} = \chi^2_{J-1}(1-0.001)$. For those probesets where there are difference across rows, look for rows with more than $C_\tau J$ outliers and call these outlier probes. Similarly, for those probesets where there are differences across columns, look for columns with more than $C_\tau I$ outliers and call these outlier arrays. A useful value of $C_\tau = 0.45$. In otherwords, flag probes or arrays where almost half or more are outliers. While this procedure was not particularly sophisticated, more complex implementations would have involved perhaps the sum of weights, it highlights how robust multi-chip models allowed troublesome probes and probesets to be identified.

Figure 4.5 shows three of the probesets identified as having outlier probes, using the procedure in Table 4.7. The Huber weighting scheme was used. In each of these plots the weights are on

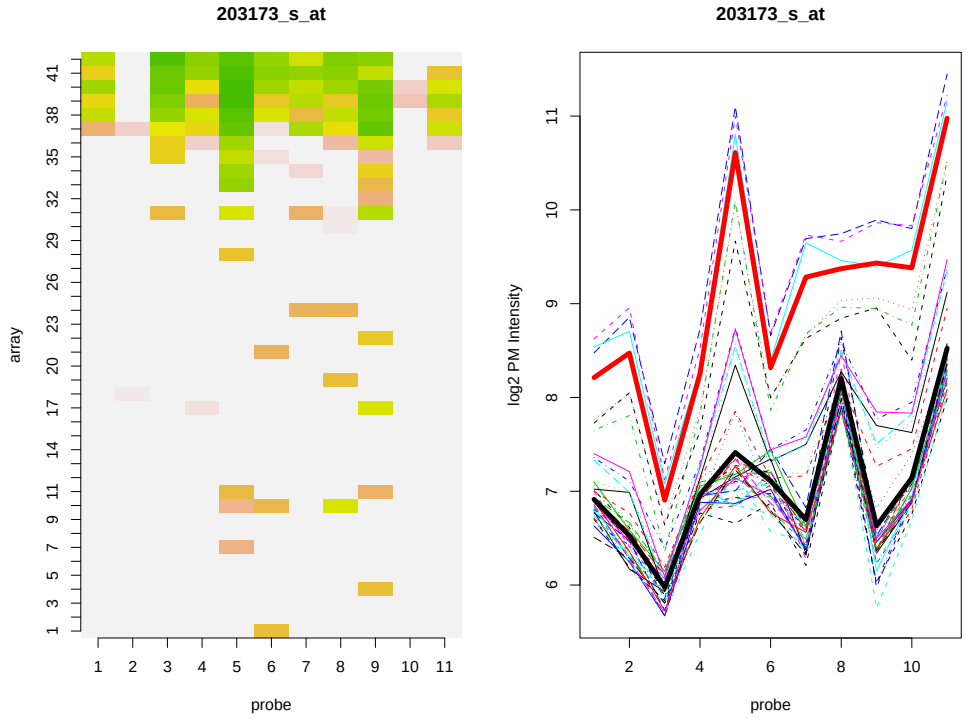

Figure 4.6: A probeset with five outlier chips. The two emphasized lines are averages over the outlier and non-outlier groups.

the left hand side and the probe-pattern is shown on the right. Darker areas in the weights plots had lower weights with arrays on the vertical axis and probes on the horizontal axis. The first probeset 200066_s_at was a non-spikein probeset which had two somewhat noisy probes (5, 7) as was visible in both the weights plot and the probe-pattern plot. A spike-in probeset, 204513_s_at, had probe 4 non-responsive except at very high concentrations. The third probeset 213441_x_at was also a non-spikein and expected to be non-differential. However, the first two probes seemed to be cross-hybridizing with one of the spike-in probesets. These three cases highlighted typical cases of probesets with probe-outliers.

A probeset with array outliers is shown in Figure 4.6. Specifically, this probeset had five arrays which had a number of low weight probes. The averaged probe-response pattern for the five outlier arrays is given by one wide line and the other wider line is the averaged probe-response for the other 37 arrays. It was interesting to note that the probe-response patterns differed between the two groups of arrays. Particularly, probe five was the highest or near highest for the five outlier arrays while considerably less than the maximum for the remaining arrays. Also the pattern across probes

Summary	Number Flagged	Total Available
Outlier Probes	103	248152
Outlier Arrays	14	936600
Outlier Probesets	103	22300

Table 4.8: Outlier statistics for HGU-133A dataset.

6,7,8,9 did not seem to agree between the two groups. In essence, the outlier arrays that were chosen in this manner are those where on a particular array the probeset has a different response pattern.

Given a method by which to identify outlier probes and arrays within probesets a sensible approach was to summarize the dataset using these flags. The number of outlier probes, the number of outlier arrays and the number of probesets with such outliers were useful quantities. Table 4.8 shows these quantities for the HGU133A spikein dataset. The number of probes, arrays and probesets flagged did not seem particularly large in either absolute or relative terms. Without doing a thorough comparison of many datasets we could not determine whether this represents good or poor quality data. However, it would seem sensible to assume that Affymetrix would produce public datasets with great care under ideal laboratory conditions.

4.4 Discussion

In this chapter, we presented a number of methods of combining multiple probe intensities to produce probeset summary values. These methods were either single array methods, such as log Average, Median log and 2nd largest PM or multi-array methods such as the median polish and linear model approaches. Some methods were robust like the Tukey biweight or robust linear model approach and others like Average Log had no provision for dealing with outliers. However, in the comparisons made no significant difference in the performance of the methods at estimating FC or detecting differential expression was found. The only method which could be said to have performed poorly was 2nd largest PM.

The multichip models modelled the data very well, specifically the residuals from these models were much smaller than from the single chip methods. In addition, a method for detecting outlier probes across arrays and arrays across probes based upon output from the robust linear model approach was proposed and its performance demonstrated.

An important caveat about the conclusions of this chapter is that data could reasonably be expected to be of the highest quality and also that there are only 42 probesets that should be changing between arrays. In other words, there is not sufficient evidence to suggest that using a robust multi-chip approach is unreasonable or unnecessary. Also this chapter looked at the summarization methods in isolation from the other steps in preprocessing. Chapter 5 will explore how summarization interacts with both background correction and normalization methodologies.

Chapter 5

Expression Measures as a Three-step Process

Earlier chapters discussed pre-processing as isolated steps, while this chapter considers combining the different stages together. Section 5.1 introduces a three-step procedure for computing gene expression values, Section 5.2 compares expression values computed using various pre-processing methodologies, and Section 5.3 discusses the results.

5.1 Introduction

This chapter examines the process of producing a gene expression measure. Chapters 2, 3 and 4 were concerned with comparing different methodologies at each of the three pre-processing stages (background, normalization, summarization), but did not consider how these methods interact. Expression values were only examined in the context of the RMA expression measure.

The procedure for generating expression measures can be considered as a three-step process. Let **X** be raw probe intensities across all arrays, in the form of a matrix with probes in rows and arrays in columns, and **E** be probeset expression measures with probesets in rows and arrays in columns. Let B be an operation which background corrects probes on each array. In general, a background method should operate in an array by array manner, that is each of the columns of **X** are treated individually. Next, let N be the operation which normalizes across arrays. In other words it reduces

variability between the columns of **X**. Finally, let S be the operation which combines probes together to compute an expression measure. This operation could involve a possible transformation and may work across arrays. The process of computing measures of expression can be written as

$$\mathbf{E} = S(N(B(\mathbf{X}))). \tag{5.1}$$

Many popular expression measures can be put into this three-step framework. In the case of RMA expression values (Irizarry et al., 2003a), (Irizarry et al., 2003b) B is the convolution model method described in Section 2.2.1, N is carried out by quantile normalization (Bolstad et al., 2003) and S is an operation which takes $\log_2$ of the probes and fits a robust multi-chip linear model using the median polish algorithm as described in Section 4.2.2.

For the MAS 5.0 expression measure (Affymetrix, 2001a), B is the location specific background correction followed by subtracting the ideal mismatch from the PM probes, N is identity operation which leaves the data unchanged (in the MAS 5.0 framework normalization takes place after summarization) and S is the process which takes $\log_2$ of the data and then uses the 1-Step Tukey biweight.

Finally, for the MBEI (Li and Wong, 2001a), B is a modified version of the location specific correction outlined in Section 2.2.2 for the PM only model and both the location specific correction and subtraction of MM from PM in the PM-MM model. The normalization N is a version of the non-linear method outlined in Section 3.2.2. Summarization is carried out by fitting a multi-chip multiplicative model with additive errors.

Any of the background, normalization and summarization methods discussed in this dissertation can be combined together to produce an expression measure. However, for brevity, this chapter will examine only a subset of possible combinations. Of particular interest will be how robustness handles the noisier background methodologies and the effect that normalization has on the multi-array summarization methods.

5.2 Results

In this section, the Affymetrix HGU133A spike-in experiment described in Appendix A.2 received further consideration. This dataset was also considered in Chapter 4. An analysis based on the

Dilution/Mixture dataset, described in Appendix A.5, where meaningful biological changes were expected was also considered. Three background options were considered: no correction, the convolution method (a small background adjustment) and the location specific followed by the ideal mismatch methodologies (a more serious background adjustment). Four normalization methodologies were used: none, scaling (a linear method), quantile (a non-linear approach), and quantile probeset (a non-linear approach which seeks to preserve probe patterns). Finally, four options were considered for the summarization process: average log, Tukey biweight (a robust average), median-polish and robust linear model. Expression values were computed for all combinations of these methods resulting in 48 different sets of expression values.

5.2.1 Analyzing the Spike-in Data

Expression Values, Fold-change and Detecting Differential Expression

An examination of slopes for regression of observed expression on spike-in concentration reaffirmed the observations in Chapters 2, 3 and 4. The choice of background method had the largest effect on the slope. Specifically, the median overall slope was around 0.52 for no background correction, 0.67 for the convolution method and 0.75 for the Affymetrix ideal MM methods. There were clear differences between methods when partitioning into low, middle and high concentrations. With the low concentrations the Affymetrix methodology had the highest slopes, while at the highest concentrations the other two methods did slightly better. After removing the effect of background method, there was no clear difference between slopes when partitioning by normalization method. There was more difference between summarization methods, with the Tukey biweight being slightly better at the low end, but not in the middle and high ranges. However, the differences between summarization methods were of a much smaller magnitude than between background methods. Thus, it was not possible to declare one summarization method a clear winner.

Similarly, when examining observed fold-change against expected fold-change, the highest slopes were those where the location specific correction and ideal mismatch were used. There was no appreciable difference in slopes between normalization methods or between summarization methods. Ideally, non-differential probesets would have small fold-change estimates of little variability. Figure 5.1 shows boxplots of the IQRs of the fold-change for the non differential probesets stratified by pre-processing method. It was clear that the background method had a large effect on the variability

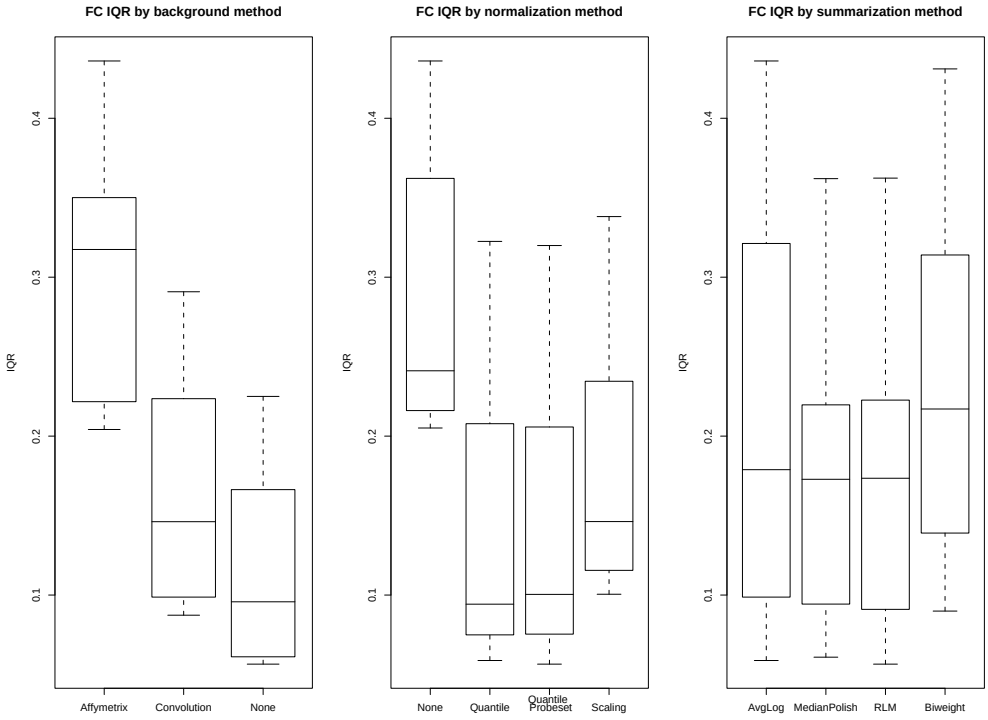

Figure 5.1: Boxplots of the IQR of FC for non-differential probesets stratified by pre-processing method. Lower values are better.

Normalization	Background	Number of Probesets Flagged			Number of outliers	
		Probe	Chip	Both	Probe Outliers	Chip Outliers
None	None	97	5	0	103	14
Quantile	None	87	7	0	93	14
Quantile Probeset	None	97	5	0	103	14
Scaling	None	97	5	0	103	14
None	Convolution	350	25	5	370	25
Quantile	Convolution	363	13	0	371	13
Quantile Probeset	Convolution	350	25	5	370	25
Scaling	Convolution	350	25	5	370	25
None	Affymetrix	5788	2711	663	6965	4126
Quantile	Affymetrix	5746	2697	639	6887	4102
Quantile Probeset	Affymetrix	5788	2711	663	6965	4126
Scaling	Affymetrix	5788	2711	663	6965	4126

Table 5.1: Counts of probesets flagged varies across pre-processing methodologies. A large number of flagged probesets implies that the multi-array linear model is not fitting well.

of the non-differential genes, with the Affymetrix correction leading to the largest IQRs. When stratified by normalization method it is clear that non-linear normalization reduced variability the greatest, but scaling also decreased the variability. The robust linear model approaches had slightly lower IQRs than the single chip methods.

ROC curves for detecting differential expression showed that the largest differences in the curves were due to the background methodology. Figure 5.2 shows the boxplots of the AUC up to 5%, dividing the expression values by background methodology. As was seen in Chapter 2, the location specific background followed by ideal mismatch correction significantly reduced the number of differential genes detected. To adjust for these large differences, we subtracted the median AUC for each background method from the AUC computed for expression values using that background method. The adjusted AUC were compared by stratifying across normalization methods and then across summarization methods. The non-linear normalizations had the highest areas under the curve, while the linear normalization, did not do quite as well. However, it is important to note that this dataset did not require a large amount of normalization. Similarly, we found that the robust linear model approach had the highest AUC while the biweight method had the lowest AUC.

Effect of pre-processing on linear model approaches

In Chapter 4, we observed that a common probe-response pattern across arrays suggested fitting multi-array models. Pre-processing can have an effect on this probe-pattern. Clearly, the ideal situ-

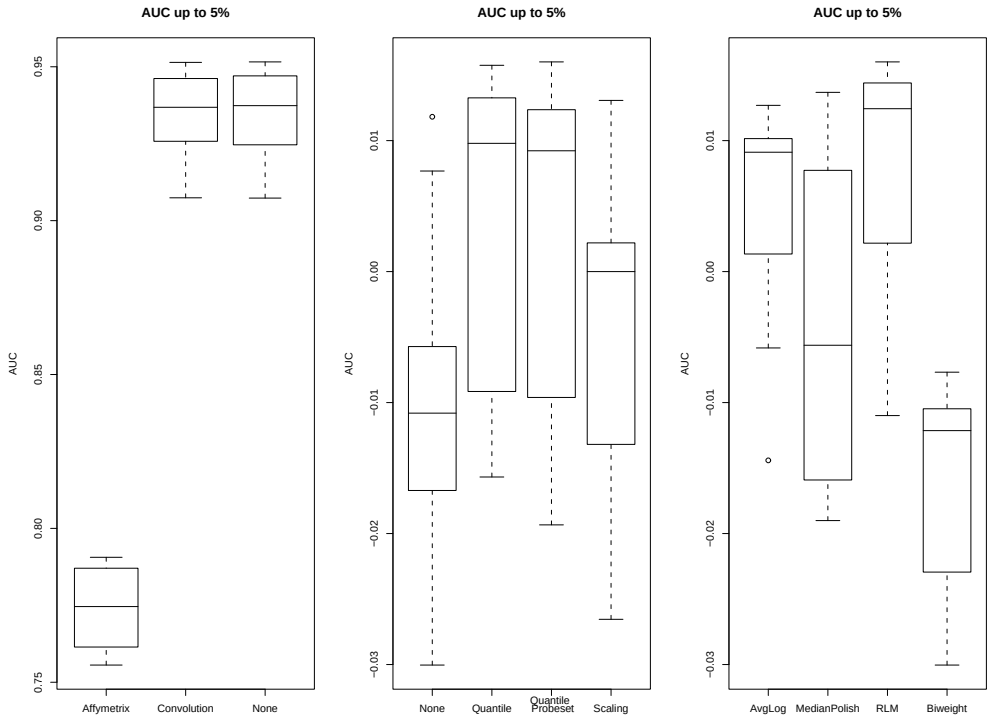

Figure 5.2: Boxplots of the AUC up to 5% for ROC curve. There are clear differences between background methods. After adjusting for the differences in background methods we find differences in area for normalization and summarization methods. Higher values are better.

ation would have been for the pre-processing to have a minimal effect on the parallelism. To investigate the possibility that the probe response was being significantly affected, the outlier detection procedure proposed in Section 4.3.2 was applied to robust linear model summarized data after using each of the background and normalization methods. If the probe-response pattern is significantly altered then a greater number of probe and chip outliers should be observed. Table 5.1 shows the number of flagged probesets, outlier probes and outlier arrays. In this comparison, the background method had the largest effect on the number of outliers, with the Affymetrix background methodology having clearly the worst effect. Looking at the summarization computed using the Affymetrix corrections without normalization, we found that after dividing the probesets by average expression value, an approximately equivalent number of outliers were found in the lowest third of the data, 3140 probesets, in the mid range, 2135 probesets, and 2561 probesets in the highest third of the data. So it can be concluded that the Affymetrix correction significantly affects the probe pattern at all intensity values. With the convolution method we found the bulk of the outliers had either low average expression or high expression. Only 14 probesets in the mid range were flagged as having some kind of outlier.

By design the quantile probeset normalization should not change the probe pattern. However, unexpectedly we found that neither the scaling or quantile methods seemed to have adverse effects on the probe-pattern. Based purely on the count of flagged probesets it appeared that quantile normalization improved the linear fit when either no background correction or the Affymetrix methods were used. Closer examination of the flagged probesets showed that a small number of borderline probesets were either being included or excluded. Examining the average weight per probe showed that on average the quantile normalized data had slightly lower weights, although the effect was not large. Since the Affymetrix U133A spike-in dataset did not demonstrate a significant need for normalization, there is not sufficient evidence to conclude that normalization does not have an effect on the probe-response pattern.

5.2.2 Analyzing the Dilution/Mixture Data

Unlike the spike-in experiment, the dilution/mixture dataset did not have a known “truth”. That is we did not know the true concentrations of any particular probeset, nor which were the truly differential genes. Instead, a truth was generated by choosing differential genes using one subset of the data and then tested on another subset of the data. Since the RMA expression measure without

Method	AUC (Rank)	Method	AUC (Rank)	Method	AUC (Rank)
None	0.81 (1)	None	0.45 (4)	Avglog	0.69 (1.6)
Convolution	0.70 (2)	Quantile	0.77 (1.6)	Median Polish	0.67 (3.0)
Affymetrix	0.50 (3)	Quantile Probeset	0.77 (1.4)	RLM	0.69 (1.8)
		Scaling	0.70 (3)	Biweight	0.63 (3.7)

Table 5.2: Effect of pre-processing on detecting differential expression as judged by AUC up to 5% using the dilution data. AUC values are averaged across all other pre-processing methods. Ranks (in parentheses) are averages of ranks across other pre-processing methods.

background correction was observed to perform well at detecting differential expression in *affycomp* Cope et al. (2004), it was used to choose a set of differential genes. More sophisticated methods of detecting differential expression are discussed in Chapter 6. First, we considered the 5 Liver arrays and 5 CNS arrays both at 10μg. Arbitrarily, the 400 probesets with the most extreme FC estimates were called differential, the remaining probesets were called non-differential. The 20 μg Liver and 1.25 μg CNS arrays were used to test the performance of the methods. Because there were such large differences in concentration, and the biological differences were confounded with this, the performance of the normalization methods was important.

Using the 400 probesets chosen as differential the performance of the different pre-processing methods can be compared. Table 5.2 compares the performance across each pre-processing stage using the AUC under the ROC curve up to 5% false positives. In each case, the AUC was averaged across all expression measures using that pre-processing method. As with the spike-in dataset, the background adjustment had a large effect on the number of true positives identified. However, a similarly sized difference in AUC was also observed for the normalization method. In particular, the two quantile methods performed the best while there was a drop off in AUC with the scaling method. The AUC was significantly lower when no normalization was applied. As with the spike-in dataset, the biweight summarization has the lowest AUC, while the robust linear model and average log methods performed well.

One method of assessing how well a linear model fits is by examining the R^2, a higher R^2 implying a better fitting model. The fit of the robust linear model was assessed in this way. For each combination of background and normalization methods, the robust linear model was fit and the R^2 value computed for each probeset. For any fixed background method, there were small differences in R^2 across normalization methods, with perhaps a very slight advantage to the standard quantile method. However, larger differences were observed for any fixed normalization method across background

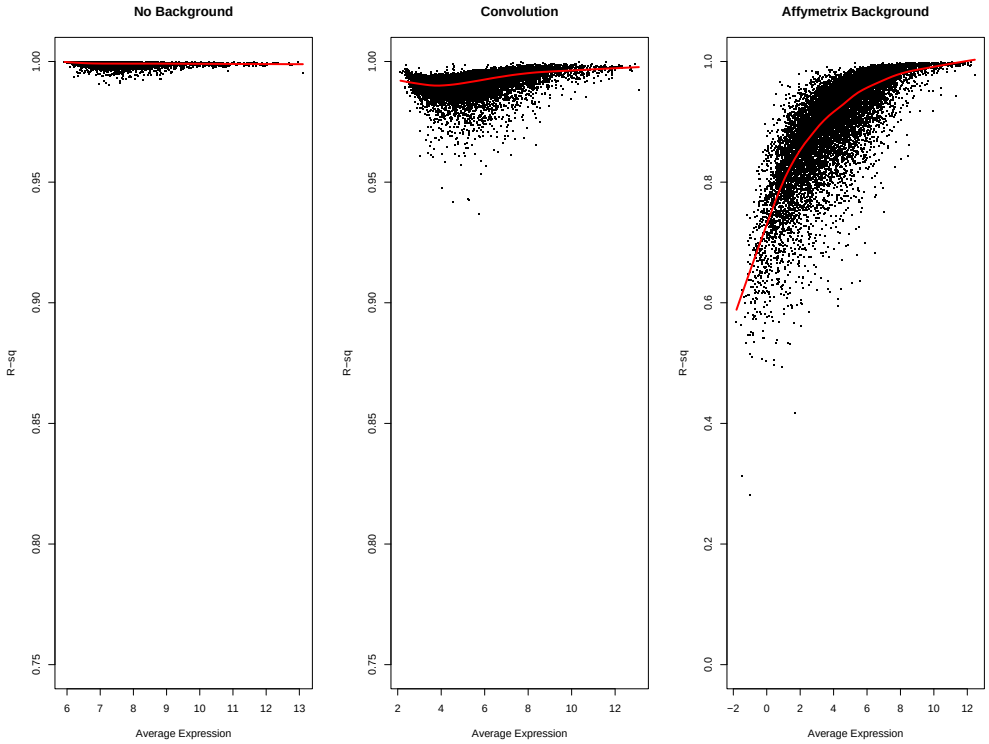

Figure 5.3: R^2 vs Average Expression. Compared for three background adjustments. Higher values are better. The vertical scale changes between the three plots.

methods. In particular, the larger the correction the smaller the R^2 with the Affymetrix background correction having the lowest R^2 in general. Figure 5.3 compares the R^2 values with average expression level for the three background adjustments. The background adjustments decreased the R^2 the most for the low intensity probesets. The convolution model, while decreasing the R^2, still allowed a very good fitting linear model to be fit across all ranges. The Affymetrix corrections had a large effect on how well the model fit.

Figure 5.4 highlights how the probe pattern is affected by the convolution background and quantile normalization. The figure shows the probe pattern for two probesets: one is non-differential and the other is differentially expressed between liver and central nervous system. Different types of lines have been used to represent the tissue source. For the non-differential probeset, there is an apparent difference in expression level between the two tissue types, but a similar probe pattern. The convolution background correction, does not remove this apparent difference in expression level, but instead makes minor adjustments to the probe-pattern. When the data is instead quantile normalized we see that the level of expression seems to coincide between all arrays and that the probe-patterns matched up very well. Similarly, quantile normalizing the background adjusted data yielded a reasonable plot with just slightly more variation in the value of each probe. Initially, the differential probeset seemed to have little difference in level between liver and CNS, and the background adjustment made little difference to this observation. After quantile normalization, there was immediately clear separation between the liver and CNS tissues and very good matching of arrays within each tissue group. There was no change in the probe pattern after normalization.

5.3 Discussion

This chapter considered the construction of an expression measure as a three-step procedure. Background correction, normalization and summarization procedures from Chapters 2, 3 and 4 were combined to produce expression measures. The comparative effect that each pre-processing stage had on the relationship between expression value and true concentration, on detecting differential genes and on the probe-pattern was examined.

With the spike-in dataset, we found that background correction made the largest difference to the slope between between computed expression value and the known concentration. More correction gave a higher slope. The background correction also had the largest effect on how many differential

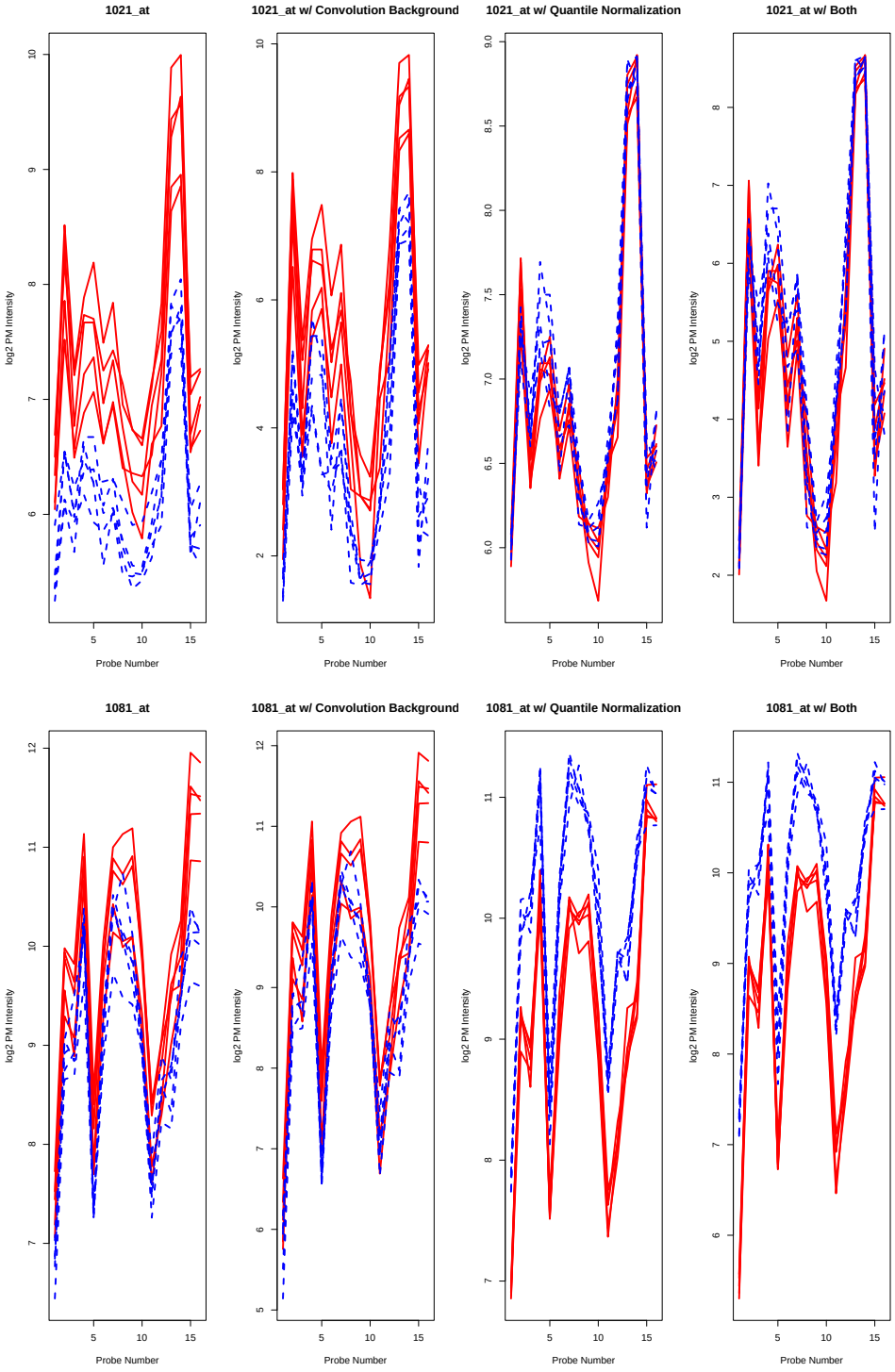

Figure 5.4: Probe-patterns for a non-differential and a differential probeset: without pre-processing, after convolution background, after quantile normalization and after both.

genes were correctly identified. Both normalization and summarization had small effects on how many differential genes were detected. However, this particular dataset did not require much normalization. The background method was also observed to have a great influence over how well the multi-array model was fitting.

The dilution/mixture dataset allowed us to assess how the methods performed with data where there were true biological differences. There were also great apparent changes in the expression level between the two tissues due to a technical difference, the quantity of cRNA hybridized, as well as between individual arrays because of scanner differences. With this dataset, normalization and background adjustment had comparable effects on the number of differential probesets correctly identified. An examination of how the probe-response pattern was affected by pre-processing showed that with background adjustment the model fit less well. The normalization procedures did not seem to reduce the ability of the multi-chip model to fit the data.

We have shown that the three-step framework is a reasonable method for producing expression measures, and that the conclusions we made about the individual pre-processing stages in the previous chapters hold true in a more general context.

Chapter 6

Probe-Level Model Based Test Statistics for Detecting Differential Expression

6.1 Introduction

Chapter 4 discussed methods for producing expression summaries using a particular robust linear model fit to probe intensity data. In this chapter that methodology is extended to more general probe-level models and test statistics are developed for differential expression. We will compare the performance of our probe-level model based test statistics against more traditional methods based upon expression values.

6.2 Methods and Data

First, we define a probe-level model (PLM) as a model of the following form

$$y_{ij}^{(k)} = f\left(X_{ij}^{(k)}\right) + \varepsilon_{ij}^{(k)} \tag{6.1}$$

where $X_{ij}^{(k)}$ are measured factors, for example treatment specific effects and probe-effects, and covariates for a particular probe and f is an arbitrary function. The indices i and j refer to probe and array respectively. We use k to index probeset. In this dissertation we restrict ourselves to linear

functions f. To fit the models, we use the M-estimation regression procedures described in Section 4.2.2.

In this chapter the focus will be on two specific PLM: the array effect model and the treatment effect model. The array effect model has a parameter for each array. For each probeset $k = 1,\ldots,K$ with $i = 1,\ldots,I_k$ probes each on $j = 1,\ldots,J$ arrays, we fit the following model

$$y_{ij}^{(k)} = \alpha_i^{(k)} + \beta_j^{(k)} + \varepsilon_{ij}^{(k)} \tag{6.2}$$

where $y_{ij}^{(k)}$ are pre-processed $\log_2$ perfect match intensities, $\alpha_i^{(k)}$ are probe effects and $\beta_j^{(k)}$ are array effects ($\log_2$ expression values). It is further assumed that $\mathrm{E}\left(\varepsilon_{ij}^{(k)}\right) = 0$ and $\mathrm{Var}\left(\varepsilon_{ij}^{(k)}\right) = \sigma^2$. To make the model identifiable we use the constraint $\sum_{i=1}^{I_k} \alpha_i^{(k)} = 0$.

The treatment effect model has a parameter for each treatment grouping. This parameter is an average expression level for that treatment group. This means that it will have fewer parameters than the array effect model. For each probeset $k = 1,\ldots,K$ with $i = 1,\ldots,I_k$ probes each on $j = 1,\ldots,J$ arrays, we fit the following model

$$y_{ij}^{(k)} = \alpha_i^{(k)} + \tau_{l_j}^{(k)} + \varepsilon_{ij}^{(k)} \tag{6.3}$$

where l_j, the group membership for array j, is from $1,\ldots,L$. Both $\alpha_{ij}^{(k)}$ and $\varepsilon_{ij}^{(k)}$ are defined in the same manner as the array effect model.

6.2.1 Test Statistics

Suppose each array can be classified by condition, for instance treatment or control, or perhaps even into multiple different treatment groups. The goal is to determine which genes are differentially expressed between conditions. We concentrate on pairwise comparisons between two treatment groups, l and m. For each gene every test statistic is computed. Our first test statistic is raw $\log_2$ fold-change which is given by

$$\mathrm{FC} = \bar{X}_l - \bar{X}_m \tag{6.4}$$

where $\bar{X}_l = \frac{\sum_{j=1}^{J} \beta_j \mathrm{Ind}(j \in l)}{\sum_{j=1}^{J} \mathrm{Ind}(j \in l)}$ is the mean expression level in group l and $\bar{X}_m$ is similarly defined.

The standard two sample t-statistic is given by

$$t_{\mathrm{std}} = \frac{\bar{X}_l - \bar{X}_m}{\sqrt{\frac{s_l^2}{n_l} + \frac{s_m^2}{n_m}}} \tag{6.5}$$

where s_l and s_m are the sample standard deviations. One drawback with this test statistic is that a stable gene with very small variability could result in a extremely large absolute t_{std} value.

We also consider a "robust" version of the two sample t-statistic as given by

$$t_{\text{robust}} = \frac{\tilde{X}_l - \tilde{X}_m}{\sqrt{\frac{\tilde{s}_l^2}{n_l} + \frac{\tilde{s}_m^2}{n_m}}} \tag{6.6}$$

where $\tilde{X}_l$ and $\tilde{X}_m$ are the median expression values in group l and group m respectively and $\tilde{s}_l$ and $\tilde{s}_m$ are the median absolute deviations. While such test statistics are statistically unappealing they are sometimes used by practitioners in the real world.

To handle the potential drawbacks of the standard two sample t-statistic, a number of procedures have been proposed to deal with the small denominator problem including Cyber-t (Baldi and Long, 2001), SAM (Tusher et al., 2001) and Broberg (2003) among others. Most of these method deal with the problem by inflating the denominator in some manner. In this dissertation we have considered two approaches to moderate the t-statistic. A simple moderated t-statistic is given by

$$t_{\text{mod}} = \frac{\bar{X}_l - \bar{X}_m}{\sqrt{\frac{s_l^2}{n_l} + \frac{s_m^2}{n_m}} + s_{\text{med}}} \tag{6.7}$$

where s_{med} is the median of $\sqrt{\frac{s_l^2}{n_l} + \frac{s_m^2}{n_m}}$ across all genes. A much more sophisticated moderation is also used, specifically, an empirical Bayes moderated t-statistics of the form described by Lönnstedt and Speed (2002) and Smyth (2004) as implemented in the Bioconductor *limma* package. We call this statistic t_{ebayes}.

The novel approach used in the chapter is to use the estimated variance covariance matrix from the model as part of the test statistic. In particular, let Σ be the portion of the variance covariance matrix related to the β from fitting the array effect model. Now consider the contrast vector $\mathbf{c}$ where element j of c is $\frac{1}{n_l}$ if array j is in group l, $-\frac{1}{n_m}$ if in group m and 0 otherwise. Then, either

$$t_{\text{PLM.1}} = \frac{\mathbf{c}'\hat{\beta}}{\sqrt{\mathbf{c}'\text{diag}(\hat{\Sigma})\mathbf{c}}} \tag{6.8}$$

or

$$t_{\text{PLM.2}} = \frac{\mathbf{c}'\hat{\beta}}{\sqrt{\mathbf{c}'\hat{\Sigma}\mathbf{c}}} \tag{6.9}$$

can be used to test for differential expression between treatment groups l and m. It is important to note that the numerator in this test-statistic is identical to that in the previous test statistics. Because

we use *M*-estimation procedures to fit our models the off-diagonal elements of Σ are non-zero. However, with an *M*-estimation with a fairly non-aggressive weighting scheme such as the method proposed by Huber (1981), the off-diagonal values are typically much smaller than the diagonal elements. In this chapter we have used equation 4.3 for the covariance matrix estimate.

6.2.2 Data

To compare the methods several datasets were used. First, several spike-in datasets where we have known "truth" given by the spike-in information were considered. We then used data from a dilution/mixture study where it could be reasonably expected that a large fraction of genes would be differential.

The HGU95A spike-in dataset from Affymetrix consists of 59 arrays upon which 14 probesets have been spiked-in at a wide range of known concentrations in a Latin square design. A common background RNA was hybridized to all 59 arrays and therefore aside from the spike-in probesets all other probesets should be non differential across arrays. Other authors, such as Cope et al. (2004) and Wolfinger and Chu (2002), have argued that several non-documented differential probesets exist in this data. For the purposes of this comparison we restricted ourselves to the 14 documented probesets. More details about this dataset are given in Appendix A.1.

We also used two spike-in datasets from GeneLogic. The first consisted of 34 HGU95A arrays upon which 11 control probesets were spiked-in at varying concentrations, again in a latin square design, against a common background of AML RNA. In most cases each concentration profile was repeated on 3 replicate arrays. The second was composed of 36 HGU95A arrays where the same 11 control probesets were spiked-in, but with a slightly different set of concentration profiles. A common human tonsil mRNA was hybridized to all 36 arrays. Each concentration profile was repeated on 3 arrays. More details about these datasets are provided in appendices A.3 and A.4.

The Dilution/Mixture datasets from GeneLogic were comprised of 75 HGU95Av2 arrays. There were 30 arrays from liver tissue, with 5 replicate arrays at each of 6 different concentrations (1.25, 2.5, 5, 7.5, 10, 20 μg of total cRNA) and a similar arrangement of 30 arrays from CNS tissue. In addition 15 arrays were hybridized using a mixture of cRNA from the two sources, at a total quantity of 10 μg , with 5 replicate arrays at each of 75:25, 50:50 and 25:75 ratios of liver:CNS. More details for this dataset can be found in Appendix A.5.

6.3 Results

6.3.1 Comparing PLM based test statistics with probeset summary based test statistics

In our first comparison, we restricted ourselves to 8 arrays from the Affymetrix HGU95A spike-in dataset. Specifically, we considered only the arrays from wafer number 1532 in experimental groups M, N, O, P, Q, R, S, T, where the first four and second four experimental groups all had the same concentration profiles. After using the standard RMA preprocessing steps (convolution background, quantile normalization), the model shown in equation (6.2) was fitted. Then, all pairwise comparisons between spike-in concentration groups were considered, using 1 vs 1 (16 comparisons), 2 vs 2 (36 comparisons), 3 vs 3 (16 comparisons) or all 4 arrays in each group. For each comparison, we computed each of the test statistics. It should be noted that when using only a single array in each group, only FC, $t_{\text{PLM.1}}$ and $t_{\text{PLM.2}}$ are possible. When choosing differential genes we thresholded on the absolute value of the test statistic. Methods were compared using an ROC curve, the true positive (a differential gene is identified as differential) rate was on one axis and the false positive (a non-differential gene is identified as differential) rate was on the other. The ideal would be to identify all the differential genes without a single false positive. We averaged over all pairwise comparisons to generate the ROC curve. The curve for 3 vs 3 comparisons is shown in Figure 6.1. The two model based test statistics, $t_{\text{PLM.1}}$ and $t_{\text{PLM.2}}$, had the highest curves followed by the empirical Bayes statistic t_{ebayes}, simple moderated statistic t_{mod} and raw fold change FC. Both t_{Std} and the t_{robust} performed poorly in comparison to the other methods. For all four comparisons, the model based methods gave the highest curves (i.e. correctly detected more differential genes at any level of false positves). In each case we also found that $t_{\text{PLM.1}}$ and $t_{\text{PLM.2}}$ almost coincided. This would be expected using the Huber weights. A more aggressive weighting scheme, would be expected to have larger differences.

Next, the study was expanded to all 24 arrays in experimental groups M, N, O, P, Q, R, S, and T. Again, the array effect model in equation (6.2) was fitted. The primary interest was again in all possible pairwise comparisons between the two concentration groups. Since it was time prohibitive to examine all possible pairwise comparisons, 100 randomly chosen comparisons with an equal number of arrays in each concentration group (i.e. from 1 vs 1 to 12 vs 12) were examined. As previously done, the methods were compared using ROC curves. However, in this case focus was

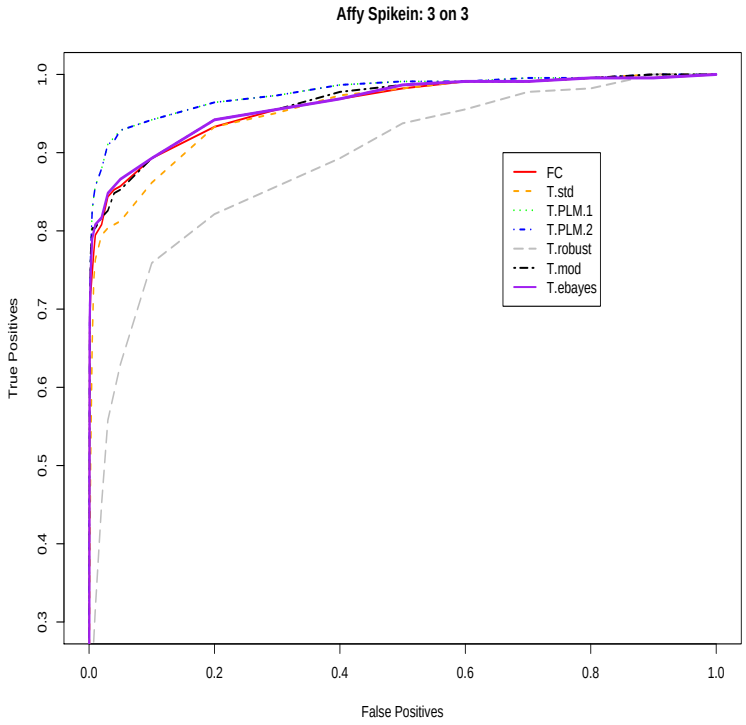

Figure 6.1: ROC curve based on all pairwise comparisons of 3 vs 3 arrays using 8 arrays from the Affymetrix HGU95A dataset. Higher curves are better. The PLM test statistics found the most differential genes with the fewest false positives.

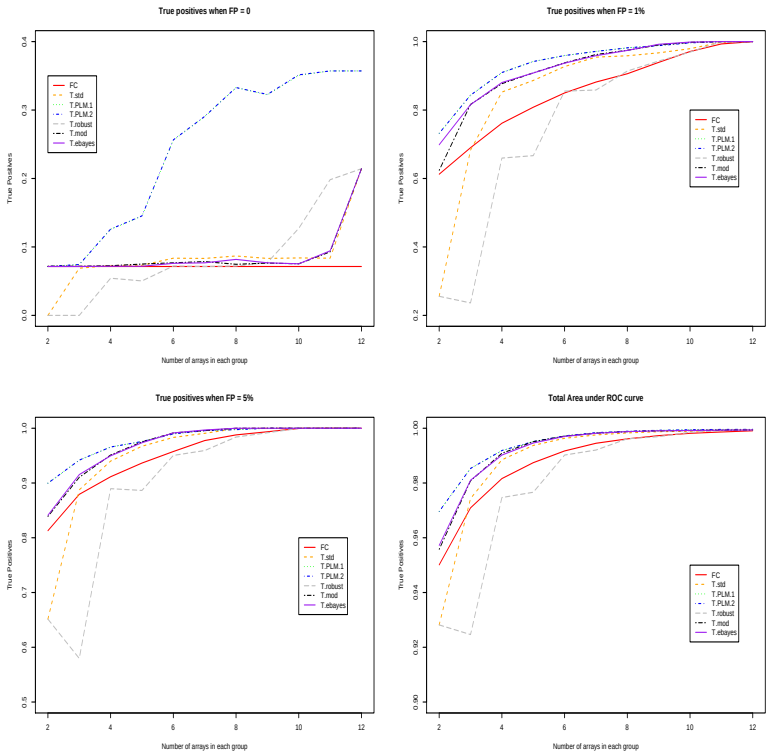

Figure 6.2: Comparing the performance of each test statistic using ROC curve quantities as the number of arrays increase. The PLM model test statistics identify more differential genes at each level of false positives. As the number of arrays increases, the t-statistics tend to outperform raw FC. The vertical axis changes scales between plots.

placed on three quantities from the ROC curves, the true positive rate (TP) when the false positive rate (FP) was 0, 0.01, and 0.05. The total area under the curve (AUC) up to 5% FP was also considered. Figure 6.2 shows these quantities against the number of arrays in the comparison for each of the different test statistics. The PLM based test statistics outperformed the other methods using all four criteria. As the number of arrays increased, the other methods tended to also outperform FC.

Next, we examined all 59 arrays in the Affymetrix HGU95A spike-in dataset and looked at all possible pairwise comparisons between concentration groups (there were 91 possible comparisons). This was done in two ways: fitting individual array effect models to each pairwise comparison and fitting a single array effect model to the entire dataset. The first case is perhaps closer to what would be done in practice when seeking to make a comparison between any two treatment groups, while the second case is what is done when producing expression summaries for a dataset involving many treatment groups. In each case, the entire dataset was preprocessed as a single group before

Method	Individual models			Single model		
	0% FP	5% FP	AUC	0% FP	5% FP	AUC
FC	0.451	0.985	0.975	0.444	0.982	0.971
Std	0.323	0.982	0.956	0.301	0.975	0.952
Robust	0.160	0.939	0.857	0.144	0.935	0.852
Mod	0.437	0.987	0.975	0.413	0.980	0.970
PLM.1	0.653	0.991	0.979	0.540	0.951	0.930
PLM.2	0.657	0.991	0.979	0.539	0.951	0.930
Ebayes	0.514	0.988	0.978	0.450	0.986	0.974

Table 6.1: Statistics for ROC curves for complete Affymetrix dataset. Figures are proportion of differential probesets identified when there is 0% or 5% false positives. AUC is area under ROC curve up to 5% false positives. Higher values are better.

fitting the models. Summary statistics for the ROC curves are shown in Table 6.1. Using individual models, the probe level model based test statistics were the best methods to use. 65.7% of the differential genes were identified before a single non-differential gene was selected. When the single model fitted to all 59 arrays was used fold-change and the two moderated t-statistics, t_{ebayes} and t_{mod} performed particularly well, while comparatively the PLM based test statistics performed poorly except at the 0% false positives. Another feature of this comparisons was that the individual models always outperformed the single model case by a small margin for all methods.

For each of the GeneLogic spike-in datasets, a single model was fitted to all the arrays. Using the Genelogic tonsil dataset, we averaged across all 66 possible pairwise comparisons to produce ROC curves. We found that raw fold-change seemed to perform the best, as depicted in Figure 6.3. The PLM t-statistics performed better than the other remaining methods. Using the GeneLogic AML dataset, one array was removed since it did not have a concentration profile repeated on any other array. Then to produce the ROC curve we averaged across the remaining 55 possible pairwise comparisons, each between groups of 3 arrays. In this case, the PLM test statistics were slightly better than raw fold-change, The remaining methods yielded fewer differential genes.

Using an overall array effect model, the PLM t-statistics were comparatively either very good (using both the GeneLogic datasets) or poor (using the Affymetrix data). Examining the residuals from the probe-level models allowed us to explain this disagreement. Figure 6.4 shows boxplots of the residuals grouped by concentration group for three spike-in probesets (36202_at, 407_at, 39058_at) and a randomly chosen non-differential probeset (41233_at) for the Affymetrix spike-in data. The three spike-ins were typical of the spike-in probesets for this dataset and the non spike-in was typical of the remaining probesets. Two observations were immediately apparent: the residuals are

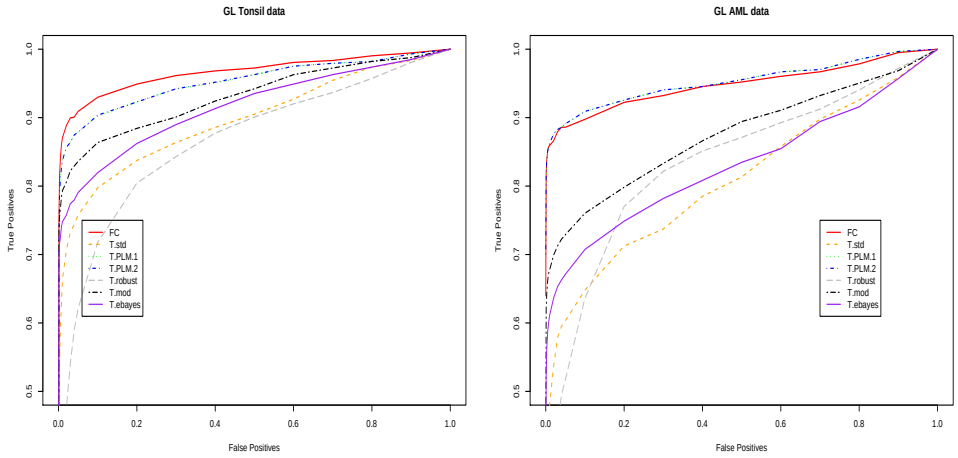

Figure 6.3: ROC curves for GeneLogic Tonsil and AML datasets.

much more variable for spike-in probesets and the spread of the residuals differed greatly between concentrations. It was not difficult to envision how the process of creating spike-in mixtures could add this variability. However, since the constant variance assumption was violated for these spike-in probesets the PLM test statistics performed poorly. Similar plots (not shown) for the GeneLogic datasets also displayed these problems, although to a much lesser degree, with the plots of residuals for the GeneLogic AML spike-ins being very similar to the non spike-ins for the same dataset.

The spike-in datasets, which allowed us to compare the performance of methods when there were only 11 or 14 probesets changing between conditions, were an artificial situation. Using the GeneLogic Dilution/Mixture data allowed us to compare the methods when more justifiable biological differences could reasonably be expected. Since we did not have "truth," as in the case of the spike-in datasets, we constructed it. Initially, we selected all 60 arrays forming the Dilution series and as a group computed RMA expression measures Irizarry et al. (2003a). The quantile normalization step of this process dealt with the differences between dilution groups and scanner differences that are a feature of this data. Two sample t-statistics were computed between the 30 liver and 30 CNS arrays. Since we have a large number of arrays we expected this to work reasonably well. The 400 probesets with the most extreme t-statistics were then called differential and give us our "truth." Using these differential probesets gave us the opportunity to compare the performance of the test statistics using the mixture data. More specifically, we combined the 5 arrays that were a 75:25 mixture and the 5 arrays that were a 25:75 mixture (both ratios are liver:CNS) and fitted the array effect model using all 10 arrays. We then compared the performance of the test statistics by examing

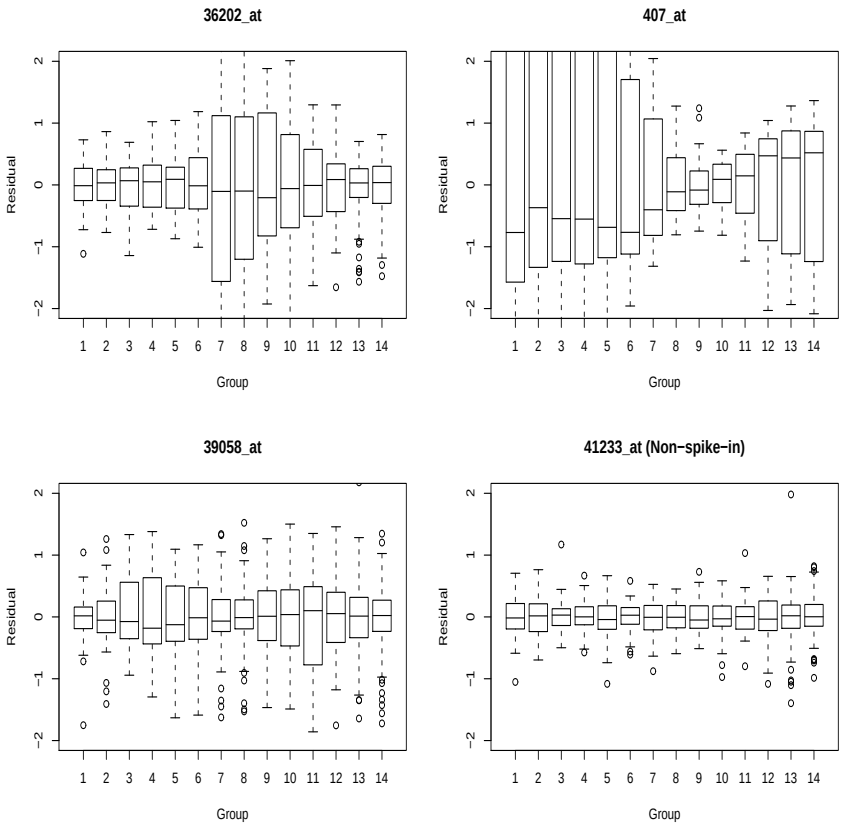

Figure 6.4: Boxplot of residuals from model by concentration group for three spike-in probesets and a typical non spike-in probeset for the Affymetrix HGU95A spike-in dataset.

Method	3 vs 3			4 vs 4			5 vs 5		
	0% FP	5% FP	AUC	0% FP	5% FP	AUC	0% FP	5% FP	AUC
FC	0.007	0.886	0.697	0.008	0.888	0.703	0.005	0.888	0.708
Std	0.004	0.793	0.530	0.008	0.872	0.626	0.018	0.902	0.675
Robust	0.002	0.485	0.271	0.005	0.747	0.490	0.010	0.743	0.488
Mod	0.007	0.908	0.697	0.002	0.932	0.735	0.000	0.948	0.760
PLM.1	0.056	0.943	0.751	0.057	0.947	0.756	0.056	0.950	0.760
PLM.2	0.057	0.943	0.752	0.057	0.948	0.758	0.058	0.950	0.761
Ebayes	0.001	0.918	0.744	0.000	0.933	0.761	0.000	0.943	0.776

Table 6.2: Summary statistics for ROC curves based upon GeneLogic Mixture dataset.

all possible 3 vs 3, 4 vs 4, and 5 vs 5 comparisons of arrays. In each case, averaging across all the pairwise comparisons allowed us to form the ROC curves. Table 6.2 summarizes the results for this comparison. The PLM test statistics identified the most differential genes at the 0% and 5% false positive levels. However, the empirical bayes statistic t_{ebayes} had a slightly higher total area under the curve then any other method in the 4 vs 4 and 5 vs 5 comparison. Closer examination of the ROC curve showed that t_{ebayes} exceeded other methods between 0.25% and 2.5% false positives. At all other values the PLM test statistics did better.

6.3.2 Moderating the PLM test statistics

Another issue that was considered was whether moderation was helpful for the PLM test statistics. To achieve a moderation, we applied a shrinkage estimation procedure to the individual variance/covariance matrices. Specifically, we shrink by updating the covariance matrix $\Sigma^{(k)}$ for any probeset k by combining it with an averaged covariance matrix Λ. The adjustment was given by

$$\Sigma^{(k)}_{\text{mod}} = p_{\text{prior}}\Lambda + \left(1 - p_{\text{prior}}\right)\Sigma^{(k)} \tag{6.10}$$

where $\Lambda = \frac{\sum_{k=1}^{K}\Sigma^{(k)}}{K}$ and p_{prior} was a mixing proportion. If $p_{\text{prior}} = 0$ then the test statistic was exactly the unmoderated version. If $p_{\text{prior}} = 1$ the test statistic was equivalent to a scaled version of fold-change.

Careful examination of the performance of the test statistic allowed us to pick a reasonably universal value for p_{prior}. Using each dataset, we examined the total area under the ROC curve up to 5% false positives as the mixing proportion was varied. The goal was to select the value for p_{prior} which maximised this area. Figure 6.5 shows this for several different datasets. Specifically, for the 4 vs 4 comparison from the Affymetrix U95A dataset, the GeneLogic AML, mixture and tonsil

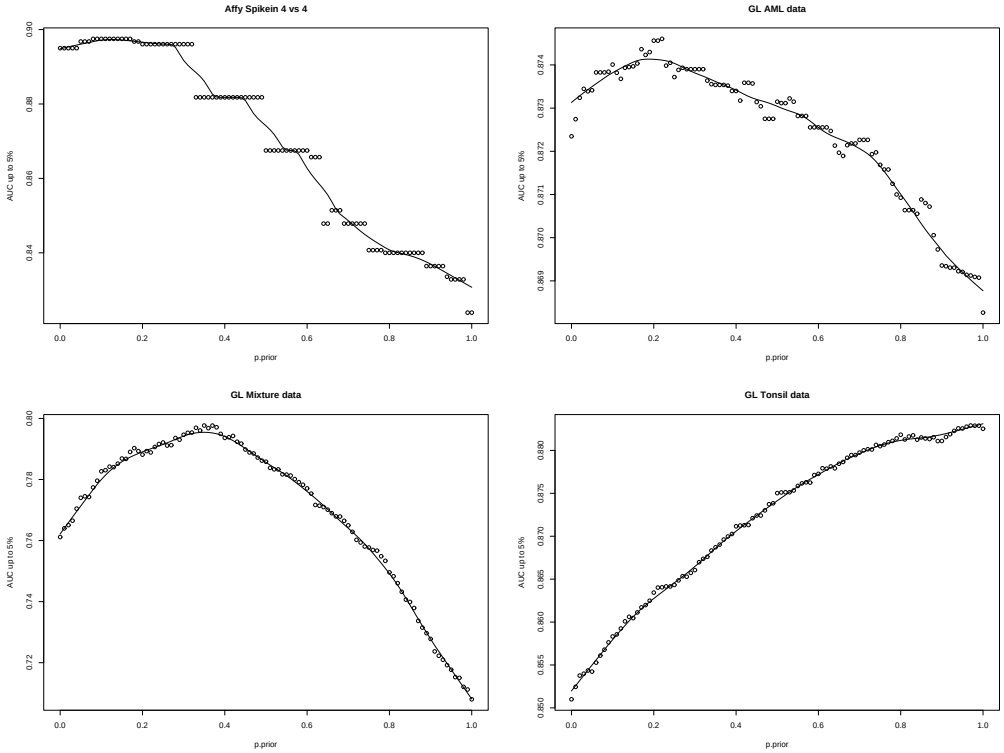

Figure 6.5: Choosing p_{prior} based on total AUC up to 5% false positives. The scale on the vertical axis changes between plots.

datasets. For the cases where the PLM test statistic performed the best(the first three cases), a p_{prior} betwen 0.2 and 0.4 seemed to maximize the AUC. When fold-change performed better than the PLM test statistic, as in the GeneLogic tonsil case, the AUC seemed to increase as p_{prior} was increased. This implied that a little moderation did not greatly reduce the AUC and often improved it. Thus, based on these plots, a reasonable choice would be $p_{\text{prior}} = 0.3$. This is because at 0.3 the moderation in the cases examined did not reduce the AUC compared with no moderation. But is not any case the optimal choice. As the number of arrays in each treatment group increases, the need for any moderation would be expected to lessen. Future work will focus on developing an adaptive procedure for choosing p_{prior}.

6.3.3 Fitting the treatment effect model

Rather than fitting the array effect model, we then considered the possibility of fitting the treatment group effect model. Using all 10 arrays of the mixture data, we fitted a probe-level model with two group effects. Restricting ourselves to the $t_{\text{PLM.2}}$ test statistic, we computed values of 0.05, 0.960 and 0.774 for the true positive rate at 0%, 5% false positives and proportion of the total area under the curve up to 5% for the 5 vs 5 comparison. This compares favorably with the previous methods. This removed the need to fit individual chip effects and since in practice fitting a model with group effects is less time consuming, it should be the favored PLM method. An important thing to note with the treatment effect model is that differences between $t_{\text{PLM.1}}$ and $t_{\text{PLM.2}}$ become greater because the off diagonal elements of the covariance matrix tend to increase to a larger proportion of the diagonal elements. In such cases, the method using the complete covariance matrix should be preferred. Moderation of the test statistic performed in an equivalent way to the array effect model.

6.4 Discussion

This analysis has shown that information from the low-level analysis can be successfully used to develop a test statistic for differential expression. This is opposed to the traditional methods which were based on probeset summaries. Except for one notable exception, the PLM based test statistics outperformed the other methods without requiring any moderation. However, our analysis also showed that some moderation was still useful for improving the properties of our test statistic.

In the one case where the PLM model performed significantly worse than the alternative methods, when using the single model for the Affymetrix HGU95A dataset, it was found that the model did not fit the data well. One interesting feature of this data is that sometimes one probe in the probeset did not significantly change across concentrations. This led to large residuals from that particular probe at some concentrations. The current model fitting procedure deals effectively with individual outlier probe intensities on a particular chip, but it does not deal with probes that do not perform over a significant number of arrays. In addition, it does not deal with probesets which have aberrant behaviour on one particular array. This is an important issue in the context of the treatment effect model. Future model fitting procedures should deal with these cases more effectively.

Chapter 7

A Study of the Effects of Pooling on Gene Expression Estimates

This chapter is a case study of the effects of pooling on gene expression estimates. Section 7.1 provides a brief introduction to the study and discusses why pooling could be used, Section 7.2 explains the experimental procedure and Section 7.3 presents an analysis of the data. Finally, Section 7.4 summarizes the conclusions of the investigation.

7.1 Introduction

High density oligonucleotide array data is being used in the study of many biomedical topics. In a typical microarray experiment, mRNA from source material is hybridized to a microarray and the resulting data is used to measure gene expression. When using multiple sources of mRNA, there are two ways that the hybridization can be prepared: hybridize the mRNA from each individual biological sample to an individual array, or combine the mRNA from multiple biological sources (this process is called pooling) and hybridize it to one or more arrays. Sometimes pooling is done because sufficient mRNA for one hybridization cannot be recovered from a single subject.

The idea of pooling sources of biological material together has long been known. Specifically, Dorfman (1943) considered the effectiveness of pooling blood samples in reducing the number of

tests required when trying to detect syphillis. Gastwirth and Hammick (1989) considered pooling of blood to preserve annonimity when testing for HIV.

More recently, pooling in the context of microarray experiments has begun to draw great attention. It has been previously been assumed that pooling reduces biological variation between arrays. Lockhart and Barlow (2001) state that "if genetically identical, inbred mice are not used, then it is necessary to do more experiments or to pool mice to effectively average out differences due to genetic inhomogeneity...the same considerations apply when using any other animal or human tissue." In addition Churchill (2002) notes that "pooling samples...increases precision by reducing the variability of the experimental material itself. When variability between individual samples is large and the units are not too costly, it may be worthwhile to pool samples." It is for this reason that the pooling process has been recommended. A number of microarray studies have used pooling including Crnogorac-Jurcevic et al. (2002), Zhu et al. (2003), Chabas et al. (2001), Waring et al. (2001), Saban et al. (2001), Enard et al. (2002) and Jiang et al. (2002).

The bias effects of pooling have been less well considered and have not been quantified. According to Hamadeh et al. (2002), "pooling may cause misinterpretation of data if one animal shows a remarkably distinct response, or lack of response." It is very important to know whether the expression level of a gene in one individual sample can drive the expression level of that gene in a pool.

Through the use of a dataset consisting of 36 MGU74av2 Affymetrix GeneChips® we examine the effects that pooling has on the detection of differential expression and the variance and bias of measures of differential expression. This is important because very often the ultimate goal of a microarray experiment is to determine a set of differentially expressed genes.

7.2 Materials and Methods

The data and details of the experimental procedures were provided by Eun Soo Han from the Department of Biological Science, University of Tulsa in Tulsa, Oklahoma. All arrays were hybridized by Yimin Wu, at the University of Texas Health Science Center in San Antonio, Texas.

7.2.1 Animal Subjects

Male C57BL/6 mice were bred in the Animal Core of the Program Project. Parent mice were purchased from Jackson Laboratories (Bar Harbor, ME). All mice were fed ad libitum (AL) Harlan Teklad LM-485 mouse/rat sterilizable diet 7912 (Madison, WI). They were kept on a cycle of 12 hour darkness 12 hour light (lights on at 0600 h). Sentinel mice were tested monthly for Endoparasites, Ectoparasites, Mouse Hepatitis Virus, Sendai Virus, Mycolpasma, Pneumonia Virus of Mice, Theiler's Mouse Encephalomyelitis Virus, and Minute Virus of Mice by a Veterinary Pathologist in Laboratory Animal Resources at the University of Texas Health Science Center at San Antonio. Every six months, the presence of murine virus antibodies: CAR Bacillus, Ectromelia Virus, Epizootic Diarrhea of Infant Mice, Lymphocytic Choriomeningitis Virus, Minute Virus of Mice, Mouse Adenovirus (M.Ad-FL), Mouse Adenovirus (M.Ad-K87), Mouse Hepatitis Virus, Murine Cytomegalovirus, Mycoplasma pulmonis, Parvovirus, Pneumonia Virus of Mice, Polyoma, Reovirus, Sendai Virus, and Theiler's Mouse Encephalomyelitis Virus was monitored with serum samples from sentinel animals by BioReliance Co. (Rockville, MD). All tests were reported as negative. All procedures involving use of the mice were approved by the Institutional Animal Care and Use Committee of the University of Texas Health Science Center and the Subcommittee for Animal Studies at the Audie L. Murphy Memorial Veterans Hospital.

7.2.2 Tissue Collection and RNA Preparation

Livers from nine 5 month-old and nine 13 month-old mice were collected between 1000 hours and 1200 hours, and total RNA was extracted from each liver using the previously described procedure (Sambrook et al., 1989).

7.2.3 Screening of mRNA by Affymetrix GeneChip Arrays

The mRNA expression of individual and pooled liver RNA from nine 5 month-old and nine 13 month-old male C57BL/6 mice were measured using Affymetrix Murine Genome U74 A Version 2 Genechips. For the pool samples, liver RNAs from 3 animals were pooled to generate three pool samples of 5 or 13 month old animals and the array hybridization was repeated three times with targets synthesized separately from the same source of pooled RNA. For individual samples, array

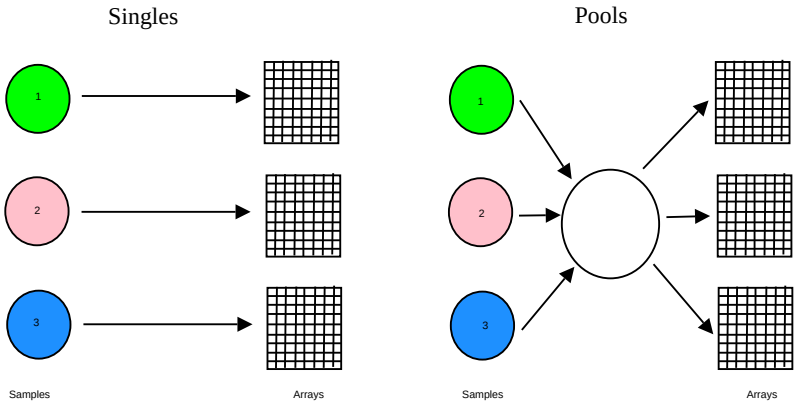

Figure 7.1: Three sources of mRNA were either individually hybridized to arrays (singles) or mixed together and hybridized to a set of arrays (pools).

hybridization was performed once for each sample of individual RNA.

For sample preparation and hybridization the vendor's protocols were followed. First, total RNA was subjected to a cleanup process using the RNeasy Total RNA Isolation Kit (Quagen, Valencia, CA). Ten μg of the cleaned total liver RNA were converted to double-stranded cDNA using SuperScript Choice System (GIBCO/BRL, Rockville, MD) and T7-(dT)24 primer (GENSET Corp, La Jolla, CA). Biotin labeled cRNA was synthesized from the cDNA with BioArray High Efficiency RNA Transcript Labeling Kit (Affymetrix, Santa Clara, CA). After being cleaned up the purified cRNA was fragmented to sizes ranging from 35 to 200 bases by incubating at 94°C for 35 min. Fifteen μg of the fragmented cRNA were hybridized to a GeneChip at 45° C with 45 rpm for 16 hours. After hybridization, the GeneChips were washed and stained with streptavidine-phycoerythrin, and then the signals were amplified with a biotinylated antibody, goat Ig G and another staining with streptavidine-phycoerythrin using the Fluidics Station (Affymetrix, Santa Clara, CA). The GeneChips were scanned with the Hewlett Packard GeneArray Scanner (Affymetrix, Santa Clara, CA).

In this experiment, we referred to the young mice by the labels 1, 2, 3, 4, 5, 6, 7, 8 and 9, and to the middle aged mice using $1'$, $2'$, $3'$, $4'$, $5'$, $6'$, $7'$, $8'$ and $9'$. The single arrays were referred to by the source of mRNA and the pooled chips were refered to by using the notation $ijk(r)$ where i,j,k designated the source of mRNA and r referred to a replicate number. The pools were all created from consecutively numbered sets of three mice. Thus, valid pools were 123, 456, 789, $1'2'3'$, $4'5'6'$ and $7'8'9'$. Figure 7.1 demonstrates how the groups of singles and pools were hybridized to arrays

in this experiment for the young mice 1, 2, and 3.

7.2.4 Data Preprocessing

As described in the earlier chapters, a three-stage process for computing expression estimates was used: background correction, normalization and probeset expression summarization. Background/signal adjustment was done for each array, using the convolution method discussed in Chapter 2. The data from all 36 arrays was normalized together using quantile normalization Bolstad et al. (2003). Expression summarization was computed using a robust linear model, which was discussed in Chapter 4. The Huber influence function Huber (1981) gave the weights used for the iteratively re-weighted least squares procedure.

7.3 Results

7.3.1 Data Quality

It is important to consider the quality of the data because poor quality data could potentially have quite serious consequences on the conclusions we make in this study. In this section, we consider two methods of judging quality based upon the modeling procedure.

The robust linear model procedure produces a weight for each (PM) probe at the conclusion of the model fitting procedure. These may be used to identify chip defects and other procedural artifacts (Collin et al., 2003). Figure 7.2 plots the robust model weights as pseudo chip images for a representative selection of chips. The lighter colors represent high weight probe intensities, while the darker colors are probe intensities that have been down weighted by the model fitting procedure. There are some minor defects visible on some of the chips, but most chips do not show evidence of serious problems. However, the pooled chip, $7'8'9'(1)$, has a darker image and is down weighted much more than the other chips, which could cause a potentially confounding increase in variability.

Another quality display is to boxplot the computed standard errors from the model and look for discordant arrays (Collin et al., 2003). For each probeset, the standard errors of the array effects are normalized to have the median equal to 1 across arrays. These are referred to as the Normalized

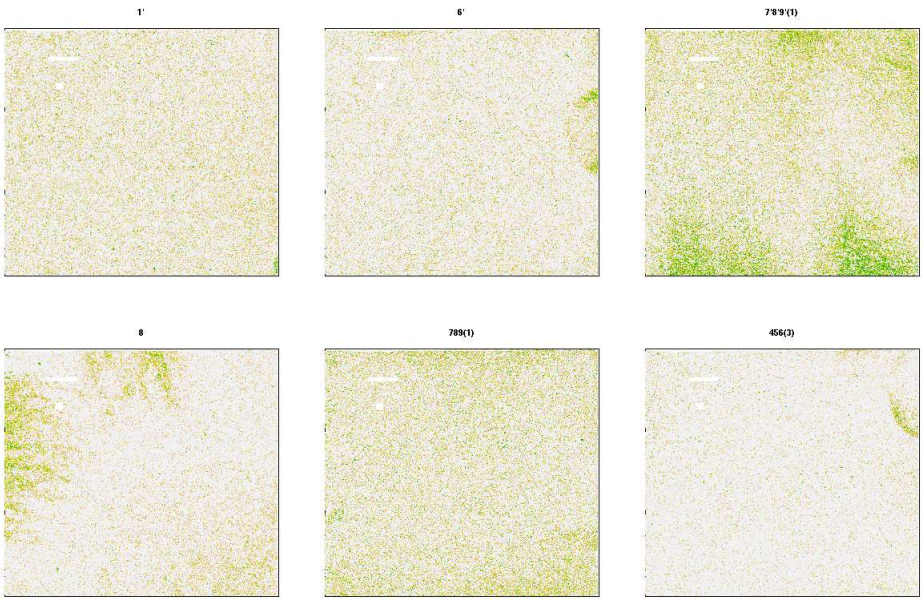

Figure 7.2: Pseudo-chip images of robust linear model weights for selected chips. Darker areas indicate areas of lower weight. Most of the arrays (not shown) are similar to $1'$, $6'$ and $456(3)$ with no or only small defects. The image plots for $7'8'9'(1)$ and $789(1)$ have a lot of down weighting indicating the possibility of poor data.

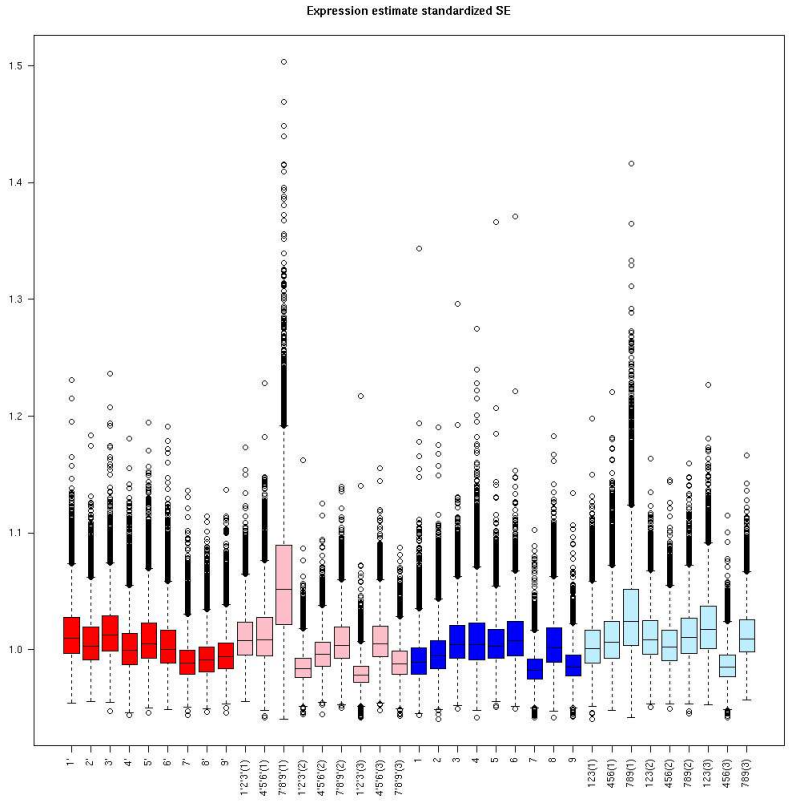

Figure 7.3: Boxplots by chip of standard errors of expression values, standardized to median 1. Two pool chips $7'8'9'(1)$ and $789(1)$ stand out as having larger standard errors relative to other chips.

Unscaled Standard Errors (NUSE). The boxplots of the normalized standard errors of the gene expression estimates for each array are shown in Figure 7.3. This plot allows identification of arrays where the standard errors for the gene expression estimates are generally larger relative to the other arrays. It is of importance to note that two pooled arrays, $7'8'9'(1)$ and $789(1)$, performed poorly using this diagnostic tool.

7.3.2 Variance

To compare the variability of expression values within an age group between single and pooled arrays, we plotted the log ratio of the variance across three singles to the variance across the three replicates of pooled chips using the same mRNA sources against the average expression value. Plots for all 6 groupings are shown in Figure 7.4. Surprisingly, the variability of the gene expression

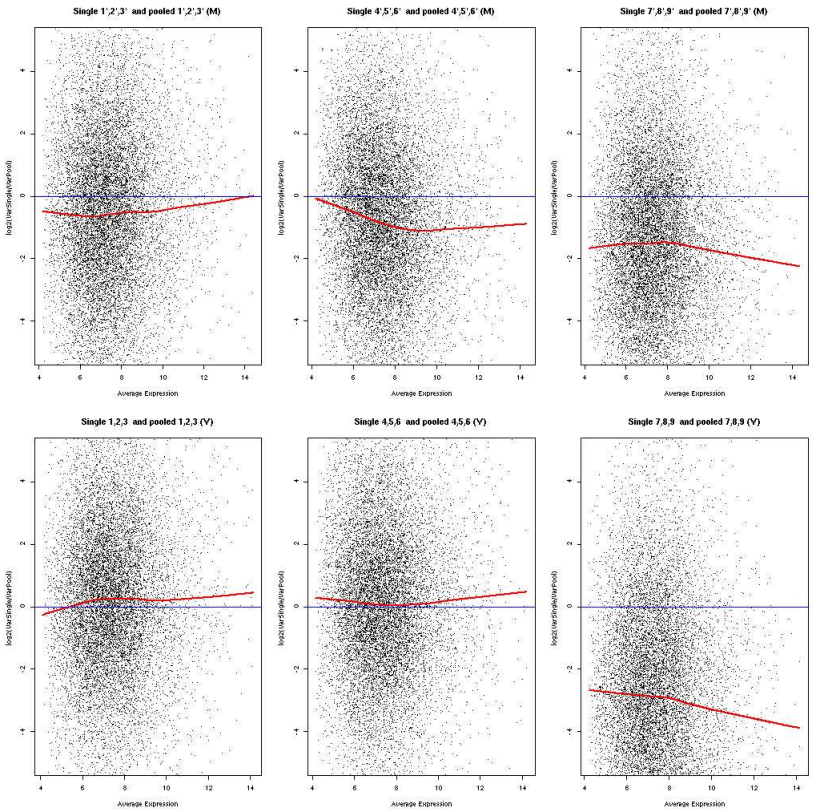

Figure 7.4: Comparison of the variability of singles to the variability (across replicates) or the corresponding pools. The figures are plots of the $\log_2$ of the variance ratio of singles to pools against the average expression value. The curve is a lowess smoother fit. Above the *x*-axis the variance of the pools is greater than the variance of the singles.

measures was higher for the pooled data than the corresponding gene expression measures of the singles in most of the cases. The particularly large differences between the single and pooled arrays for both young mice 7, 8, 9, and middle aged mice $7'$, $8'$, $9'$, were most likely driven by poorer quality data, as can be seen from the earlier discussion on data quality.

Another variance comparison that can be made is to compare the variability across all the single arrays within an age group against the variability of the pooled array from the same mRNA. For each probeset, we compared the variance of the expression value across all singles against the within group component of variation from an ANOVA fitted to the pooled arrays. Figure 7.5 shows the log ratio of the single variance to within pooled variance, with the variability of the pools and singles being similar (or slightly lower in the case of the young-aged chips). After removing the two poorer quality chips, $7'8'9'(1)$ and $789(1)$ from the analysis, we found that a little over 50% of the probesets

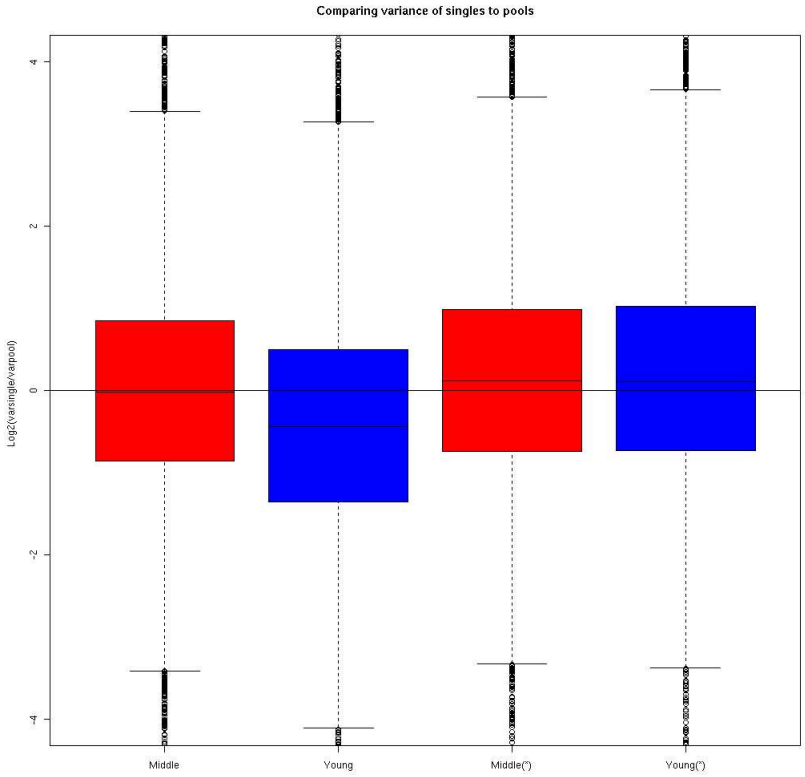

Figure 7.5: Comparing the variance of all singles to the within pool variance of the pool arrays by looking at the log ratio of the single variance to the pool variance, a ratio above zero indicates that the variance of the pools is less than the variance of the singles. After removing two poor quality arrays from the analysis, we find that the variance of the singles is higher than the variance of the pooled arrays.

had higher variability in the singles. Either way, the differences were not great.

It is also possible to look at the variability of differential expression estimates. In particular, we looked at computing the differential expression between groups of three young and three middle age mice. The three arrays could be either three arrays where RNA from an individual sample was hybridized or three replicate arrays hybridized with pooled RNA. Thus, measures of differential

expression would be

$$M^{\text{single}}_{ijk,lmn} = \frac{\sum_{c=i,j,k} \hat{\beta}_c}{3} - \frac{\sum_{c=l,m,n} \hat{\beta}_c}{3}$$

$$M^{\text{pool}}_{ijk,lmn} = \frac{\sum_{r=1}^{3} \hat{\beta}_{ijk(r)}}{3} - \frac{\sum_{r=1}^{3} \hat{\beta}_{lmn(r)}}{3}$$

which are the $\log_2$ fold-change values.

For example, to make a comparison between middle aged mice $4'$, $5'$, $6'$ and young mice 1, 2, 3, we could either use the difference between the averages of the single arrays $M^{\text{single}}_{4'5'6',123}$ or the difference between the averages of the pooled arrays $M^{\text{pool}}_{4'5'6',123}$. Since there is no direct correspondence between the mice in the young age group and those in middle age group, 1, 2, 3 do not correspond to $1'$, $2'$, $3'$. Comparisons can be made across any of the groups of three. Figure 7.6 shows the variability of the differential expression statistic for all single-to-single or pool-to-pool comparisons. The variability of the comparisons between pooled arrays was less than the variability of the corresponding comparisons between groups of single chips.

7.3.3 Bias

It is useful to examine whether the expression values of genes in a pool are driven by just one of the individuals. For each single array we compared the ratio of each gene's expression value on that chip to the average expression of all the other singles within the same age group. This allowed the identification of probe sets where the gene expression for that sample alone differed from the majority. For the purposes of this investigation such probesets were called "outliers".

To choose these probesets, a cutoff C_{single} was used. When the absolute value of the estimated fold change was greater than C_{single}, we called the probeset differentially expressed for that array within its age group. Table 7.1 shows the number of probesets chosen in this way for $C_{\text{single}} = 1$ and $C_{\text{single}} = 0.75$. The young-aged singles had more outlier genes than the middle age singles.

To perform a similar comparison for the pooled arrays, we averaged across the replicates of a particular pool and compared this to the average across both sets of three replicates of the other two pools. Differential probesets were selected and a different cutoff was used, which shall be referred to as C_{pool}.

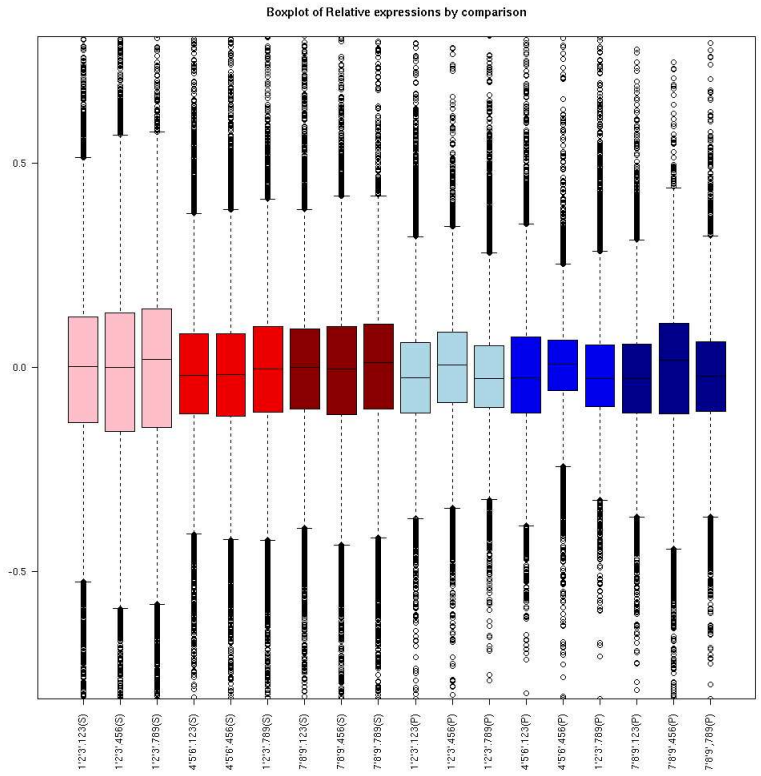

Figure 7.6: Boxplots of relative expression values for each middle age to young comparison. The expression values are less variable for the pool to pool comparisons than in the corresponding comparison between singles.

Array	Number of Probesets selected	
	Differential at $C_{\text{single}} = 1$	**Differential at** $C_{\text{single}} = 0.75$
$1'$	14	30
$2'$	11	20
$3'$	41	106
$4'$	24	59
$5'$	11	27
$6'$	9	25
$7'$	12	17
$8'$	6	16
$9'$	29	63
1	61	167
2	46	127
3	95	279
4	49	123
5	37	107
6	26	77
7	36	106
8	37	95
9	41	128

Table 7.1: Number of probesets selected when comparing expression values on one array to average expression on all other arrays from the age group. So array 3 has 41 probe sets where the relative expression of that probe set compared to the average expression in the 8 other middle age arrays has estimated fold change greater than 1. We will refer to these probe sets showing differential expression on just one array as “outliers”.

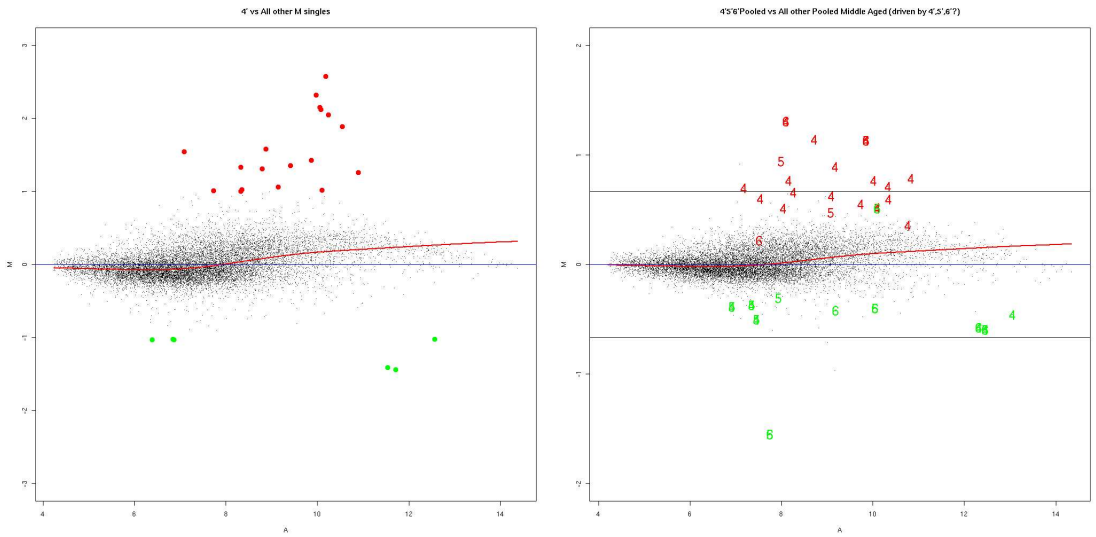

Figure 7.7: *MA*-plots: (a) using a cut-off $C_{\text{single}} = 1$ to detect outlier probesets from one individual array $4'$ vs average across all the other middle age arrays. (b) comparing the average across the replicates of a pool, $4'5'6'$, against the averages over all the replicates of pooled middle aged arrays. The numbers on the plot indicate the single array where that probeset was called an "outlier". The horizontal lines indicate the cut-off $C_{\text{pool}} = \frac{2}{3}$.

By making some assumptions, we could relate C_{single} and C_{pool}. We assumed that the observed expression-level of a probeset on any array was given by the sum of the true expression value and a random error, that is $\beta_{\text{obs},i} = \beta + \varepsilon_i'$ where ε_i' had expectation 0 and variance σ^2 . Independence between chips was also assumed.

Assuming equal variance in both singles and pools ($\sigma^2_{\text{single}} = \sigma^2_{\text{pool}}$), we found that $C_{\text{pool}} = \frac{2}{3} C_{\text{single}}$. If instead we had assumed that the mRNA averages out across a pool, an assumption in Kendziorski et al. (2003), that is $\sigma^2_{\text{pool}} = \frac{\sigma^2_{\text{single}}}{3}$, then $C_{\text{pool}} = \sqrt{\frac{4}{27}} C_{\text{single}}$. This assumption was not borne out by our data.

Figure 7.7 demonstrates the selection method for one particular comparison. First, we chose probesets that were discordant on only one single array, $4'$, as compared to the average across all singles. Then, we saw how many of these genes were also discordant when comparing the average across replicates of a pool ($4'5'6'$) to average expression across all other pooled chips in the age group. The numbers on the plot identify the individual upon which the probeset was identified as an outlier.

Table 7.2 summarizes the results for all of the possible pool to all other pools comparisons. A large

Array type	$C_{\text{pool}} = \frac{2}{3}$	$C_{\text{pool}} = \frac{1}{2}$	$C_{\text{pool}} = \sqrt{\frac{4}{27}}$	$C_{\text{pool}} = \frac{3}{4}\sqrt{\frac{4}{27}}$
$1'2'3'$	13(0.77)	27(0.78)	47(0.51)	107(0.49)
$4'5'6'$	14(0.78)	34(0.79)	83(0.33)	290(0.2)
$7'8'9'$	18(0.77)	34(0.56)	104(0.18)	286(0.2)
123	4(0.5)	12(0.5)	38(0.31)	154(0.23)
456	13(0.54)	46(0.32)	141(0.10)	495(0.08)
789	27(0.41)	62(0.37)	138(0.13)	334(0.14)

Table 7.2: Number of probesets selected when comparing average over replicates of a pool to average of all other pools. The figures in parentheses are proportions of these probe sets that have been ruled as an "outlier" in the single chip comparison in Table 7.1. The first two columns are the cut-offs given and assuming equal variances in both pooled and single arrays. The second column two columns correspond to the assumption that the mRNA averages in the pool. This assumption did not seem justified by our data.

proportion of the "outlier" probesets in the pools were also "outlier" probes in one of the singles that was a part of that pool. A greater proportion of the outliers for the middle-aged pooled arrays were accounted for by probesets that were outliers in our single array comparison.

7.3.4 Detecting Differential Expression

Selecting genes that were differentially expressed between young and middle aged mice, required computing $M^{\text{single}}_{ijk,lmn}$ and $M^{\text{pool}}_{ijk,lmn}$ for each of the possible combinations. We used a cutoff $C_{\text{diff,single}} = 1$. Keeping our assumption of equal variance it was simple to show that $C_{\text{diff,pool}} = \sqrt{\frac{16}{27}} C_{\text{diff,single}}$. Results for differential gene expression are shown in Table 7.3. The simple cutoff selected many more genes for comparisons based upon single chips than the same comparisons between pooled arrays using the same mRNA, as would have been expected from the variance comparison. Of particular importance, was the proportion of probesets chosen as differential that were accounted for by "outlier" genes from just one chip. About 80% of the probesets called differential from the pooled arrays were accounted for by a probeset that was shown to be an "outlier" in just one probeset.

7.3.5 Temporal Effects in Experimental Procedure

In the process of analyzing this dataset, it became apparent that a poor experimental design was used. In particular, the arrays were hybridized over a period of months, sometimes with replicate

Singles (a)	123	456	789
1’2’3’	157 (0.47)	174(0.21)	127(0.32)
4’5’6’	72 (0.51)	94 (0.57)	117 (0.35)
7’8’9’	99 (0.46)	91 (0.32)	84 (0.63)

Pools (b)	123	456	789
1’2’3’	20 (0.80)	24 (0.75)	20 (0.90)
4’5’6’	21 (0.81)	21 (0.81)	21 (0.90)
7’8’9’	21 (0.67)	41 (0.51)	30 (0.80)

Table 7.3: Number of differential probesets chosen using a fixed cutoff for estimated fold change for (a) comparisons between groups of singles and (b) pools. The figures in parentheses are the proportions of the differential probesets that were ruled “outliers” on one of the single arrays in the comparison.

pool arrays being hybridized many months apart. This suggested that perhaps chips hybridized at the same time may be more similar than those many months apart.

We made use of hierarchal clustering to examine whether chips were more similar because of biological source or because of hybridization order. Specifically, we used the average linkage method with Manhattan distances. A dendrogram was used to examine the clustering results. The leaf nodes for more closely related chips were closer on the tree.

Figure 7.8 shows the dendrogram for the middle-aged mice arrays. It is important to note that the middle-aged mice arrays were hybridized over a period of about three months. If the variability due to hybridization order was unimportant, then we would have expected biologically similar arrays to be grouped together. However, this was not observed. In almost every case, arrays that were hybridized on the same date were more closely grouped together than chips from the same biological source. In particular, the third replicates of each of the three pooled arrays $1'2'3'(3)$, $4'5'6'(3)$ and $7'8'9'(3)$ were all hybridized on the same date and all group togethered, instead of with other arrays from pools of the same material. The three groupings of single arrays were also hybridized on the same dates, whereas replicate pool chips were done on separate dates. For example $1'$, $2'$ and $3'$ were all hybridized on March 28, but the three replicates of $1'2'3'$ were hybridized on March 28, April 4 and May 22.

The young mice arrays were hybridized over a period of 6 months. A dendogram for clustering of the young mice arrays is shown in Figure 7.9. As with the middle-aged mice, arrays were grouped together by hybridization date. While it is less clear than with the middle-aged mice, it was still somewhat apparent that singles were hybridized on closer dates than the pool arrays using the same

Figure 7.8: Dendrogram for hierarchal clustering of middle-aged mice chips. The labels are for single or pool source and date of hybridization.

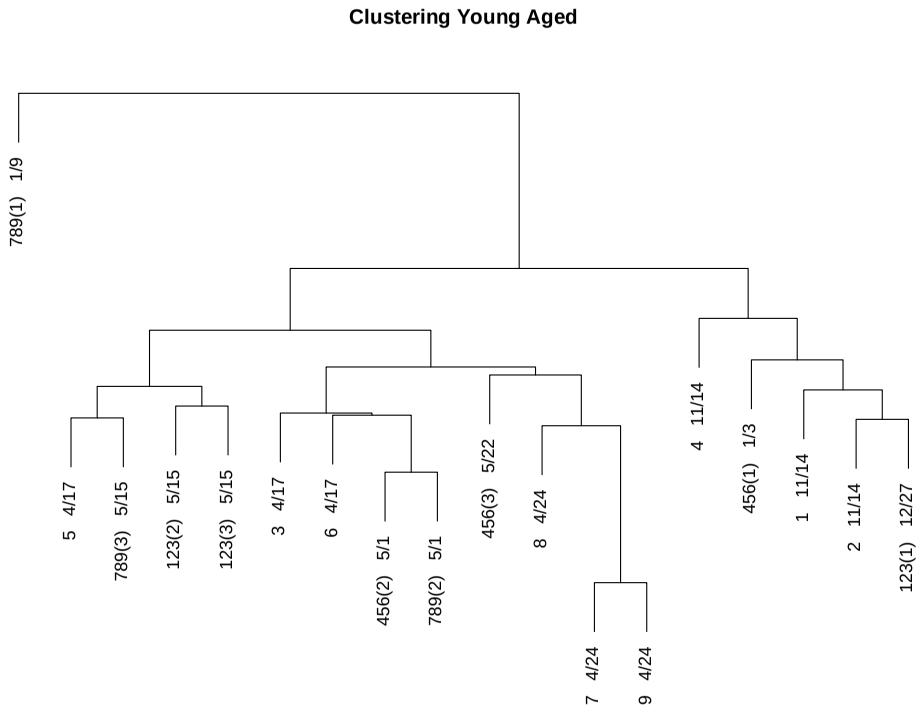

Figure 7.9: Dendrogram for hierarchal clustering of Young Aged mice chips. The labels are for single or pool source and date of hybridization.

RNA sources and thus also were more similar in expression values.

7.4 Discussion

The statistical effects of pooling in microarray experiments have been considered before in Kendziorski et al. (2003). One drawback to this study was that it drew conclusions based on RT-PCR data for only 6 genes from 5 mice. Peng et al. (2003) also considered the statistical implications of pooling. However, their study was restricted to simulated data and also some "virtual" pooled data. In contrast to these studies, our dataset was many scales of magnitude larger and contained high-density oligonucleotide microarray data where actual pooling has occured.

No clear evidence was found that the expression values for arrays using pooled mRNA were less variable than when using arrays hybridized with mRNA from only an individual. A further examination of the data suggested that because of poor experimental design, this observation was due to temporal variation between pooled arrays that was smaller or non-existent for the single mRNA source arrays. However, what should be apparent from our study is that if there were benefits in variance reduction that may have been achieved by pooling, they were not large enough to overcome poor experimental design.

Although, Kendziorski et al. (2003) considered possible benefits in variance reduction, they did not address the potential for bias to be introduced by pooling. This chapter, on the other hand, highlighted the dangers of pooling and its effect on bias. Using pooled mRNA leaves open the risk that a discordant gene from just one individual could drive the expression value of the gene in the pool. This was found to be the case, as both Figure 7.7 and Table 7.2 demonstrated.

In addition, this study examined the effects of pooling on differential expression, a topic not addressed in Kendziorski et al. (2003). We saw that the same outlier genes could also have significant effects when determining differential expression in a pool. In particular, we found that a much larger proportion of genes determined to be differential using the pooled arrays were also genes where the expression value from just a single individual was discordant.

Appendix A

Datasets

A.1 Affymetrix HGU95A Spike-in dataset

This was provided by Affymetrix for the purposes of developing and comparing expression algorithms. It was used as part of the process of developing and validating the Affymetrix Microarray Suite (MAS) 5.0 expression algorithm.

This dataset had a Latin Square design consisting of 14 spiked-in gene groups in 14 experimental groups. The concentrations used were 0, 0.25, 0.5, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, and 1024 pM. Table A.1 shows how the spike-in concentrations were applied to each of the experimental groups. The dataset consists of a total of 59 arrays of which there are 3 replicate arrays for each group except group C for which there was only 2 replicates. The known spike-in concentrations give us a "truth" by which to judge methods of computing expression summaries.

Affymetrix states that two of the probesets, 407_at and 36889_at, have poorly behaving probe pairs and should be excluded from the analysis. However, we shall make no such exclusions.

A.2 Affymetrix HGU133A Spike-in Dataset

This was a second spike-in dataset provided by Affymetrix. It consisted of 42 HG_U133A arrays. The 42 spike-in transcripts were organized into a Latin square design. Table A.2 shows the design

Grp	37777_at	684_at	1597_at	38734_at	39058_at	36311_at	36889_at
A	0	0.25	0.5	1	2	4	8
B	0.25	0.5	1	2	4	8	16
C	0.5	1	2	4	8	16	32
D	1	2	4	8	16	32	64
E	2	4	8	16	32	64	128
F	4	8	16	32	64	128	256
G	8	16	32	64	128	256	512
H	16	32	64	128	256	512	1024
I	32	64	128	256	512	1024	0
J	64	128	256	512	1024	0	0.25
K	128	256	512	1024	0	0.25	0.5
L	256	512	1024	0	0.25	0.5	1
M	512	1024	0	0.25	0.5	1	2
N	512	1024	0	0.25	0.5	1	2
O	512	1024	0	0.25	0.5	1	2
P	512	1024	0	0.25	0.5	1	2
Q	1024	0	0.25	0.5	1	2	4
R	1024	0	0.25	0.5	1	2	4
S	1024	0	0.25	0.5	1	2	4
T	1024	0	0.25	0.5	1	2	4

Grp	1024_at	36202_at	36085_at	40322_at	407_at	1091_at	1708_at
A	16	32	64	128	0	512	1024
B	32	64	128	256	0.25	1024	0
C	64	128	256	512	0.5	0	0.25
D	128	256	512	1024	1	0.25	0.5
E	256	512	1024	0	2	0.5	1
F	512	1024	0	0.25	4	1	2
G	1024	0	0.25	0.5	8	2	4
H	0	0.25	0.5	1	16	4	8
I	0.25	0.5	1	2	32	8	16
J	0.5	1	2	4	64	16	32
K	4	2	4	8	128	32	64
L	4	4	8	16	256	64	128
M	4	8	16	32	512	128	256
N	4	8	16	32	512	128	256
O	4	8	16	32	512	128	256
P	4	8	16	32	512	128	256
Q	8	16	32	64	1024	256	512
R	8	16	32	64	1024	256	512
S	8	16	32	64	1024	256	512
T	8	16	32	64	1024	256	512

Table A.1: Concentrations in pM for spike-in probesets in Affymetrix HG_U95A dataset. There were three replicates for every group except group C making a total of 59 arrays.

Group	1	2	3	4	5	6	7
EXP 1	0	0.125	0.25	0.5	1	2	4
EXP 2	0.125	0.25	0.5	1	2	4	8
EXP 3	0.25	0.5	1	2	4	8	16
EXP 4	0.5	1	2	4	8	16	32
EXP 5	1	2	4	8	16	32	64
EXP 6	2	4	8	16	32	64	128
EXP 7	4	8	16	32	64	128	256
EXP 8	8	16	32	64	128	256	512
EXP 9	16	32	64	128	256	512	0
EXP 10	32	64	128	256	512	0	0.125
EXP 11	64	128	256	512	0	0.125	0.25
EXP 12	128	256	512	0	0.125	0.25	0.5
EXP 13	256	512	0	0.125	0.25	0.5	1
EXP 14	512	0	0.125	0.25	0.5	1	2
Group	8	9	10	11	12	13	14
EXP 1	8	16	32	64	128	256	512
EXP 2	16	32	64	128	256	512	0
EXP 3	32	64	128	256	512	0	0.125
EXP 4	64	128	256	512	0	0.125	0.25
EXP 5	128	256	512	0	0.125	0.25	0.5
EXP 6	256	512	0	0.125	0.25	0.5	1
EXP 7	512	0	0.125	0.25	0.5	1	2
EXP 8	0	0.125	0.25	0.5	1	2	4
EXP 9	0.125	0.25	0.5	1	2	4	8
EXP 10	0.25	0.5	1	2	4	8	16
EXP 11	0.5	1	2	4	8	16	32
EXP 12	1	2	4	8	16	32	64
EXP 13	2	4	8	16	32	64	128
EXP 14	4	8	16	32	64	128	256

Table A.2: Concentrations in pM for spike-in probesets in Affymetrix HG_U133A dataset. There were three replicates for every experimental group.

and table A.3 shows the names of the spike-in probeset.

A.3 GeneLogic AML Spike-in Dataset

Provided as part of a group of spike-in datasets by GeneLogic, and made publically available. This dataset consists of 34 HG_U95A arrays. There are 11 probesets that have been spiked in at 12 different concentrations ranging from 0.5 pM to 100 pM. Table A.4 shows the 12 distinct experimental groups and the corresponding spike-in concentrations. The experiment used a latin square design to assign spike-in probeset concentrations. Each group had three replicate arrays except group 1, for

Group	Probesets
1	203508_at, 204563_at, 204513_s_at
2	204205_at, 204959_at, 207655_s_at
3	204836_at, 205291_at, 209795_at
4	207777_s_at, 204912_at, 205569_at
5	207160_at, 205692_s_at, 212827_at
6	209606_at, 205267_at, 204417_at
7	205398_s_at, 209734_at, 209354_at
8	206060_s_at, 205790_at, 200665_s_at
9	207641_at, 207540_s_at, 204430_s_at
10	203471_s_at, 204951_at, 207968_s_at
11	AFFX-r2-TagA_at, AFFX-r2-TagB_at, AFFX-r2-TagC_at
12	AFFX-r2-TagD_at, AFFX-r2-TagE_at, AFFX-r2-TagF_at
13	AFFX-r2-TagG_at, AFFX-r2-TagH_at, AFFX-DapX-3_at
14	AFFX-LysX-3_at, AFFX-PheX-3_at, AFFX-ThrX-3_at

Table A.3: Names of probesets in each spike-in group for Affymetrix HG_U133A dataset.

Grp	BioB-5_at	DapX-M_at	DapX-5_at	CreX-5_at	BioB-3_at	BioB-M_at	BioDn-3_at	BioC-5_at	BioC-3_at	DapX-3_at	CreX-3_at
1	25	37.5	50	75	100	3	5	12.5	0.5	1	1.5
2	37.5	50	75	100	3	5	12.5	0.5	1	1.5	2
3	50	75	100	3	5	12.5	0.5	1	1.5	2	25
4	75	100	3	5	12.5	0.5	1	1.5	2	25	37.5
5	100	3	5	12.5	0.5	1	1.5	2	25	37.5	50
6	3	5	12.5	0.5	1	1.5	2	25	37.5	50	75
7	5	12.5	0.5	1	1.5	2	25	37.5	50	75	100
8	12.5	0.5	1	1.5	2	25	37.5	50	75	100	3
9	0.5	1	1.5	2	25	37.5	50	75	100	3	5
10	1	1.5	2	25	37.5	50	75	100	3	5	12.5
11	1.5	2	25	37.5	50	75	100	3	5	12.5	0.5
12	2	25	37.5	50	75	100	3	5	12.5	0.5	1

Table A.4: Concentrations for GeneLogic AML Dataset in pM for the 11 spike-in transcripts. Each group has three replicates except group 1.

which there is only one array.

These arrays were all hybrized with common acute myeloid leukemia (AML) complex background, with only the concentrations of the control spike-in probesets differing.

A.4 GeneLogic Tonsil Spike-in dataset

This dataset is also part of a the same series of spike-in experiments as the dataset discussed in Appendix A.3. It also contains 11 control spike-in probesets at concentrations ranging from 0.5 to 100

Grp	BioB-5_at	DapX-M_at	DapX-5_at	CreX-5_at	BioB-3_at	BioB-M_at	BioDn-3_at	BioC-5_at	BioC-3_at	DapX-3_at	CreX-3_at
1	0.5	0.75	1	1.5	2	3	5	12.5	25	50	75
2	0.75	1	1.5	2	3	5	12.5	25	50	75	100
3	1	1.5	2	3	5	12.5	25	50	75	100	0.5
4	1.5	2	3	5	12.5	25	50	75	100	0.5	0.75
5	2	3	5	12.5	25	50	75	100	0.5	0.75	1
6	3	5	12.5	25	50	75	100	0.5	0.75	1	1.5
7	5	12.5	25	50	75	100	0.5	0.75	1	1.5	2
8	12.5	25	50	75	100	0.5	0.75	1	1.5	2	3
9	25	50	75	100	0.5	0.75	1	1.5	2	3	5
10	50	75	100	0.5	0.75	1	1.5	2	3	5	12.5
11	75	100	0.5	0.75	1	1.5	2	3	5	12.5	25
12	100	0.5	0.75	1	1.5	2	3	5	12.5	25	50

Table A.5: Concentrations for GeneLogic Tonsil Dataset in pM for the 11 spike-in transcripts. Each group has three replicates.

pM arranged in a Latin square design, as shown in Table A.5. Each group had three replicates making for a total of 36 arrays. These arrays were hybridized with a common complex RNA produced from pooled tonsil tissue samples.

A.5 GeneLogic Dilution/Mixture dataset

This dataset consists of 75 HGU95A_v2 arrays. It consists of arrays to which RNA from either Liver, Central Nervous System (CNS) or a mixture of the two sources has been hybridized. Table A.6 shows the concentrations that were used. In each case there were 5 replicate arrays each scanned on a different scanner.

Type	Conc. Liver	Conc. CNS
Dilution	20	0
Dilution	10	0
Dilution	7.5	0
Dilution	5	0
Dilution	2.5	0
Dilution	1.25	0
Dilution	0	20
Dilution	0	10
Dilution	0	7.5
Dilution	0	5
Dilution	0	2.5
Dilution	0	1.25
Mixture	7.5	2.5
Mixture	5	5
Mixture	2.5	7.5

Table A.6: GeneLogic Dilution/Mixture study. 5 arrays were used at each concentration level. Concentrations are in μg.

Bibliography

Affymetrix (1999). *Affymetrix Microarray Suite Users Guide*. Affymetrix, Santa Clara, CA, version 4.0 edition.

Affymetrix (2001a). *Affymetrix Microarray Suite Users Guide*. Affymetrix, Santa Clara, CA, version 5.0 edition.

Affymetrix (2001b). Statistical algorithms reference guide. Technical report, Affymetrix, Santa Clara, CA.

Affymetrix (2002). Statistical algorithms description document. Technical report, Affymetrix, Santa Clara, CA.

Affymetrix (2003). *GeneChip® Expression Analysis Technical Manual*. Affymetrix, Santa Clara, CA, rev 4.0 edition.

Altman, D. G. and Bland, J. M. (1983). Measurement in medicine: The analysis of method comparison studies. *The Statistician*, 32(3).

Amaratunga, D. and Cabrera, J. (2001). Analysis of data from viral dna microchips. *Journal of the American Statistical Association*, 96(456):1161–1170.

Astrand, M. (2003). Contrast normalization of oligonucleotide arrays. *J Comput Biol*, 10(1):95–102.

Baldi, P. and Long, A. D. (2001). A Bayesian framework for the analysis of microarray expression data: regularized t -test and statistical inferences of gene changes. *Bioinformatics*, 17(6):509–519.

Ballman, K., Grill, D., Oberg, A., and Thurneau, T. (2003). Faster cyclic loess: normalizing dna arrays via linear models. Technical Report 68, Mayo Foundation, Rochester, MN.

Begg, C. B. (1991). Advances in statistical methodology for diagnostic medicine in the 1980's. *Statistics in Medicine*, 10:1887–1895.

Berg, J. M., Tymoczko, J. L., and Stryer, L. (2002). *Biochemistry*. W.H. Freeman, New York.

Bland, J. M. and Altman, D. G. (1986). Statistical methods for assessing agreement between two methods of clinical measurement. *Lancet*, 1(8476):307–310. Clinical Trial.

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(2):185–193.

Broberg, P. (2003). Statistical methods for ranking differentially expressed genes. *Genome Biology*, 4(6):R41. A previous version of this manuscript was made available before peer review at http://genomebiology.com/2002/3/9/preprint/0007.

Brown, P. O. and Botstein, D. (1999). Exploring the new world of the genome with DNA microarrays. *Nat Genet*, 21(1 Suppl):33–37.

Campbell, G. (1994). Advances in statistical methodology for the evaluation of diagnostic and laboratory tests. *Statistics in Medicine*, 13:499–508.

Chabas, D., Baranzini, S. E., Mitchell, D., Bernard, C. C. A., Rittling, S. R., Denhardt, D. T., Sobel, R. A., Lock, C., Karpuj, M., Pedotti, R., Heller, R., Oksenberg, J. R., and Steinman, L. (2001). The Influence of the Proinflammatory Cytokine, Osteopontin, on Autoimmune Demyelinating Disease. *Science*, 294(5547):1731–1735.

Chambers, J. M., Cleveland, W. S., Kleiner, B., and Tukey, P. A. (1983). *Graphical Methods of Data Analysis*. Wadsworth, Belmont, CA.

Chicurel, M. E. and Dalma-Weiszhausz, D. D. (2002). Microarrays in pharmacogenomics–advances and future promise. *Pharmacogenomics*, 3(5):589–601.

Chin, K.-V. and Kong, A. N. T. (2002). Application of DNA microarrays in pharmacogenomics and toxicogenomics. *Pharm Res*, 19(12):1773–1778.

Churchill, G. A. (2002). Fundamentals of experimental design for cDNA microarrays. *Nat Genet*, 32 Suppl:490–495.

Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74:829–836.

Cleveland, W. S. and Devlin, S. J. (1988). Locally-weighted regression: An approach to regression analysis by local fitting. *J. Am. Stat. Assoc.*, 83:596–610.

Collin, F., Brettschneider, J., Bolstad, B. M., and Speed, T. P. (2003). Quality assessment of gene expression data for affymetrix genechips. *Poster at 2003 Affymetrix Workshop on Low Level Analysis*.

Cope, L. M., Irizarry, R. A., Jaffee, H. A., Wu, Z., and Speed, T. P. (2004). A benchmark for Affymetrix GeneChip expression measures. *Bioinformatics*, 20(3):323–331.

Crnogorac-Jurcevic, T., Efthimiou, E., Nielsen, T., Loader, J., Terris, B., Stamp, G., Baron, A., Scarpa, A., and Lemoine, N. R. (2002). Expression profiling of microdissected pancreatic adenocarcinomas. *Oncogene*, 21(29):4587–4594.

Cui, X., Kerr, M. K., and Churchill, G. A. (2003). Transformations for cdna microarray data. *Statistical Applications in Genetics and Molecular Biology*, 2(1):Article 4.

DeRisi, J., Penland, L., Brown, P. O., Bittner, M. L., Meltzer, P. S., Ray, M., Chen, Y., Su, Y. A., and Trent, J. M. (1996). Use of a cDNA microarray to analyse gene expression patterns in human cancer. *Nat Genet*, 14(4):457–460.

Dorfman, R. (1943). The detection of defective members of large populations. *The Annals of Mathematical Statistics*, 14(4):436–440.

Dudoit, S., Yang, Y. H., Callow, M. J., and Speed, T. P. (2002). Statistical methods for identifying genes with differential expression in replicated cdna microarray experiments. *Stat. Sin.*, 12(1):111–139.

Enard, W., Khaitovich, P., Klose, J., Zollner, S., Heissig, F., Giavalisco, P., Nieselt-Struwe, K., Muchmore, E., Varki, A., Ravid, R., Doxiadis, G. M., Bontrop, R. E., and Paabo, S. (2002). Intra- and interspecific variation in primate gene expression patterns. *Science*, 296(5566):340–343.

Fodor, S. P., Rava, R. P., Huang, X. C., Pease, A. C., Holmes, C. P., and Adams, C. L. (1993). Multiplexed biochemical assays with biological chips. *Nature*, 364(6437):555–556.

Fodor, S. P., Read, J. L., Pirrung, M. C., Stryer, L., Lu, A. T., and Solas, D. (1991). Light-directed, spatially addressable parallel chemical synthesis. *Science*, 251(4995):767–773.

Gastwirth, J. L. and Hammick, P. A. (1989). Estimation of the prevalence of a rare disease, preserving the anonymity of the subjects by group testing: application to estimating the prevalence of aids antibodies in blood donors. *Journal of Statistical Planning and Inference*, 22(1):15–27.

Gautier, L., Cope, L., Bolstad, B. M., and Irizarry, R. A. (2004). affy–analysis of Affymetrix GeneChip data at the probe level. *Bioinformatics*, 20(3):307–315.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D., and Lander, E. S. (1999). Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science*, 286(5439):531–537.

Gonick, L. and Wheelis, M. (1991). *The Cartoon Guide to Genetics*. HarperPerenial, updated edition.

Hamadeh, H. K., Bushel, P. R., Jayadev, S., Martin, K., DiSorbo, O., Sieber, S., Bennett, L., Tennant, R., Stoll, R., Barrett, J. C., Blanchard, K., Paules, R. S., and Afshari, C. A. (2002). Gene Expression Analysis Reveals Chemical-Specific Profiles. *Toxicol. Sci.*, 67(2):219–231.

Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., and Stahel, W. A. (1986). *Robust Statistics: The Approach Based On Influence Functions*. John Wiley & Sons, Inc, New York, New York.

Han, E.-S., Wu, Y., McCarter, R., Nelson, J. F., Richardson, A., and Hilsenbeck, S. G. (2004). Reproducibility, Sources of Variability, Pooling, and Sample Size: Important Considerations for the Design of High-Density Oligonucleotide Array Experiments. *J Gerontol A Biol Sci Med Sci*, 59(4):B306–315.

Hartemink, A., Gifford, D., Jaakkola, T., , and Young, R. (2001). Maximum likelihood estimation of optimal scaling factors for expression array normalization. In *SPIE BIOS 2001*.

Hawkins, D. M. (2002). Diagnostics for conformity of paired quantitative measurements. *Stat Med*, 21(13):1913–1935.

Hubbell, E., Liu, W.-M., and Mei, R. (2002). Robust estimators for expression analysis. *Bioinformatics*, 18(12):1585–1592.

Huber, P. J. (1981). *Robust statistics*. John Wiley & Sons, Inc, New York, New York.

Ihaka, R. and Gentleman, R. (1996). R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5(3):299–314.

Irizarry, R. A., Bolstad, B. M., Collin, F., Cope, L. M., Hobbs, B., and Speed, T. P. (2003a). Summaries of Affymetrix GeneChip probe level data. *Nucleic Acids Res*, 31(4):e15.

Irizarry, R. A., Hobbs, B., Collin, F., Beazer-Barclay, Y. D., Antonellis, K. J., Scherf, U., and Speed, T. P. (2003b). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostat*, 4(2):249–264.

Jiang, Y., Harlocker, S. L., Molesh, D. A., Dillon, D. C., Stolk, J. A., Houghton, R. L., Repasky, E. A., Badaro, R., Reed, S. G., and Xu, J. (2002). Discovery of differentially expressed genes in human breast cancer using subtracted cDNA libraries and cDNA microarrays. *Oncogene*, 21(14):2270–2282.

Jin, W., Riley, R. M., Wolfinger, R. D., White, K. P., Passador-Gurgel, G., and Gibson, G. (2001). The contributions of sex, genotype and age to transcriptional variance in Drosophila melanogaster. *Nat Genet*, 29(4):389–395.

Kendziorski, C., Zhang, Y., Lan, H., and A.D. Attie, A. (2003). The efficiency of mrna pooling in microarray experiments. *Biostatistics*, 4:465–477.

Li, C. and Wong, W. H. (2001a). Model-based analysis of oligonucleotide arrays: expression index computation and outlier detection. *Proc Natl Acad Sci U S A*, 98(1):31–36.

Li, C. and Wong, W. H. (2001b). Model-based analysis of oligonucleotide arrays: model validation, design issues and standard error application. *Genome Biol*, 2(8):RESEARCH0032.

Lipshutz, R. J., Fodor, S. P., Gingeras, T. R., and Lockhart, D. J. (1999). High density synthetic oligonucleotide arrays. *Nat Genet*, 21(1 Suppl):20–24.

Lockhart, D. J. and Barlow, C. (2001). Expressing what's on your mind: DNA arrays and the brain. *Nat Rev Neurosci*, 2(1):63–68.

Lockhart, D. J., Dong, H., Byrne, M. C., Follettie, M. T., Gallo, M. V., Chee, M. S., Mittmann, M., Wang, C., Kobayashi, M., Horton, H., and Brown, E. L. (1996). Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nat Biotechnol*, 14(13):1675–1680.

Lockhart, D. J. and Winzeler, E. A. (2000). Genomics, gene expression and DNA arrays. *Nature*, 405(6788):827–836.

Lönnstedt, I. and Speed, T. P. (2002). Replicated microarray data. *Statistica Sinica*, 12:31–46.

Lusted, L. B. (1971). Signal detectability and medical decision-making. *Science*, 171(3977):1217–1219.

Mei, R., Hubbell, E., Bekiranov, S., Mittmann, M., Christians, F. C., Shen, M.-M., Lu, G., Fang, J., Liu, W.-M., Ryder, T., Kaplan, P., Kulp, D., and Webster, T. A. (2003). Probe selection for high-density oligonucleotide arrays. *Proc Natl Acad Sci U S A*, 100(20):11237–11242.

Naef, F., Lim, D. A., Patil, N., and Magnasco, M. A. (2001). From features to expression: High-density oligonucleotide array analysis revisited. Downloaded from http://xxx.lanl.gov/abs/physics/0102010.

Nuwaysir, E. F., Bittner, M., Trent, J., Barrett, J. C., and Afshari, C. A. (1999). Microarrays and toxicology: the advent of toxicogenomics. *Mol Carcinog*, 24(3):153–159.

Pease, A., Solas, D., Sullivan, E., Cronin, M., Holmes, C., and Fodor, S. (1994). Light-Generated Oligonucleotide Arrays for Rapid DNA Sequence Analysis. *PNAS*, 91(11):5022–5026.

Peng, X., Wood, C. L., Blalock, E. M., Chen, K. C., Landfield, P. W., and Stromberg, A. J. (2003). Statistical implications of pooling RNA samples for microarray experiments. *BMC Bioinformatics*, 4(1):26.

Peterson, W., Birdsall, T., and Fox, W. (1954). The theory of signal detectability. *IEEE Transactions on Information Theory*, 4(4):171– 212.

Pitman, E. J. G. (1939). The estimation of the location and scale parameters of a continuous population of any given form. *Biometrika*, 30(3/4):391–421.

Quackenbush, J. (2002). Microarray data normalization and transformation. *Nat Genet*, 32 Suppl:496–501.

Regalado, A. (1999). Inventing the pharmacogenomics business. *Am J Health Syst Pharm*, 56(1):40–50.

Rousseeuw, P. J. and Leroy, A. M. (1987). *Robust Regression and Outlier Detection*. John Wiley & Sons, Inc, New York, New York.

Saban, M. R., Hellmich, H., Nguyen, N. B., Winston, J., Hammond, T. G., and Saban, R. (2001). Time course of LPS-induced gene expression in a mouse model of genitourinary inflammation. *Physiol Genomics*, 5(3):147–160.

Sambrook, J., Fritsch, E., and Maniatis, T. (1989). *Molecular Cloning: A Laboratory Manual*. Cold Spring Harbor Press, Spring Harbor, NY, 2nd edition edition.

Schadt, E. E., Li, C., Ellis, B., and Wong, W. H. (2001). Feature extraction and normalization algorithms for high-density oligonucleotide gene expression array data. *J Cell Biochem Suppl*, Suppl 37:120–125.

Schadt, E. E., Li, C., Su, C., and Wong, W. H. (2000). Analyzing high-density oligonucleotide gene expression array data. *J Cell Biochem*, 80(2):192–202.

Sidorov, I. A., Hosack, D. A., Gee, D., Yang, J., Cam, M. C., Lempicki, R. A., and Dimitrov, D. S. (2002). Oligonucleotide microarray data distribution and normalization. *Information Sciences*, 146(1-4):67–73.

Simon, R. (2003). Using DNA microarrays for diagnostic and prognostic prediction. *Expert Rev Mol Diagn*, 3(5):587–595.

Smyth, G. K. (2004). Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, 3(1):Article 3.

Smyth, G. K., Yang, Y. H., and Speed, T. P. (2003). Statistical issues in cdna microarray data analysis. In Brownstein, M. J. and Khodursky, A. B., editors, *Functional Genomics: Methods and Protocols - Methods in Molecular Biology*, volume 224. Humana Press, Totowa, NJ.

Swets, J. A. (1988). Measuring the accuracy of diagnostic systems. *Science*, 240(4857):1285–1293.

Tukey, J. W. (1977). *Exploratory Data Analysis*. Addison-Wesley, Reading, Massachusetts.

Tusher, V. G., Tibshirani, R., and Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *PNAS*, 98(9):5116–5121.

Waring, J. F., Jolly, R. A., Ciurlionis, R., Lum, P. Y., Praestgaard, J. T., Morfitt, D. C., Buratto, B., Roberts, C., Schadt, E., and Ulrich, R. G. (2001). Clustering of hepatotoxins based on mechanism of toxicity using gene expression profiles. *Toxicol Appl Pharmacol*, 175(1):28–42.

Warrington, J. A., Dee, S., and Trulson, M. (2000). Large-scale genomic analysis using affymetrix genechip®. In Schena, M., editor, *Microarray Biochip Technology*, chapter 6, pages 119–148. BioTechniques Books.

Winzeler, E. A., Shoemaker, D. D., Astromoff, A., Liang, H., Anderson, K., Andre, B., Bangham, R., Benito, R., Boeke, J. D., Bussey, H., Chu, A. M., Connelly, C., Davis, K., Dietrich, F., Dow, S. W., El Bakkoury, M., Foury, F., Friend, S. H., Gentalen, E., Giaever, G., Hegemann, J. H., Jones, T., Laub, M., Liao, H., Liebundguth, N., Lockhart, D. J., Lucau-Danila, A., Lussier, M., M'Rabet, N., Menard, P., Mittmann, M., Pai, C., Rebischung, C., Revuelta, J. L., Riles, L., Roberts, C. J., Ross-MacDonald, P., Scherens, B., Snyder, M., Sookhai-Mahadeo, S., Storms, R. K., Veronneau, S., Voet, M., Volckaert, G., Ward, T. R., Wysocki, R., Yen, G. S., Yu, K., Zimmermann, K., Philippsen, P., Johnston, M., and Davis, R. W. (1999). Functional Characterization of the S.cerevisiae Genome by Gene Deletion and Parallel Analysis. *Science*, 285(5429):901–906.

Wolfinger, R. and Chu, T. (2002). Who are those strangers in the latin square? In *Proceedings of CAMDA 2002*.

Workman, C., Jensen, L. J., Jarmer, H., Berka, R., Gautier, L., Nielser, H. B., Saxild, H.-H., Nielsen, C., Brunak, S., and Knudsen, S. (2002). A new non-linear normalization method for reducing variability in DNA microarray experiments. *Genome Biol*, 3(9):research0048.

Wu, Z. and Irizarry, R. A. (2004). Stochastic models inspired by hybridization theory for short oligonucleotide arrays. In *Proceedings of RECOMB 2004*.

Wu, Z., Irizarry, R. A., Gentleman, R., Murillo, F. M., and Spencer, F. (2003). A model based background adjustment for oligonucleotide expression arrays. Technical Report Working Paper 1, Johns Hopkins University, Dept. of Biostatistics Working Papers.

Yang, Y. H., Dudoit, S., Luu, P., Lin, D. M., Peng, V., Ngai, J., and Speed, T. P. (2002). Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Res*, 30(4):e15.

Zhu, G., Reynolds, L., Crnogorac-Jurcevic, T., Gillett, C. E., Dublin, E. A., Marshall, J. F., Barnes, D., D'Arrigo, C., Van Trappen., P. O., Lemoine, N. R., and Hart, I. R. (2003). Combination of microdissection and microarray analysis to identify gene expression changes between differentially located tumour cells in breast cancer. *Oncogene*, 22(24):3742–3748.

Zien, A., Aigner, T., Zimmer, R., and Lengauer, T. (2001). Centralization: a new method for the normalization of gene expression data. *Bioinformatics*, 17(90001):323S–331.