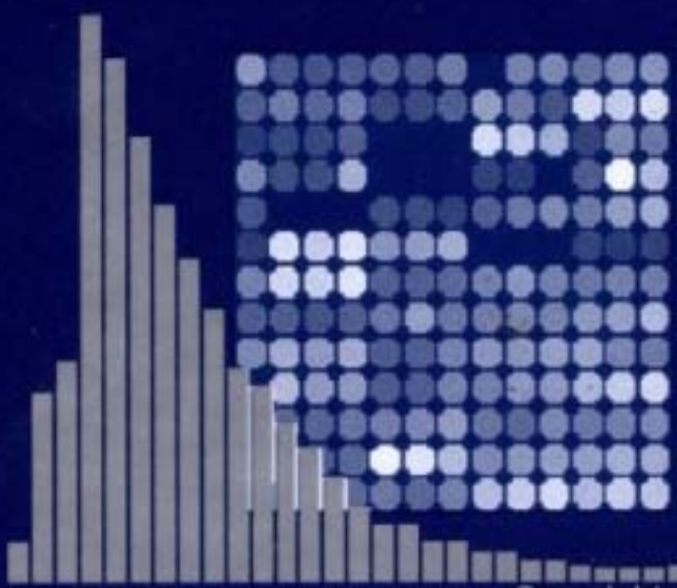
WILEY

EXPLORATION AND ANALYSIS OF DNA MICROARRAY AND PROTEIN ARRAY DATA

DHAMMIKA AMARATUNGA
JAVIER CABRERA

Wiley Series in Probability and Statistics



Exploration and Analysis of DNA Microarray and Protein Array Data

WILEY SERIES IN PROBABILITY AND STATISTICS

Established by WALTER A. SHEWHART and SAMUEL S. WILKS

A complete list of the titles in this series appears at the end of this volume.

Exploration and Analysis of DNA Microarray and Protein Array Data

DHAMMIKA AMARATUNGA

Johnson & Johnson Pharmaceutical R&D
Raritan, NJ

JAVIER CABRERA

Rutgers University
Piscataway, NJ

A JOHN WILEY & SONS, INC., PUBLICATION

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

Library of Congress Cataloging-in-Publication Data:

Amaratunga, Dhammika, 1956–
Exploration and analysis of DNA microarray and protein array data / Dhammika Amaratunga, Javier Cabrera.
p. cm.
Includes bibliographical references and index.
ISBN 0-471-27398-8 (cloth)
1. DNA microarrays—Statistical methods. 2. Protein microarrays—Statistical methods. I. Cabrera, Javier. II. Title.
QP624.5.D726A45 2004
572.8′636—dc21 2004050097

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

To our families
in America, Sri Lanka, and Spain

Contents

Preface

In August 1999, at the Joint Statistical Meetings in Baltimore, the two of us were invited to present a paper on the analysis of DNA microchip data. This was the first presentation in any Joint Statistical Meetings on the topic of DNA microarrays, as they are now called. In just a few years, the field has exploded, and in August 2002, at the Joint Statistical Meetings in New York City, there were over a hundred presentations related to DNA microarrays!

Our Baltimore paper outlined many of the issues that are still being discussed today, including intensity-dependent normalization, the use of methods that are robust to outliers and improving the sensitivity of the analysis by borrowing strength across genes. However, there have been many developments since then, and this book represents our effort at organizing this material into a semi-coherent whole.

Both of us were trained as statisticians at Princeton University, and while we became skilled at data analysis, we did not learn much biology there, little did we realize how much we would need it later on in our careers. Thus the last few years have been an educational period for us, learning about molecular biology and having to rethink some of what we learned in statistics. Incidentally, the Statistics Department at Princeton is no more (a sad reflection on our field and we earnestly hope it is not a trend), while the Molecular Biology Department, which did not exist then, is growing.

We would like to thank several bioinformaticians, scientists, and statisticians at Johnson and Johnson Pharmaceutical Research and Development LLC who patiently helped educate us about genomics and microarrays and in writing this book. We would particularly like to acknowledge Jim Colaianne, for his whole-hearted support throughout the entire project, Gordon Pledger for introducing us to microarrays, Gayatri Amaratunga, who organized the contents and references, Maria Drelich, who helped with the composition of the figures, Harindra Abeysinghe, who read through the molecular biology sections of the book and provided many useful comments (any remaining errors are

ours), and Albert Lo and the students at SBM of Hong Kong University of Science and Technology for their inspiration and support.

DHAMMIKA AMARATUNGA
JAVIER CABRERA

Exploration and Analysis of DNA Microarray and Protein Array Data
Edited by DHAMMIKA AMARATUNGA and JAVIER CABRERA

CHAPTER 1

A Brief Introduction

Data analysis has, quite suddenly, begun to assume a prominent role in the life sciences. From being a science that generally produced relatively limited amounts of quantitative data, biology has, in the space of just a few years, become a science that routinely generates enormous amounts of it.

To a large part, this metamorphosis can be attributed to two complementary advances. The first is the successful culmination of the Human Genome Project and other genome-sequencing efforts, which have generated a treasure trove of information about the DNA sequences of the human genome and the genomes of several other species, large and small. Biologists are now confronted with a huge number of genes being newly identified and the daunting, but exhilarating, task of ascertaining their functions.

This is where the second advance, the emergence of modern experimental technology, such as microarray technology, comes in. Currently the most widely used form of this technology is the DNA microarray, which offers scientists the ability to monitor the behavior patterns of several thousands of genes simultaneously, allowing them to study how these genes function and follow how they act under different conditions. Another form of microarray technology, the protein array, provides scientists the capability of monitoring thousands of proteins simultaneously, for similar purposes. And this is just the beginning. Emerging technical innovations, such as bead-based arrays, have the potential to increase throughput even much more.

These developments have ushered in a thrilling new era of molecular biology. Traditional molecular biology research followed a "one gene per experiment" paradigm. This tedious and inherently exhausting approach was capable of producing only limited results in any reasonable period of time. Although it has, without question, logged a series of remarkable achievements over the years, this approach does not allow anything close to a complete picture of gene function and overall genome behavior to be readily determined.

Exploration and Analysis of DNA Microarray and Protein Array Data
By Dhammika Amaratunga and Javier Cabrera
ISBN 0-471-27398-8

The advent of microarray technology has created an opportunity for doing exactly this by fast tracking research practice away from a "one gene" mode to a "thousands of genes per experiment" mode and allowing scientists to study how genes function, not just each on its own, but jointly as well.

In fact the way microarray technology is revolutionizing the biological sciences has been likened to the way microprocessors transformed the computer sciences toward the latter part of the twentieth century (through miniaturization, integration, parallel processing, increased throughput, portability, and automation) and the way the computer sciences, in turn, transformed many other disciplines just a few years later. Microarray technology has been brought into play to characterize genomic function in genome systems spanning all the way from yeast to human.

Microarray experiments are conducted in such a manner as to profile the behavior patterns of thousands of nucleic acid sequences or protein simultaneously. Plus, they are capable of being automated and run in a high-throughput mode. Thus they can, and do, generate mountains of data at an ever-increasing pace. The proper storage, analysis and interpretation of these data have turned out to be a major challenge.

Our focus is on the analysis part. After all, the data alone does not constitute knowledge. It must be first analyzed, relationships and associations studied and confirmed, in order to convert it into knowledge. By doing so, it is hoped that a complete picture of the intermeshing patterns of biomolecular activity that underlie complex biological processes, such as the growth and development of an organism and the etiology of a disease, would emerge.

One issue is that the structure of the data is singular enough to warrant special attention. The raw data from a DNA microarray experiment, for example, is a series of scanned images of microarrays that have been subjected to an experimental process. The general plan for analyzing this data involves converting these images into quantitative data, then preprocessing the data to transform it into a format suitable for analysis, and finally applying appropriate data analysis techniques to extract information pertinent to the biological question under study. Application of statistical methodology is feasible as these experiments can be run on replicate samples, although, by and large, the amount of replication tends to be limited. Thus a complexity is that while there is data on thousands and thousands of genes, the information content per gene is small. As a result there is a sense that much of the data collected in microarray experiments remains to be fully and properly interpreted.

It should therefore not be a surprise that statistical and computational approaches are beginning to assume a position of greater prominence within the molecular biology community. While these quantitative disciplines have a rich and impressive array of tools to cover a very broad range of topics in data analysis, the structure of the data generated by microarrays is sufficiently unique that either standard methods have to be tailored for use with microarray data or an entirely fresh set of tools has to be developed specifically to handle such data. What has happened, of course, is a confluence of the two.

The purpose of this book is to present an extensive, but, by no means, exhaustive, series of computational, visual, and statistical tools that are being used for exploring and analyzing microarray data.

1.1 A NOTE ON EXPLORATORY DATA ANALYSIS

Early statistical work was essentially enumerative and exploratory in nature. Statisticians were concerned with developing effective ways of discerning patterns in quantitative data. Then, from about a fourth of the way into the twentieth century, mathematics-driven confirmatory techniques began to dominate the field of statistics, driving data exploration into the background. The focus began to be the development of optimal ways to analyze data rigorously, but under various sets of fairly restrictive assumptions.

Fortunately, toward the latter part of the twentieth century, data exploration began to make a comeback as an imperative aspect of statistics, having been revitalized almost single-handedly by Tukey (1962, 1977, 1986), who likened it to detective work. *Exploratory data analysis* (EDA), as the modern incarnation of statistical data exploration is called, is an approach for data analysis that employs a range of techniques (many graphical), in a strategic fashion, in order to:

- Gain insight into a data set
- Discover systematic structures, such as clusters, in the data
- Flag outliers and anomalies in the data
- Assess what assumptions about the data are reasonable

The last of these guides the data analyst to an approach or a model that should be suitable for a more formal phase in the analysis of the data. This *confirmatory data analysis* (CDA) phase, which may involve inferential procedures such as confidence interval estimation and hypothesis testing, allows the data analyst to probabilistically model the uncertainties of a situation to assess the reproducibility of the findings. CDA ensures that chance patterns are not mistaken for real structure. Even at this phase, EDA stresses the importance of running diagnostic checks to assess the validity of any underlying assumptions (e.g., Anscombe and Tukey, 1963; Daniel and Wood, 1971).

EDA is particularly well suited to situations where the data is not well understood and the problem is not well specified, such as screening. For this reason EDA techniques have found their way into the world of data mining (Fayyad, Piatetsky-Shapiro, and Smyth, 1996). In data mining, broad-based methods that have the capability to discover and illustrate essential aspects of the data are of most value. Proper data visualization tools, for instance, are highly effective both at revealing facets of the data that otherwise may not be apparent and at challenging assumptions about the data that otherwise may be taken for granted.

It could be argued that EDA is as much an attitude or a philosophy about how a data analysis should be conducted as an assortment of techniques. The EDA approach suggests strategies for carefully scrutinizing a data set: how to examine a data set, what to look for, and how to interpret what has been observed. The key is that EDA permits the data itself to reveal its underlying structure and model without the data analyst having to make too many possibly indefensible assumptions.

Over the years the popularity of EDA has been boosted by a number of noteworthy publications by Tukey and his students and colleagues, such as Mosteller and Tukey (1977), Velleman and Hoaglin (1981), Hoaglin (1982), Hoaglin, Mosteller, and Tukey (1983), Tukey (1986), Brillinger, Fernholz, and Morgenthaler (1997), Fernholz, Morgenthaler, and Stahel (2001), and has gained a large following as the most effective way to seek structures in data. Hoaglin, Mosteller, and Tukey (1983) provide an excellent introduction to EDA. Cabrera and McDougall (2002) give a wide range of applications of EDA to real world problems.

That is not to forget CDA. Tukey (1980) argues that exploratory and confirmatory analyses must both be components of a good data analysis. This is the approach we will take in this book.

1.2 COMPUTING CONSIDERATIONS AND SOFTWARE

The data analyst must have access to computing resources, both hardware and software, that are capable of dealing with the huge amounts of data that must be analyzed. Holloway et al. (2002) is a review of some of the issues related to this topic.

A number of software packages offer the data analyst powerful tools for EDA and CDA, including interactive graphics and a large collection of statistical procedures. Two that are commonly used in the analysis of microarray data are R (Ihaka and Gentleman, 1996) and SPLUS. Other statistical packages that are good for EDA include SAS, JMP, DataDesk, Matlab, MINITAB, and STATISTICA.

In addition libraries of routines specially designed for analysis of microarray data have begun to spring up. Some of these are in the public domain; others are only available commercially. A few are listed below:

- DNAMR (`http://www.rci.rutgers.edu/~cabrera/DNAMR.`), which stands for "DNA Microarray Routines," is a collection of R and SPLUS programs developed by the authors of this book. Implementations of many of the procedures described in this book are available in the DNAMR package and can be downloaded from the book's web page.
- The Bioconductor project (`http://www.bioconductor.org`), based at the Biostatistics Unit of the Dana Farber Cancer Institute at the Harvard

Medical School and the Harvard School of Public Health, produces open source R software for scientists and statisticians working in bioinformatics, with primary emphasis on inference using DNA microarrays.

- MA-ANOVA (`http://www.jax.org/research/churchill/software/anova`) is a set of functions written in Matlab by the Statistical Genetics group at the Jackson Laboratory for analysis of variance of microarray data.
- DRAGON (Database Referencing of Array Genes ONline) is a series of tools for analyzing and interpreting microarray data that has been developed by a group of researchers at the Johns Hopkins University (`http://pevsnerlab.kennedykrieger.org/index.html`). It has an annotate tool that can be used to add any type of biological information to the lists of genes. The DragonView suite of tools can be used to visualize microarray data relevant to the information derived from the annotate tool. SNOMAD is a collection of R programs for normalizing DNA microarray data.
- The Stanford University Laboratory for the Statistical Analysis of Microarray Data (`http://www-stat.stanford.edu/~tibs/lab`) has software called SAM: Significance Analysis of Microarrays, ScanAnalyze, Cluster, and TreeView.

Although such packages are adequate for routine analyses, for more complex experiments the greater flexibility afforded by software developed in-house may be more desirable. In addition care must be taken that this software is not blindly (mis)used by an individual who does not have enough understanding of the details of the procedures—using the wrong methods to analyze data from an experiment may produce meaningless "findings" or, at the very least, be less than optimal. Unfortunately, few off-the-shelf packages offer a comprehensive data-handling system that integrates all of the data-related needs, such as data acquisition, storage, extraction, quality assurance, and analysis, that are essential for even a moderate-sized microarray laboratory.

1.3 A BRIEF OUTLINE OF THE BOOK

Exploratory and confirmatory data analysis techniques can be applied to microarray data to:

- Assess the quality of a microarray
- Assess the quality of the individual spots on a microarray
- Determine which genes are differentially expressed
- Classify genes based on how they co-express
- Classify samples based on how genes co-express

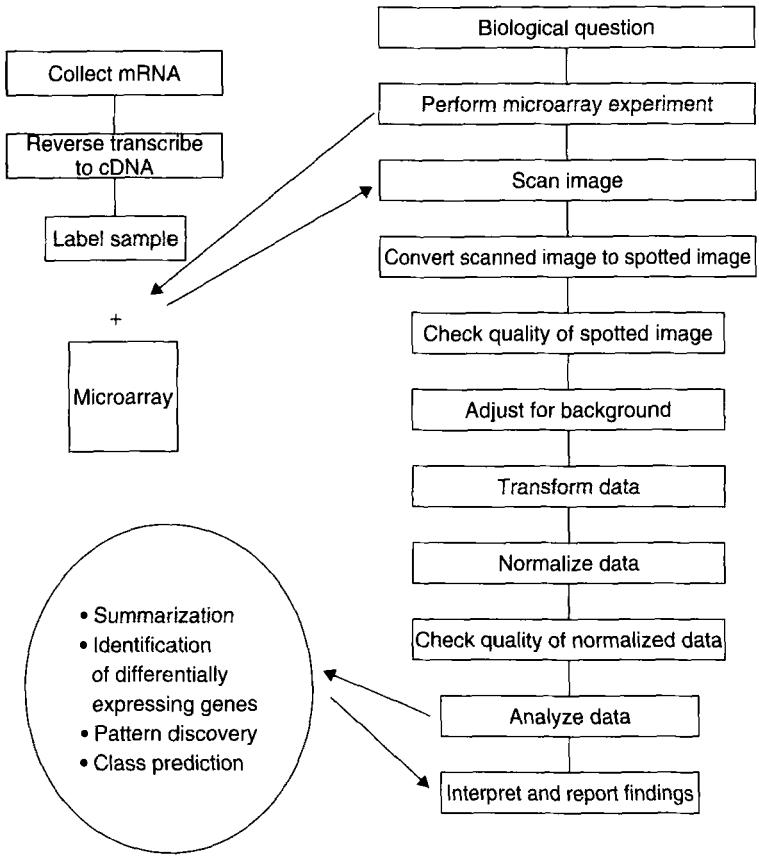


Figure 1.1 Schematic of a typical microarray data analysis.

Following this, the investigator will generally try to:

- Connect differentially expressed genes to sequence databases
- Locate differentially expressed genes on pathway diagrams
- Relate expression levels to other cell-related information
- Determine the roles of genes based on patterns of co-expression.

Often this process will culminate in an insight of interest.

Figure 1.1 shows schematically the path of a typical microarray data analysis. The reader may find it useful to periodically refer to it. In this book we will present a collection of techniques for analyzing microarray data. Before we embark on our journey, a brief road map of where we are going may be helpful.

Chapter 2 is a brief introduction to molecular biology and genomics. Chapter 3 describes DNA microarrays, what they are, how they are used, and how

a typical DNA microarray experiment is performed. Chapter 4 outlines how the output of a DNA microarray experiment, the scanned image, is processed and quantitated and how image and spot quality checks are done. Chapter 5 discusses preprocessing microarray data, which typically involves transforming the data and then applying a normalization. Chapter 6 discusses summarization of data across replicates. Chapter 7 describes statistical methods used for analyzing the simplest comparative experiments, those involving just two groups. Chapter 8 discusses more complex experiments and issues related to their design. The next two chapters deal with multivariate methods: Chapter 9 discusses unsupervised classification methods and Chapter 10 discusses supervised classification methods. Chapter 11 describes protein arrays. A typical protein array experiment is outlined and methodology for analyzing protein array data is described.

The website `http://www.rci.rutgers.edu/~cabrera/DNAMR` will function as a companion to this book. It contains color versions of the figures, software, updates and any amendments related to the book.

Exploration and Analysis of DNA Microarray and Protein Array Data
Edited by DHAMMIKA AMARATUNGA and JAVIER CABRERA

CHAPTER 2

Genomics Basics

It is useful to review the basic concepts of modern molecular biology before fully immersing ourselves in the world of microarrays. We are sure that the reader who has had limited exposure to this fast-developing field will appreciate this review; others may skip ahead. Genomics is a fascinating subject; after all, it is the story of life, and can occupy a multi-volume book just by itself. In the interest of space, of course, it is necessary that we confine our discussion to those topics that are essential to an understanding of the science underlying microarrays, leaving other topics for interested readers to explore on their own. Some excellent general references that we, not being trained as molecular biologists ourselves, have found useful are listed at the end of the chapter.

2.1 GENES

From ancient times it was suspected that there existed some sort of a hereditary mechanism that carried information from parent to child. It is because of this mechanism that family members tend to exhibit similar characteristics or *traits*. For example, they tend to resemble each other in terms of appearance and physical characteristics such as skin color, they tend to be predisposed toward certain diseases such as diabetes, cancer, and heart disease, and so on. However, inheritance is clearly not a perfect copying process. For example, a child of brown-eyed parents could turn out to be blue-eyed. Despite the efforts over the years of many leading scientists and thinkers to understand the hereditary mechanism, its precise nature remained an intriguing mystery until quite recently.

Following centuries of speculation and research, the existence of discrete hereditary units, which we now call *genes*, has been firmly established. Each gene, either by itself or in combination with some other genes, provides a clear

Exploration and Analysis of DNA Microarray and Protein Array Data
By Dhammika Amaratunga and Javier Cabrera
ISBN 0-471-27398-8

and unambiguous set of instructions for producing some property of its organism. The complete set of genes in an organism, essentially the master blueprint for that organism, is referred to as its *genome*. This blueprint contains all the hereditary instructions for building, operating, and maintaining the organism, and for passing life in like form on to the next generation of that organism.

Until the twentieth century, there was hardly any concrete information as to what genes were and how they operated. Then a panoply of innovative research work and pathbreaking discoveries over (roughly) the first half of the twentieth century gave genes a chemical (molecular) existence. This culminated in the pivotal realization that genes are made of *deoxyribonucleic acid* (DNA).

2.2 DNA

A DNA molecule consists of two long strands wound tightly around each other in a spiral structure known as a *double helix*. The structure has been likened to a twisted ladder, whose sides are made of sugar and phosphate and whose rungs are made of bases.

Each strand of the DNA molecule (i.e., each side of the ladder once it has been untwisted and straightened out) is a linear arrangement of repeating similar units called *nucleotides*. Every nucleotide has three components: a sugar (deoxyribose), a phosphate group, and a nitrogenous base. The base is one of: *adenine*, *thymine*, *guanine*, *cytosine* (A, T, G, C, for short). The bases on one strand are paired with the bases on the other strand according to the *complementary base pairing rules* (also called the *Watson–Crick base pairing rules*): adenine only pairs with thymine, guanine only pairs with cytosine. The pairs so formed are called *base pairs* (bp, for short); they form the coplanar rungs of the ladder. The force that holds a base pair together is a weak hydrogen bond. Although each individual bond is weak, their cumulative effect along the strands is strong enough to bind the two strands tightly together. As a result DNA is chemically inert and is a stable carrier of genetic information.

The sequences of bases along each of the two strands of DNA are complementary to each other as they are matched by the complementary base-pairing rules. This *complementary sequencing* has an important consequence. It was recognized from very early on that whatever the entity was that was a hereditary unit, it must be able to self-replicate so that information could be passed on from generation to generation. At the time that the structure of DNA was deduced, there was a lot of excitement, as it was clear that the complementary structure of the DNA molecule would allow every DNA molecule to create an exact replica of itself, thus fulfilling this requirement.

The *DNA replication* process is, in principal, quite straightforward. First, the DNA molecule unwinds and the "ladder" unzips, thereby disrupting the weak bonds between the base pairs and allowing the strands to separate. Then, each strand directs the synthesis of a brand new complementary strand, with free nucleotides matching up with their complementary bases onto each separated

strand, a process that produces two descendant DNA molecules. Each descendant consists of one old and one new DNA strand. The constraints imposed by the complementary base-pairing rules ensure that each new strand is an exact copy of the old one with the order of the bases along the strands being faithfully preserved.

The preservation of the base order is crucial. The particular order of the bases arranged along any one strand, its *DNA sequence*, is the mechanism that specifies the exact genetic instructions required to create the traits of a particular organism.

Many genes are located along each long DNA molecule. A gene is a specific contiguous subsequence of the DNA sequence whose A–T–G–C sequence is the *code* required for constructing a *protein*. Proteins are giant complex molecules made of chains of amino acids and it is they that are actually both the building blocks and the workhorses of life. Proteins also regulate most of life's day-to-day functions; even the DNA replication process is mediated by *enzymes*, proteins whose job is to catalyze biochemical reactions.

2.3 GENE EXPRESSION

An organism's DNA is located in its cells. *Cells* are the fundamental units of all living organisms, both structurally and functionally. A cell is a microscopic, yet extraordinarily complex, structure that contains a heterogeneous mix of substances essential to life.

There are many substructures within a cell. The most prominent one is a highly protected subcompartment called the *nucleus*, in which resides the organism's DNA. Enclosing the nucleus is the *nuclear membrane*, the protective wall that separates the nucleus from the rest of the cell, which is called its *cytoplasm*. The entire cell is enclosed by the *plasma membrane*. Embedded within this membrane is a variety of protein structures that act as *channels* and *pumps* to control the movement of molecules into and out of the cell.

The set of protein-coding instructions in the DNA sequence of a gene resembles a computer program. A computer program must first be compiled and executed in order for anything to happen. In much the same way a gene must be *expressed* in order for anything to happen. A gene expresses by transferring its coded information into proteins that dwell in the cytoplasm, a process called *gene expression*.

The transmission of genetic information from DNA to protein during gene expression is formulated by the *central dogma of molecular biology*, which can be stated in oversimple terms as DNA → mRNA → protein. This postulates that the protein-coding instructions from a gene are transmitted indirectly through *messenger ribonucleic acid* (mRNA), a transient intermediary molecule that resembles a single strand of DNA. There are a few differences between mRNA and DNA, three being that mRNA is single-stranded, its sugar is ribose, and it has the base *uracil* (U) rather than the base thymine.

When a gene is expressed, the DNA double helix splits open along its length. One strand of the open helix remains inactive, while the other strand acts as a template against which a complementary strand of mRNA forms (a process called *transcription*). The sequence of bases along the mRNA strand is identical to the sequence of bases along the inactive DNA strand (except that mRNA has uracil where DNA has thymine). The mRNA strand then separates from the DNA strand and transports out of the nucleus, across the nuclear membrane, and into the cellular cytoplasm. There it serves as the template for protein synthesis, with consecutive (nonoverlapping) triplets of bases (called *codons*) acting as a code to specify the particular amino acids that make up an individual protein. The sequence of bases along the mRNA is thus converted into a string of amino acids that constitutes the protein molecule for which it codes (a process called *translation*).

Each possible triplet of mRNA bases codes for a specific amino acid, one of the 20 amino acids that make up proteins. For example, GCC codes for alanine, CAC for histidine, AUC for isoleucine, and GAG for glutamic acid—the complete list is referred to as the *genetic code*. As there are four possible bases, there are $4^3 = 64$ possible triplets, but only 20 possible amino acids. This means that there is room for redundancy: for example, GCU, GCC, GCA, and GCG all code for alanine. This redundancy is a valuable feature of the genetic code as it provides a safeguard against small errors that might occur during transcription.

In addition the genetic code has specific triplets to signal the start and the end of a coding sequence. The *start codon*, AUG, is the triplet of mRNA bases that signals the initiation of a sequence that is to be translated, while the *stop codon* is a triplet of mRNA bases—UGA, UAG, or UAA—that signals the termination of a coding sequence. The sequence of mRNA bases in between and including these two is called an *open reading frame* (ORF). All sequence information of coding interest lies in ORFs (but not every ORF codes for a gene). Since the codes for the start and stop codons are known, given an mRNA sequence, it is a simple matter to read off all of its ORFs.

Scientists involved in gene expression research usually find it easier to work with *expressed sequence tags* (ESTs) instead of the whole gene. An EST is a unique short subsequence (only a few hundred base pairs long), generated from the DNA sequence of a gene, that acts as a "tag" or "marker" for the gene. An advantage of ESTs are that they can be back-translated into genetic code that is coded for or expressed as an exon as opposed to an intron or other noncoding DNA. A short (typically 5 to 50 bp long) fragment of single-stranded DNA, not necessarily associated with a gene, is called an *oligonucleotide* (oligo for short).

Although every cell in an organism has a copy of the exact same genome (more or less), not all cells express the same genes, which is why different cells perform different functions. For instance, genes that are expressed in a brain cell may not be expressed in a stomach cell. In addition, even within the same cell, different genes will be expressed at different times, and perhaps at different

levels, depending on the phase of the cell and perhaps as a response to different stimuli. There are, however, a few exceptions: these are genes, called *housekeeping genes*, which are in constant use to maintain basic cell functions.

2.4 HYBRIDIZATION ASSAYS AND OTHER LABORATORY TECHNIQUES

Two single-stranded DNA molecules whose sequences are complementary to each other will exhibit a tendency to bind together to form a single double-stranded DNA molecule, a process known as *hybridization*. Two DNA strands (or one DNA strand and one mRNA strand) will hybridize with each other, regardless of whether they originated from a single source or from two different sources, as long as their base pair sequences match according to the complementary base-pairing rules. Even when the sequences on the two strands do not match perfectly, as long as there is sufficient similarity, it is likely that some base pairing will occur and that a hybrid DNA molecule will be formed.

The tendency of DNA strands of complementary sequence to hybridize preferentially is exploited in *hybridization assays*. In these assays a *probe* consisting of a homogenous sample of single-stranded DNA molecules, whose sequence is known, is prepared and labeled with a reporter chemical, usually a radioactive or fluorescent substance. An immobilized *target*, usually a heterogeneous mixture of single-stranded DNA molecules of unknown composition is challenged by the probe. As the probe will hybridize only to sequences complementary to its sequence, DNA sequences in the target that are complementary to the probe DNA sequence can be identified by the presence of reporter molecules.

This concept is applied in *blotting* techniques. In *Southern blotting* the target DNA is separated by electrophoresis (see below) and transferred onto a filter, where it is exposed to the probe. *Northern blotting* is a variant in which the target is mRNA instead of DNA. As mRNA is the intermediary molecule in gene expression, Northern blotting provides a means of studying the expression patterns of specific genes. DNA microarrays can be regarded as a massively parallel version of Northern blotting.

In *in situ hybridization*, denatured DNA (DNA in which the two strands are unwound and separated) is kept in place in the cell and is then challenged with mRNA or DNA extracted from another source and labeled with a reporter chemical, usually a fluorescent substance. By retaining the DNA in the cell, the specific chromosome containing the DNA sequence of interest can be identified by observing, under a microscope, the location of the fluorescence.

Besides hybridization assays there are several laboratory techniques that have had, and continue to have, an enormous impact on progress in genomics research. Since they play an important role in microarray experiments, we will outline them briefly.

Electrophoresis is a method of using an electric field to separate large molecules, such as DNA, RNA, and proteins, from a mixture of similar molecules.

An electric current is passed through a porous medium containing the mixture, usually a gel. The different kinds of molecules separate as different molecules will travel through the medium at different rates, depending on their electrical charge and size (e.g., small molecules typically travel farther through the medium than large molecules).

Cloning is the process of using specialized DNA technology to produce multiple exact copies of a single gene or other segment of DNA to obtain enough material for further study. These clones can be grown in bacteria to produce multiple copies and large amounts of a given DNA molecule. The resulting cloned collections of DNA molecules are called *clone libraries.*

Polymerase chain reaction (PCR) is a rapid and versatile procedure for generating multiple copies of (i.e., for *amplifying*) virtually any fragment of DNA. The number of copies is limited only by rate-limiting factors such as the number of cycles and the amount of enzymes, bases, and other reagents required.

PCR is a cyclic process that involves repeating three basic steps over and over. The three basic steps are as follows: First, the two strands of the target DNA are unwound and separated by heating (a process called *denaturing*). Next, primers, short strands of single-stranded DNA that match the sequences at either end of the target DNA, are bound to their complementary bases on the now single-stranded DNA in a process called *annealing*. Finally, DNA is synthesized by a *polymerase*, an enzyme that is present in all organisms and whose job is to copy and, where necessary, repair genetic material. Starting from the primer, the polymerase reads a template strand and matches it with free complementary bases. This produces two descendant DNA strands, each of which consists of one old and one new DNA strand. As in DNA replication, the complementary base-pairing rules ensure that each new strand is an exact copy of the old one. Cycling through these three basic steps over and over generates more and more copies of the target DNA. The amount of DNA grows exponentially as it doubles with every cycle. Since each cycle takes only a few minutes, a laboratory scientist can generate millions of copies of the target DNA in less than an hour. For this reason and because of its specificity, its versatility, and its easy automatability, PCR has had a major impact on molecular biology and many related sciences in less than two decades.

Reverse transcription is a procedure for reversing, in a laboratory, the process of transcription. It is accomplished by isolating mRNA, which is unstable and subject to degradation, and using it as a template to synthesize a *complementary DNA* (cDNA) strand, which is stable and is not easily degraded. cDNA is so called because its sequence is complementary to the original mRNA sequence. This process utilizes the enzyme *reverse transcriptase.* The resultant single-stranded cDNA molecule is considerably shorter than the parent DNA sequence, as it will have only its coding exon sequences; the noncoding intron sequences would have been excised during the formation of the original mRNA. Incidentally, as far as is known, the process of translation cannot be reversed.

The cDNA generated by reverse transcription can, if needed, be amplified

by PCR. The process is then called *reverse transcriptase polymerase chain reaction* (RT-PCR). RT-PCR is the one of the most sensitive techniques for detecting and quantifying target mRNA sequences. Among other uses, RT-PCR can be utilized to provide information on gene expression.

2.5 THE HUMAN GENOME

A few words now about our own genome. DNA in the human genome is made up of roughly three billion base pairs and is partitioned into 46 molecules, each of which resides in a threadlike cellular structure called a *chromosome*. Chromosomes come in pairs (except for the sex chromosomes): one of these is one of the father's two corresponding chromosomes, the other is one of the mother's two corresponding chromosomes. The two members of a pair of chromosomes are called *homologous chromosomes*.

Chromosomes range in length from about 50 to 250 million bp. Each chromosome contains many genes. In total, the human genome is estimated to contain somewhere around 40,000 genes. Genes vary widely in length, from a few hundred bp to several thousand bp. Only a tiny percentage of human DNA includes *exons*, the protein-coding sequences of genes. Interspersed within many genes are *introns*, sequences that have no coding function and that are excised during transcription. In between many genes are other noncoding regions whose functions remain largely obscure.

Every single human being has almost the exact same genome. In fact, at the genome level, we are 99.9% identical! However, genomes do vary slightly from person to person, a phenomenon known as *genome variation* (or *genetic variation*). It is this subtle variability in our genomes that is responsible for the evolution and diversity of the human race. Some genome variations are unique to a person, while others are passed on generation through generation via reproductive cells.

The existence of genome variation means that some genes will differ slightly from person to person. When this happens, each alternate version of a gene is called an *allele*. In fact every person carries two alleles of each gene, one in each of a pair of homologous chromosomes. When both alleles are the same, the person is said to be *homozygous* for that gene; otherwise, the person is said to be *heterozygous* for that gene. In the latter case only one of the alleles (called the *dominant allele*) may be expressed, the other one (called the *recessive allele*) may not be. The presence of two versions of each gene is another protective mechanism provided by nature; if one copy should happen to be defective, the other copy is there to compensate.

Besides physical characteristics, a familiar example of genome variation is blood type. We are all of us classified as being A, B, AB, or O. The ABO gene that controls the blood group has three alleles, which are designated as A, B, and O. All three alleles have generally the same DNA sequence except for dif-

ferences at a few nucleotides. Alleles A and B, which code for proteins A and B, respectively, are co-dominant. Everyone is assigned a blood type according to which two alleles of the ABO protein he or she is carrying. Anyone who has AA or AO (and therefore has protein A only) is said to have blood type A. Anyone who has BB or BO (and therefore has protein B only) is said to have blood type B. Anyone who has AB (and therefore has both proteins) is said to have blood type AB. Anyone who has OO (and therefore has neither protein) is said to have blood type O.

Since everyone has almost the exact same genome and any person-to-person genome variation is relatively minor, it is reasonable to try to establish a *consensus* human genome sequence; in other words, to *sequence* the entire human genome. This is exactly the stated goal of the much-publicized massive international undertaking known as the Human Genome Project. A near-complete catalog of the human genome is now available and a complete catalog is only a few years away.

2.6 GENOME VARIATIONS AND THEIR CONSEQUENCES

Most genome variations are small and simple, and involve only a few bases—for example, one person might have a G where another has a C, or one person might be missing a T that another person has, and so on. Such genome variations are due to *mutations* and *polymorphisms*, alterations in a DNA sequence. Some common alterations are one base being replaced by another (*substitution*), a base being excised (*deletion*), a base being added (*insertion*), a small subsequence of bases being removed and then reinserted in the opposite direction (*inversion*), and a small subsequence of bases being removed and then reinserted in a different place (*translocation*).

A genome variation may be inherited or acquired. An inherited genome variation is present in the DNA of almost all of the organism's cells and could be passed on to the next generation of that organism. Acquired genome variations are mutations that occur spontaneously during DNA replication or are caused by an external environmental factor such as exposure to a toxic substance. Such variations will only be present in the DNA of the affected cells and their direct descendants. Thus an acquired mutation will be passed on to the next generation of that organism only if it affects a reproductive cell, in which case a new line of hereditary gene mutation would be initiated.

In practice, the terms "mutation" and "polymorphism" tend to be used interchangeably, but technically a polymorphism is a genome variation in which every possible sequence is present in at least 1 percent of people, whereas a mutation refers a genome variation that is present in less than 1 percent of people. Thus a location in a DNA sequence where 95 percent of people have an A and 5 percent have a T is a polymorphism, while a T in a location in a DNA sequence where 99.5 percent of people have an A and only 0.5 percent have a

T is a mutation. The common and properly functioning version of a gene is referred to as its *wild-type allele*; a version with a mutation is called a *mutant allele*.

Many genome variations do not produce any noticeable effects, even at the cellular level. An obvious way for this to happen is for a variation to occur outside the genome's coding regions. What may be somewhat surprising is that it can happen even when a variation occurs within a coding region. This is because of the redundancies in the genetic code that allow the same protein to be produced from two slightly different sequences. In addition, if that is not enough, cells have mechanisms that are capable of repairing certain types of damaged DNA.

A small percentage of genome variations do produce noticeable effects, some deleterious, some beneficial. This is the genetic basis of biological diversity and the evolutionary process.

Many polymorphisms that produce noticeable effects are, in general, harmless; if not, they would not survive the natural selection process. However, this is not a hard and fast rule. For example, people with blood type O are more susceptible to peptic ulcers and cholera than others, yet the trait did not die out (in fact, almost half the world's population has this blood type), perhaps because they also are less susceptible to malaria and certain types of cancer.

Certain mutations can be harmful with no obvious beneficial features. They could either cause a disease or increase a person's susceptibility to a disease or even lead to death. For example, mutations in the p53 gene, which, in its wild type, codes for a protein that suppresses abnormal cell proliferation, may cause it to lose its ability to block abnormal cell growth, leading to cells dividing uncontrollably and forming tumors. Not surprisingly, mutations in the p53 gene have been found to be strongly associated with cancer.

It has been conjectured that most human genome variation may be attributable to *single nucleotide polymorphisms* (SNPs), polymorphisms that involve just one nucleotide. Blood grouping is an example: the only difference between the genes for blood types A and O is that the gene for the former has a G base that has been deleted in the gene for the latter. SNPs are frequent in our genomes: it has been estimated that, on average, about one in every one thousand nucleotides is a SNP.

Many scientists believe that SNPs underlie the susceptibility of certain people to certain diseases. An often-cited example is the association between the apolipoprotein E gene (ApoE, for short) and Alzheimer's disease. ApoE has three alleles (called E2, E3, E4), each of which differs from any other by a SNP (there are two SNPs in all). It appears that those who have at least one copy of the E4 allele have a greater risk of developing Alzheimer's disease (and earlier on in life), whereas those who have at least one copy of the E2 allele have a lesser risk of developing Alzheimer's disease.

Given a specific DNA sequence, there are, in theory, a huge number of possible combinations of SNPs. However, SNPs are not randomly scattered along a chromosome. Instead, many of them occur in groups, called *haplotypes*, and

relatively few of the countless number of theoretically possible haplotypes are observed with any significant frequency. The SNPs defining a haplotype tend to be inherited together over generations and serve as more reliable genetic markers for diseases and other traits than any of the individual SNPs.

As research progresses, the genomic basis of health and disease is being better and better understood. Clearly, the central theme of this effort is the better elucidation of genotype-phenotype relationships, such as the association between ApoE and Alzheimer's disease. *Genotype* refers to the genetic makeup of an individual. *Phenotype* refers to the outward characteristics of the individual. They are, naturally, connected, as the phenotype essentially results, functions, and develops based on the information provided by and encoded in the genotype. Despite this, the association between the genotype and the phenotype is by no means perfect. Environmental effects and other external factors tend to appreciably modify the actual manifestation. Statistical procedures that measure association play a significant role in analyzing these complex relationships.

2.7 GENOMICS

Genomics is the branch of biology that studies the structure and function of genes. Much progress has been made in the area of *structural genomics*. Structural genomics refers to the application of sequencing technologies to establish representative genome sequences for different organisms, particularly humans. Nowadays the term is increasingly being used to also refer to methods for determining protein structures as a primary tool for discovering the biological functions of genes and proteins and their interrelationships.

The other key area is *functional genomics*, which, as its name implies, is the study of the functions of genes. It seeks to understand the behavior of all the genes in a genome (for all genomes). It is important to realize that just knowing the sequence of a gene does not imply that its function is also known. In addition genes do not function in isolation. Instead, genes (and proteins) operate collectively in *pathways*, as coordinated sequences of genetic and molecular activities. Such pathways underlie all cellular processes. Therefore studying each gene as a separate discrete entity tells only part of a story, like a still from a film. On top of that, a plethora of external factors can alter or disrupt a pathway. This constant interplay between genes, proteins, and external factors makes functional genomics a complex subject, one that was almost intractable until technologies, such as microarrays, emerged that allowed large numbers of molecular entities (perhaps even entire genomes) to be studied simultaneously.

Among the important questions in functional genomics are:

- Which genes are expressed in which tissues?
- How is the expression of a gene affected by extracellular influences?
- Which genes are expressed during the development of an organism?

- How does gene expression change during development and differentiation?
- What is the effect of misregulated expression of a gene?
- What patterns of gene expression cause a disease or lead to disease progression?
- What patterns of gene expression influence response to treatment?

Over the last decade a great deal of progress was made in all of the various branches of genomics, and it is likely that this trend will continue for decades to come and benefit medicine, agriculture, and everyday life.

2.8 THE ROLE OF GENOMICS IN PHARMACEUTICAL RESEARCH

The immediate benefits of the progress in genomics will be seen in the discovery and development of novel pharmaceutical products (Lennon, 2000). For example, much can be learned from studying general genotype-phenotype relationships and how these, in turn, affect drug response. This is the key aspect of *pharmacogenomics*, the study of pharmacologically relevant genes. Research in pharmacogenomics attempts to elucidate how these genes manifest their variations, how these variations interact to produce phenotypes, and how these phenotypes and environmental factors combine to affect drug response.

That genome variation does contribute to different individuals experiencing different pharmacological and toxicological reactions to medication has been amply demonstrated. For example, variations in the CYP2D6 gene, which codes for an enzyme involved with the metabolism of many commonly prescribed drugs, including analgesics, antiarrhythmics, beta-blockers, neuroleptics, and antidepressants, have been found to seriously affect the therapeutic response to these drugs. Severe adverse drug reactions have also been associated with these variations. The wild-type allele of this gene is referred to as CYP2D6*1. Two variant alleles are CYP2D6*3 and CYP2D6*4. Both are due to SNPs: the CYP2D6*3 polymorphism is a deletion; the CYP2D6*4 polymorphism is a substitution. Both truncate the protein that they code for, which results in functional CYP2D6 protein being absent. Those who have inherited two copies of variant alleles in any combination are likely to be poor metabolizers. Drugs, like codeine, which need CYP2D6 for activation, will not be effective in these patients. Other drugs, like lidocaine, are known to cause serious side effects, even heart failure, in these patients. On the other hand, those who have one wild-type allele and one polymorphic allele are likely to be fast metabolizers, in whom the drugs are ineffective or unsafe.

The fact that it is now possible to gather this sort of knowledge has led to the hype that the ultimate goal of pharmacogenomics will be "personalized medicine," the ability to target a drug specifically to a patient based on his or her genotype, so that he or she will have maximal response with maximal safety. Needless to say, if pharmacogenomics ever lives up to this promise,

medicine would be revolutionized, as most currently available drugs are fully effective in only about half the patients to whom they are prescribed, and moreover a subset of these patients will experience undesirable side effects. Still, personalized medicine is a long way off, and to be realistic, it is uncertain as to whether it is even possible that given environmental factors, diet, age, lifestyle, life history, and state of health, all have the potential to influence an individual's response to medication.

Thus the true long-term promise that pharmacogenomics offers is likely to be the ability to stratify patients and diseases based on genotype and to develop better strategies for therapy and prevention based on these stratifications. An example of a potential genotype-based therapy is pravastatin, which appears to be more effective in lowering cholesterol levels in people with the B1B1 variant of the CETP gene than in other people. An example of potential genotype-based prevention is tamoxifen, which appears to prevent breast cancer among women with BRCA1 and BRCA2 gene mutations.

Clearly, such knowledge is useful for the development of novel pharmaceutical products. Therefore it is hardly surprising that the pharmaceutical industry has embraced genomics and greatly expanded their investment in genomics-related research. The greatest impact has been on the drug discovery process. Genomics has begun to play a pivotal role in drug discovery particularly through pharmacogenomics and through improving the processes of *drug target identification* and *drug target validation*.

A *drug target* is typically a protein that is intimately associated with a disease process and that is the intended site of drug activity. For example, the protein, immunoglobulin or IgE, is a target for allergy, it having been established that the allergy response is mediated by it. Information obtained from studying correlations between genome variations and disease information and from studying correlations between gene expression differences and disease information can be used to identify target molecules that directly underlie the disease processes themselves, rather than just the symptoms. Statistical methods play a significant role in this endeavor.

Once a target has been identified, it must be *validated* to prove that inhibiting the target has the desired pharmacological effect. Gene expression studies can be used to validate a target by demonstrating that target genes are indeed expressed differently in different disease states. A more complex validation approach is to make a *knockout mouse* (a mouse lacking the gene that produces the target) and check whether it shows the desired behavior. For example, an IgE knockout mouse exhibits no allergic reactions, validating IgE as a target. Once an identified target has been adequately validated, an assay can be developed to screen a number of chemicals, perhaps in a high throughput mode, for potential activity with it.

Protein research can also contribute to better *drug design*. Drugs generally work by binding with a target protein at a particular site on the protein, thereby inhibiting its normal function. If the structure of the target protein were

known, it may be possible to construct a drug specifically to interact with it, for example, by using a technology such as X-ray crystallography to examine a three-dimensional protein structure and then designing a small molecule that will be able to fit and bind into the pockets of the structure.

The more specific a target site is the better. A drug's toxic side effects usually stem from *nonselectivity*, the affinity of the drug to more than just the intended site of activity. Drugs that aim specifically for the molecular differences between diseased and normal cells are likely to be less toxic and, therefore, more useful clinically.

These are only some of the ways genomics and associated sciences can contribute to pharmaceutical research.

2.9 PROTEINS

All living organisms are composed largely of proteins. Proteins perform and regulate most of life's basic functions. Thus *structural proteins* form part of a cellular structure, *enzymes* catalyze almost all the biochemical reactions occurring within a cell, *regulatory proteins* control the expression of genes or the activity of other proteins, and *transport proteins* carry other molecules across membranes or around the body.

Structurally proteins are giant complex chains of amino acids. A protein's sequence of amino acids is determined by the DNA sequence of the gene that produced it. Proteins belong to a class of large compounds that are called polypeptides as the amino acids that comprise them are held together by peptide bonds. Polypeptide chains, in general, and protein chains, in particular, have a tendency to fold up into complex three-dimensional structures. A protein's particular function in the cell is determined not only by its amino acid sequence but also by the specific structure into which it folds. In addition, it is likely to be affected by other proteins present in the same cell at the same time. Thus proteins are much harder to study than genes.

Interestingly there are far more proteins than there are genes. This is partly due to *post-translational modifications* (proteins, once synthesized at the translation step of gene expression, are subject to a multitude of modifications) and partly due to *alternative splicing* (different ways of splicing the exons together after they are separated during transcription produces different mRNA sequences and thereby different proteins).

The multitude of all proteins generated by a genome of an organism is called its *proteome*, and the study of protein structure and behavior, which is getting more and more attention, is called *proteomics*. Proteomics encompasses the identification of proteins in tissues, the characterization of their physicochemical properties (e.g., their sequences and post-translational modifications), and the description of their behavior (e.g., what functions they perform and how they interact with one another and their environment).

2.10 BIOINFORMATICS

As stated in Chapter 1, an inevitable consequence of the modern technology-driven research effort in genomics and genomics-related sciences is a steadily growing mountain of data, which is neither easy to examine nor straightforward to understand. Given the sequence of the human genome, for instance, it is already a colossal task just to identify the individual genes. Ascertaining the function of the many thousands of genes and proteins identified and determining how in this constellation genes and proteins interact among themselves (and under what circumstances) is a mind-boggling task that will challenge those working in this area for many years to come. Issues lie in data storage, in the querying and analysis of this data, in effective communication of these results, and in organizing them to infer functional relationships.

The steady influx of genomics information has spawned a new discipline called *bioinformatics* that has become an integral part of genomics research. In bioinformatics, scientists in the biological and computational sciences, together with significant contributions from other disciplines, collaborate to provide insight into biological processes. Statistics is an essential component of many of these activities. As a fledgling discipline, bioinformatics does not yet have a well-defined charter, but some common bioinformatics activities are given below.

Creation and Maintenance of Databases. As a first step, the magnitude and complexity of the data being collected has led to the creation of large relational databases to store, organize, and index such data. At the moment DNA sequences (and protein sequences derived from them) comprise the majority of such catalogs. Some well-known examples are GenBank (a database that contains the totality of public DNA and protein sequence data), SWISS-PROT (a protein sequence database), and PDB (a database of three-dimensional biological macromolecular structure data).

Analysis of Sequence Information. In parallel with the development of large sequence databases, specialized tools (e.g., BLAST) are being devised to efficiently search, view, and analyze the data in these databases. This includes the development of methods for finding the genes in the DNA sequences of various organisms, clustering sequences into families of related sequences, aligning similar genes and proteins, and examining evolutionary relationships. Probability and statistical techniques, such as hidden Markov models, can efficiently and automatically build representations of related sequences. They form the basis of several of the more sensitive database searching tools. Statistical methodology can also be brought into play to assess the significance of any match found.

Prediction of Three-Dimensional Structure. Knowledge of physics and chemistry, and information gathered from similar molecules, is being used to deduce the three-dimensional structure of proteins and other large molecules.

Expression Analysis. Pattern analysis of gene expression data (mostly obtained from DNA microarrays) using statistical and data mining tools is a major effort in bioinformatics.

Modeling Dynamic Life Processes. The ultimate challenge in bioinformatics is to develop ways of putting together the information gathered from all the diverse areas of research in order to understand fundamental life processes.

SUPPLEMENTARY READING

The book by Gonick and Wheelis (1991) is an excellent introduction to genetics presented in an amusing and informal style. The book by Clark and Russell (1997) provides a more in-depth introduction. More detailed treatment can be found in the molecular biology textbooks by Alberts et al. (1994) and Strachan and Read (1999).

Vingron (2001) argues the importance of applying statistical thinking to bioinformatics. The book by Ewens and Grant (2001) offers an introduction to statistical methods employed in bioinformatics.

EXERCISES

2.1. What are the complementary base-pairing rules? Describe their role in **(a)** DNA replication, **(b)** gene expression, **(c)** hybridization assays, **(d)** polymerase chain reaction.

2.2. Explain the function of **(a)** DNA, **(b)** mRNA, **(c)** start and stop codons in protein synthesis.

2.3. Explain how a child of parents, both of whom are blood type A, could be blood type O.

2.4. Discuss some ways in which developments in genomics could alter the practice of medicine.

Exploration and Analysis of DNA Microarray and Protein Array Data
Edited by DHAMMIKA AMARATUNGA and JAVIER CABRERA

CHAPTER 3

Microarrays

The state of a cell at any given time is governed by which subset of its genes is expressed at that time. Recall that according to the central dogma of molecular biology (Section 2.3), the first step in gene expression is transcription, in which expressed DNA sequences are transcribed into mRNA. Thus it is reasonable to conjecture that from knowledge of what mRNAs are present in the cell and in what quantities, a scientist can make some inferences regarding the state of that particular cell. This line of reasoning has led to a considerable effort to measure and compare the levels of mRNA in cells in various states. The complete collection of mRNAs (including their alternative splicing variants) is referred to as the organism's *transcriptome*.

It could be argued that it is more pertinent to study the end products of gene expression, the proteins, rather than mRNA, which is an intermediate molecule. After all, it is these proteins that are responsible for most biological activities in the body. However, the function of a protein is determined not only by its amino acid sequence but also by the specific structure it folds up into. Furthermore proteins are difficult to purify. Thus an added inducement for working with mRNA levels, in order to investigate cell state, is that unlike proteins, they are relatively simple to study with current technology, even in a high-throughput mode.

The *DNA microarray* (*microarray* or *array*, for short) has now become the most widely used technology for studying mRNA levels. DNA microarrays were developed as a general means of monitoring the expression patterns (or more precisely, the transcription patterns) of large numbers of genes (perhaps even entire genomes) at once, thereby bringing about a tremendous improvement over the tedious "one gene per experiment" paradigm that prevailed until then.

In brief, a typical DNA microarray experiment proceeds as follows: Take a small glass slide. Suppose that the surface of the slide has been divided into

Exploration and Analysis of DNA Microarray and Protein Array Data
By Dhammika Amaratunga and Javier Cabrera
ISBN 0-471-27398-8

series of imaginary square cells to form a rectangular grid. Onto each square cell, stick a tiny amount of liquid that contains DNA corresponding to a gene of known sequence. Different cells will have different genes. Separately prepare a solution that contains a mixture of mRNAs whose sequences are unknown. Add to this solution a substance that fluoresces when excited by light. Pour the solution onto the slide. The mRNA molecules will diffuse over the slide and, wherever they find a matching (i.e., complementary) DNA sequence, such as the one taken from the gene from which the mRNA was transcribed, they will hybridize to each other and the solution will stick to the slide. Without a match the solution will not stick to the slide and can be washed away. Use a laser scanner to detect and measure the fluorescent signal being emitted at each cell.

In a comparative microarray experiment, different slides containing the same set of genes will be exposed to different mRNA samples. By comparing the intensity levels of the fluorescent signals across the multiple mRNA samples, a scientist will be able to understand how the expression profile of a set of genes differs across the different mRNA samples.

3.1 TYPES OF MICROARRAY EXPERIMENTS

The simple microarray idea has enormous potential. Microarrays have already been heavily used in biological research to address a wide variety of questions. To motivate our subsequent discussion, we begin by presenting a few examples of some such research. We emphasize that this discussion is by no means exhaustive and in fact represents only a fraction of the types of experiments that a scientist could envision addressing with this technology.

3.1.1 Experiment Type 1: Tissue-Specific Gene Expression

Cells from different tissues perform different functions. Although it is a simple matter to distinguish cells from different tissues by their phenotypes, the details of precisely why cells from one tissue behave differently from cells from another tissue remains a fertile topic for research. Since it is the individual proteins, particularly enzymes, within each cell that control all the various intermeshing biochemical reactions within that cell, a cell's functions are determined by which proteins are produced by the cell, and this in turn depends on which genes are expressed by the cell. Microarray experiments can be used to identify which genes are preferentially expressed in which tissues. This would enable scientists to gain valuable insight into the mechanisms that govern the functioning of genes and cells.

3.1.2 Experiment Type 2: Developmental Genetics

The genes in an organism's genome express differently at different stages of its developmental process. Interestingly it has been found that there is a subset of

genes involved in early development that is used and reused at different stages in the development of the organism, generally in different order in different tissues, with each tissue having its own combination. Crucial to these processes are growth factors; the same growth factors that can, later in an organism's development, be involved in causing or promoting cancer; these genes are known as *proto-oncogenes*. Microarrays can, in principle, be used to track the changes in the organism's gene expression profile, tissue by tissue, over the series of stages of the developmental process, beginning with the embryo and up to the adult.

Supplementary applications of this line of research include deducing evolutionary relationships among species and assessing the impact of environmental changes on the developmental process of an organism.

3.1.3 Experiment Type 3: Genetic Diseases

There are many diseases called *genetic diseases* that are the result of mutations in a gene or a set of genes. A gene that is thus altered is called a *mutant gene*. The result can be a disease as these genes express inappropriately or do not express at all. Cancer, for example, could occur when certain regulatory genes, such as the p53 tumor suppressor gene, are deleted, inactivated, or become constitutively active (i.e., become always transcribed, regardless of any regulatory factors).

Microarray experiments can be used to identify which genes are differentially expressed in diseased cells versus normal cells. This would enable scientists to identify genes associated with the disease process, such as the tumor suppressor genes and the oncogenes (i.e., normal cellular genes that, when inappropriately expressed or mutated, can transform normal cells into tumor cells) associated with the onset of cancer and the genes associated with the development of a cancer form a low-grade malignancy through to a high-grade malignancy. This would enable the development of drugs aimed directly at the difference between diseased and normal cells. Such drugs can be designed to specifically target a particular gene, protein, or signaling cascade, and they are therefore less likely to cause undesirable side effects. One way in which this knowledge would be useful is in the development of target assays for screening new compounds in high-throughput mode to assess their potential efficacy as treatments for the disease.

There are certain diseases that have subtypes that are clinically indistinguishable but are genetically heterogeneous. As they are different subtypes, it is most likely that they will call for different treatments. A case in point is acute lymphoblastic leukemia and acute myeloid leukemia (ALL and AML, for short). It is crucial for proper therapy that a correct clinical diagnosis be quickly made. However this can be extremely difficult due to the clinical similarity of the two diseases. Microarray experiments can be used to identify which genes are differentially expressed in the two different types of cancer patients, thereby creating specific disease profiles by virtue of their gene expression pat-

terns. The information gleaned from these studies could lead to diagnostic procedures.

Sometimes such experiments also uncover disease subtypes that were not even known to exist. This would happen, for instance, if, during the course of studying a group of patients thought to be homogeneous, it is found that they exhibit two very distinct gene expression profiles, indicating two different disease subtypes, as may explain why some patients were responding well to treatment while the others were not.

Thus these types of experiments will afford scientists the capability of grouping diseases into classes. Eventually more precise, but less invasive, clinical diagnosis procedures could be developed.

3.1.4 Experiment Type 4: Complex Diseases

There are many diseases called *complex diseases* that are not caused by a few errors in genetic information but are caused instead by a combination of small genetic variations (polymorphisms) predisposing an individual to a serious problem. The risk of such an individual contracting a complex disease tends to be amplified by nongenetic factors, such as environmental influences, diet, and lifestyle. Coronary artery disease, multiple sclerosis, diabetes, and schizophrenia are complex diseases where the genetic makeup of the individual plays a major role in predisposing the individual to the disease. The genetic component of these diseases is responsible for their increased prevalence among certain groups, such as within families, within ethnic groups, within geographic regions, and within genders. Microarray experiments can be used to identify the genetic markers, usually a combination of SNPs, that may predispose an individual to a complex disease.

3.1.5 Experiment Type 5: Pharmacological Agents

Some genes alter their expression patterns when the organism is exposed to an external stimulus such as a pharmacological agent or a substance present in the environment. Microarray experiments can be used to identify genes that express differently in response to such exposure. The information obtained from such experiments will be useful for target identification and target validation.

The simplest such experiment is one in which a sample of cells is exposed to the pharmacological agent and permitted to reach a steady state of transcription. The mRNA levels in the treated cells can then be compared to those in a control sample.

A potentially more informative experiment would be a *temporal study*. A temporal study is an experiment in which a sample of cells is exposed to the pharmacological agent and subsamples of the cell sample are drawn at successive points in time. This allows the scientist to monitor the gradual change in gene expression profiles from the old steady state through to the new steady state. Such temporal studies provide information not only on which genes

undergo expression profile changes but also the order in which these changes occur.

Microarrays are also useful as a means of assessing toxicity that evokes changes in gene expression. A toxicologist would expose cells or tissues or a few animals to a class of chemicals whose toxicity is known and, from this, establish a *signature*, a common set of changes in gene expression produced by this class of toxic agents. Then another set of cells or tissues or animals is exposed to a chemical whose toxicity is unknown and the results matched against the signature. From this, the toxicologists should be able to make a prediction regarding the potential toxicity of that chemical. If successful, this procedure could be automated to allow for high-throughput toxicity screening of new molecular entities and should reduce the need for lengthy, expensive, and unpleasant animal testing of potential drugs.

3.1.6 Experiment Type 6: Plant Breeding

For centuries, researchers in plant breeding have been trying, with some success, to improve cultivated plant species and their products. For example, given that crops are heavily influenced by the environmental conditions to which they are continuously exposed, researchers in plant breeding have attempted to induce greater tolerance for environmental stressors such as extreme weather conditions. Some other goals of plant breeding are to boost the resistance of plants to infections, to reduce insect predation, to maximize the productivity of plants, to improve the quality of plant products, to increase the nutrition level of foods processed from plants, and to develop characteristics of plant products that are valued by consumers (e.g., fruits that stay ripe for long periods of time).

Microarray experiments can be used to identify the genes responsible for various traits of interest and to determine the conditions under which these traits are expressed. This information would enable scientists to create plant varieties with exact combinations of desirable traits.

3.1.7 Experiment Type 7: Environmental Monitoring

Environmental factors are known to affect gene expression, both as to whether or not a particular gene is expressed, and the degree to which it is expressed if, in fact, it is. Should a normal biological pathway be disrupted as a result of a gene expressing differently, the health of the affected organism could suffer. Thus it is important to assess the genome-level impact of exposure to environmental stressors, especially contamination of air, food, and water. Microarrays can be used to compare and contrast gene expression patterns across affected versus unaffected organisms, whether they be flora or fauna, taking into account natural effects such as seasonal fluctuations. One goal of these experiments is the characterization of environmental changes that may be a hazard to health.

Another goal of environmental monitoring is the detection of pathogens in food and water. This is generally done by examining the DNA in potentially contaminated samples, as each pathogen possesses a DNA sequence unique to it. The traditional approach tends to be a slow and laborious process, which is highly undesirable in situations where rapid intervention may be critical. Using microarrays, monitors can simultaneously and swiftly screen for several different strains of pathogens. To do this, a microarray containing the DNA of a number of different pathogens would be prepared, DNA would be extracted from an environmental sample, and this DNA would be applied to the microarray. If a pathogen is present in the sample, it will hybridize to the microarray and its presence would be detected. With this information, scientists can assess whether or not there is a hazard to health.

3.2 A VERY SIMPLE HYPOTHETICAL MICROARRAY EXPERIMENT

It is easiest to explain the principal behind microarray experiments with a very simple hypothetical example.

Suppose that we have obtained some cancerous liver tissue and some normal liver tissue from a liver cancer patient and that we want to know which genes are expressed differently in the two. We will begin by extracting mRNA from each tissue so that we have two mRNA samples. In each sample only mRNA corresponding to any genes that were expressed (i.e., transcribed) would be present. We will reverse transcribe the mRNA to cDNA and add some fluorescent dye to each sample. These two labeled samples are sometimes called *targets*. (Sometimes, however, they are called "probes" because they are used to probe the collection of spots on a microarray, but this usage appears to be now less standard—in order to avoid confusion, we will call them the *labeled samples*.)

Now suppose that we have prepared a DNA microarray containing the entire human genome (there is no such microarray as of yet, which is one reason why this example is hypothetical). Suppose that there are 36,000 genes. A DNA microarray for this experiment would be a tiny glass slide on which the 36,000 genes are printed in, say, a 300×120 rectangular array of spots, one gene per spot. Each gene printed on the microarray is called a *probe*. (In the confusion of terminology, they are sometimes called "targets.") Two such microarrays are prepared.

We will now flood one of the microarrays with the labeled sample from the cancerous tissue and flood the other microarray with the labeled sample from the normal tissue. We allow enough time for any cDNA in the samples to recognize and hybridize to its complementary sequence in the microarray. Once we are satisfied that this has happened, we will wash off any excess labeled sample from the microarrays and dry them.

Each spot on the microarrays where the labeled sample bound to the spot

would identify a gene that corresponds to some reverse transcribed mRNA in the sample. Such spots can be easily recognized, as they are the only ones that will fluoresce. In this way every spot on the microarray functions much like an independent assay for the presence of a particular mRNA.

We will then scan the microarrays and measure the intensity level of fluorescence at each spot. By comparing these intensities across the two microarrays, we will be able to tell which genes are differentially expressed in cancerous liver tissue versus normal liver tissue.

Example. Let X_g and Y_g denote the intensities measured for the gth gene in the normal liver tissue microarray and cancerous liver tissue microarray respectively, and let the ratio of these intensities be $R_g = Y_g/X_g$. This ratio is usually called the *fold change*. Figure 3.1*a* shows a scatterplot of Y_g versus X_g and Figure 3.1*b* shows a histogram of $\{R_g\}$. It is impossible to discern any

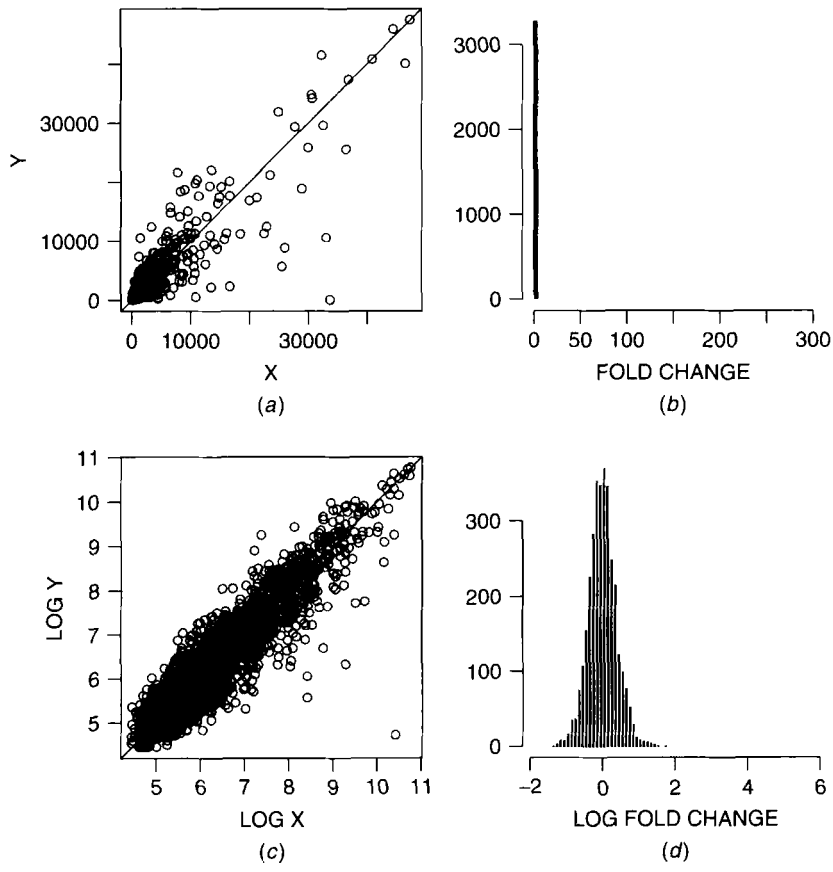


Figure 3.1 Comparing two samples by (*a*) scatterplot of intensities, (*b*) histogram of fold changes, (*c*) scatterplot of log intensities, (*d*) histogram of log fold changes.

structure in these graphical displays because the data is so heavily skewed. However, by taking logs, we are better able to see structure. Figure 3.1*c* shows a scatterplot of $\log(Y_g)$ versus $\log(X_g)$ and Figure 3.1*d* shows a histogram of $\{\log(R_g)\}$. In Figures 3.1*a* and 3.1*c*, most genes fall along the $Y = X$ line, indicating that they are expressed to the same degree in both tissues. The differentially expressing genes are those that lie far away from the $Y = X$ line. In Figure 3.1*b* most genes have R_g values close to one (and, correspondingly, in Figure 3.1*d* most genes have $\log(R_g)$ values close to zero), again indicating that they are expressed to the same degree in both tissues. The differentially expressing genes are those whose R_g is large (e.g., $R_g > 2$), indicating genes that are *overexpressed* or *upregulated* in the cancer cells, and those whose R_g is small (e.g., $R_g < 0.5$) indicating genes that are *underexpressed* or *downregulated* in the cancer cells. Of the 3300 genes in this example, 145 genes are upregulated ($R_g > 2$) and 124 genes are downregulated ($R_g < 2$).

This is the general idea behind microarray experiments.

3.3 A TYPICAL MICROARRAY EXPERIMENT

The very simple hypothetical example that was given above outlined the five basic steps of a typical actual microarray experiment. The five steps are:

1. Preparing the microarray
2. Preparing the labeled sample
3. Hybridizing the labeled sample to the microarray and washing the microarray
4. Scanning the microarray
5. Interpreting the scanned image

We will now describe each of these steps in greater detail.

3.3.1 Microarray Preparation

To start with, we must have a collection of purified single-stranded DNAs. A drop of each type of DNA in solution is placed onto a specially prepared glass microscope slide by a robotic machine called an *arrayer*. This process is called *arraying* or *spotting*. The arrayer can quickly produce a regular grid of thousands of spots in a dime-sized area, small enough to fit under a standard slide's coverslip. The DNA in the spots is bonded to the glass to keep it from washing off during the hybridization reaction and subsequent wash. This then is the DNA microarray for the experiment.

The DNA spotted on the microarray may be either cDNA, in which case the microarray is called a *cDNA microarray*, or oligonucleotides, in which case the microarray is called an *oligonucleotide array*.

The DNA spotted on cDNA microarrays are cloned copies of cDNA, amplified by PCR, corresponding to whole or part of a fully sequenced gene or putative ORF; ESTs are commonly arrayed. These microarrays are widely applicable as their manufacture requires only that a large library of cDNAs be available as a source of clones. The sequence of the cDNA could be several hundred to a few thousand base pairs long. When only a part of a gene is spotted, the subsequence that is spotted is carefully chosen for maximal specificity.

The DNA spotted on oligonucleotide arrays are synthesized chains of oligonucleotides corresponding to part of a known gene or putative ORF; each oligonucleotide is usually only about 25 base pairs long. In oligonucleotide arrays, a gene is represented by several different oligonucleotides; the oligonucleotides are carefully chosen for maximal specificity.

The selection of DNA probes to be spotted on the microarray determines which genes can be studied in the experiment in which it is used. For organisms whose genomes have been completely sequenced, including several bacteria, viruses, and yeast, it is possible to array genomic DNA from every known gene or putative ORF in the organism. For these organisms enough DNA must be produced to make as many arrays as needed. One way to do this is to amplify each gene or putative ORF from total genomic DNA by PCR. However, one disadvantage of using PCR to make multiple copies for array spotting is that PCR can induce mutations, especially at higher cycles. An alternative is to clone fragment cDNAs, make large amounts of identical DNA copies by growing them in bacteria, and then extract plasmid, excising out the specific cDNA fragments.

For organisms with larger and more complex genomes, such as the human genome, that have not yet been completely sequenced, a comprehensive array for the entire genome cannot yet be produced. Of course, in the case of the human genome, the location and sequence of a large percentage of human genes is now known, chiefly as a result of the Human Genome Project. Therefore the same method as above can be used to produce an incomplete but substantial human genome microarray, with a complete one perhaps only a few years away. In addition there are methods for producing arrayable DNA even for unknown genes.

There are a few different robotic technologies that have been developed for arraying microarrays. One method uses a robotic arm to touch and spot nanoscale droplets of the solution containing the cDNA or oligonucleotide. Another method uses ink jet technology to eject the solution onto the surface of the glass slide without the robot actually touching it. Other technologies concurrently synthesize oligonucleotides on the slide in situ, using either photolithography (a proprietary method developed by Affymetrix) or ink jet technology (a method developed by Rosetta Inpharmatics).

The DNA probes arrayed on the microarrays are frequently referred to as "genes" even though this may not be quite accurate.

3.3.2 Sample Preparation

The labeled sample is prepared separately. The first step here is to purify mRNA from total cellular contents. The experimenter must contend with several challenges here: (1) mRNA accounts for only a small fraction (less than 3%) of all mRNA in a cell, (2) the more heterogenous the cells (e.g., the cells of solid tumors), the more difficult it is to isolate mRNA specific to the study, and (3) captured mRNA degrades very quickly. As far as the latter is concerned, in order to prevent the experimental samples from being lost, the mRNA is immediately reverse-transcribed into more stable cDNA (for cDNA microarrays) or cRNA (for oligonucleotide arrays—cRNA is synthetic RNA produced by transcription from a single-stranded DNA template).

Even here there is a small problem: not all mRNAs are reverse-transcribed with the same efficiency. As this effect is gene-specific, the fluorescence intensity that is measured for a gene at the end of the study may not be a true reflection of original mRNA level. Consequently it would not be correct to compare fluorescence intensities for different genes across a single sample. Fortunately, however, it would not be incorrect to compare fluorescence intensities across several samples.

In order to be able to detect which cDNAs are bound to the microarray, the sample is labeled with a reporter molecule that flags their presence. The reporters currently used in microarray experiments are fluorescent dyes, called *fluors* or *fluorophores*, chemicals that fluoresce when exposed to a specific wavelength of light. The labeled sample is the target for the experiment.

The number of fluor molecules that label each cDNA depends on its length and also possibly its sequence composition. This is another reason why fluorescent intensities for different cDNAs cannot be quantitatively compared. However, identical cDNAs from different labeled samples will still be comparable as long as the same number of label molecules is added to the same DNA sequence in each labeled sample.

3.3.3 The Hybridization Step

The labeled sample is poured onto the microarray and allowed to diffuse uniformly all over it. Then it is sealed in a hybridization chamber and incubated at a specific temperature for enough time to allow the hybridization reactions to complete. The experimental conditions should ensure that all areas of the microarray are exposed to a uniform amount of labeled sample throughout this time.

A single-stranded DNA molecule will bind with highest affinity to another single-stranded DNA molecule with a precisely matching sequence and with significantly lower affinity to one with an imperfect match. The stringency of the hybridization depends on experimental conditions such as temperature. If the labeled sample contains a cDNA whose sequence is complementary to the

DNA on a given spot on the microarray, that cDNA will hybridize to the spot. Enough incubation time should be allowed for the hybridization reactions to complete.

The microarray is then removed from the hybridization chamber and thoroughly, but carefully, washed to eliminate any excess labeled sample. Finally the microarray is dried using a centrifuge or by blowing with clean compressed air.

The quality of the hybridization can be assessed experimentally by spotting the probes for a set of hybridization control genes, spiking the labeled sample with a known amount of these controls prior to exposure to the array, and verifying that these control genes are, indeed, showing up as having been hybridized.

3.3.4 Scanning the Microarray

The microarray is scanned to determine the amount of labeled sample bound to each spot. Recall that the sample was labeled with fluorescent reporter molecules that emit detectable light when stimulated by a laser. The emitted light is captured by a scanner, such as a charge-coupled device or a confocal microscope, that records its intensity. Spots with more bound sample will have more reporters and will therefore fluoresce more intensely.

Although it is only supposed to pick up light emitted by the target cDNAs bound to their complementary spots, the scanner will inevitably also pick up light from various other sources, including the labeled sample hybridizing nonspecifically to the glass slide, residual (unwashed) labeled sample adhering to the slide, various chemicals used in processing the slide, and even the slide itself. This extra light is called *background.*

Scanner settings can affect both the precision of the intensity measurements as well as the lower and upper threshold intensity levels that can be measured. Intensities outside this range, called the *dynamic range*, cannot be properly quantified and are often set to the corresponding threshold level. When intensities exceed the upper threshold, *saturation* is said to have occurred. There is a trade-off between the precision and the dynamic range: increasing one will decrease the other, and vice versa, so a balance must be struck.

3.3.5 Interpreting the Scanned Image

The end product of a microarray experiment is a scanned gray scale image (see Fig. 3.2) whose intensity measurements range from 0 to 2^{16}. The image is usually stored in 16-bit tagged image file format (tiff, for short). Image-processing software will convert the image into spot intensity measurements, which will then be analyzed for gene expression differences.

Figure 3.2 shows a typical microarray image. The whiter spots are of higher intensity and can be associated with higher hybridization activity. The very dark spots occur at locations where there was little or no hybridazation.

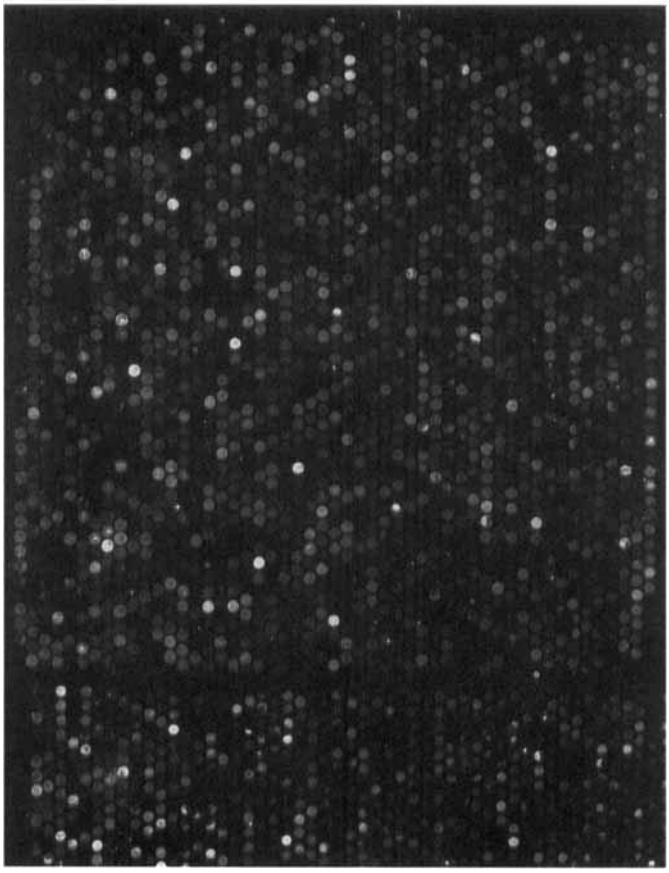


Figure 3.2 A single-channel scanned image.

3.4 MULTICHANNEL cDNA MICROARRAYS

It has become a common practice among those who use cDNA microarrays to fashion the labeled sample out of two or more mRNA samples mixed together. Each mRNA sample in the mixture is labeled with a different fluorescent dye. At the scanning stage, the slide is scanned as many times as there are samples. Such microarrays are called *multichannel cDNA microarrays*.

Figure 3.3 shows the two scanned images from a two-channel cDNA microarray, in which one of the channels was exposed to a control mRNA sample and the second channel was exposed to a treated mRNA sample. Any spot whose intensity is different between the two channels (e.g., dark in channel 1 and white in channel 2) corresponds to a spot that was differentially hybrized

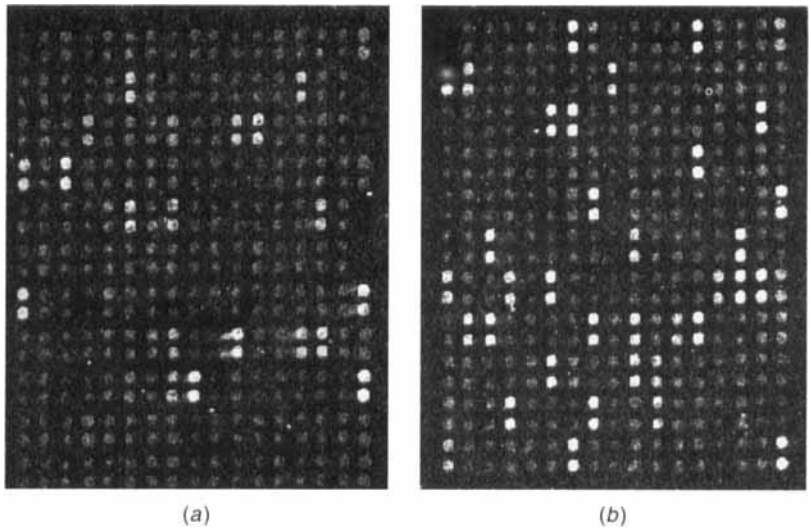


Figure 3.3 Scanned images from a small two-channel microarray. (*a*) Channel 1 (control), (*b*) channel 2 (treated).

and, by inference, to a gene that was differentially expressed in treated versus control due to a treatment effect.

Multichannel cDNA microarrays have some advantages that have led them to becoming the standard technology for cDNA microarrays over single-channel cDNA microarrays. For one thing it is often difficult to tightly control the amount of DNA that is spotted onto the slides, and this could vary from array to array for the same gene. The effect of this variation on downstream analysis can be reduced by the natural matching of samples in multichannel microarrays. In addition some economy is gained as data on expression levels of several mRNA samples can be gathered using just one slide.

However, there are some drawbacks as well: (1) There is an overall dye effect, although this can usually be corrected by normalization (see Chapter 5). (2) If the objective is to compare a large number of mRNA samples, the logistics of setting it up become more complex with multichannel microarrays. (3) A more serious problem is that some genes may incorporate certain dyes better than other dyes, so gene-specific dye effects could occur.

3.5 OLIGONUCLEOTIDE ARRAYS

The technology for the production of high-density oligonucleotide arrays (Lockhart et al., 1996) was pioneered by Affymetrix and remains proprietary to this day. In an oligonucleotide array, a gene is represented by a set of 20 or so

oligonucleotides, called *perfect match probes* (PM). The multiple oligonucleotides that represent a gene are designed in such a way as to hybridize to different regions of the RNA corresponding to an expressed gene and act a series of multiple independent detectors for the gene.

Each perfect match probe is paired with an artificially created *mismatch probe* (MM) that is fashioned by changing the middle base of the corresponding perfect match probe to its complementary base. The mismatch probe is intended to play the role of an internal control for hybridization specificity peculiar to its particular hybridization site. The hybridization to the gene by the perfect match probe represents specific hybridization and should be stronger than any nonspecific hybridization to the mismatch probe. In addition, if the PM intensities are consistently larger than the MM intensities for a probe set, this global effect is more likely to be indicative of actual presence of mRNA corresponding to that gene in the sample as opposed to being a random chance event. At least, that's the theory—in practice, there is a great deal of controversy about the use of the mismatch probes.

Affymetrix refers to each PM-MM pair as a *probe pair* and the entire set of probe pairs for a gene is called a *probe set*. High-density oligonucleotide microarrays are manufactured by synthesizing the oligonucleotides directly onto the surface of a silicon chip. The process is highly elaborate and involves defining the exposure sites on the chip with a series of semiconductor-based photolithographic masks and following this with a light-directed chemical synthesis of the oligonucleotides guided by their DNA sequences. The nature of the process is such that a very large number of oligonucleotides can be densely arrayed at the same time.

3.6 BEAD-BASED ARRAYS

New technologies are constantly emerging in an effort to extend the throughput and potential of microarrays. One of the most promising is *bead-based microarray technology*.

A bead-based fiber-optic microarray is a bundle of optical fibers. Microscopic wells are etched onto the end of each fiber. These wells hold the probe DNA sequences in bead form. The array is exposed to the fluorescently labeled sample. Wherever the labeled sample finds a matching (i.e., complementary) DNA sequence on the microarray, hybridization takes place. Without a match, the labeled sample does not hybridize to the probe. The array is illuminated with a lamp. This triggers fluorescence in the tagged samples, which causes a signal to be passed through the optical fiber to a detector, which indicates which probe DNA sequences match some sequence in the labeled sample.

The throughput of three-dimensional bead-based microarrays is a great deal higher than conventional two-dimensional microarrays. In fact the number of DNA sequences tested could be in the hundreds of thousands, or even millions, range.

3.7 CONFIRMATION OF MICROARRAY RESULTS

Microarray technology is still a dynamic and evolving entity. As such, the state of the technology at this time is that microarray experimental results could be rather variable. The value of microarray technology as a high-throughput screen for gene expression information is without question, but investigators should interpret any results from microarray experiments with some circumspection (e.g., see Kothapalli et al., 2002). Thus the key is to utilize microarrays as a means of screening and prioritizing a large number of genes, but any findings pertaining to genes of special interest should be independently confirmed. This is generally done on a gene-by-gene basis using methods such as Northern blots or quantitative RT-PCR.

SUPPLEMENTARY READING AND ELECTRONIC REFERENCES

1. *Animations.* The Web sites `http://www.bio.davidson.edu/courses/genomics/chip/chip.html` and `http://darwin.bio.uci.edu/~faculty/wagner/array2.html` have animations that demonstrate how a DNA microarray experiment is performed.
2. *The Chipping Forecast* (2001) and *The Chipping Forecast II* (2002) are special supplementary issues of the journal *Nature Genetics* that carry several excellent review articles by several researchers who either pioneered or significantly advanced the field of DNA microarrays. *The Chipping Forecast* is freely available online at `http://www.nature.com/ng/chips_interstitial.html`.
3. The Web site `http://www.cs.wustl.edu/~jbuhler/research/array` has an excellent introduction to cDNA microarrays and comparative hybridization by J. Buhler.
4. The book by Schena (1999) discusses various aspects of microarray experiments.
5. Nguyen et al. (2002) provides an excellent review of the biological and technological aspects of microarray experiments in a format suitable for data analysts.

EXERCISES

3.1. What is the difference between a genetic disease and a complex disease? How would a microarray experiment to discover the genes involved in a genetic disease differ from an experiment to discover the genes involved in a complex disease?

3.2. What is the advantage of doing a temporal study?

3.3. Outline the various steps of a typical microarray experiment.

3.4. Explain the terms: background, saturation.

3.5. Discuss the advantages and disadvantages of a two-channel microarray versus a single-channel microarray.

3.6. What is **(a)** a probe pair **(b)** a probe set?

3.7. In what way does quantitative RT-PCR complement microarrays?

Exploration and Analysis of DNA Microarray and Protein Array Data
Edited by DHAMMIKA AMARATUNGA and JAVIER CABRERA

CHAPTER 4

Processing the Scanned Image

When microarrays are scanned at the end of an experiment, the result is a series of images, one image per channel. Thus a one-channel microarray, such as an oligonucleotide array, yields one image per array, whereas a two-channel microarray yields two images per array, one image per channel.

The scanner "reads" a microarray by dividing it up into a very large number of pixels and recording the intensity level of the fluorescence at each pixel. The resulting rectangular array of pixels and their associated intensities constitutes the *image* of the microarray.

The image must be converted into spot intensities for analysis (see the schematic in Fig. 4.1). The purpose of this conversion is to assign to every DNA sequence that was spotted on the microarray an intensity measure, called the *spot intensity*, reflecting the amount of labeled sample that hybridized to it.

Following this, it is generally advisable to perform a series of quality checks on the data and, if necessary, generate warnings about possible problems, such as aberrant spots and defective microarrays, so that the investigator could take appropriate action.

Finally, the spot intensity data should be adjusted for background fluorescence.

4.1 CONVERTING THE SCANNED IMAGE TO THE SPOTTED IMAGE

The task of quantifying a scanned image is often carried out in three steps. First, the location of each spot in the array is defined by assigning coordinates to the center of each spot—this is called *gridding*. Second, the *signal*, the set of pixels that correspond to labeled cDNA hybridizing to its complementary DNA sequence spotted on the microarray, is separated from the *background*,

Exploration and Analysis of DNA Microarray and Protein Array Data
By Dhammika Amaratunga and Javier Cabrera
ISBN 0-471-27398-8

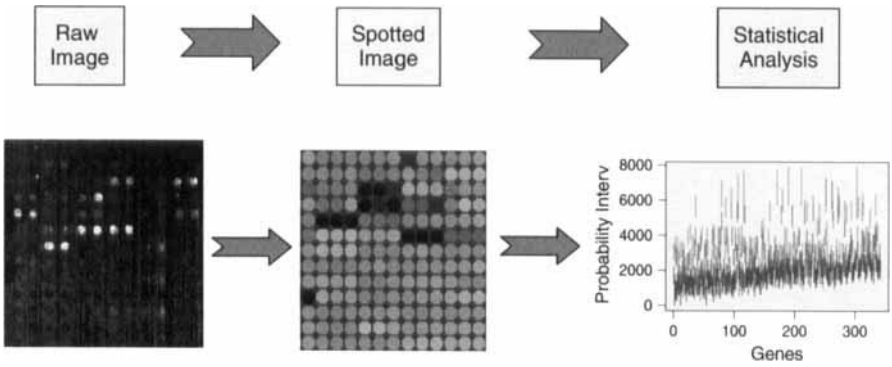


Figure 4.1 Data processing steps starting with the raw image.

the set of pixels that correspond to labeled cDNA hybridizing nonspecifically to the microarray—this is called *segmentation*. Third, each spot is assigned an intensity value—this is called *quantification*. We now mention some aspects of each step; Yang et al. (2000, 2001) provide more detailed accounts.

4.1.1 Gridding

If the arraying process arranged the spots in a perfect rectangular grid, as it should, the task of defining the spots by assigning coordinates to the center of each circular spot would be a simple matter: just overlay an appropriately sized grid on the microarray and move it around until it is properly aligned. In practice, however, the arraying process is not perfect, so the grid that is actually arrayed tends to be a slightly deformed version of the target regular rectangular grid. As a result the overlaid grid will need some fine-tuning, which can be done by manipulating the rows and columns of the overlaid grid until the it is satisfactorily aligned. Care must be taken that speckles and dust, which can fluoresce as brightly as a spot, do not confuse the procedure.

A somewhat more rigorous method would be to first locally smooth the image using a Gaussian kernel, designate the modes of the smoothed regions as the spot centers, and then modify the grid so that the distances from the spot centers to the centers of the rectangular or square regions containing each spot are minimized.

4.1.2 Segmentation

Once the locations of the centers of the spots have been determined, the next step is to separate from the background the spot, that is, the region of the slide on which cDNA was actually arrayed. This should not be too difficult if all the spots were circular with a well-defined boundary. The procedure would involve either fitting a circle with a constant diameter to all the spots on the image

(*fixed circle segmentation*) or fitting circles with different diameters to different spots on the image (*adaptive circle segmentation*).

In practice, neither of these segmentation procedures works particularly well as the spots tend to vary considerably in size, shape, and regularity due to a number of factors, such as the quality of the spotting tip of the arrayer (which degrades with use), how long the tip stays on the slide, the deposition of the cDNA causing a bowl-like depression, the coating on the slide, the surface tension and viscosity of the solution being arrayed, the ambient temperature and humidity, and the posthybridization processing of the microarray. Work in computer vision has suggested some ways of dealing with this problem.

One such method is the *seeded region growing algorithm*. The algorithm consists of the following steps:

Step 1. *Seed specification.* To get this algorithm started, a set of pixels called *seeds* have to be specified. One simple way to do this is to let the seeds for signal be the estimated spot locations from the gridding step and to let the seeds for background could the midpoints.

Step 2. *Region growing.* For each spot we have a seed for a signal region and a seed for a background region. The seeds are then "grown" into regions by allocating the remaining pixels to either signal or background region, depending on their intensity and their closeness to a seeded region. A pixel that is adjacent to an allocated pixel is considered a candidate for allocation into that region. At each step, among the pixel candidates for all regions, the pixel that is the closest in intensity to the average intensity of the corresponding region is assigned to that region.

Step 3. *Stopping rule.* This process continues until all the pixels have been allocated to one of the regions.

Another method is *histogram segmentation*. A mask is placed over each spot. The mask should be larger than the spot. The histogram of pixel intensities within the mask is examined to determine a threshold value. Each pixel within the mask is then classified as signal or background depending on whether its intensity is above or below this threshold.

Therneau et al. (2002) discuss a method, based on the EM algorithm (Dempster et al., 1977), for sharpening spots to correct for the bleeding of one spot onto another.

4.1.3 Quantification

At each spot the average intensity of the pixels is measured. This observation is complemented by a number of other spot-related statistics that allow the quality of the spot to be assessed (e.g., see Kuklin et al., 2001; Wang et al., 2001; Brown et al., 2001, recommend a pixel-by-pixel analysis of individual spots for two-channel microarrays). The following list outlines some typical spot-related statistics that are reported:

- *Spot intensity.* The end product of the conversion process is an array $\{X_{rc}\}$ of spot intensities: here X_{rc} denotes the intensity of the spot located at the rth row and cth column of the array. At each spot the average intensity of the pixels designated as signal is taken to be its spot intensity value. The average used is often the mean, because it should be representative of the number of labeled mRNA molecules hybridizing to the DNA spotted on the array. However, because the distribution of pixel intensities might be irregular, other measures of location, such as the median, trimmed mean, biweight, and mode, are also sometimes used.
- *Spot background.* This is the average intensity of the pixels around the spot that were designated as background. The average used is often the mean or median. The background intensities are represented as an array $\{B_{rc}\}$ of the same dimension as $\{X_{rc}\}$.
- *Pixel intensity distribution.* In general, the distribution of pixel intensities in and around a spot does not resemble a normal. Instead, for example, it could be peaked with one or two long tails. This is one reason why the segmentation process can sometimes be imprecise. *Spot cv* and *background cv* are the coefficients of variation of the intensities of the pixels assigned to signal and background respectively. The higher the spot *cv*, the more variable are the intensities of the pixels that make up the spot and, possibly, the lower is the quality of the spot. Moreover the closer the spot intensity is to spot background or to saturation, the less reliable it is.
- *Spot morphology.* Measures associated with geometric characteristics of the spot, such as the size of the spot, also provide information on the quality of the spot. When seeking circular spots, two other such measures are *circularity*, which is 4π times the area of the spot divided by its perimeter, and *regularity*, which is the proportion of pixels designated as signal by the segmentation procedure among the pixels falling within the circle designated by the gridding procedure. The closer these measures are to unity, the more circular is the spot.

4.2 QUALITY ASSESSMENT

Once the spotted image and related statistics are obtained, it is advisable to (1) assess the quality of the array and (2) evaluate the quality of the individual spots on the array. This is because sometimes the array could have a region of generally increased or decreased intensity and some spots might be defective. Figure 4.2*a* shows a scanned image that has a few such blemishes. There are some speckles that could be dust. The background seems to be nonuniform. Figure 4.2*b* shows some spots with high-contrast background effects, and Figure 4.2*c* shows some defective spots.

Artifacts such as these, introduced perhaps by the experimental process or some odd random event (e.g., dust particles settling on the array), could seri-

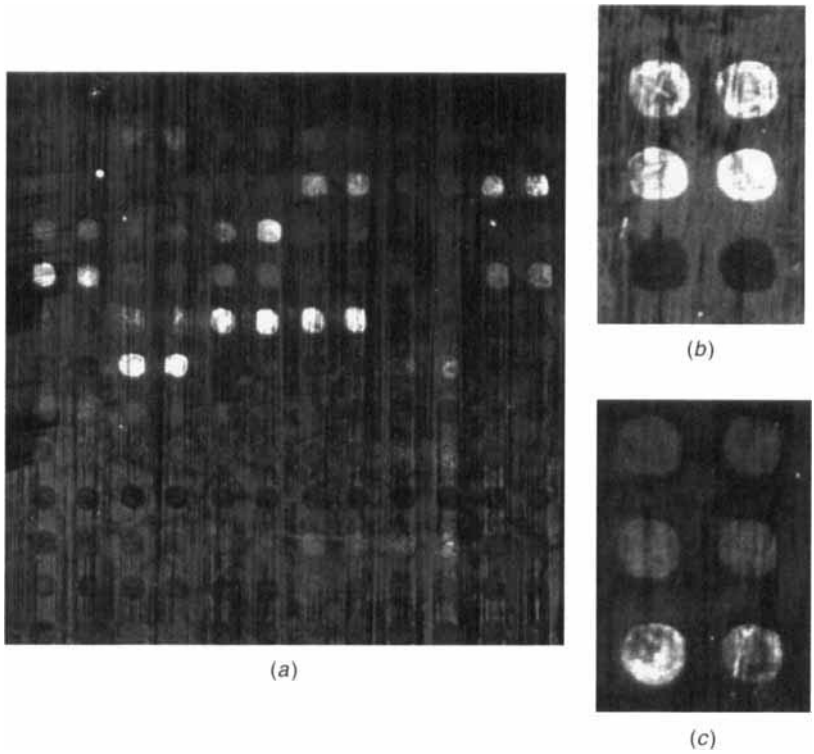


Figure 4.2 (*a*) Scanned image with a few blemish like speckles and a nonuniform background and a few spots that show, (*b*) high-contrast background effects, and (*c*) defective spots.

ously compromise the corresponding spot intensities. If the affected arrays and/or spots are not identified and removed or otherwise adequately downweighted, they could mask true experimental effects. Several steps are involved in assessing the quality of an array and the spots within an array.

4.2.1 Visualizing the Spotted Image

Visual inspection of the data is a first attempt to appraise the quality of the spotted image. This can be done using a typical *image plot* (referred to as a *heat map* in the computer science literature), in which each image pixel corresponds to a spot. The image plot is then examined and searched for obvious nonrandom patterns that would suggest poor data quality. If none is observed, the image is passed on to the next step.

A basic graph for visualizing a spotted image is shown in Figure 4.3. The graph contains a central panel showing a color image with a color scale underneath. The central panel may be subdivided into groups from top to bottom giving the images of subpanels or clusters with a bar on the left indicating

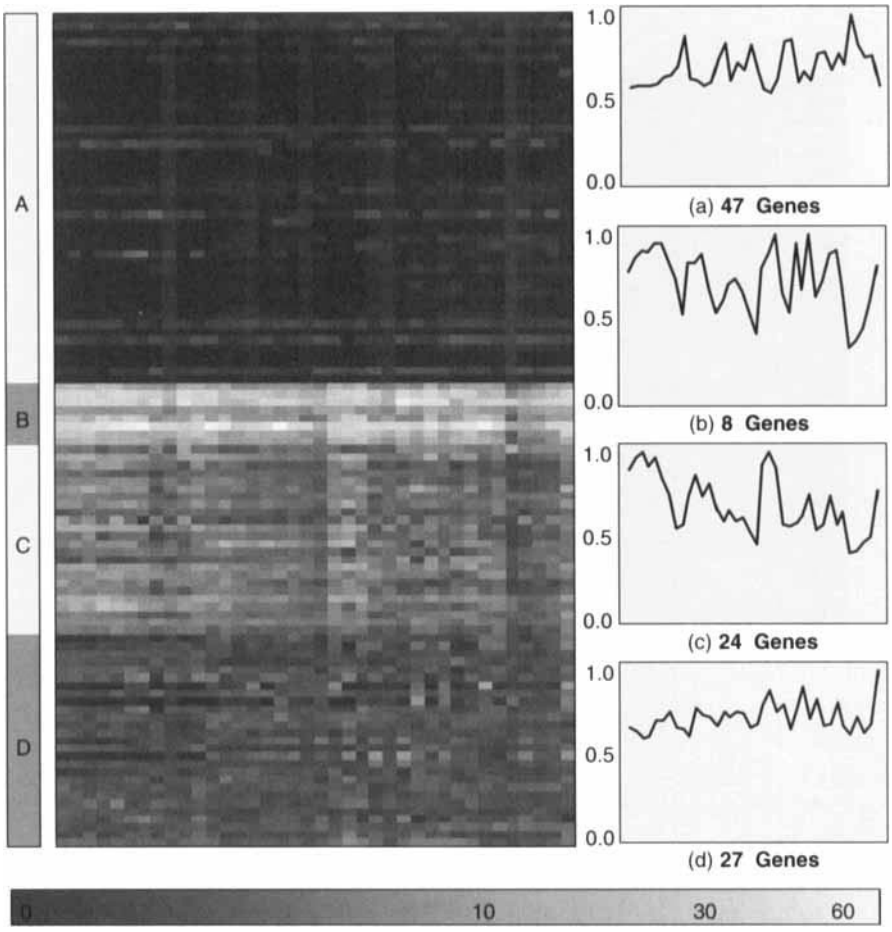


Figure 4.3 Microarray graph.

the subgroups. Finally, the right side of the graph shows the average profiles of each of the groups or subpanels. The average profiles are computed by taking the average of the subgroups across the columns and then normalizing each profile by dividing it by its maximum.

4.2.2 Numerical Evaluation of Array Quality

In addition to visual methods for checking for patterns in the spotted image, numerical methods should also be considered. This is because:

- It is possible that some of the more subtle spatial patterns in a spotted image are not visible in the image graph because the variation is small

enough not to show on the color scale or the color scale may not be sensitive enough to show the pattern.

- Automated methods are crucial for processing a large number of microarray images, such as those that occur in a high-throughput environment, without the need for tedious individual visual inspection.

4.2.3 Spatial Problems

In order to ensure the integrity of the data, the spot and background intensities, $\{S_{rc}\}$ and $\{B_{rc}\}$, must satisfy some quality criteria related to the spatial distribution of the intensities. The first one is that the background intensities must be uniformly distributed. We expect this to be approximately correct, or at a minimum, we expect that the background intensities will not display clear nonuniform patterns.

A few nonuniform patterns appear quite often because they may be related to specific problems with the experimental process, such as hybridization artifacts, inconsistent washing across the slide and other technical problems, that introduce topographical variation:

Case 1. A large smudge covering a substantial part of the area of the background image. These smudges are areas of the array that show higher or lower intensities compared to the rest of the image.

Case 2. Vertical or horizontal strips on the background image that show higher or lower intensities.

Case 3. Diagonal strips again showing higher or lower background intensities.

Case 4. A gradient in the background intensities going across the array.

Case 5. A row or column effect such as an edge effect.

Case 6. Bleeding in the spotted image, namely a series of consecutive spots that are blurred together forming a horizontal or vertical line.

In order to detect such patterns, Amaratunga and Cabrera (2003b) propose a method that separates pixel intensities into high and low and separates them using only the two coordinates: row number and column number. If the separation is successful, the array has an spatial problem and should be discarded; otherwise, the array is accepted. The steps of the algorithm they proposed are as follows:

Step 1. Split the image into high-intensity and low-intensity spots. This is a binary split similar to the ones performed by a regression tree algorithm at a single node. The CART procedure (Breiman et al., 1984) does this by identifying the cutoff that minimizes the within-group sum of squares. However this split is not robust against outliers. A simple alternative that is resistant to outliers is to set the cutoff to the mid point between to quantiles (e.g., 5%

and 95% quantiles). Then define the response at the spot at row r and column c: $Y_{rc} = 1$ for high-intensity spots and $Y_{rc} = 0$ for the rest.

Step 2. Fit a quadratic discriminant function (see Section 10.3) to the binary response $\{Y_{rc}\}$ using the spot coordinates (r, c) on the microarray as predictors. Suppose that Z_{rc} are the predicted responses by the discriminant function. In order to assess the goodness of the fit, calculate the proportion π of correctly predicted spots, that is, the proportion of spots with $Y = Z$. The null distribution of the π statistic can be simulated by a simulation the images. To do this, generate a large number (e.g., 300) of images by random permutations of the spot intensities and calculate the value of π for each image, resulting in the set $\{p_1, \ldots, p_{300}\}$. Estimate the p-value as the proportion of π_i greater than the observed π. This p-value measures the performance of the quadratic discriminant analysis and is used to determine the overall quality of the microarray.

The outcome of the procedure above can be summarized in an image quality graph such as the one shown in Figure 4.4. The figure consists of a central panel showing a color image and a set of four graphs on the right side of the figure. The main panel displays an image representing the background intensities that are being analyzed. The color or gray scale corresponding to the main panel is shown on a narrow horizontal strip below the main panel. The right side of Figure 4.4 shows a column of four graphs:

- The two graphs at the top of the right side show the average profiles of the rows and columns of the main panel respectively.
- The third and fourth graphs show the image graphs of the arrays $\{Y_{rc}\}$ and $\{Z_{rc}\}$ respectively.

4.2.4 Spatial Randomness

Another way of assessing whether any part of an array is emitting higher signals compared to the rest of the array is to check whether the "outliers" in either the signal or the background are randomly scattered throughout the array or clustered together or distributed according to some pattern. In the algorithm in the previous section this check could be used as in the graphs of Figure 4.4. The assessment could be made using a simple test of complete spatial randomness, such as that proposed for a problem in ecology by Clark and Evans (1954).

Suppose that the array has G spots, r of which are outliers. For the ith outlier, let d_i be the distance to the "outlier" closest to it, so that $\bar{d} = \sum_{i=1}^{G} d_i/G$ is the average nearest-neighbor distance between the outliers. The test statistic for complete spatial randomness is $\bar{d}$ or its standardized form

$$T_{CSR} = \frac{\bar{d} - 1/(2\sqrt{\rho})}{\sqrt{(4-\pi)/4G\pi\rho}},$$

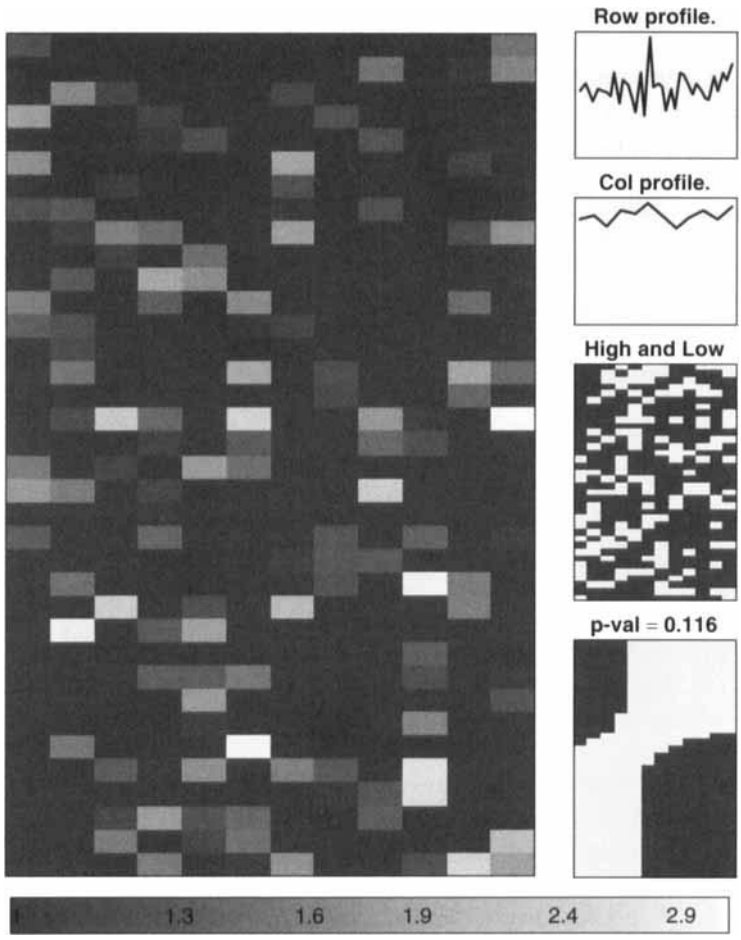


Figure 4.4 Image quality graph.

which has a standard normal distribution under complete spatial randomness. The parameter ρ can be estimated as $\hat{\rho} = r/G$. Note that two aspects of the data are being ignored in doing this test: nonindependence of some nearest-neighbor distances and edge effects. More complex tests that adjust for these aspects of the data have been developed in the spatial data analysis field.

This method is fast and would produce good results for arrays where the outliers appear in small clusters, such as the bleeding spots, case 6 in the list of Section 4.2.3. If the smudge covers a large part of the array, then it would help to smooth the image, but it may be harder to detect with this approach.

4.2.5 Quality Control of Arrays

Rigorous quality assurance ensures that the accuracy and precision of an experimental process is maintained over time. Besides continuously making

sure that the quality of the various individual steps of the microarray experimental protocol is being preserved, the experimenter should use the data being collected to monitor the stability, consistency, and overall performance of the experimental process as a whole.

Microarray experiments are usually performed over time. It is important to take this temporal effect into account because experimental conditions tend to be affected by time. For example, the sample could vary (perhaps degrade) over time, operators of varying ability may run the experiment over several days, and various day effects (e.g., temperature and humidity) could affect the materials and the results. All these could potentially have a significant impact on experimental results. Therefore such effects should be monitored carefully. Some of these effects, if reasonably small, may be accounted for at the modeling stage of the analysis. However, it is useful to be able to detect when an experiment may be going out of control in the early stages of the experiment, or a series of experiments as it is being run, so that the experimenter can intervene immediately and address any experimental problems.

The process of data acquisition starts with the outcome of the experiment that is the microarray. The microarray is then scanned and the scanned raw image is processed to generate the spotted image. The spotted image is stored in the database.

A simple quality control procedure can be established at the moment when the spotted image is stored in the database by running a procedure that produces the following items:

1. An image quality graph, such as the one shown in Figure 4.4, could be used to detect specific problems with the array.
2. A side-by-side display of boxplots of the sequence of arrays that have been observed up to this point, or a set of summaries based on them, could be used to check whether there are any changes from the previous arrays to the current one.

This quality control process requires that the process of data acquisition be automated as mach as possible, in order to avoid unnecessary delays on the experimental side. Figure 4.5 shows an experiment where a change in operator produced a shift in the scale of the observations in the last four arrays.

4.2.6 Assessment of Spot Quality

Once an array is deemed to be of satisfactory quality to be included in an analysis, the quality of the individual spots should be assessed. Actually spot quality assessments can be done at two different stages of an analysis.

First, at the image processing stage, the quality of a spot can be assessed by studying the properties of its pixel intensity distribution or its spot morphology (see Section 4.1.3). Since it is unlikely that a single quantity could capture everything that could go wrong with a spot or a spot intensity measure, some

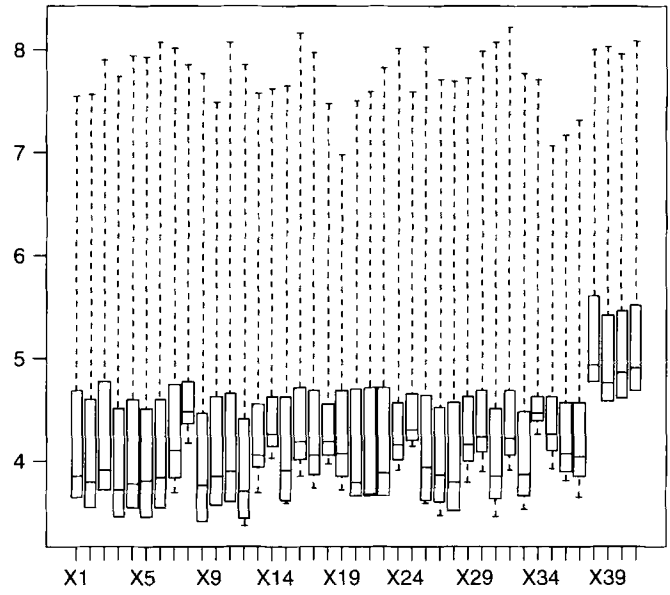


Figure 4.5 Quality control graph.

composite index must be formed from the above-mentioned quality metrics to flag suspect spots. These flagged spots can then be individually examined visually, if necessary.

Then, at the post normalization stage, if replicates are available, each set of replicate spots can be analyzed to check whether any value in the set is markedly different from the others (this is described in Section 5.7). The replicates may be repeated spots on an array, technical replicates, or biological replicates. Exclusion or downweighting of spots that are considered low quality is likely to improve downstream analyses.

4.3 ADJUSTING FOR BACKGROUND

In principal, the intensities of those pixels not corresponding to spots should be zero. However, this never happens. Instead, because of various reasons such as nonspecific binding of the labeled sample to the array substrate and substrate fluorescence, these pixels emit a low, but not insubstantial, level of fluorescence that may vary with location.

The concern is that the spot intensities may also contain a certain amount of this nonspecific fluorescence, called the *background* fluorescence. It is customary therefore to estimate a background intensity from data and, assuming that the spot signal intensity is an additive combination of the true spot intensity

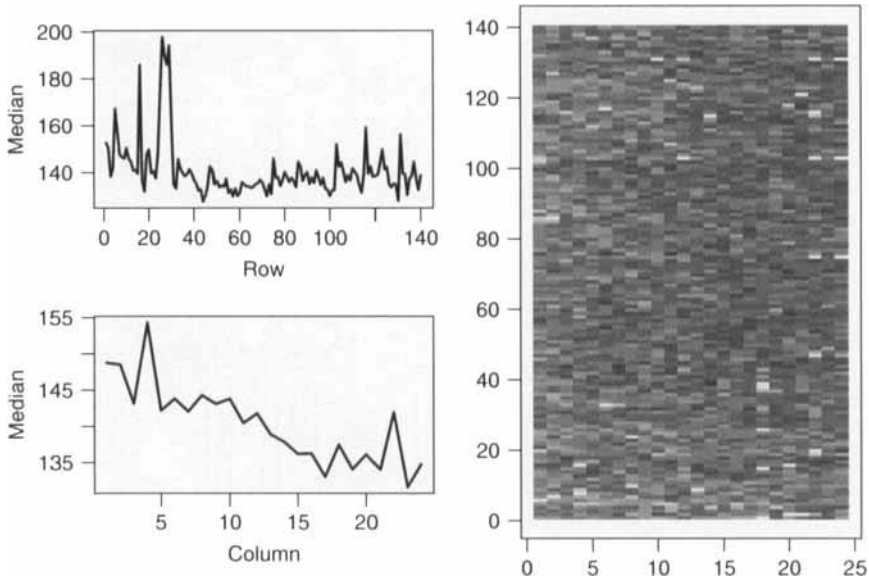


Figure 4.6 *Left*: Line graphs of background showing the row and column medians. *Right*: Image plot of the background.

and the background, to subtract the background from the raw spot intensity values to yield a set of *background-adjusted spot intensity values.*

4.3.1 Estimating the Background

There are a few different ways in which the background is estimated.

Global Background Adjustment. A very simple estimate of the background is the average intensity of all the pixels not belonging to spots. However this naïve approach is rarely effective, because the background often tends not be uniform over the entire microarray.

Example. Figure 4.6 is an image graph of the background intensities for a microarray of 140 rows and 24 columns. The left side of the graph shows the row and column medians, while the right side shows an image plot of the background. Clearly, there is some topographical variation across the slide. The intensities decrease moderately from the left to right along the columns, and a few rows on the bottom of the array have higher intensities than the rest.

Spot Background Adjustment. The spot background can be subtracted from the spot intensity value to yield a *spot background-adjusted spot intensity values.* However, the segmentation process, which separates spot from background, is usually imperfect, and the spot background often contains a contribution from

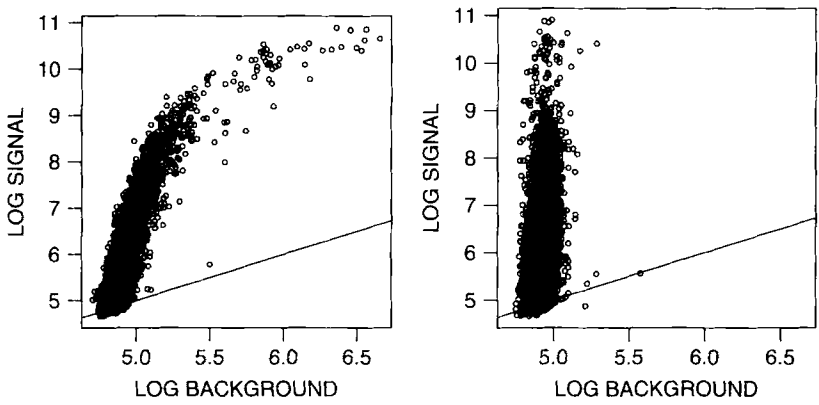


Figure 4.7 Scatterplots of signal versus background before and after local smoothing.

the signal. This manifests itself as a nonzero correlation between spot intensity and background intensity; spots with high intensity tend to have high spot local background, whereas spots with low intensity tend to have low spot background. In this case it is evident that subtracting the spot local background would not be the right idea.

Example. Figure 4.7*a* shows a scatterplot of spot intensity versus background intensity, both on a log scale. Spearman's rank correlation coefficient (see Section 5.6) is 0.92, indicating a strong monotone relationship.

Smoothed Background Adjustment. The true variation in background across an array should be smooth as it is due to experimental effects—such as hybridization artifacts, the washing process, and scanning variation—that vary gradually across the slide. The background may be smoothed by running a simple smoothing procedure through the array. For example, one simple smoothing procedure is to take the median of the 49 values in the 7×7 subgrid surrounding a spot as the smoothed background value at the spot. Yang et al. (2000) describe a more sophisticated smoothing procedure called *morphological opening*.

Example. Figure 4.7*b* shows a scatterplot of spot intensity versus smoothed background intensity, both on a log scale. The smoothed background values are now uniformly distributed across the spot intensity values. Nevertheless, certain spatial features, notably the left to right gradient and the rows of high intensities in the bottom rows, still remain as shown by Figure 4.8, which is the smoothed background analogue of Figure 4.6. In other words, while the sporadic high background intensities that were associated with the high signal spots have been dampened by the smoothing, the background is not uniform throughout the slide.

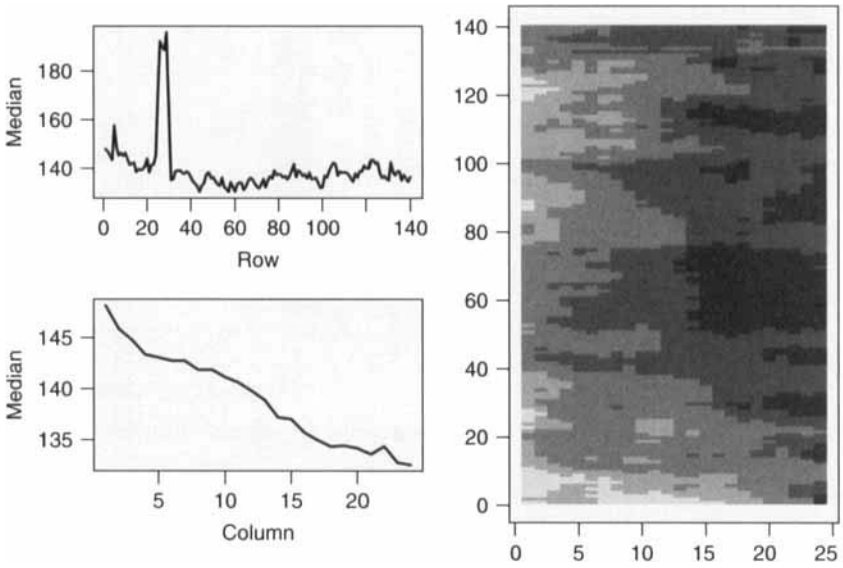


Figure 4.8 *Left*: Line graphs of locally smoothed background showing the row and column medians. *Right*: Image plot of the smoothed background.

Zonal Background Adjustment. Affymetrix uses a variation of smoothed background adjustment, called *zonal background adjustment*, for its oligonucleotide microarrays. This approach can also be used with cDNA microarrays.

First, the microarray is split up into K rectangular zones, Z_k, $k = 1, \ldots, K$ (Affymetrix uses the default $K = 16$). For each zone a low percentile of the spot intensities, SI_g, is chosen as the background, B_k, for that zone (Affymetrix uses the second percentile as its default; this is the value such that 98 percent of spot intensity values are larger than it and 2 percent are smaller).

If we were to just use B_k as the background for zone Z_k, there could be a sharp transition in background estimates when crossing a zone. As this is unlikely to reflect reality, a smoothed version of these background estimates is used instead. The background for a given spot then is a weighted sum of all the B_k values, where the weights are inversely proportional to how far the spot is from the various zone centers. That is, if the distance between the gth spot and the center of Z_k is d_{gk}, the gth spot would be assigned a weight: $w_{gk} = 1/d_{gk}^2$. In practice, a small positive factor, d_0, is added to the denominator to ensure that it will never be zero, so $w_{gk} = 1/(d_{gk}^2 + d_0)$. The background, B_g^*, for the gth spot is then the weighted sum

$$B_g^* = \frac{\sum_{k=1}^{K} w_{gk} BZ_k}{\sum_{k=1}^{K} w_{gk}}.$$

4.3.2 Adjusting for the Estimated Background

Most arrays generally have a background that is not spatially uniform even after smoothing. This background could be sizable enough to affect the spot intensity distribution and therefore any downstream analysis. For this reason it is generally removed from the spot intensities prior to formal analysis.

Suppose that the spot intensity at the gth spot is SI_g and the background intensity was estimated to be BI_g. The *background-adjusted spot intensity value*, AI_g, is obtained by shifting the spot intensity down by the background intensity:

$$AI_g = SI_g - BI_g.$$

In principle, SI_g should be larger than BI_g. However, either due to some problem or perhaps purely due to random variability, this may not always be the case as BI_g can exceed SI_g, leading to a negative value for AI_g. As this is not desirable, sometimes a small additional adjustment is made.

One very simple way to do this is to set a threshold. For example, if T is a low percentile of the SI_g values (e.g., the fifth percentile), take the *background-adjusted thresholded spot intensity value*, AI_g, to be

$$AI_g = \max(SI_g - BI_g, T).$$

While these suggestions regarding background adjustment are reasonable, how to properly adjust for spatially nonuniform background still appears to be an open research problem.

4.4 EXPRESSION LEVEL CALCULATION FOR TWO-CHANNEL cDNA MICROARRAYS

Let the adjusted spot intensities for the two channels of a two-channel array be $\{AI_{gR}\}$ and $\{AI_{gG}\}$, where the letters R and G refer to the colors red and green, respectively, that are typically used to label the channels. If channel G is a reference channel, then the expression level of the gth gene in channel R is usually stated as a *gene expression ratio*:

$$R_g = \frac{AI_{gR}}{AI_{gG}}.$$

Usually, however, there is a systematic effect due to the use of two different dyes that need to be removed from $\{AI_{gR}\}$ and $\{AI_{gG}\}$ by normalization prior to calculating these ratios (see Section 5.5.4).

4.5 EXPRESSION LEVEL CALCULATION FOR OLIGONUCLEOTIDE ARRAYS

In high-density oligonucleotide arrays, such as those manufactured by Affymetrix, a gene is represented by a probe set, which is a set of 20 or so oligonucleotides called the perfect match probes, along with a set of paired mismatch probes. The expression level for a gene, which Affymetrix refers to as the gene's *signal*, is therefore not directly measured but rather obtained by combining the perfect match and mismatch intensities of the probe set for the gene in some way. There are several ways in which a composite value can be calculated.

4.5.1 The Average Difference

Let PM_{gi} and MM_{gi} denote the (untransformed) background-adjusted spot intensity measurements for the ith perfect match probe and mismatch probe respectively for gene g ($i = 1, \ldots, m_g$, $g = 1, \ldots, G$). Noting that $Y_{gi} = PM_{gi} - MM_{gi}$ functions as a measure of the hybridization level of the gth gene's ith probe, the most natural estimate of the signal, S_g, for the gth gene is

$$S_g = \frac{\sum_{i=1}^{m_g}(PM_{gi} - MM_{gi})}{m_g} = \frac{\sum_{i=1}^{m_g} Y_{gi}}{m_g}.$$

In other words, S_g is the arithmetic mean of the Y_{gi} values. One of Affymetrix's early approaches was exactly this, with one modification: in order to lessen the impact of gross outliers on this estimate, any Y_{gi} value further than three standard deviations away from the mean is discarded from the calculation of S_g. Affymetrix called this estimate the *average difference* (an older version used a trimmed mean, which Affymetrix referred to as "Olympic scoring"). However, recognizing problems with this estimator, Affymetrix has recently modified their algorithm.

4.5.2 A Weighted Average Difference

One problem with the average difference is that although the mismatch probes are placed on the array to provide probe-specific estimates of any stray signal due to nonspecific hybridization that may affect the perfect match probe, it can happen that MM_{gi} also contains some portion of the true target signal. Thus a nonlinear relationship between the PM_{gi} and MM_{gi} intensities can often be detected. Therefore each MM_{gi} value should be adjusted to give an *ideal mismatch value*, IM_{gi}, prior to subtracting it from its corresponding PM_{gi}.

If PM_{gi} exceeds MM_{gi}, Y_{gi} represents a possible measure of the true hybridization level for the ith probe for gene g and IM_{gi} is usually set to MM_{gi}. On the other hand, if MM_{gi} exceeds PM_{gi}, which can happen either due to some biological or physical effect or due to random variability, Y_{gi} is negative and

no longer represents a possible measure of the hybridization level. In this case Affymetrix recommends using an algorithm they developed for calculating a value for IM_{gi} that satisfies $0 < IM_{gi} < PM_{gi}$ based on the behavior of the totality of probes in the gth probe set (the Affymetrix web site offers details).

Once this is done, S_g is calculated as an average of the Y_{gi} values, where now $Y_{gi} = PM_{gi} - IM_{gi}$ (and all $Y_{gi} > 0$). However, as the Y_{gi} values may contain outliers, instead of merely taking their arithmetic mean as above, they are log transformed (let $X_{gi} = \log(Y_{gi})$), then averaged using their one-step biweight mean, and finally exponentiated back to the original scale:

$$S_g = \exp(T_{biwt}\{X_{gi}\}).$$

The log transformation reduces the skewness of distribution of $\{Y_{gi}\}$ and the use of the one-step biweight mean reduces the influence of outliers on the final estimate.

The *one-step biweight mean* is a weighted mean that offers efficiency as well as resistance to outliers. It is calculated as follows: Let M_g and MAD_g denote, respectively, the median and the MAD (median absolute deviation from the median) of the $\{X_{gi}\}$. For each observation, X_{gi}, calculate $u_{gi} = (X_{gi} - M_g)/\tau MAD_g$, which indicates how "unusual" it is, then assign it a weight w_{gi} based on the *biweight weighting function*: $w(u) = (1 - u^2)^2$ if $|u| < 1$ and $w(u) = 0$ otherwise. The weighting process is such that observations relatively close to the center of the data will be assigned high weights, whereas any observations relatively far from the center of the data, namely outliers, will be assigned low weights. The *tuning constant*, τ, determines the amount of efficiency and resistance desired. The larger τ is, the more efficient the estimator is if the $\{X_{gi}\}$ are normally distributed, but the more vulnerable it is to being affected by outliers. The smaller τ is, the less efficient the estimator is if the $\{X_{gi}\}$ are normally distributed, but the less influenced it is by outliers. A compromise between these two extremes offers both high efficiency at the normal distribution and resistance should the data contain outliers. The one-step biweight mean is the weighted mean:

$$T_{biwt}(\{X_{gi}\}) = \frac{\sum_{i=1}^{m_g} w_{gi} X_{gi}}{\sum_{i=1}^{m_g} w_{gi}}.$$

Various other averaging methods have been proposed. For instance, Efron et al. (2001) explored the possibility of averaging $\{\log(PM_{gi}) - c \log(MM_{gi})\}$, with a preference for the compromise value for c of 0.5.

4.5.3 Perfect Matches Only

Concerned that MM_{gi} contains too much target signal to function as a true measure of nonspecific hybridization, some investigators prefer to avoid utilizing them altogether (e.g., Naef et al., 2001). These investigators calculate S_g as

an average of the PM_{gi} values. However, as the distribution of $\{PM_{gi}\}$ is usually skewed, instead of merely taking their arithmetic mean as in Section 4.4.1, it is better to log transform them prior to averaging, and use as signal either

$$S_g = \exp\left(\frac{\sum_{i=1}^{m_g} \log(PM_{gi})}{m_g}\right)$$

or the one-step biweight mean described in Section 4.4.2 with $IM_{gi} = 0$.

4.5.4 Background Adjustment Approach

Irizarry et al. (2002) examined the distribution of $\{MM_{gi}\}$ on an array and, observing that it was consistent with a mixture of low background intensities and high signals corresponding to probes detecting transcript, concluded that the mode of this distribution constituted a natural estimate of background for the array. Estimating the mode using a density kernel estimate and using it as the background, an average of the background-adjusted perfect match values is then the estimate of signal:

$$S_g = \exp\left(\frac{\sum_{i=1}^{m_g} \log(PM_{gi} - \text{mode}(\log(MM_{gi})))}{m_g}\right).$$

4.5.5 Model-Based Approach

Li and Wong (2001b) proposed a model-based approach. Their model

$$Y_{gi} = PM_{gi} - MM_{gi} = \theta_g \phi_{gi} + \varepsilon_{gi}$$

postulates that the perfect match to the mismatch difference is, except for random error, $\varepsilon_{gi} \sim \mathrm{N}(0, \sigma_g^2)$, the product of a model-based expression index θ_g, whose estimate functions as S_g, and a probe-specific sensitivity index ϕ_{gi}. The model parameters are estimated using maximum likelihood, and S_g is estimated as a weighted mean:

$$S_g = \frac{\sum_{i=1}^{m_g} \phi_{gi}(PM_{gi} - MM_{gi})}{m_g}.$$

The Li and Wong model is most useful when several replicates are available (see Section 6.2).

4.5.6 Absent-Present Calls

The availability of several PM_{gi} and MM_{gi} intensities for a probe set allows the reliability of the measurement of the signal corresponding to that probe set to

be assessed. The rationale behind the procedure is that a probe set whose probe pairs consistently exhibit PM_{gi} intensities greatly exceeding their corresponding MM_{gi} intensities is more likely displaying a reliable signal than one whose PM_{gi} intensities are all close to their corresponding MM_{gi} intensities.

Based on this rationale, Affymetrix reports, for each gene on the array, an *absolute call*, which indicates whether the transcript (mRNA) for that gene was likely to have been present, absent, or marginal in the sample. A *present call* (P) indicates a gene for which there was enough transcript in the sample to quantify the abundance of that transcript to an acceptable degree of reliability. In most cases the genes that are so designated can be considered to be expressed. On the other hand, an *absent call* (A) indicates the exact opposite. This does not necessarily imply that the transcript was absent in the sample but rather that the amount of transcript could not be established reliably. A *marginal call* (M) indicates that the detectable level of transcript for that gene was close to negligible.

One simple way to assign an absolute call is to calculate the t statistic (see Chapter 7):

$$T_S = \frac{S_g}{\text{se}(S_g)},$$

and check whether it exceeds a specific cutoff.

Affymetrix's algorithm is based on a rank-based statistic. For probe set g, each probe pair, i, is assigned a score R_{gi}:

$$R_{gi} = \frac{PM_{gi} - MM_{gi}}{PM_{gi} + MM_{gi}}.$$

A probe pair whose PM_{gi} intensity greatly exceeds its MM_{gi} intensity will have an R_{gi} score close to unity, whereas a probe pair whose PM_{gi} and MM_{gi} intensities are roughly similar will have an R_{gi} score close to zero. The scores for a probe set are then ranked from 1 to m_g according to their distance from r, a specified low threshold (e.g., 0.15). The sum, R_{g+}, of the ranks of all probe pairs whose R_{gi} exceeds r is the one-sided *Wilcoxon's signed rank test* statistic for determining whether the R_{gi} scores are consistently below r. The p-value, p_a, associated with this statistic can be obtained. A probe set that has many R_{gi} scores near unity and is therefore considered more reliable will have a large R_{gi} and therefore a low (more significant) p-value, whereas one with many R_{gi} scores near zero will have a small R_{gi} and therefore a high (less significant) p-value. An absolute call is then made based on this p-value. For example, if $p_a < 0.04$, a present call is made; if $0.04 < p_a < 0.06$, a marginal call is made; and if $p_a > 0.06$, an absent call is made.

Absolute calls are often used for gene-filtering purposes.

SUPPLEMENTARY READING

The ScanAlyze user manual written by M. Eisen (available online at `http://www.microarrays.org/software.html`), user manuals of other image processing software, such as Genepix, QuantArray, Genespring, and Imagene, and the Affymetrix white papers (available online at `http://www.affymetrix.com/index.affx`) provide useful information about the topics covered in this chapter.

EXERCISES

Use the dataset `p4.txt` that is found in the book's Web site. This dataset consists of 432 genes by 14 columns. The first two columns are labeled *row* and *column* representing the positions of the spots on the slide. The next twelve columns are labeled $S1, \ldots, S6$, $B1, \ldots, B6$, representing data from six microarrays. The column labeled $S1$ has the signal data and the column labeled $B1$ has the background data for microarray number 1. In the same way Si and Bi correspond to the ith microarray for $i = 1, \ldots, 6$. Each microarray is made of 36 rows and 12 columns that correspond to the positions of the spots on the chip.

4.1. Extract the data and place it in 12 separate arrays of dimension 36×12, six for signal arrays and six for background arrays. Verify that the array rows and columns are correctly placed by manually checking a few individual spots. As a check of the quality of the arrays, explore the individual arrays using image plots and compare the intensities and backgrounds using scatterplots. Try to evaluate the data quality with the information that you have collected at this stage.

4.2. Some of the six background images in the dataset may show that the corresponding sample is defective, from the point of view that each one may have one or more of the nonrandom patterns listed in Section 4.2. Identify the samples that you think are defective and that appear to pass the quality check. *Note*: If you use `R` or `SPLUS`, functions for performing several quality checking routines are available in the DNAMR package.

4.3. Use the complete spatial randomness (CSR) criterion to check for nonrandom arrays within the set, and compare the results of the test to the results that you obtained in Problem 4.2.

4.4. Perform a background correction of the six arrays in two ways:

a. Take the difference between the signal and background and take the ratio between signal and background. Which one gives the most reasonable results?

b. Draw pairwise scatterplots of the microarrays before and after you remove the background, and use the graph to justify whether or not to adjust and which way is better.

4.5. Another way to remove the background is to smooth the background using a spatial smoother. You may use the function provided for this purpose in our `R/SPLUS` library DNAMR.

a. Once the background intensities have been smoothed, repeat the same tasks as in Problem 4.4, but using the smoothed background.

b. Write a brief summary of the results of both problems. State which background correction method appears to work best and why.

Exploration and Analysis of DNA Microarray and Protein Array Data
Edited by DHAMMIKA AMARATUNGA and JAVIER CABRERA

CHAPTER 5

Preprocessing Microarray Data

Once the experiment has been run and spot intensity data collected, it is necessary to preprocess this data prior to formally analyzing it. Preprocessing is needed to address several data-related issues:

1. To transform the data into a scale suitable for analysis
2. To remove the effects of systematic sources of variation
3. To identify discrepant observations and arrays

Preprocessing can greatly enhance the quality of any downstream analyses. We will now discuss each of these issues in turn.

Example. The methods in this chapter will be illustrated using Example E5, which is data from an experiment involving 10 pairs of microarrays, C1A, C1B, C2A, C2B, ..., C10A, C10B. Each pair of microarrays corresponds to a single mRNA sample (labeled C1, C2, ..., and C10), which was taken from a mouse following treatment and hybridized to two separate microarrays (labeled A and B). The two microarrays in each pair are technical replicates as they are exposed to the same biological sample. The five mice from which samples C1 to C5 were drawn are controls, so they are biological replicates, while each of the other five was treated with one of five drugs. There were 3300 genes arrayed on the microarrays.

5.1 LOGARITHMIC TRANSFORMATION

Often spot intensity data is initially transformed for analysis by a *logarithmic transformation*, $X \rightarrow \log(X)$. It is preferable to work with logged intensities

Exploration and Analysis of DNA Microarray and Protein Array Data
By Dhammika Amaratunga and Javier Cabrera
ISBN 0-471-27398-8

rather than absolute intensities for a number of reasons: the variation of logged intensities tends to be less dependent on the magnitude of the values, taking logs reduces the skewness of highly skewed distributions, and taking logs improves variance estimation.

Moreover logged intensities facilitate visual inspection of the data. The raw data is often very heavily clumped together at low intensities followed by a very long tail. More than 75% of the data may lie in the lowest 10% range of intensities. The details of such configurations are impossible to discern. After the log transformation the data is spread out more evenly, making it easier to examine visually.

Often logarithms of base 2 are used.

Other simple *power transformations* (i.e., transformations of the form $X \to X^{\beta}$ for some $\beta > 0$) have been found to be useful for certain datasets (e.g., Amaratunga and Cabrera, 2001a, 2001b, use a square root transformation: $X \to \sqrt{X}$ and Tusher et al., 2001, use a cube root transformation: $X \to X^{1/3}$), but the log transformation is, by far, the most widely used.

Example. Figure 5.1 shows a histogram and normal probability plot of the data for C1A before and after log transformation. It is clear that the transformation has greatly reduced the skewness of the distribution but it has not eliminated it altogether. We will use the log transformed data in the remainder of this chapter.

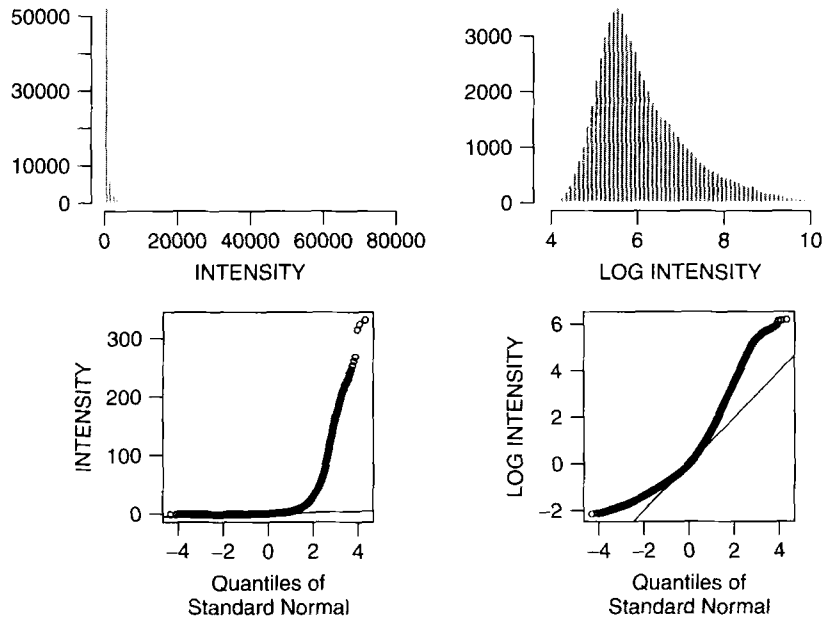


Figure 5.1 Histograms and normal probability plots of spot intensities before and after log transformation. The straight lines in the normal probability plots are identity lines.

5.2 VARIANCE STABILIZING TRANSFORMATIONS

Several data analysts observed that more complex transformations, such as the *started log transformation*, $X \rightarrow \log(X + c)$, appeared to better achieve the dual objective of symmetrizing the spot intensity data and stabilizing their variances (e.g., Sapir and Churchill, 2000, use such a transformation).

The rationale for this was investigated in greater detail by Rocke and Durbin (2001) using data from experiments involving arrays with replicate spots. In analogy with models used for estimating the actual concentration of an analyte in a sample for a given response, they found that it was appropriate to model spot intensity data as

$$X = \alpha + \mu e^{\eta} + \varepsilon,$$

where α is the mean background, μ is the true expression level, and the terms η and ε represent normally distributed error terms with mean zero and variances σ_η^2 and σ_ε^2 respectively. Spot intensity data often manifests the distributional features implied by this model:

- At very low expression levels, where μ is close to zero, the measured spot intensity is dominated by the first term in the model, so that $X \cong \alpha$, and X is approximately normally distributed with mean α and variance σ_ε^2.
- At very high expression levels, where μ is large, the measured spot intensity is dominated by the second term in the model, so that $X \cong \mu e^{\eta}$, and X is approximately lognormally distributed with variance $\mu^2 S_\eta^2$, where $S_\eta^2 = e^{\sigma_\eta^2}(e^{\sigma_\eta^2} - 1)$. Thus the variance of X varies linearly with μ^2. However, on the log scale, $\log(X) \cong \log(\mu) + \eta$, indicating that the variance of $\log(X)$ is constant.
- At moderate expression levels, the measured spot intensity is in between the two extremes above and behaves as a mixture of a normal and lognormal distribution with variance $\mu^2 S_\eta^2 + \sigma_\eta^2$, which, again, varies with μ.

Durbin et al. (2002) showed that the *generalized log transformation*

$$X \rightarrow \log\left((X - \alpha) + \sqrt{(X - \alpha)^2 + \left(\frac{\sigma_\varepsilon^2}{S_\eta^2}\right)}\right)$$

stabilizes variance in that the transformed data has constant variance equal to S_η^2. A similar transformation was suggested by Huber et al. (2002).

In order to apply this transformation, the parameters α, σ_η^2, and σ_ε^2 must be estimated from the spot intensity data. If replicate blanks or negative controls are available, the background parameters α and σ_ε^2 can be estimated as their mean and variance. If not, they can be estimated as the mean and variance of a

set of unexpressed genes. The parameter σ_η^2 can be estimated as the mean and variance of a set of highly expressed genes. Details of the procedure are provided by Rocke and Durbin (2001).

Unfortunately, by using the generalized log transformation, the convenient interpretation of log ratios as log fold changes (see Chapter 7), which is possible with an ordinary log transformation, is lost. Rocke and Durbin (2002) demonstrate that the started log transformation, $X \rightarrow \log(X + c)$, with $c = (\sigma_\varepsilon^2 / S_\eta^2) - \alpha$, is a reasonable compromise.

5.3 SOURCES OF BIAS

The complexities and intricacies of the microarray experimental process often introduce systematic effects into the intensity measurements. These effects can be substantial enough to dilute the effects that the experimenter is trying to detect. Among other sources of variability systematic effects have been attributed to:

> the concentration and amount of DNA placed on the microarrays, arraying equipment such as spotting pins that wear out over time, mRNA preparation, reverse transcription bias, labeling efficiency, hybridization efficiency, lack of spatial homogeneity of the hybridization on the slide, scanner settings, saturation effects, background fluorescence, linearity of detection response, ambient conditions.

In addition *dye bias* is present in almost all multichannel experiments. Generally, the Cy5 (red) intensities tend to be higher than the Cy3 (green) intensities but the magnitude of the difference generally depends on the overall intensity. The reason for the imbalance between the channels is the difference between the physicochemical properties of the dyes, the labeling efficiencies, and the scanning properties of the dyes and the scanner settings.

Some of these sources of variability can be controlled to a limited extent with due diligence on the part of the experimenter. However, few can be completely eliminated.

Because systematic variation will generally affect different microarrays to different extents, in order to be able to make valid comparisons across microarrays, we need to try and remove the effects of such systematic variations and bring the data from the different microarrays onto a common scale.

5.4 NORMALIZATION

Early microarray researchers noticed substantial differences in intensity measurements even among microarrays that were treated exactly alike. Differences still persist despite huge improvements in the technology, but their magnitude is

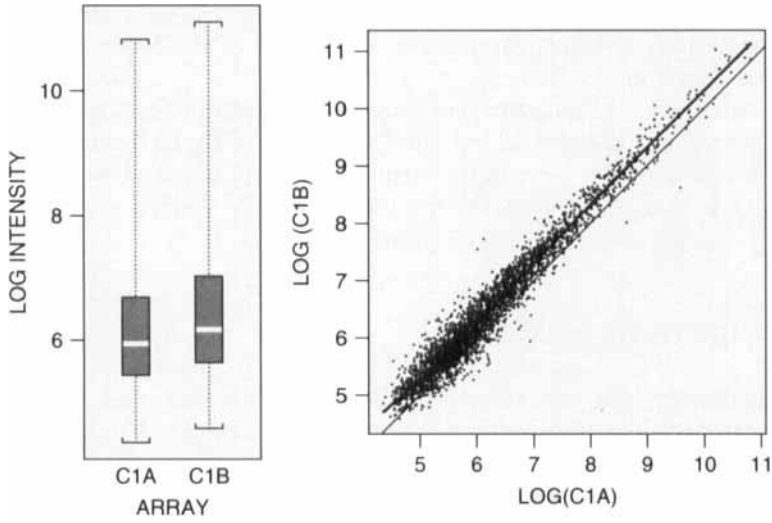


Figure 5.2 Side-by-side boxplot display and scatterplot of arrays C1A and C1B. The thinner line on the scatterplot is the identity line, and the thicker line is a smooth of the plot.

not as high as in the early days. The differences can generally be traced to systematic effects as described in Section 5.3 above. The purpose of *normalization* is to remove, by data processing, as much as possible, the effects of any systematic sources of variation. Normalization can be regarded as a sort of calibration process that improves the comparability among microarrays treated alike.

Example. In Figure 5.2 the data from microarrays C1A and C1B are plotted against one another. Although both were hybridized to the same sample, it is clear that the intensities are systematically higher in microarray C1B compared to microarray C1A. In Figure 5.3, the data from microarrays C1B and C5B are plotted against one another. These microarrays were hybridized to different samples, but because the samples were taken shortly after treatment, it is unlikely that more than a few genes would be differentially expressed in the two. Yet the plot shows most of the intensities are generally higher in microarray C1B compared to microarray C1A.

Early efforts at normalization used simple methods. One such method is *normalization by the sum.* In this method the sums of the intensities of the k microarrays being normalized are forced to be equal to one another. The rationale for doing this is that the total mRNA content should be roughly the same across samples. Suppose that the k original sums were $X_{1+}, \ldots, X_{k+}$. If we divide all the observations in the ith microarray by X_{i+}, their sum will be 1. Doing this for all the microarrays would make all the sums equal (to 1), as desired.

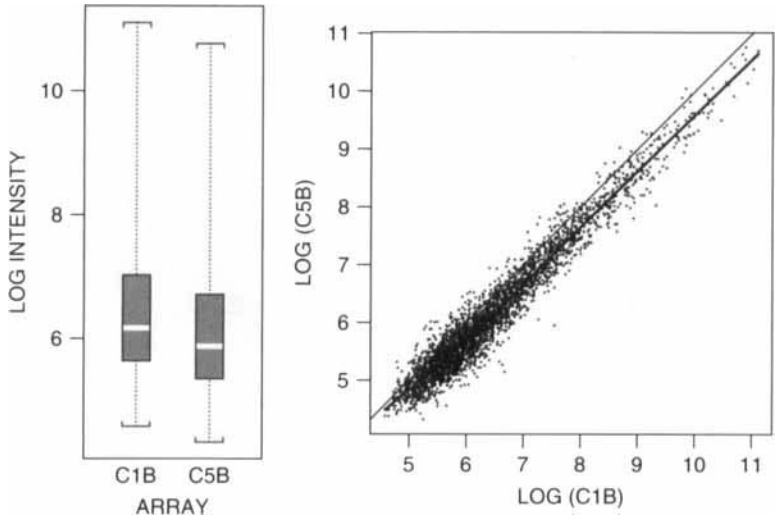


Figure 5.3 Side-by-side boxplot display and scatterplot of arrays C1A and C5B. The thinner line on the scatterplot is the identity line, and the thicker line is a smooth of the plot.

Example. Figure 5.4 shows the data from microarrays C1A and C1B plotted against each other after normalization by the sum. Now the observations are more in agreement.

An entirely equivalent method is *normalization by the mean*, in which the arithmetic means of the microarrays are equated. A similar, but not equivalent, idea is *normalization by the median*, in which the microarray medians are equated. *Q3 normalization*, in which third quartiles are equated, is on the same lines and is reasonable when it is expected that about half of the genes are unexpressed and the third quartile is then roughly the median intensity of the expressed genes.

All of these are examples of *global* or *linear normalization* schemes. The common feature of these normalization schemes is that they assume that the spot intensities on every pair of arrays being normalized are linearly related with no intercept so that the lack of comparability can be corrected by adjusting every single spot intensity on any microarray by the same amount, called the *normalizing factor*, regardless of its intensity level.

5.5 INTENSITY-DEPENDENT NORMALIZATION

The relationship between the spot intensities in Figure 5.2 is clearly nonlinear. It suggests that the factor necessary to adjust low-intensity measurements should be different from the factor necessary to adjust high-intensity measurements. In other words, an *intensity-dependent normalization* method, a nor-

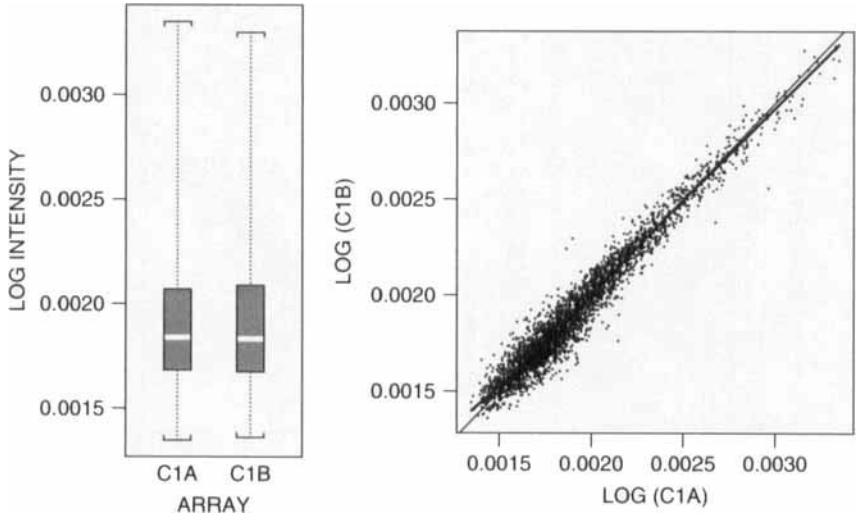


Figure 5.4 Side-by-side boxplot display and scatterplot of arrays C1A and C1B after mean normalization. The thinner line on the scatterplot is the identity line, and the thicker line is a smooth of the plot.

malization scheme in which the normalizing factor is a function of intensity level, should be preferable to any global normalization method. In intensity-dependent normalization the transformed spot intensity data is normalized using a nonlinear *normalization function*: $X \rightarrow f(X)$.

As it arises naturally by studying plots such as Figure 5.2, the need for intensity-dependent normalization was recognized independently by a number of different data analysts, including Amaratunga and Cabrera (2001a, b), Dudoit et al. (2002), Li and Wong (2001a), Schadt et al. (2001), and Yang et al. (2001). Other papers on this topic, including Astrand (2001), Bolstad et al. (2002), Colantuoni et al. (2002), Hoffman et al. (2002), Irizzary et al. (2002), Quackenbush (2002), Tseng et al. (2001) and Yang et al. (2002). Hoffman et al. (2002), and Yang et al. (2002), have demonstrated that normalization can have a profound effect on downstream analysis.

For an intensity-dependent normalization there must be a *reference* or *baseline microarray* to which all the microarrays are normalized. In the absence of a universal standard against which the arrays can be calibrated, this is usually some sort of average microarray, a mock array fashioned out of the averages of the arrays being normalized. One possibility is the *median mock array*. If X_{gi} denotes the transformed spot intensity measurement for the gth gene ($g = 1, \ldots, G$) in the ith microarray ($i = 1, \ldots, I$), the median mock array will have as its gth observation:

$$M_g = \text{median}\{X_{g1}, \ldots, X_{gI}\}.$$

Prior to constructing the reference microarray, it is generally a good idea to first perform a median or Q3 normalization so that all the microarrays are brought to a common overall level to start with and each can contribute to the construction of the reference microarray.

One key issue for any normalization is the selection of an *invariant gene set*, the subset of genes that will be used to estimate the normalization functions. This set of genes should exhibit the following characteristics:

1. Their expression levels should remain constant across the arrays being normalized so that they can be used to estimate the normalization functions.
2. Their expression levels should span the entire range of expression levels observed in the experiment so that it will not be necessary to extrapolate the estimated normalization functions.
3. The normalization relationship for these genes should be representative of the normalization relationship for the all the genes so that they can be used to normalize all.

The invariant gene set could be:

- *Control genes.* A small number of DNA sequences could be specially arrayed onto the microarray specifically for normalization purposes. Synthetic or cross-species DNA sequences have been used for this purpose. Then, if necessary, DNA sequences complementary to these sequences would be spiked into the probe at a known concentration. One concern with this procedure is whether characteristic 3 is satisfied.
- *Housekeeping genes.* A small number of housekeeping genes could be arrayed onto the microarray. If it can be assumed that these genes express at about the same level across the set of arrays being normalized, these genes will form an invariant gene set. However, a number of the more commonly used housekeeping genes have been found to express differentially across various samples, so whether they satisfy characteristic 1 is debatable. If they are all low to moderate expressing genes, they also may not adequately satisfy characteristic 2.
- *Unchanging genes.* Metrics from the raw data could be used to select a subset of genes that appear to be the least likely to be differentially expressed. One way to do this is to rank the spot intensities on each array, including the reference microarray, from smallest to largest, and then to select as the invariant gene set those genes whose ranks across the microarrays are the least different from the reference microarray. If the gene set is not carefully chosen, characteristics 1 and 2 may not be adequately satisfied.
- *All the genes on the array.* It is reasonable to expect that only a very small percentage of the genes will be differentially expressed across the arrays

being normalized, as is the case with many microarray experiments. Then the entire set of genes on the microarray can be used as the invariant gene set, since most normalization schema are robust to small perturbations. This assumption will be more realistic (and characteristic 1 more likely to be satisfied), the larger the number of genes on the arrays and the smaller the percentage of genes differentially expressed across the arrays being normalized.

As with global normalization, intensity-dependent normalization can be performed in several different ways.

5.5.1 Smooth Function Normalization

In *smooth function normalization* each microarray is normalized as follows: First, the inverse, $g_i = f_i^{-1}$, of the monotone normalization function, f_i, for the ith microarray, is estimated by fitting the model

$$X_{gi} = g_i(M_g) + \varepsilon_{gi},$$

where ε_{gi} is a random error term, to the (X_{gi}, M_g) data, for the invariant gene set. The normalized values for the ith microarray are then obtained from

$$X'_{gi} = f_i(X_{gi}).$$

In *spline normalization*, the function g_i is a smooth but flexible function such as a cubic spline with a small number (e.g., 7) of degrees of freedom; the smaller the degrees of freedom, the smoother is the fit. In *lowess normalization*, the function g_i is estimated by fitting a lowess smooth (Cleveland, 1979) to the invariant gene set. The lowess smooth is essentially a series of locally linear fits, each fitted robustly so as to limit the influence of outliers. A user-specified parameter, *span*, denotes the fraction of data (e.g., span $= \frac{1}{3}$) used for smoothing at any data point; the larger it is, the smoother the fit. Note that neither of these methods is affected by a small percentage of outliers. Alternative smoothers such as a multilinear continuous function, a piecewise running median or kernel-based methods may also be used.

Example. Figure 5.5 shows the data from microarrays C1B and C5B plotted against each other after spline normalization. As these are biological replicates and, other than natural variability, no differential expression was expected between the two, all 3300 genes were used as the invariant gene set. The observations are now in good agreement.

5.5.2 Quantile Normalization

The objective of quantile normalization is to make the distributions of the transformed spot intensities, $\{X_{gi}\}$, as similar as possible across the micro-

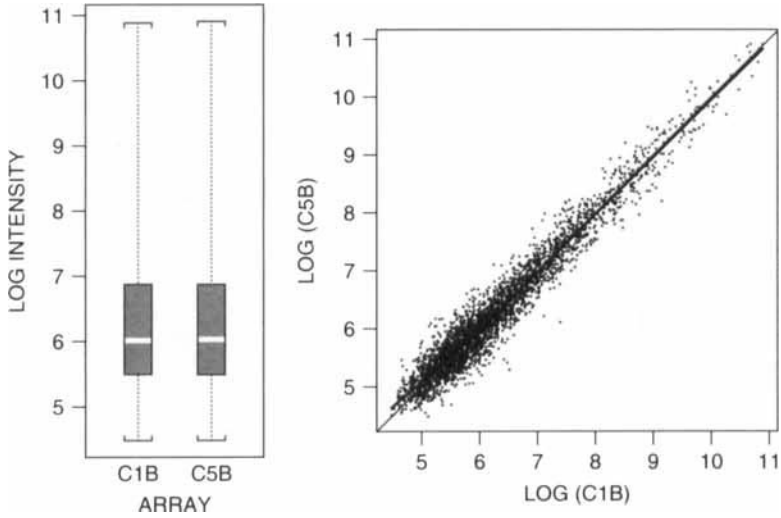


Figure 5.5 Side-by-side boxplot display and scatterplot of arrays C1A and C5B after spline normalization. The thinner line on the scatterplot is the identity line, and the thicker line is a smooth of the plot.

arrays, or, at least as similar as possible to the spot intensity distribution of the median mock array. Either a subset of quantiles or all the quantiles may be equated.

To equate a subset of quantiles, say the percentiles, as in Amaratunga and Cabrera (2001b), calculate the percentiles $(Q_{i0}, \ldots, Q_{i100})$ of the ith array and the percentiles $(Q_{M0}, \ldots, Q_{M100})$ of the median mock array. For any value X_{gi}, find the interval, $[Q_{ih}, Q_{i(h+1)}]$, to which it belongs and obtain its normalized value, X'_{gi}, by linearly interpolating between the pair of points (Q_{Mh}, Q_{ih}) and $(Q_{M(h+1)}, Q_{i(h+1)})$.

Bolstad et al. (2002) give the following algorithm to equate all the quantiles: Arrange the transformed spot intensity $\{X_{gi}\}$ into a $G \times I$ matrix X. Sort each column of X to give X_{sort}. Take the means across the rows of X_{sort} and assign this mean to each element in the row to get X_{*sort}. Obtain the normalized version X' of X by rearranging each column of X_{*sort} to have the same ordering as the original X.

Quantile normalization is useful for normalizing across a series of conditions where it is believed that a small but indeterminate number of genes may be differentially expressed, yet it can be assumed that the distribution of spot intensities does not vary too much.

Example. Figure 5.6 shows the data from microarrays C1B and C5B plotted against each other after quantile normalization with, again, all the genes used as the invariant gene set. As with spline normalization, the observations are in agreement. In fact both methods appear to perform similarly.

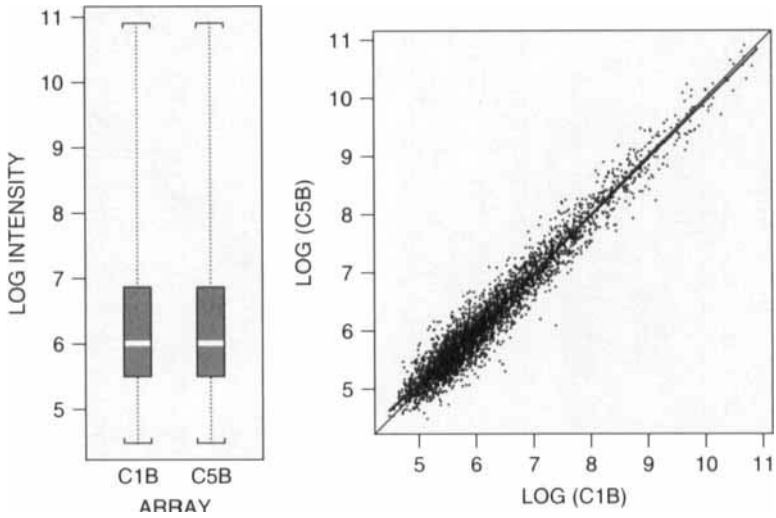


Figure 5.6 Side-by-side boxplot display and scatterplot of arrays C1A and C5B after quantile normalization. The thinner line on the scatterplot is the identity line, and the thicker line is a smooth of the plot.

5.5.3 Normalization of Oligonucleotide Arrays

Oligonucleotide arrays can be normalized using any of the methods described above. Normalization can be carried out either at the probe level or at the signal level. Using data from a spike-in experiment, Irizarry et al. (2002) demonstrate that probe level normalization is the more effective, as it reduces bias and variability with the benefits carrying over to the expression level.

5.5.4 Normalization of Two-Channel Arrays

Consider the log-transformed spot intensities, $\{X_{gR}\}$ and $\{X_{gG}\}$, for the channels of a two-channel array, where the letters R and G refer to the colors, red and green respectively, that are typically used to label the channels. If there is no systematic dye bias, the data points on a scatterplot of X_{gR} versus X_{gG} should generally lie along the $Y = X$ line. If this is not the case, then it is necessary to normalize the two channels.

Yang et al. (2001) argue that it is easier to assess this with an *MVA plot*, a scatterplot of M_g versus A_g, where $M_g = X_{gR} - X_{gG}$ and $A_g = (X_{gR} + X_{gG})/2$. Here $\{A_g\}$ is analogous to the median mock array that was used as the reference array above. If there is no systematic dye bias, the points on the MVA plot would be scattered around the $M = 0$ line. Otherwise, normalization can be done using any of the methods described above.

For an intensity-dependent normalization the normalization function is fitted to the MVA plot, and the fitted values, $\hat{M}_g$, which function as the nor-

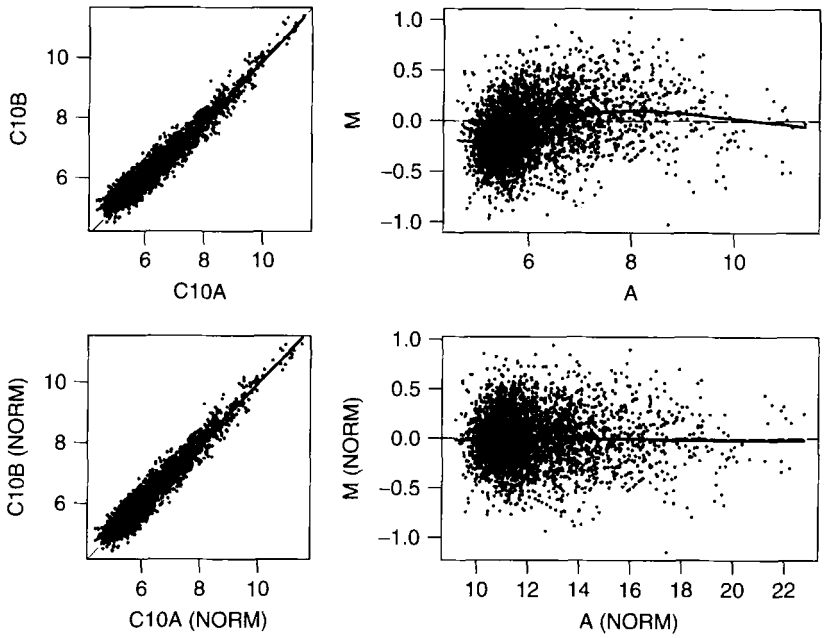


Figure 5.7 Scatterplot and MVA plot of arrays C10A and C10B before and after lowess normalization. The thinner line on the scatterplots is the identity line (in the plots on the left) or the zero line (in the plots on the right), and the thicker line is a smooth of the plot.

malization adjustments, are calculated. The normalized values are taken to be $X'_{gR} = X_{gR} - \hat{M}_g/2$ and $X'_{gG} = X_{gG} + \hat{M}_g/2$.

After normalization, the expression ratios, $R_g = \exp(X'_{gR} - X'_{gG})$, should be scattered around unity if few genes are differentially expressed across the two channels.

Example. Figure 5.7*a* shows the data from microarrays C10A and C10B plotted against each other. Even though the arrays appear to be reasonably in agreement, Figure 5.7*b*, the MVA plot shows more clearly that they are not. A lowess normalization with span $= \frac{1}{3}$, with all the genes used as the invariant gene set, produces normalized arrays that are more in agreement with each other, as is clear from Figures 5.7*c* and 5.7*d*. Figure 5.8 shows histograms of the expression ratios before and after the normalization. It can be observed that this distribution is more centralized at the null value of unity after normalization as no genes should be differentially expressed between C10A and C10B.

5.5.5 Spatial Normalization

Sometimes the arraying equipment or the experimental conditions can introduce systematic spatial effects within a single slide. In such cases a within-slide normalization should be considered (Colantuoni et al. 2002; Schuchhardt et al.

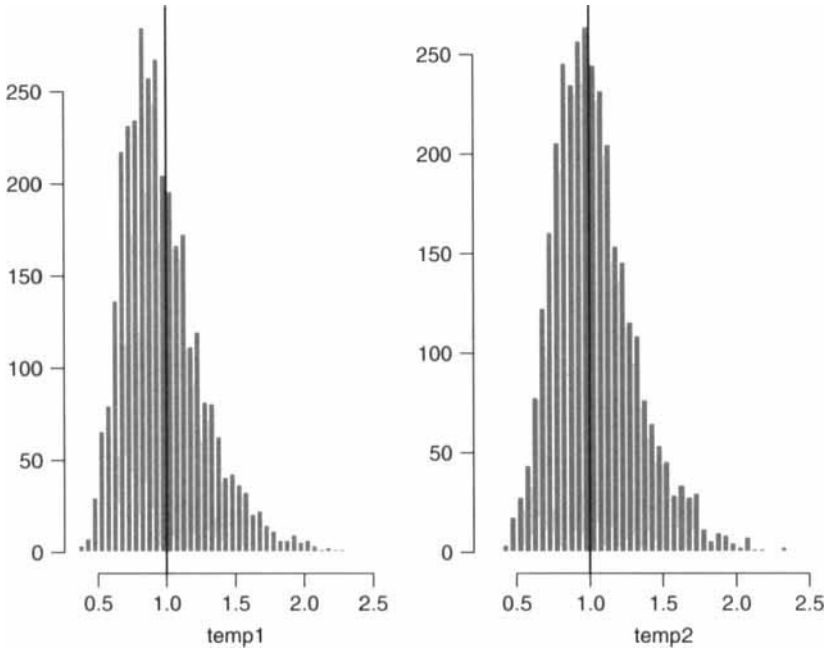


Figure 5.8 Histogram of gene expression ratios for array C10A and C10B before and after lowess normalization. The vertical line is the unit line.

2000; Yang et al. 2001). This can be done by subdividing the slide into a grid. A natural grid is the grid determined by the print tip of the arrayer. Normalization across the subsections of the grid can be done using any of the methods described above. Care must be taken not to normalize out array defects and other artifacts.

5.5.6 Stagewise Normalization

When the data includes both technical replicates as well as biological replicates, it is most effective to carry out the normalization in stages. The technical replicates can be normalized using smooth function normalization and the biological replicates can be normalized using quantile normalization. If the biological replicates fall into groups, such as treatment groups, each group can be normalized separately using quantile normalization, and then all the arrays in all the groups can be normalized across all the arrays using quantile normalization. Figure 5.9 is a schematic of a stagewise normalization.

Example. Figure 5.10 shows the results of a stagewise normalization. Figure 5.10*a* shows a side-by-side boxplot display of the data before any normalization is done (S0). Figure 5.10*b* is the data after normalizing the technical repli-

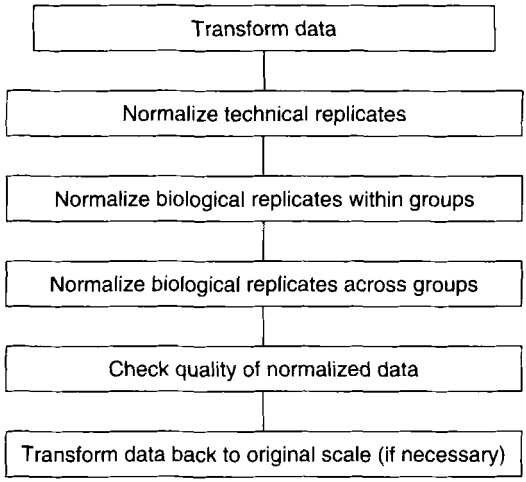


Figure 5.9 Schematic of a stagewise normalization.

cates via spline normalization (S1). Figure 5.10*c* is the data after normalizing the control biological replicates via quantile normalization (S2). Figure 5.10*d* is the data after normalizing across all 20 microarrays via another quantile normalization (S3).

5.6 JUDGING THE SUCCESS OF A NORMALIZATION

Consider the normalization of two arrays, whose log-transformed spot intensities are $\{Y_{g1}\}$ and $\{Y_{g2}\}$. A normalization based on a monotone normalizing procedure (as those described in Section 5.5) will be truly successful in bringing them into agreement only if they are, more or less, monotonically related to each other. Whether this holds for $\{Y_{g1}\}$ and $\{Y_{g2}\}$ can be assessed by calculating their *Spearman's rank correlation coefficient*:

$$\hat{\rho}_S = \frac{12\sum_{g=1}^{G}\{R_{g1} - \frac{1}{2}(G+1)\}\{R_{g2} - \frac{1}{2}(G+1)\}}{G(G^2-1)},$$

where R_{gi} is the rank of Y_{gi} when the $\{Y_{gi}\}$ are ranked from 1 to G.

Spearman's rank correlation coefficient is a measure of monotone (not necessarily linear) association between two variables. The value of $\hat{\rho}_S$ lies in between -1 and $+1$, with values close to $+1$ indicating that the two sets of values are positively associated to each other, values close to -1 indicating that the two sets of values are negatively associated to each other, and values close to 0 indicating that the two sets of values not associated with each other. Thus, if $\hat{\rho}_S$ is high (i.e., close to one), it is likely that a normalization of the sort

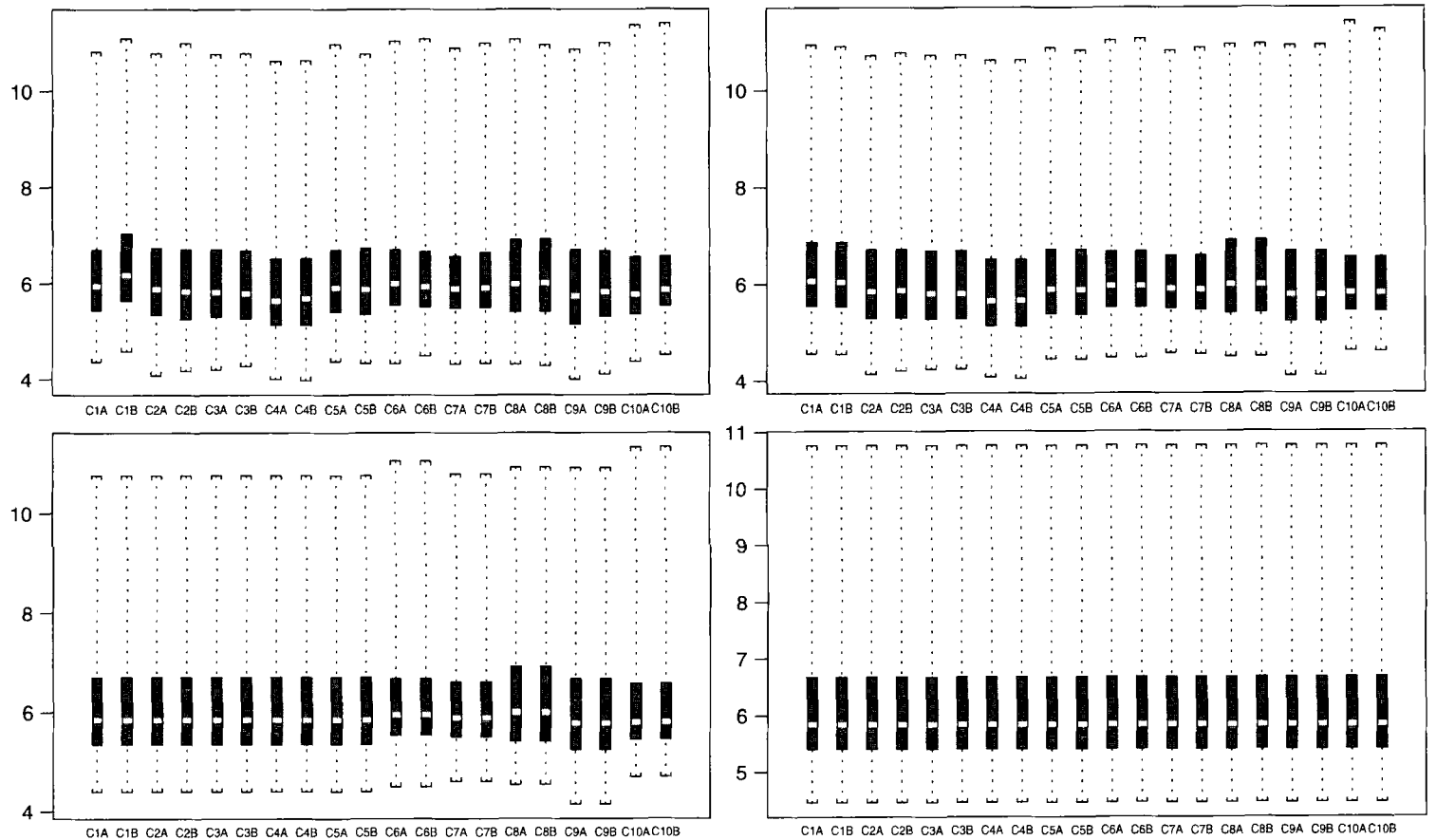


Figure 5.10 Side-by-side boxplot displays of arrays C1A and C10B at various stages of stagewise normalization (S0) before normalization, (S1) after normalization of technical replicates, (S2) after normalization of biological replicates in control group, and (S3) after quantile normalization.

described above would be able to bring the two sets of values into agreement, whereas, if $\hat{\rho}_S$ is low (i.e., much lower than one), it is unlikely that a normalization of the sort described above would be able to bring the two sets of values into agreement.

The value of Spearman's rank correlation coefficient is unchanged by a monotone normalization procedure. Therefore, while it is a good measure of whether a normalization would be successful, it cannot be used to judge the success of a monotone normalization procedure.

Instead, once a normalization has been performed, the degree of success of the normalization can be assessed via the *concordance correlation coefficient* (Lin, 1989), an index that quantifies the degree of agreement between two sets of numbers. The concordance correlation coefficient, $\hat{\rho}_c$, is defined as

$$\hat{\rho}_c = \frac{2s_{12}}{s_1^2 + s_2^2 + (\bar{Y}_1 - \bar{Y}_2)^2},$$

where $\bar{Y}_c = \sum_{g=1}^{G} Y_{gc}/G$ and $s_c^2 = \sum_{g=1}^{G}(Y_{gc} - \bar{Y}_c)^2/G$ are, respectively, the mean and variance of the cth microarray ($c = 1, 2$) and $s_{12} = \sum_{g=1}^{G}(Y_{g1} - \bar{Y}_1) \cdot (Y_{g2} - \bar{Y}_2)/G$ is the covariance. ρ_c is a standardized measure of $\mathrm{E}[(Y_{g1} - Y_{g2})^2]$ and $\rho_c = 1$ if and only if $\{Y_{g1}\}$ and $\{Y_{g2}\}$ are in perfect agreement. Otherwise, $\rho_c < 1$.

Spearman's rank correlation coefficients and concordance correlation coefficients can be used together to assess the need for normalization:

- If, for a pair of arrays, $\hat{\rho}_c$ is very high (as a rough rule of thumb, greater than 0.99), normalization may not be necessary.
- On the other hand, if $\hat{\rho}_c$ is not very high and $\hat{\rho}_S$ is high (as a rough rule of thumb, greater than 0.8), indicating a monotone, but not strongly concordant relationship, normalization is very likely to be highly beneficial.
- When both $\hat{\rho}_c$ and $\hat{\rho}_S$ are low, indicating that the relationship between the arrays is not strong, it may be worth looking further to see whether there was a problem with either of the arrays before doing any normalization.

When normalizing across a series of arrays, it is instructive to display, on image plots, the pairwise Spearman's rank correlation coefficients (the resulting display is called a *Spearman map*) and the pairwise concordance correlation coefficients (the resulting display is called a *concordance map*).

Example. Figure 5.11*a* shows a Spearman map of the data. It shows a strong monotone relationship among the 10 control replicates and also among 2 other technical replicate pairs. All the Spearman's rank correlation coefficients are greater than 0.85, indicating that the arrays can be normalized. Figure 5.11*b* shows a concordance map of the data, with several concordance correlation coefficients below 0.9, so there is a need for a normalization. Figure 5.11*c*

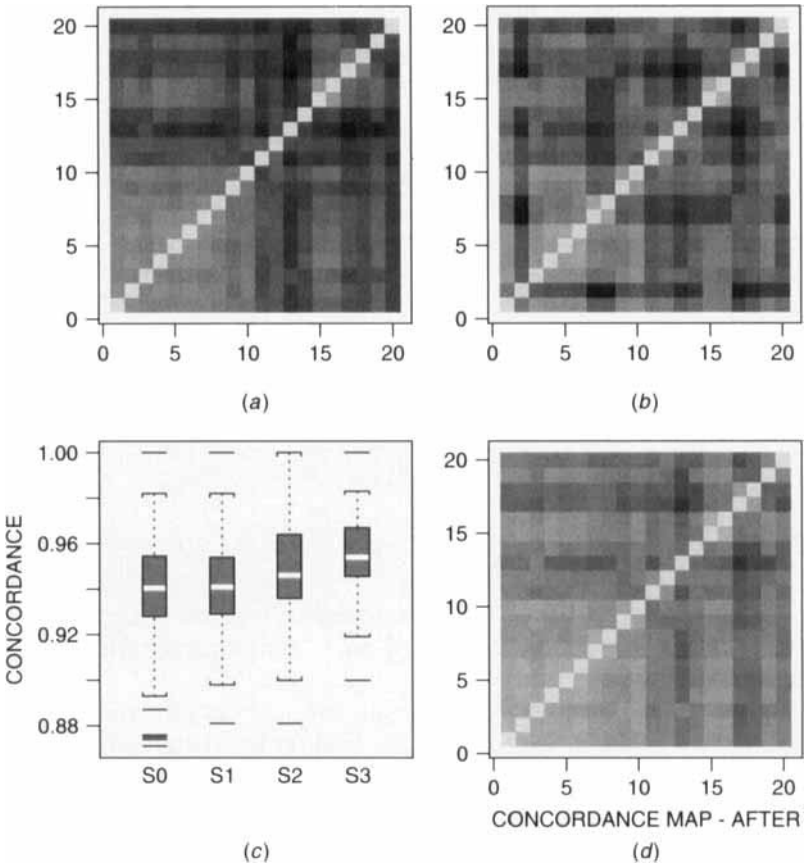


Figure 5.11 (*a*) Spearman map, (*b*) concordance map before normalization, (*c*) concordance correlation coefficients at various stages of normalization, and (*d*) concordance map after normalization [on gray scale, low to high correlations go from black to white].

shows a side-by-side boxplot display of the concordance correlation coefficients after each stage of the stagewise normalization (S0 to S3 are defined in Section 5.5.6). Figure 5.11*d* shows a concordance map after the complete normalization; all the concordance correlation coefficients are now above 0.9. Observe how the concordance improves at each stage, culminating in the substantially higher concordance in the control group after normalization.

Some comments regarding various correlation coefficients:

1. If the distributional properties of the values change substantially during a normalization (e.g., the skewness is decreased), it is possible that the concordance correlation coefficients might increase, but this may only be an artificial improvement.

2. The more familiar Pearson's correlation coefficient,

$$\hat{\rho} = \frac{s_{12}}{s_1 s_2},$$

measures how close $\{Y_{g1}\}$ and $\{Y_{g2}\}$ are to linearity rather than to agreement.

3. For a pair of microarrays that have been normalized by equating all the quantiles, the concordance correlation coefficient will equal Pearson's correlation coefficient. This is because, after such a normalization, the quantiles of both microarrays are identical, and therefore both means are equal, $\bar{Y}_1 = \bar{Y}_2$, as are both variances, $s_1^2 = s_2^2$.
4. Spearman's rank correlation coefficient is equal to (a) Pearson's correlation coefficient calculated on the ranks of the data and (b) the concordance correlation coefficient calculated on the ranks of the data.

5.7 OUTLIER IDENTIFICATION

Outliers are observations that appear to be inconsistent with the majority of the data. When there are replicate arrays, the replicates could be used to identify discrepant spot intensities in the data. Let X_{gi} denote the transformed and normalized spot intensity measurement for the gth gene on the ith array. An *outlier* is an observation, X_{gj}, that is substantially different from a majority of the other values X_{gi} for that same gene. The same observation may or may not have been discovered as an unusual observation in the spot quality checks performed at the preprocessing stage (see Section 5.2.6).

Many ways of identifying outliers in replicate observations have been suggested (Barnett and Lewis, 1994, provide an extensive review of the vast literature on this topic).

5.7.1 Nonresistant Rules for Outlier Identification

Some common approaches to outlier identification are as follows:

The z-score rule (Grubbs' test). Calculate a *z-score*, z_{gi}, for every observation:

$$z_{gi} = \frac{X_{gi} - \bar{X}_g}{s_g},$$

where $\bar{X}_g$ and s_g are the mean and standard deviation of the gth gene. Call X_{gj} an outlier if $|z_{gj}|$ is large, say, greater than five.

The *CV* rule. Call the furthest observation X_{gj} from the mean, $\bar{X}_g$, an outlier if the coefficient of variation, $CV_g = s_g/\bar{X}_g$ exceeds some prespecified cutoff.

Neither of these rules is a particularly reliable tool for detecting outliers. Their most serious drawback is that they are based on statistics that are themselves influenced by outliers. For example, if there is a large outlier in the data, both the mean and the standard deviation will be inflated by its influence and both the z-score of the outlier itself and the CV will appear normal. This phenomenon is known as *masking*, an outlier remaining undetected because it is hidden either by itself or by some other, usually adjacent, outliers. A related effect is *swamping*, which happens when a normal observation is classified as an outlier because of the presence of an unrelated outlier or outliers.

5.7.2 Resistant Rules for Outlier Identification

As in the discussion above, the outlier detection method must be based on statistics that are *resistant* to outliers, meaning not influenced by them, such as the median and the MAD (median absolute deviation from the median and scaled to be consistent at the normal distribution). Both measures can tolerate up to almost 50 percent outliers without being affected. Thus a more reliable outlier detection rule is as follows:

The resistant z-score rule. Calculate a *resistant z-score*, z^*_{gi}, for every observation

$$z^*_{gi} = \frac{X_{gi} - \tilde{X}_g}{\tilde{s}_g},$$

where $\tilde{X}_g$ and $\tilde{s}_g$ are the median and MAD of the gth gene. Call X_{gj} an outlier if $|z^*_{gj}|$ is large, say, greater than five.

One remaining problem is that microarray experiments usually have little replication. With very few replicates, the median and MAD (in particular, the latter) are not dependable estimates of the location and scale of the data. The estimation of the scale can be improved by observing that with microarray data, there is a relationship between the median and the MAD across all the genes. Assuming that this is a true relationship, $\sigma_g^2 = f(\mu_g)$, use it to calculate a smoothed version of MAD, $\widetilde{MAD}_g$, that will be more stable as it "borrows strength" from similar expressing genes. To calculate $\widetilde{MAD}_g$, first calculate the absolute deviations from the median: $AD_{gi} = |X_{gi} - \tilde{X}_g|$. Then run a smoother, such as a smoothing spline, through the relationship of AD_{gi} versus $\tilde{X}_g$, and use the fitted value, $\widetilde{MAD}_g$, as an estimator of scale for the gth gene. The following revised rule can be used to identify outliers:

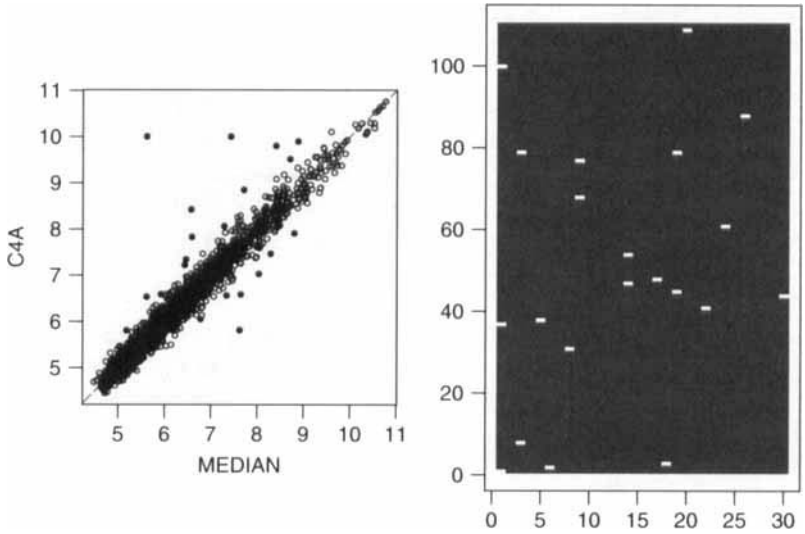


Figure 5.12 Scatterplot of (normalized) array C4A with the outliers plotted as filled circles and an image plot of the array showing, in white, the positions of the outliers.

The revised *z*-score rule. Calculate a *revised z-score*, z_{gi}^{**}, for every observation

$$z_{gi}^{**} = \frac{X_{gi} - \tilde{X}_g}{\tilde{s}_g'}.$$

Call X_{gj} an outlier if $|z_{gj}^{**}|$ is large, say, greater than five.

Example. When the control group data (arrays C1A to C5B) were screened for outliers using the revised *z*-score rule with a critical value of five, 119 observations were designated as outliers. None of the arrays had a substantially higher proportion of outliers than the others. Therefore there is no evidence that any of the arrays is particularly different from the rest. The outliers found on array C4A are shown in Figure 5.12, both on a scatterplot of C4A versus the median mock array and on an image plot of the array. The latter shows that the outliers are randomly scattered through the array. Therefore, there is no evidence that C4A has any spatial problems.

5.8 ASSESSING REPLICATE ARRAY QUALITY

Methods for assessing the quality of a single array or a series of arrays were discussed in Section 4.2. Replicate arrays (e.g., the set of arrays in the control

group in the example) can be also used to judge array quality by seeing whether any of them is different from the others:

- If the Spearman correlation coefficients between one of the arrays and each of the other arrays is substantially lower than the other Spearman correlation coefficients, then that array is suspect.
- When a procedure in Section 5.7 discovers substantially more outliers on one of the arrays more than on any of the other arrays, then that array is suspect.

In addition, when there are groups of arrays, such as one group of arrays hybridized to control samples and another group of arrays hybridized to treatment samples, the extent of the differences can be roughly assessed from a Spearman map or a concordance map (e.g., the lighter 10×10 square area in the lower left quadrant of Figs. 5.9*a* and 5.9*d*, separating out the control group). The more obvious the separation, the more substantive will be the difference between groups. However, small and subtle differences between groups are not be evident in these displays.

EXERCISES

5.1. A crude, but resistant, estimate of the skewness of a distribution is

$$K = \frac{Q_3 - M}{M - Q_1},$$

where Q_1, M, and Q_3, are the first quartile, median, and third quartile respectively. Calculate the value of K for the data from C1A in dataset E5, before transformation and after transformation by $X \to \log(X - c)$, for $c = 0, 10, \ldots, 50$. Comment.

5.2. Construct a MVA plot for the data from arrays C9A and C9B in dataset E5. Normalize the data using a spline normalization, and redraw the MVA plot. Compare the gene expression ratios before and after normalization. Comment.

5.3. Consider the data from C1A, C1B, C10A, and C10B in dataset E5. Carry out a stagewise normalization for this data. The average expression level in the "control group" can be estimated as the averages of C1A and C1B. The average expression level in the "treated group" can be estimated as the averages of C10A and C10B. Plot these averages against each other before and after normalization. Comment.

5.4. Determine whether there any outliers in the following:

a.
```
10.07 10.11 10.27 10.10  9.88
 9.99 10.13  9.76  9.22 10.04
```

b.
```
10.07 10.11 10.27 10.10  9.88
 9.99 10.13  9.76  2.22 10.04
```

using **(i)** the z-score rule and **(ii)** the resistant z-score rule.

5.5. Average the data for each mouse, for example, $C1 = (C1A + C1B)/2$. Identify any outliers in the set of averages for C1, C2, C3, C4, C5 using **(a)** the z-score rule **(b)** the resistant z-score rule. Is there any evidence that any of the mice is an outlier?

5.6. Use the postnormalization MVA plot in Problem 5.2 to determine whether there are any discrepant pairs of observations between arrays C9A and C9B.

Exploration and Analysis of DNA Microarray and Protein Array Data
Edited by DHAMMIKA AMARATUNGA and JAVIER CABRERA

CHAPTER 6

Summarization

Once the spot intensity data have been preprocessed, statistical estimation techniques can be applied to summarize the data across replicates and determine the expression level of each gene.

6.1 REPLICATION

Replication, the execution of an experiment on more than one unit, is an important consideration when performing any experiment (Fisher, 1951, remains a key reference). There is a sound scientific rationale for replication. In the first place, by averaging over replicates, the underlying parameters of interest can be estimated with greater precision, as replication followed by averaging dampens the effect of chance variations on parameter estimates. The higher the number of replicates, the greater the precision. Second, replication provides information that allows the extent of experimental variation to be estimated. This is crucial for evaluating the statistical significance of any findings from the experiment.

Unfortunately, there is often confusion about what constitutes replication in a microarray experiment. This is because there are several types of replication in a microarray experiment, each giving information regarding a different source of variability. Some examples are as follows:

- Spotting a gene several times on an array allows the gene's variability within the array to be assessed.
- Hybridizing a number of microarrays to the same labeled mRNA sample allows the variability across arrays to be assessed. In this case the microarrays can be regarded as *technical replicates*.

Exploration and Analysis of DNA Microarray and Protein Array Data
By Dhammika Amaratunga and Javier Cabrera
ISBN 0-471-27398-8

- Hybridizing a number of microarrays to different labeled mRNA samples prepared from the same mRNA sample allows the variability of the labeling and sample preparation procedure to be assessed. Here, again, the microarrays can be regarded as *technical replicates*, but the source of variability they assess is different from the technical replicates above.
- Collecting several mRNA samples from a number of different but similar subjects allows biological variability (e.g., animal to animal or tissue to tissue differences) to be assessed. In this case the replicate microarrays can be regarded as *biological replicates*.

It is important to realize that any type of replication offers information only regarding the particular source of variability associated with that type of replication and no other. Thus, for instance, increasing the number of technical replicates merely because they are less costly than biological replicates will not offer an increase in information about biological variability.

Careful consideration should go into what type of replication to include in an experiment. To increase the overall precision of an experiment, it is most effective to add replication where there is greatest variability and, therefore, least precision. Thus, if there is high subject to subject variability and the measurements taken across the technical replicates are very precise, increasing the number of subjects will increase the overall precision of the experiment more than increasing the number of technical replicates. In fact, as the technical aspects of microarray experiments improve, biological variability is likely to constitute the highest percentage of variability in an experiment. The drawback is that it is usually the costliest. In any case, the number of biological replicates to include in an experiment should be carefully assessed, as without enough biological replicates, the overall sensitivity of the experiment will be low and reliably extending experimental findings beyond the limited confines of the experiment may prove to be difficult.

Churchill (2002) and Lee et al. (2000) give further guidance regarding replication in microarray experiments.

6.2 TECHNICAL REPLICATES

We will first discuss technical replicates. These are used to deal with technical variation, which arises from the handling steps, such as mRNA extraction, amplification, labeling, hybridization, and scanning. This variation introduces uncertainty to the intensity measurements associated with a gene. Using technical replicates and averaging across them allows gene expression levels to be estimated with greater precision. The higher the number of replicates, the greater is the precision.

The summarized intensity level of a gene on the microarrays that are exposed to the sample is an average of its intensity levels across the technical

replicates. The average could be a simple estimator such as the arithmetic mean or the median or a more sophisticated estimator such as a biweight mean.

Let us examine this in some detail. Let X_{gi} denote the (suitably transformed and normalized) spot intensity measurement for the ith technical replicate of the gth gene, $g = 1, \ldots, G$, $i = 1, \ldots, n$. The random variable X_{gi}, with mean μ_g, represents the (true) mean expression level of the gth gene and the (true) variance σ_g^2. We write this using statistical notation as $X_{gi} \sim (\mu_g, \sigma_g^2)$. The *model parameters* μ_g and σ_g^2 are estimated using observed data.

An alternative, and entirely equivalent, formulation is to write this as: $X_{gi} = \mu_g + \varepsilon_{gi}$. Here ε_{gi} is the *error* introduced by the ith technical replicate for the gth gene. Note that the statistical interpretation of the word "error" differs from its conventional meaning: it is used to denote the difference between an observed value (here X_{gi}) and its value as expected according to a statistical model (here μ_g). The error ε_{gi} is a random variable with mean zero and variance σ_g^2, namely $\varepsilon_{gi} \sim (0, \sigma_g^2)$.

The usual estimators of the model parameters, μ_g and σ_g^2, are respectively, the sample mean $\bar{X}_g$ and the sample variance s_g^2, for the gth gene:

$$\hat{\mu}_g = \bar{X}_g = \frac{\sum_{i=1}^{n} X_{gi}}{n},$$

$$\hat{\sigma}_g^2 = s_g^2 = \frac{\sum_{i=1}^{n} (\bar{X}_{gi} - \bar{X}_g)^2}{n-1}.$$

The standard error of $\hat{\mu}_g$ (i.e., the standard deviation of $\hat{\mu}_g$) is $\sigma/\sqrt{n}$, which is estimated by $\hat{\sigma}_g/\sqrt{n}$.

These estimators are all optimal in many senses if the underlying distribution of X_{gi}, or equivalently, ε_{gi}, is normal, that is, if $X_{gi} \sim \mathrm{N}(\mu_g, \sigma_g^2)$, or equivalently $\varepsilon_{gi} \sim \mathrm{N}(0, \sigma_g^2)$. However, if not, the estimators might have undesirable properties. In particular, if X_{gi} contains outliers, as is often the case with microarray data, both $\hat{\mu}_g$ and $\hat{\sigma}_g^2$ will be suboptimal, and perhaps seriously so.

It may therefore be preferable to estimate the values of μ_g and σ_g in such a way that the extent to which they are influenced by outliers is limited. Such estimators are said to be *resistant*. The most resistant reasonable estimators of μ_g and σ_g are the *median* and the *median absolute deviation from the median* (MAD):

$$M_g = \mathrm{median}(X_{gi}),$$

$$MAD_g = \mathrm{median}\{|X_{gi} - M_g|\}.$$

These estimators are so resistant that almost half the observations have to be bad before the estimators themselves are affected. However, there is a price to pay for so much resistance. These estimators are not very efficient, meaning they tend to have high variability.

It is possible to obtain reasonably high efficiency at the normal distribution coupled with reasonably high resistance should the data contain outliers. Such estimators are said to be *statistically robust*. The price is that they are not 100% efficient (but the efficiency can exceed 90%) and they come at a slight computational cost. *Biweight means* and *biweight standard deviations* are statistically robust estimators of μ_g and σ_g.

The biweight mean is defined as the value $\tilde{X}_g$ that maximizes

$$\sum_{i=1}^{n} \rho\left(\frac{X_{gi} - \tilde{X}_g}{\tau s_g^O}\right),$$

where the *objective function* ρ is defined by $\rho(u) = (1 - (1 - u^2)^3)/6$ if $|u| < 1$ and $\rho(u) = 1/6$ otherwise, s_g^O is a resistant estimate of σ_g, and the *tuning constant* τ determines the amount of efficiency and resistance desired. The larger τ is, the more efficient the estimator is if the distribution is truly normal, but the less resistant it is. The smaller τ is, the less efficient the estimator is if the distribution is truly normal, but the more resistant it is. A compromise between these two extremes offers both high efficiency at the normal distribution and resistance should the data contain outliers.

There is no closed form expression for the biweight mean. This is where the computational cost comes in. The biweight mean has to be calculated using an iterative process. The iteration is begun at M_g, the median of the gth gene. For each observation X_{gi}, calculate $u_{gi} = (X_{gi} - M_g)/\tau s_g^O$, which indicates how unusual it is; then assign it a weight $w_{ig} = w(u_{gi})$ based on the *biweight weighting function*: $w(u) = (1 - u^2)^2$ if $|u| < 1$ and $w(u) = 0$; otherwise (note that $w(u) = \rho'(u)/u$). The weighting process ensures that observations relatively near the center of the data will be assigned high weights. If there are any observations relatively far from the center of the data (i.e., outliers), they will be assigned low weights. Calculate a weighted mean

$$M_g' = \frac{\sum_{i=1}^{n} w_{gi} X_{gi}}{\sum_{i=1}^{n} w_{gi}}$$

using these weights.

These steps can be iterated, now beginning with M_g', until there is, for all practical purposes, no change in M_g'. The resulting estimator is the biweight mean, as desired. However, it has been shown that just doing a single iteration usually produces an estimator that inherits the high resistance of the median and gains substantially in efficiency. Therefore this estimator, called the *one-step biweight mean*, is sometimes used instead of the fully iterated version.

The usual choice for s_g^O is MAD_g. This is the natural choice because the estimate of σ_g used here must be resistant, but it does not necessarily have to be particularly efficient. However, this choice may not work very well with microarray data. The problem is that the number of technical replicates tends to be very small and the MAD is too unreliable in such instances.

An alternative choice for s_g^O can be obtained by exploiting the fact that σ_g^2 is usually a function of μ_g, such that $\sigma_g^2 = f(\mu_g)$. This involves first modeling the $\log(MAD_g)$ versus $\log(M_g)$ relationship using, for example, a spline with a few degrees of freedom; the spline is an estimate of f. Then a value for s_g^O is the value associated with $\log(M_g)$ as predicted by the fit.

The biweight mean, originally proposed by Tukey, belongs to a particular class of statistically robust estimators called *M-estimators* (the book by Hoaglin, Mosteller, and Tukey, 1983, gives a good review of robust estimation). Other estimators in this class are obtained by using different objective functions with bounded derivatives. Huber proposed $\rho(u) = u^2/2$ if $|u| < 1$ and $\rho(u) = |u| - 1/2$ otherwise. Hampel proposed $\rho(u) = u^2/2$ if $|u| < h_1$, $\rho(u) = h_1|u| - h_1^2/2$ if $h_1 < |u| < h_2$, $\rho(u) = h_4 + h_5(h_3 - |u|^2)$ if $h_2 < |u| < h_3$, and $\rho(u) = h_6$ otherwise. Andrews proposed $\rho(u) = (1 - \cos(\pi u))/\pi^2$ if $|u| < 1$, $\rho(u) = 2/\pi^2$ otherwise. Note that the median can be obtained by setting $\rho(u) = |u|$. Finally it can be observed that the mean corresponds to $\rho(u) = u^2/2$, but since its derivative, $\rho'(u) = u$, is not bounded, it is not, strictly speaking, an M-estimator, as it is the boundedness of the derivative that is the key to resistance.

The idea of weighting observations according to how distant they are from the center of the data is also used for calculating robust standard deviation estimates. The biweight standard deviation estimate, $\tilde{s}_g$, is defined as

$$\tilde{s}_g = \frac{\tau_A MAD_g \sqrt{n} [\sum_{i=1} \psi^2(u_i')]^{1/2}}{\sum_{i=1} \psi(u_i')},$$

where τ_A is a tuning constant that determines the resistance and efficiency desired, $u_i' = (X_{gi} - \tilde{X}_g)/(\tau_A s_g^O)$, with s_g^O set to MAD_g or smoothed MAD_g as before, is a measure of how unusual X_{gi} is, and $\psi(u) = u(1 - u^2)^2$ if $|u| < 1$ and $\psi(u) = 0$ otherwise (note that $\psi(u) = \rho'(u)$). The biweight standard deviation belongs to a particular class of statistically robust estimators of standard deviation called *A-estimators*. Another class of robust variance estimators, with perhaps slightly better properties, are known as *τ-estimators*.

Yet another robust estimator of μ_g is the *trimmed mean*. The α% trimmed mean of a set of observations is obtained by ordering the observations from smallest to largest, removing (i.e., trimming) the prespecified percentage, α%, of observations from each end of the ordered list, and taking the mean of the rest. The ordinary mean is, of course, a 0% trimmed mean; the median is something like a 50% trimmed mean; and the 25% trimmed mean is called a *midmean*.

6.3 BIOLOGICAL REPLICATES

Biological replicates are used to deal with biological variation, which is the natural variability among subjects due to genetic diversity, environmental effects and other causes. This variation also contributes uncertainty to the intensity measurements associated with a gene. Using biological replicates and averaging across them allows gene expression levels to be estimated with greater

biological precision. The higher the number of replicates, the greater is the precision.

In this case the average intensity measurement of each gene can be estimated in an analogous way to the case where there were technical replicates, so we will not go into details here.

6.4 EXPERIMENTS WITH BOTH TECHNICAL AND BIOLOGICAL REPLICATES

In certain experiments there will be both biological replicates as well as technical replicates.

Example. The first five pairs of microarrays in Example E5: C1A, C1B, C2A, C2B, ..., C5A, C5B correspond to five control samples. Each pair of microarrays corresponds to a single mRNA sample (labeled C1, C2, ..., C5), which was taken from a mouse following treatment and hybridized to two separate microarrays (labeled A and B). The two microarrays in each pair are technical replicates as they are exposed to the same biological sample. The five mice from which samples C1 to C5 are biological replicates. There were 3300 genes arrayed on the microarrays.

In the example above the number of technical replicates was the same (i.e., two) for every biological replicate. In such cases the experiment is said to be *balanced* with respect to the replication. If the number of technical replicates was not the same across the biological replicates, the experiment is said to be *unbalanced* with respect to the replication. Balanced experiments have several advantages.

When there are both biological replicates and technical replicates, the estimated average intensity measurement would be subject to biological variation as well as technical variation and some of the calculations change. In order to study this situation, let X_{gij} denote the intensity of the jth technical replicate within the ith biological replicate for the gth gene. Here g indexes the genes ($g = 1, \ldots, G$), j indexes the biological replicates ($j = 1, \ldots, a$) and i indexes the technical replicates ($i = 1, \ldots, n$).

The statistical model for this situation shows the presence of both sources of variability:

$$X_{gij} = \mu_g + \alpha_{gi} + \varepsilon_{gij}.$$

In this model, μ_g is the overall (true) mean, α_{gj} is the effect of the jth biological replicate ($\alpha_j \sim (0, \sigma^2_{BIOL;g})$), ε_{gij} is the effect of the ith technical replicate within the jth biological replicate ($\varepsilon_{gij} \sim (0, \sigma^2_{TECH;g})$).

Let $\bar{X}_g = \sum_{j=1}^{a} \sum_{i=1}^{n} X_{gij}/an$ denote the overall mean and $\bar{X}_{gj} = \sum_{i=1}^{n} X_{gij}/n$ denote the mean for the jth biological replicate. The expected value for the

overall mean is

$$\mathrm{E}(\bar{X}_g) = \mu.$$

The mean squared error across biological replicates

$$MS_g^{BIOL} = \frac{\sum_{j=1}^{a}(\bar{X}_{gj} - \bar{X}_g)^2}{a-1}$$

measures the variation across biological replicates. It also has a contribution due to the variation across technical replicates as shown by its expected value:

$$\mathrm{E}(MS_g^{BIOL}) = n\sigma^2_{BIOL;g} + \sigma^2_{TECH;g}.$$

The mean squared error across technical replicates

$$MS_g^{TECH} = \frac{\sum_{j=1}^{a}\sum_{i=1}^{n}(\bar{X}_{gij} - \bar{X}_{gj})^2}{a(n-1)}$$

measures variation across technical replicates. Its expected value is

$$\mathrm{E}(MS_g^{TECH}) = \sigma^2_{TECH;g}.$$

We can use the expected values to derive estimators for the model parameters:

$$\hat{\mu}_g = \bar{X}_g,$$

$$\hat{\sigma}^2_{TECH;g} = MS_g^{TECH},$$

$$\hat{\sigma}^2_{BIOL;g} = \frac{MS_g^{BIOL} - MS_g^{TECH}}{n}.$$

The mean has expected value and variance given by

$$\mathrm{E}(\bar{X}_g) = \mu_g$$

$$\mathrm{var}(\bar{X}_g) = \frac{\sigma^2_{BIOL;g}}{a} + \frac{\sigma^2_{TECH;g}}{an}$$

The variance of $\bar{X}_g$ can be estimated by plugging in the estimates of σ^2_{BIOL} and σ^2_{TECH}:

$$\widehat{\mathrm{var}}(\bar{X}_g) = \frac{MS_g^{BIOL} - MS_g^{TECH}}{an} + \frac{MS_g^{TECH}}{an} = \frac{MS_g^{BIOL}}{an}$$

$$= \frac{1}{a}\frac{\sum_{j=1}^{a}(X_{gj} - \bar{X}_g)^2}{(a-1)} = \frac{1}{a}\widehat{\mathrm{var}}(\bar{X}_{gj}).$$

In other words, the variance of $\bar{X}_g$ is estimated by dividing by a the variance across the biological replicates of the means obtained by averaging across the technical replicates.

The overall mean $\bar{X}_g$ could be affected by outliers among the biological replicates as well as outliers among the technical replicates. Therefore a resistant version should offer protection against both types of outliers. To begin with, observe that $\bar{X}_g$ is a mean of means; that is, it is the mean across the biological replicates of the a means across the technical replicates:

$$\bar{X}_g = \frac{\sum_{j=1}^{a} \sum_{i=1}^{n} X_{gij}}{an} = \frac{\sum_{j=1}^{a} \bar{X}_{gj}}{a},$$

or in simple terms,

$$\bar{X}_g = \text{mean}_j \, \text{mean}_i(X_{gij}).$$

Resistant and robust estimators of μ_g can be obtained by replacing the means with resistant analogues.

Working along these lines, Amaratunga and Cabrera (2001a, b) proposed a highly resistant estimator for μ_g called the *median-of-medians* (MOM):

$$M_g = \text{median}_j \, \text{median}_i(X_{gij}).$$

However, no simple resistant estimators of σ^2_{BIOL} or σ^2_{TECH} analogous to MAD are readily available. A standard error for M_g can be estimated using a resampling procedure.

Example. Since the control group in Example E5 has both biological replicates as well as technical replicates, median-of-medians were calculated to summarize the data across the replicates for the genes in the control group. Medians were calculated for the genes in each treatment group. Figure 6.1 shows the treatment group average plotted against the control group average for the first four treatments. The filled circles represent genes that showed a threefold or greater increase or decrease in expression compared to the control group.

A robust estimate of μ_g can be obtained by replacing the means with robust analogues:

$$\tilde{X}_g = \text{biweightmean}_j \, \text{biweightmean}_i(X_{gij}).$$

In other words, we first calculate the biweight mean across the technical replicates for each biological replicate. Then the overall biweight mean is the biweight mean across the a biological replicates.

Recall now that the variance of $\bar{X}_g$ is estimated by dividing by a the variance across the biological replicates of the means obtained by averaging across the

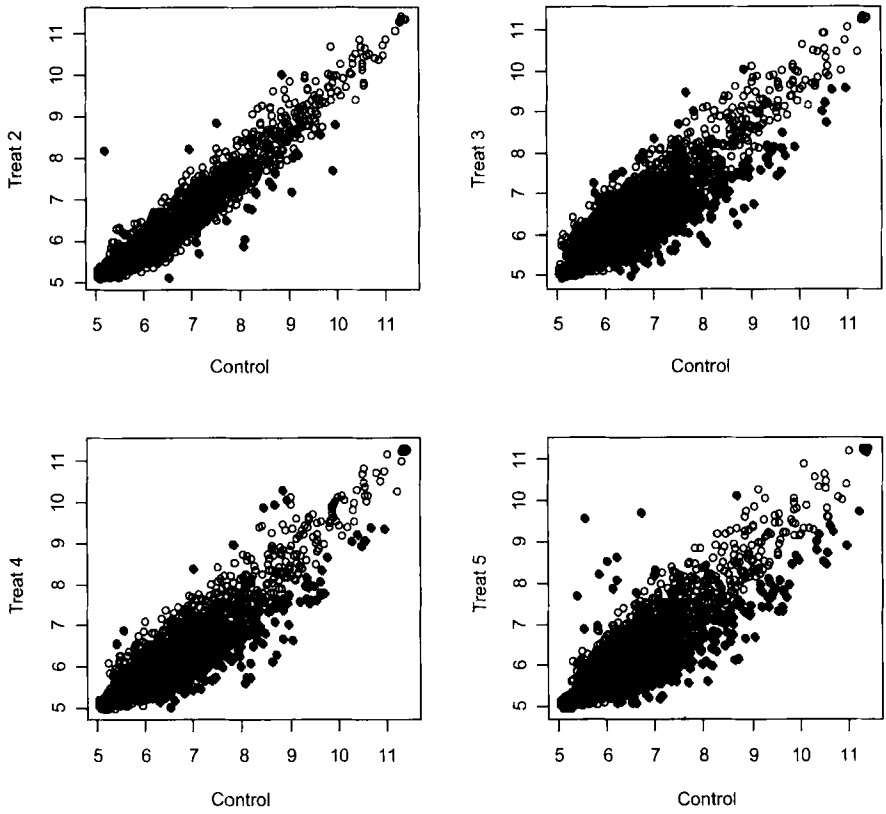


Figure 6.1 Treatment group average plotted against the control group average for the first four treatments.

technical replicates. Analogously the variance of $\tilde{X}_g$ is estimated by dividing by a the variance across the biological replicates of the means obtained by averaging across the technical replicates. In equation form

$$\widetilde{\text{var}}(\tilde{X}_g) = \frac{1}{a}\widetilde{\text{var}}(\tilde{X}_{gj}).$$

This then produces a robust estimate of the variance of $\tilde{X}_g$ as an estimate of μ_g.

6.5 MULTIPLE OLIGONUCLEOTIDE ARRAYS

The expression level of a gene, summarized across multiple oligonucleotide arrays, can be calculated using the methods described.

A model-based alternative estimation approach was proposed by Li and Wong (2001b) for one probe set in multiple oligonucleotide arrays. For the jth

probe pair in this probe set, let PM_{ij} and MM_{ij} denote the (untransformed) expression level measurements for the perfect match probe and mismatch probe respectively in the ith microarray.

Let θ_i denote the true expression level of the probe set in the ith array. The model postulates that (1) the observed measurements for MM_{ij} and PM_{ij} are linear functions of θ_i and (2) for a truly expressed gene the strength of the PM_{ij} versus θ_i relationship is greater than the strength of the MM_{ij} versus θ_i relationship. Algebraically the *Li–Wong model* can be written

$$MM_{ij} = v_j + \theta_i \alpha_j + \varepsilon,$$
$$PM_{ij} = v_j + \theta_i(\alpha_j + \phi_j) + \varepsilon.$$

The parameters, v_j, α_j, and ϕ_j, are all assumed to be nonnegative and (2) above implies that ϕ_j should be strictly positive for a truly expressed gene.

Subtracting the equation for MM from the equation for PM indicates that the PM–MM differences can be modeled by the simpler *reduced Li–Wong model*:

$$Y_{ij} = PM_{ij} - MM_{ij} = \theta_i \phi_j + \varepsilon.$$

In words, the perfect match to mismatch difference is, except for random error, the product of a model-based expression index θ_i and a probe sensitivity index ϕ_j. Although some information is lost in using this simpler model, it is easier to use and is often adequate for most practical purposes. The model is overparametrized and some constraint, usually the constraint that $\Sigma\phi_j^2 = J$, is imposed in order to make the model identifiable. The model is fitted by estimating the parameters of this model that minimize the ordinary least squares criterion. They can be obtained by iterative application of a standard least squares routine alternating between estimation of θ_i and ϕ_j until convergence.

Statisticians will recognize the PM–MM difference model as an exponentiated form of the two-way ANOVA linear model with a probe pair effect, an array effect, no intercept term, and no interaction term:

$$\log(Y_{ij}) = \log(\theta_i) + \log(\phi_j) + \varepsilon.$$

Whereas writing the model in its raw form allows negative values of Y_{ij} to be accommodated, there are certain clear advantages to writing the model the ANOVA way. One is that if the arrays correspond to different types of samples (e.g., some are from normal tissue and the rest are from diseased tissue), more complex models that include such experimental factors and their interactions can be postulated as in Chapter 8, whereas the Li–Wong model cannot be simply extended beyond the one-way treatment design. For designs where both models can be fitted, Chu et al. (2002b) compared the two model fits and found that they produced comparable results. On the other hand, the linear model

inherently has certain operational advantages due to its linearity; for example, it is easier to fit and the distributional properties of its error term are nicer.

Whichever model is used, there are certainly several benefits to taking a model-based approach over just averaging. By modeling, overly influential observations, such as outliers, can be automatically flagged, and the effects of various experimental factors can be statistically assessed.

Efron et al. (2001) discuss a different approach for summarizing data in a 2^3 factorial experiment involving eight oligonucleotide arrays.

6.6 ESTIMATING FOLD CHANGE IN TWO-CHANNEL EXPERIMENTS

We now consider two-channel microarray experiments. For the gth gene on the ith microarray ($i = 1, \ldots, n$), let X_{1gi} and X_{2gi} refer to the log-transformed and normalized intensity measurements referent to the samples labeled with channels 1 and 2 respectively. Let $X_{cgi} \sim (\mu_{cg}, \sigma_{cg}^2)$, $c = 1, 2$, $i = 1, \ldots, n$. One of the principal objectives of two-channel microarray experiments is to estimate the true differential expression for the gth gene, $\rho_g = \mu_{1g} - \mu_{2g}$, and to pick out those that appear to be the most differentially expressed.

The natural estimate of the differential expression is

$$\hat{\rho}_g = \frac{\sum_{i=1}^{n} (X_{1g} - X_{2g})}{n} = \bar{X}_{1g} - \bar{X}_{2g}.$$

This is called the *log fold change* as it is the estimated differential expression on a logarithmic scale and, when transformed back to the original scale,

$$\hat{\phi}_g = \exp(\hat{\rho}_g),$$

gives the *fold change*.

The fold change (or log fold change) is a very reasonable, easily understood and readily interpretable estimate of the true differential expression. Consequently it is also, by far, the most widely used. Nevertheless, it is wise to be cautious when interpreting fold changes across a multitude of genes because a given fold change may have a different interpretation for a gene whose expression level is low in both channels as compared to a gene whose expression level is high in both channels.

The standard error of $\hat{\rho}_g$ is

$$\text{se}(\hat{\rho}_g) = \frac{s_g}{\sqrt{n}},$$

where s_g is the standard deviation of $\{Y_{gi} = X_{1gi} - X_{2gi}\}$. The standard error of

$\hat{\phi}_g$ is, approximately

$$\text{se}(\hat{\phi}_g) = \frac{\hat{\phi}_g s_g}{\sqrt{n}}.$$

6.7 BAYES ESTIMATION OF FOLD CHANGE

Newton et al. (2001) attempt to overcome the problem of different interpretations being necessary for genes with different expression levels by using an Bayesian hierarchical modeling approach to estimate the true differential expression. The models are based on the gamma distribution. This is because gamma distributions have several features pertinent to this situation: their support is $(0, \infty)$, they are skewed, their coefficient of variation can be controlled, they are easy to work with, and it can be argued that they may be meaningful biologically.

Suppose that X_{gi} and Y_{gi} are the intensity measurements from the sample labeled with channels 1 and 2 respectively. Assume that they have been suitably transformed and normalized and then transformed back to the original scale. Let $X_{gi} \sim (\mu_{Xg}, \sigma^2_{Xg})$ and $Y_{gi} \sim (\mu_{Yg}, \sigma^2_{Yg})$.

The intensities X_{gi} and Y_{gi} can be modeled as independent gamma random variables: $X_{gi} \sim \text{gamma}(a, \theta_X)$, $Y_{gi} \sim \text{gamma}(a, \theta_Y)$. Since their means are $\mu_x = \text{E}(X_{gi}) = a/\theta_x$ and $\mu_x = \text{E}(Y_{gi}) = a/\theta_Y$, the ratio of their means is $\rho = \mu_x/\mu_Y = \theta_Y/\theta_X$. Their variances are $\text{var}(X_{gi}) = a/\theta_x^2$ and $\text{var}(Y_{gi}) = a/\theta_y^2$, so that X_{gi} and Y_{gi} both have the same coefficient of variation, $1/\sqrt{a}$, regardless of whether or not they have the same variance, which is reasonable to expect after normalization. The hierarchical aspect of the model is that the parameters (θ_x, θ_Y) themselves are modeled as $(\theta_x, \theta_Y) \sim \text{gamma}(a_0, v)$.

The posterior distribution of true differential expression can be derived using Bayes's theorem:

$$p(\rho \mid X, Y, a, a_0, v) \propto \rho^{-(a+a_0+1)} \left\{ \frac{1}{\rho} + \frac{X+v}{Y+v} \right\}^{-2(a+a_0)}.$$

The statistic

$$\hat{\rho}_g^{EB} = \frac{X_g + v}{Y_g + v},$$

which lies somewhere in between the mean and the mode of this distribution, is used as the empirical Bayes estimator of the true differential expression. Those familiar with the concept of regularization (see Section 10.3) will recognize that $\hat{\rho}_g^{EB}$ has the form of a shrinkage estimator. For a gene whose expression level is high in both channels, $\hat{\rho}_g^{EB}$ will be quite close to $\hat{\rho}_g$, whereas for a gene whose

expression level is low in both channels, $\hat{\rho}_g^{EB}$ will be shrunk toward 1; the amount of shrinkage is governed by v. Thus the empirical Bayes estimator is able to reflect the decreased variation in differential expression with increasing intensity.

Unfortunately, there is no closed form expression for v. The unknown parameters, (a, a_0, v), in the model are estimated by marginal maximum likelihood, whose details are provided by Newton et al. (2001). Besides this, another slight drawback to $\hat{\rho}_g^{EB}$ is that it does not inherit the natural fold change interpretation of $\hat{\rho}_g$.

Not surprisingly, the ordering of the most significantly expressed genes using empirical Bayes estimates will generally be quite different from that using regular estimates.

EXERCISES

6.1. Explain the need to have both technical replicates and biological replicates in a study.

6.2. Read Efron et al. (2001) and outline the summarization procedure used there.

6.3. For the data in Example E5, calculate the median-of-medians for the control microarrays and the medians for the treatment group C10. Plot them against one another. Do any genes appear to be differentially expressed in the treatment group compared to the control group?

Exploration and Analysis of DNA Microarray and Protein Array Data
Edited by DHAMMIKA AMARATUNGA and JAVIER CABRERA

CHAPTER 7

Two-Group Comparative Experiments

Many microarray experiments are comparative in nature. That is, their objective is to compare the expression levels of a set of genes across two or more conditions, in particular, to identify genes that are significantly differentially expressed across these conditions. For example, an experiment might be conducted to compare the expression levels of several genes in cancerous liver cells versus those in normal liver cells in an attempt to identify those genes that are expressed in cancerous liver cells but not in normal liver cells, and vice versa. As another example, an experiment might be conducted to compare the expression levels of several genes in cancerous liver cells in a group of patients treated with a particular test drug versus those in a group of untreated patients in an attempt to identify those genes that are expressed in treated cancerous liver cells but not in untreated cancerous liver cells, and vice versa.

The simplest way to analyze comparative experiments is to consider each gene in isolation and to compare its expression levels across the groups. At a higher level of complexity, genes can be analyzed in combination, comparing the expression levels of clusters of genes across the groups. The clusters may be prespecified or identified as part of a clustering exercise. Besides finding individual differentially expressing genes, any collection of genes that is found to be differentially expressed across the groups could be used to deduce the regulatory pathways involved in the situation under investigation.

We will begin by considering the simplest and most common case: a comparative experiment in which two groups are being compared to one another. In the first few sections of this chapter we will discuss methods for analyzing each gene on its own. The concepts introduced in this initial discussion are important in their own right and will also lay the foundation for more complex and refined analyses, which are discussed in the later sections of this chapter.

Exploration and Analysis of DNA Microarray and Protein Array Data
By Dhammika Amaratunga and Javier Cabrera
ISBN 0-471-27398-8

Example. In an example that will recur throughout the chapter, we consider a comparison between the gene expression profiles of two groups of four mice. The first group of mice was treated with a vehicle control while the other group was treated with a test compound. Several hours post-treatment, a mRNA sample was extracted from the liver of each animal and placed on a microarray containing 4077 genes. Intensity measurements were taken, log transformed and normalized. A scatterplot matrix showing pairwise scatterplots of the gene expression profiles of the eight mice is shown in Figure 7.1.

We will use the following notation in this chapter. Suppose that we are comparing the expression levels of a set of G genes in two groups of microarrays, which we will refer to as Group 1 and Group 2. There are n_1 microarrays in Group 1 and n_2 microarrays in Group 2, and the total sample size is $N = n_1 + n_2$. Let x_{gij} denote the intensity measurement for the gth gene in the ith microarray in the jth group, where $i = 1, \ldots, n_j$; $j = 1, 2$; and $g = 1, \ldots, G$. When emphasis on the gene is unnecessary, we will omit the first subscript and denote the intensity measurements x_{ij}. It is assumed that the data has already been suitably transformed and normalized. In addition, let $\bar{x}_j, \tilde{x}_j, s_j, \tilde{s}_j$ denote, respectively, the mean, median, standard deviation, and median absolute deviation (MAD) from the median of the jth group.

7.1 BASICS OF STATISTICAL HYPOTHESIS TESTING

We digress now to review briefly the fundamentals of statistical hypothesis testing. We expect readers with statistics backgrounds to skip this section.

In statistical hypothesis testing, the conjecture that there is no difference between groups is called the *null hypothesis*. With microarray data, there are G null hypotheses being tested, the gth null hypothesis, for $g = 1, \ldots, G$, being that the gth gene is differentially expressed across the groups.

The result of a hypothesis test is its *decision*. There are two possible decisions: *reject* the null hypothesis and claim that there is a difference between the groups (which can be thought of as a *positive* finding) or *do not reject* the null hypothesis and declare that there is insufficient evidence to detect a difference between the groups (which can be thought of as a *negative* finding).

If the decision of the test is to reject the null hypothesis, it may be correctly rejecting a null hypothesis that is false (called a *true positive*) or it may be incorrectly rejecting a null hypothesis that is true (called a *false positive* or a *Type I error*), we do not know which. On the other hand, if the decision of the test is not to reject the null hypothesis, it may be correctly not rejecting a null hypothesis that is true (called a *true negative*) or it may be incorrectly not rejecting a null hypothesis that is false (called a *false negative* or a *Type II error*), we do not know which. Table 7.1 is a simple tabular representation of these four possibilities.

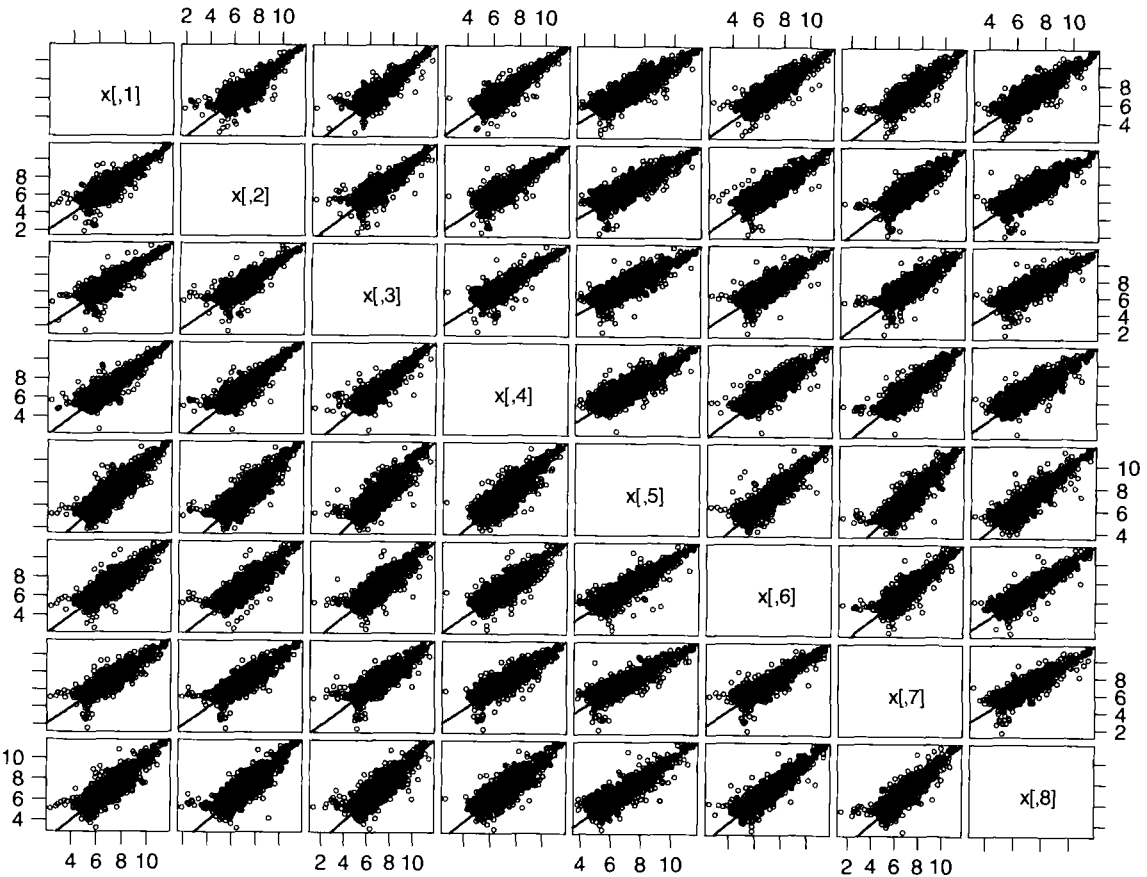


Figure 7.1 Scatterplot matrix showing the pairwise scatterplots of the log-transformed and normalized spot intensities for the eight mice.

Table 7.1

	Null hypothesis true	Null hypothesis false
Null hypothesis not rejected	True negative	False negative (Type II error)
Null hypothesis rejected	False positive (Type I error)	True positive

In practice, we do not know whether the null hypothesis is true or false. Thus we really have no way of knowing whether the test might have made the right decision or reached a false positive or false negative decision. However, what is fascinating is that it *is* possible to set up the test to reduce the chances of making such errors. This is what distinguishes a "good" test from a "bad" test.

The key to a good test is a good *test statistic*. The test statistic is generally a sample statistic that reflects how far the observed data is from the situation described by the null hypothesis. Many test statistics, T, have the form $T = r/s$. Here r is an estimate of the size of the biological effect being tested; the further the data is from the null hypothesis (i.e., the more likely that the null hypothesis is false), the larger the value of r. The denominator, s, is a standard error that measures the variability of r. Thus T measures how large the biological effect r is relative to its variability s. It is no accident that T has the form of a signal-to-noise ratio with r as the "signal" and s as the "noise."

The probability distribution of the test statistic under the null hypothesis is called its *null distribution*. Based on the null distribution, we can calculate the *p-value*, the probability of observing a value as extreme as that observed if the null hypothesis was true. Clearly, the smaller the p-value, the greater is the weight of evidence against the null hypothesis. A typical decision rule for a test states that the null hypothesis is rejected if and only if the p-value is less than a specified value called the *significance level* (or just the *level*) of the test.

The probability of the test reaching a false positive decision is called the *false positive rate* (or the *Type I error probability*) and the probability of the test reaching a false negative decision is called the *false negative rate* (or the *Type II error probability*). Also the "true positive rate" is called the *specificity* and the "true negative rate" is called the *sensitivity* or *power*. Naturally we would like both the false positive rate and the false negative rate to be zero, but this is impossible. On top of that, decreasing one tends to increase the other; in other words, increasing the specificity lowers the sensitivity, and vice versa. Thus we have to arrive at some compromise between the two.

The most popular such compromise is the *Neyman–Pearson approach* to statistical hypothesis testing. In this approach the false positive rate is controlled at a specified small value, called the *size* of the test, and then the test is set up to have as small a false negative rate (or, equivalently, as high a true

negative rate) as possible—in other words, "fix the size, maximize the power." Generally, the size of the test is bounded above by the significance level.

Of course, life is not so simple. In order to select or develop a good test for a particular situation, it is necessary to make some *assumptions* about that situation. Different assumptions for the same situation will generally lead to quite different tests and, what is more unsettling, perhaps even quite different test results. Thus it is important to consider one's assumptions carefully and to keep in mind that if the assumptions being made are not correct, the size and power properties one expected the test to display might not be achieved. This is why it is always a good idea to run some *diagnostics* to check whether the assumptions underlying the test seem to hold. If they do not appear to hold, it is wise to rethink the testing procedure.

7.2 FOLD CHANGES

Early analyses of microarray data declared a gene differentially expressed if its fold increase or fold decrease exceeded a specified cutoff. For example, in their seminal paper on using microarrays to study gene expression in *Arabidopsis thaliana*, Schena et al. (1995) declared a gene differentially expressed if its expression level showed a fivefold difference between the two mRNA samples.

On a logarithmic scale, the decision rule that declares that changes of h-fold or greater are significant translates into asserting that a gene should be declared differentially expressed if $|\bar{x}_2 - \bar{x}_1| > \log(h)$

Example. For the example dataset, Figure 7.2*a* shows a histogram of the $D = \bar{x}_2 - \bar{x}_1$ values, which range from -1.66 (i.e., a 5.26-fold downregulation compared to control) to 1.72 (i.e., a 5.58-fold upregulation compared to control) with a near zero median of 0.01 with 119 genes showing a twofold or greater upregulation compared to control and 144 genes showing a twofold or greater downregulation compared to control.

The reliance on fold change alone to designate significance has, rightly, been criticized. Keep in mind that the means are merely estimates of the true but unknown mean expression levels and hence are subject to variability. Genes with high variability have a reasonable probability of having a large fold change and looking deceptively interesting. The problem with the fold change approach is that it utterly fails to take this uncertainty into account. It is entirely possible, after all, that a gene might exhibit a tenfold change and yet not be significant because it has high variability, whereas another gene might exhibit a twofold change and be highly significant, both statistically and biologically, because its expression level measurements had low variability and were therefore more precise.

The variability of the estimates can be assessed and should be used to adjust the threshold (an early paper on microarrays, Chen et al., 1997, developed

some distribution theory in this regard). This is the idea behind the t-test discussed in Section 7.3 and extensions of the t test that are discussed later. Applying the same arbitrarily chosen threshold to all the genes in the study is just not appropriate.

7.3 THE TWO-SAMPLE t TEST

The most basic statistical test for comparing two groups is the *two-sample t test*. The two-sample t test statistic is given by

$$T_e = \frac{|\bar{x}_1 - \bar{x}_2|}{s_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}},$$

where

$$s_p^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{(n_1 + n_2 - 2)}$$

is the pooled estimate of variance.

If the data is drawn from a normal distribution (the normal distribution is sometimes called the Gaussian distribution) and is homoscedastic (i.e., has equal variances): $x_{ij} \sim \mathrm{N}(\mu_j, \sigma^2)$, the null distribution of T_e is a t-distribution with degrees of freedom $\nu = n_1 + n_2 - 2$. If the observed value of T_e is $T_{e;obs}$, then the p-value is given by the probability $p_e = \mathrm{Prob}(|T_e| > T_{e;obs})$. A gene is declared significantly differentially expressed at level of significance α if $p_e < \alpha$.

Example. Of the 4077 genes in the example, 998 are significantly differentially expressed at the 5 percent level according to the two-sample t test above; 523 are upregulated compared to control, while 475 are downregulated compared to control. Figure 7.2*b* shows a scatterplot of the difference in means versus the cube root of the pooled variance, with the genes found significantly differentially expressed by the t test plotted as filled circles and the others as clear circles. Figures 7.2*c* and 7.2*d* show scatterplots of the p-values versus the differences in means and the cube roots of the pooled variances, respectively. These plots indicate that the t test has a tendency of ignoring those genes that have large differences in means (i.e., large fold changes on the raw scale) if they also should happen to have high variances. This is reasonable, but its inclination to focus on genes with small variances may be too strong when the variances are estimated from small samples from which variance estimates cannot be well estimated. This behavior of the t test is addressed in Sections 7.10 and 7.11.

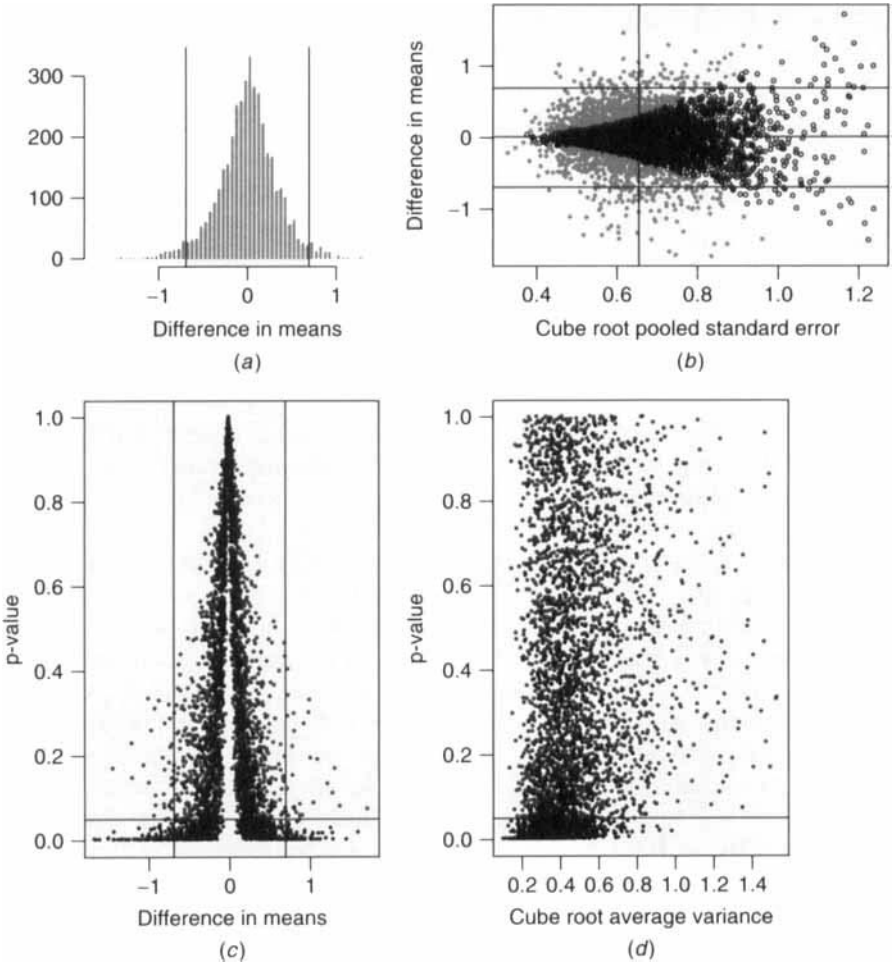


Figure 7.2 (a) Histogram of the difference in means; the two vertical lines refer to twofold changes. (b) Scatterplot of the difference in means versus the cube root of the pooled standard error with those genes found significant by the two-sample t test shown as filled circles; the central vertical and horizontal lines pass through the medians of the axes while the other two horizontal lines refer to twofold changes. (c) Scatterplot of the p-value versus the difference in means. (d) Scatterplot of the p-value versus the cube root of the pooled standard error. Genes found significant by the two-sample t test are shown as filled circles. The two vertical lines in (c) refer to twofold changes, and the horizontal lines in (c) and (d) refer to a p-value cutoff of 5%.

Observe that the t test statistic has the form of a signal-to-noise ratio as mentioned in Section 7.1. The "signal" is the numerator that reflects the difference we are trying to discover; the "noise" is the denominator that reflects the variability of the system.

The two-sample t test can be modified to test whether the average intensity of the first group is greater or less than that of the second group by some

specified value, Δ (e.g., $\Delta = \log(2)$ for a twofold difference). The test statistic for this is

$$T_\Delta = \frac{|\bar{x}_1 - \bar{x}_2| - \Delta}{s_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}.$$

The null distribution of T_Δ is, again, a t-distribution with degrees of freedom $\nu = n_1 + n_2 - 2$. Since very small group differences are usually of no interest and can come up significant due to having unbelievably small variances as outlined above, it is better to use this form of the t test to focus in on more meaningful differences.

Example. Of the 4077 genes in the example, 223 are upregulated by more than twofold compared to control, while 175 are downregulated by more than twofold compared to control.

The assumptions of normality and homoscedasticity are critical to the t test's functioning properly. For instance, if the underlying distribution has longer tails than a normal distribution, the denominator of the t test statistic will be inflated, and it will generally be harder to reject the null hypothesis by way of the t test. Therefore the t test should have a lower false positive rate than expected (because of which it is sometimes claimed that the t test is robust) but a much higher false negative rate (i.e., lower power) than expected.

When the assumption of normality is tenable but that of homoscedasticity is not, the t test will tend to have a higher false positive rate than expected. In an attempt to alleviate this problem, an unequal-variance form of the t test, called *Welch's test*, has been proposed. The test statistic for Welch's test is given by

$$T_u = \frac{|\bar{x}_1 - \bar{x}_2| - \Delta}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}},$$

with $\Delta = 0$ when trying to detect any differences. The null distribution of T_u is approximately a t-distribution with degrees of freedom:

$$\nu = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{1}{(n_1 - 1)}\left(\frac{s_1^2}{n_1}\right)^2 + \frac{1}{(n_2 - 1)}\left(\frac{s_2^2}{n_2}\right)^2}.$$

If the observed value of T_u is $T_{u;obs}$, then the p-value is given by the probability $p_u = \text{Prob}(|T_u| > T_{u;obs})$. A gene is declared significantly differentially expressed at level of significance α if $p_u < \alpha$.

A test based on T_u can be shown to have, more or less, the correct test size even if $\sigma_1^2 \neq \sigma_2^2$. There is a drawback, however: having fewer degrees of freedom than T_e, T_u also tends to be less powerful. The loss of power may be substantial enough, particularly when n_1 and n_2 are very small, to consider using T_e, even when a moderate amount of heteroscedasticity is present.

Example. In the example, Welch's test finds 872 genes that are significantly differentially expressed. The same genes were also found to be significantly differentially expressed by the t test, demonstrating that the latter flags more differences as being significant; 455 are upregulated compared to control, and 417 are downregulated compared to control.

7.4 DIAGNOSTIC CHECKS

The *residuals*, $r_{ij} = x_{ij} - \tilde{x}_j$, form the basis for checking the validity of the assumption that the data follow a normal distribution. Here the residuals are centered at the median rather than at the mean because, with the median being relatively unaffected by outliers, it provides a more resistant estimator of the center of the distribution than the mean. When the residuals are sorted and plotted against the quantiles of a normal distribution, the resulting plot, called a *normal probability plot*, should be roughly linear if the underlying distribution was a normal distribution. With microarrays, it is usual to perceive some tapering away at the ends, indicating some degree of long-tailedness in the data.

If the variances appear to differ across the groups, the *standardized residuals*, $r_{ij}^* = (x_{ij} - \tilde{x}_j)/\tilde{s}_j$, may be used instead. Again a resistant measure of scale, $\tilde{s}_j$, is used instead of the traditional measure of scale, s_j. Large absolute values of r_{ij}^* indicate outliers.

With microarray data, it is in fact useful to gather all the residuals across all the genes $\{r_{gij}^*\}$ for making the normal probability plot.

Example. A normal probability plot of the standardized residuals for the example is shown in Figure 7.3. The plot indicates that the central portion of the distribution of the residuals resembles a normal distribution, but the tails of the residual distribution are considerably longer than those of a normal distribution.

This graphical check is often enough, but there are several formal statistical tests for assessing the normality of the underlying distribution as well. One of the most effective is the Shapiro–Wilk test. Other tests include the Kolmogorov–Smirnov test and its modifications, such as the Anderson–Darling test. With a very large number of observations, however, these tests will indicate nonnormality even with trivial departures from perfect normality. Therefore we will not use them here.

Formal tests for unequal variances across groups, such as Bartlett's test and Levene's test, require larger sample sizes than are generally available in micro-

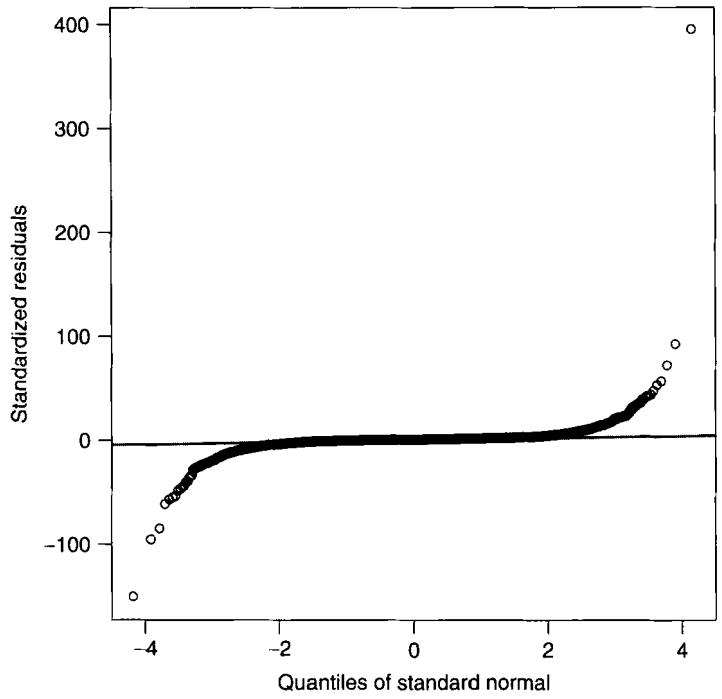


Figure 7.3 Normal probability plot of the standardized residuals. The straight line is the identity line.

array studies. Hence they are not described here. However, it is possible to get some idea as to how close the variances in the two groups are by plotting $\{s_{gi1}^{2/3}\}$ versus $\{s_{gi2}^{2/3}\}$. Here we use the cube root transformation which brings the distribution of variances closer to a normal distribution (Wilson and Hilferty, 1931).

Example. Figure 7.4*a* shows a scatterplot of $\{s_{gi1}^{2/3}\}$ versus $\{s_{gi2}^{2/3}\}$. This plot shows that, on an individual gene basis, the variances in the two groups can differ quite markedly. However the normal probability plot of $\{s_{gi1}^{2/3}\}$ versus $\{s_{gi2}^{2/3}\}$, shown in Figure 7.4*b*, indicates that the distributions of the variances are the same across the two groups.

7.5 ROBUST *t* TESTS

If the t test is applied when the data is normally distributed except for a few outliers, these outliers will tend to degrade the power of the test. What happens is that the outliers will inflate the denominator of the test statistic more than its numerator, so the test statistic is less likely to be large and its propensity for being large when the null hypothesis is false will be dampened. Consequently

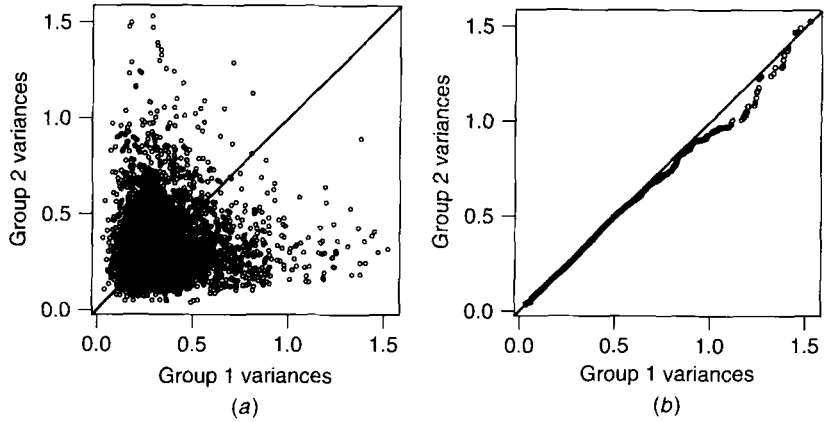


Figure 7.4 (*a*) Scatterplot (on a cube root scale) and (*b*) normal probability plot of the variance estimates for Group 1 versus the variance estimates for Group 2. In each plot the straight line is the identity line.

the false positive rate of the test will be low (called *robustness of validity*), which is fine, but the false negative rate of the test will be high (called lack of *robustness of efficiency*).

The t test can be rendered *robust* (i.e., it can be made to be less influenced by outliers) by replacing the means and variances in the test statistic with robust versions of these sample statistics. One robust form of the t test is obtained by replacing the means by biweight means (or their one-step counterparts) and the variances by A-estimators or τ-estimators (these estimators are described in Section 6.2).

Example. The robust t test, with a tuning constant set so that it is very resistant to outliers, finds that 228 genes are upregulated compared to control and 224 genes are downregulated compared to control. The reason for the relatively small number of significant genes is the loss of power due to the high resistance. Raising the tuning constant will give results closer to the t test. Figure 7.5 illustrates the resistance of the robust t test. It shows the data for three genes that are declared not significant by the t test but significant by the robust t test at the 5% level of significance and one gene that is the reverse. It can be seen that the first three all have a single extreme outlier that prevents them from turning up significant. This is why it is so important with microarray data to use methods that are not heavily influenced by outliers.

7.6 RANDOMIZATION TESTS

Randomization tests are resampling-based procedures for assessing how reasonable the null hypothesis is in the face of the observed data. As in any

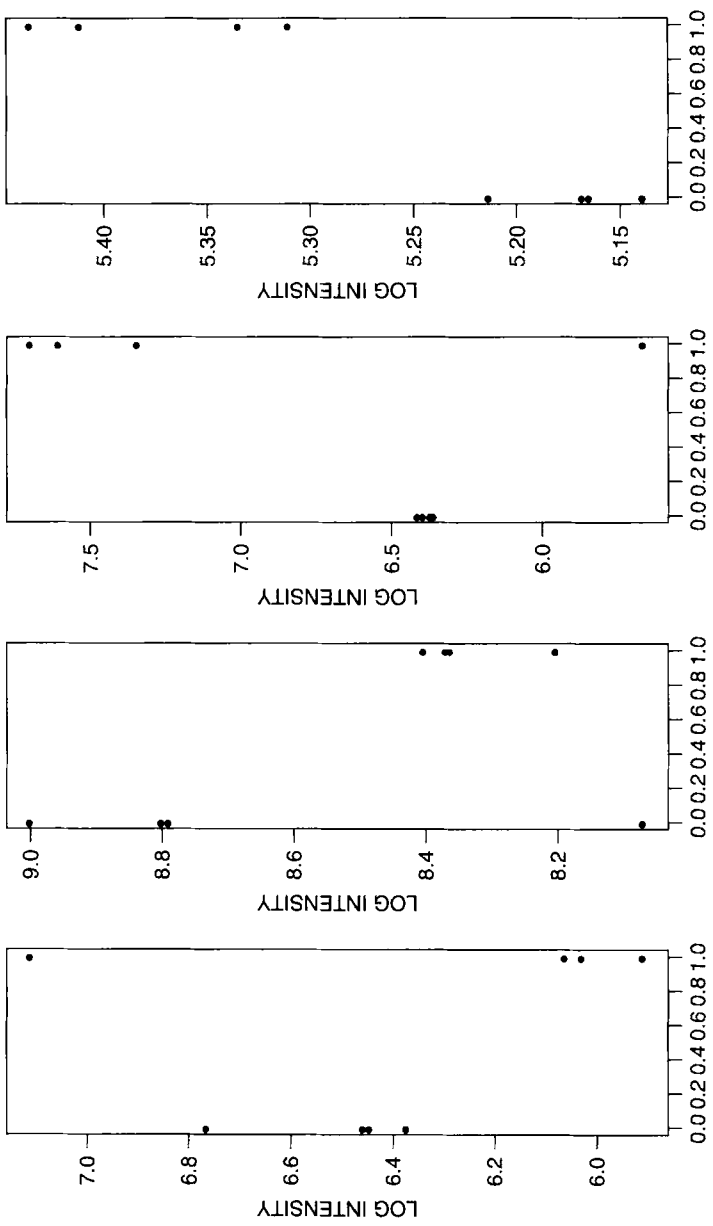


Figure 7.5 Log intensities for the control group and the treatment group for four genes, the first three of which are significant by the robust *t* test but not significant by the *t* test, the fourth of which is not significant by the robust *t* test but significant by the *t* test.

hypothesis-testing situation, a randomization test proceeds by selecting a test statistic, T, which measures how far the observed data is from the situation described by the null hypothesis.

For the two-group situation the most natural candidate for a test statistic is the t test statistic. The observed value of T, t_{obs}, is compared to the distribution of values of T that are obtained by randomly reassigning the data to the two groups, keeping the sample sizes the same. In other words, the procedure is to repeat the following for every possible permutation of the data:

Step 1. Permute the data.

Step 2. Assign the first n_1 observations to the first group and the remaining n_2 observations to the second group.

Step 3. Calculate the test statistic (which we will denote t_p^*) for the permuted data.

Step 4. Count the number of the permutations whose t_p^* value exceeds t_{obs}, and divide this count by the total number of permutations to get the proportion, p_{Perm}, of times the value of the t statistic on the permuted data exceeded the value of the t statistic on the data we actually obtained.

This proportion, p_{Perm}, is an estimate of the probability of such an extreme result under the null. In other words, it functions as a p-value. A gene is declared significantly differentially expressed at level of significance α if $p_{Perm} < \alpha$. Performed this way, this test is also referred to as a *permutation test*. If it is impractical to perform all possible permutations, one can get by doing a substantial number of random permutations instead.

The idea that motivates permutation tests is that, if the null hypothesis were true, then all possible permutations of the data are equally likely to have occurred. The order of the data that we observe would be just one of the equally likely orders and t_{obs} should appear as a typical value of the randomization distribution of T. If this does not seem to be the case, then that should be regarded as evidence against the null hypothesis.

Incidentally, for the two-group situation, both the difference between the means and the mean of the first group could be used as the test statistic and are in fact equivalent to using the t statistic.

Example. There are two groups of four, making for 35 possible permutations. We use the difference in means, $T_d = \bar{x}_2 - \bar{x}_1$, as the test statistic and regard T_d as significant if the observed value $|T_{d;obs}|$ of $|T_d|$ exceeds $|T_d|$ in at most one permutation, which constitutes a two-sided test of level $2/35 = 5.7\%$, which we will call 5% without quibbling over the extra 0.7%. This test finds that 1384 genes are significantly differentially expressed; 651 are upregulated compared to control, while 733 are downregulated compared to control.

The advantage of a randomization test is that it does not require specification of the underlying distribution to be valid. However, a randomization test is robust to outliers only if the test statistic itself is resistant. Thus a randomiza-

tion test based on a difference of means is not robust, but a randomization test based on a difference of medians, suitably standardized, is robust.

7.7 THE MANN–WHITNEY–WILCOXON RANK SUM TEST

When it is clear that the underlying distribution is far from normal, it may still be reasonable to assume that the distributions for the two groups are identical except for a location effect, so that $x_{i1} \sim F(\mu)$, $x_{i2} \sim F(\mu + \theta)$, where $F(\mu)$ denotes a distribution centered at μ. The Mann–Whitney–Wilcoxon test can be used to test the hypothesis that location parameter $\theta = 0$.

Once the observations have been ranked from 1 to N in increasing order, the test statistic for the Mann–Whitney–Wilcoxon test is the *rank sum statistic*, R, the sum of the ranks corresponding to the observations in Group 1. This statistic measures the degree of overlap between the two groups, the smaller the overlap, the further the value of R is from its null value of $n_1(N+1)/2$, indicating a group difference.

The null distribution of R has been tabulated (e.g., see Hollander and Wolfe, 1999) for small values of n_1 and n_2 using an argument similar to that of permutation tests. For larger values of n_1 and n_2, the fact that

$$\frac{\left|R - \dfrac{n_1(N+1)}{2}\right|}{\sqrt{\dfrac{n_1 n_2 (N+1)}{12}}}$$

has, approximately, a standard normal distribution under the null hypothesis can be used to obtain p-values. If the observed value of R is R_{obs}, then the p-value is given by is the probability $p_R = \text{Prob}(|R| > R_{obs})$. A gene is declared significantly differentially expressed at level of significance α if $p_R < \alpha$.

Rank-based tests, like the Mann–Whitney–Wilcoxon test, are referred to as *nonparametric tests* or *distribution-free tests*, as they do not depend on strong distributional assumptions holding to be valid and can be used in a wide range of situations. However, they are less powerful than their parametric counterparts; in other words, their p-values tend to be higher, making it harder to detect real differences as being statistically significant. If the sample sizes are large, the difference in power is minor. On the other hand, with small sample sizes, as in typical microarray experiments, nonparametric tests have very little power to detect differences.

Example. The Mann–Whitney–Wilcoxon rank sum test finds 1117 genes are significantly differentially expressed at the 5% level (the actual level is 5.7%, but as with the randomization test, we will not quibble over the extra 0.7%), 952 of which were also found to be significantly differentially expressed by the t test; 588 are upregulated compared to control, while 529 are downregulated compared to control. The genes that were found to be significantly differentially

expressed by the rank sum test but not the t test had large variance differences across the two groups, demonstrating the t test's loss of power when this happens.

Chen et al. (1997), one of the earliest microarray papers to apply a formal statistical test, used the Mann–Whitney–Wilcoxon rank sum test for the segmentation step in image processing. Chambers et al. (1999) apply the Mann–Whitney–Wilcoxon rank sum test to analyze microarray data from a study of human cytomegalovirus infection.

For microarray data, Zhang et al. (2002) propose a different nonparametric scheme. They suggest sorting and scoring the intensities on each array from 1 (for the lowest intensities) to, say, 10 (for the highest intensities) based on a clustering algorithm and then comparing the scores across the groups of arrays. Their rationale is that as much as half the data on an array could be referring to nonexpressing genes and any differences among them is due to experimental variability. This way changes in scores across arrays would be better than changes in raw values as indicators of differential expression.

7.8 MULTIPLICITY

Analyzing microarray data involves performing a very large number of statistical tests, as a test is being run on each and every gene. One drawback of doing so many tests is that the more the number of statistical tests performed, the higher the overall false positive rate and the higher the expected number of false positives. Therefore the microarray researcher must beware of attaching too much importance to all the findings labeled "significant," without making a suitable allowance for multiple testing.

In the case of G statistical tests, each performed at level α, if the tests are independent, the probability of making at least one false positive is $1 - (1 - \alpha)^G$, which is very close to unity for large G, and the expected number of false positives is αG, which is very large for very large G. Thus the number of false positives can be so high as to overwhelm and totally obscure any actual effects.

It is possible to alleviate this problem by adjusting the individual p-values of the tests for multiplicity. Indeed, a number of ways of doing so exist in the statistics literature. One major drawback, though, is that such procedures could lower the sensitivity as drastically as they raise the specificity. Indeed, in microarray experiments, G is so large and the number of replicates is so small that the power of the multiplicity adjusted tests is likely to be very small. In other words, aggressively adjusting for multiplicity could seriously impede the ability of the tests to find truly differentially expressing genes.

7.8.1 A Pragmatic Approach to the Issue of Multiplicity

This dilemma can be resolved by taking a pragmatic view as to how the overall objective of the study demands that the p-values be adjusted for multiplicity

(e.g., Nadon and Shoemaker, 2002, make a similar suggestion). For instance, a researcher in a screening study may be willing to accept a fairly large number of false positives in order to improve his or her chances of identifying some truly differentiating genes. In another instance, a researcher's resources may be so limited that he or she would be able to follow up on only a handful of genes that appeared to be the most interesting. In such cases stochastic multiplicity considerations are useful to the extent that they protect the researchers from assiduously following up on random patterns. However, there is no reason to slavishly adhere to classical cutoffs like the one that demands that a p-value should be less that 5% after multiplicity adjustment to declare significance.

In such cases, as long as there is some evidence that the experiment is picking up differences (i.e., the experiment is not a failure, as can be assessed by a quality check of the arrays as briefly outlined in Section 5.8), a reasonable approach is to rank the genes from 1 to G according to some criterion, such as the t statistic, and to select the H genes with the best ranks for further study. The second researcher would want H to be quite small and would be more selective about how the H are chosen, whereas the first researcher would take a larger H. Multiplicity considerations may help in choosing H. The gene ranking could be based on one or more factors, but it is always preferable to rank a statistic that takes experimental variability into account, such as the t test statistic, or equivalently, the p-value associated with the t test statistic, rather than one that does not, like the fold change.

A modification of this approach is to rank the G genes, select a moderate size H of them, and then run these H through a cluster analysis (Chapter 9), with the intention of picking either one gene or a very small subset of genes from each tight cluster as the "most interesting" genes. The rationale for doing this is that, since genes mostly express along genetic pathways, an assemblage of co-expressing genes that express differentially across the treatment groups are more interesting than a single gene with a unique expression profile that is differentially expressed across the treatment groups. Annotation information, if available, should also be useful in so picking a subset of interesting genes (Bouton and Pevsner, 2000, 2002). In practice, the most satisfactory gene selection procedure is likely to be some blend of all these considerations.

In later studies, particularly confirmatory studies or studies that are to be submitted for external publication, the researcher would want to protect against an excessively high number of false positives. In such cases a formal multiplicity adjustment should be applied.

7.8.2 Simple Multiplicity Adjustments

We will now outline several ways of adjusting the p-values for the increased false positive rate due to multiple testing. Consider a situation in which G statistical tests have been performed. Let $p_1, \ldots, p_G$ be the G observed p-values. Suppose that according to the rejection rule, R of the G tests led to rejection of

the individual hypotheses they were testing and, unbeknown to us, V of those were actually false positives.

If we make no adjustments for multiple testing, we are controlling the *per-comparison error rate* (PCER): $\text{PCER} = \text{E}(V)/G$. This tends to be too permissive in practice, as described above. Most conventional multiplicity adjustments attempt to control the *familywise error rate* (FWER), the probability, $\text{Prob}(V > 0)$, of committing at least one false positive among all the hypotheses tested.

Classical p-value adjustments are *single-step* procedures in that the same adjustment is applied to each p-value regardless of their ordering.

- Bonferroni: The *Bonferroni p-value* for the kth test is simply $\tilde{p}_k^B = Kp_k$. If it exceeds 1, it is set to 1. The Bonferroni adjustment is highly *conservative* in that it produces large adjusted p-values that make it difficult to reject many null hypotheses, and consequently the adjusted tests have low power.
- Sidak: The *Sidak p-value* (Sidak, 1967) for the kth test is $\tilde{p}_k^S = 1 - (1 - p_k)^K$. Sidak p-values are slightly less conservative than Bonferroni p-values.

Example. The Bonferroni p-values for the example data set can be obtained by multiplying each individual p-value by 4077, the number of genes. When this is done with the t test p-values, only 12 remain significant, 5 are upregulated, and 7 are downregulated compared to control. In this case the Sidak method is only slightly more liberal: it finds one additional upregulated gene.

7.8.3 Sequential Multiplicity Adjustments

While such adjustments certainly offer full protection against too many false positives being committed, they are so strong that they result in too many false negatives being committed. An alternative approach is sequential p-value adjustment, a technique pioneered by Holm (1979) and extended by a number of others. These methods take the order of the observed p-values into account with smaller p-values being adjusted more than larger p-values. For instance, with step-down sequential testing, successively smaller adjustments are made at each step of the procedure. These methods retain control of the FWER and are generally more powerful than single-step p-value adjustments as they do not inflate the p-values as much as the single-step procedures.

Suppose that the unadjusted p-values have been ordered so that $p_1 < p_2 < \cdots < p_G$. Sequential methods can be either *step-down* or *step-up*. We now outline a few of the proposed methods.

- Holm–Bonferroni: The *Holm–Bonferroni step-down p-values* are determined as

$$\tilde{p}_1^{HB} = Kp_1,$$
$$\tilde{p}_k^{HB} = \max(\tilde{p}_{k-1}^{HB}, (K-1)p_k).$$

As always, if any adjusted p-value exceeds 1, it is set to 1.

- Holm–Sidak: The *Holm–Sidak step-down p-values* are determined similarly as

$$\tilde{p}_1^{HS} = 1 - (1-p_1)^K,$$
$$\tilde{p}_k^{HB} = \max(\tilde{p}_{k-1}^{HB}, (K-1)p_k).$$

- Hochberg: Assuming that the G p-values are independent and uniformly distributed under their respective null hypotheses, Hochberg (1988) demonstrated that Holm's step-down adjustments control the FWER even when calculated in a step-up fashion. The *Hochberg step-up p-values* are determined in reverse order to the step-down Bonferroni as

$$\tilde{p}_G^{H} = p_{G1},$$
$$\tilde{p}_{G-g}^{HB} = \max(\tilde{p}_{G-g+1}^{HB}, kp_{G-g}).$$

The advantage of doing the adjustments step-up instead of step-down is that the adjustments are uniformly smaller for the former than for the latter. Therefore the step-up technique is more powerful and the number of false negatives is reduced. However, this improved power comes at the cost of having to make the assumption of independence.

- Westfall–Young: The *Westfall and Young step-down p-values* (1993) are determined as

$$\tilde{p}_g = \max_{k=1,\ldots,g} \left\{ \text{Prob}\left(\max_{l=k,\ldots,G} P_l \le p_k \mid H_0 \right) \right\}.$$

These adjusted p-values usually have to be estimated by simulation and this is, as a result, a computationally much more intensive method than the others. On the other hand, it has a couple of advantages: (1) Unlike the other methods, it takes into account the possibility that the tests may not be independent of one another, a valuable consideration for microarray data as genes rarely act in isolation. (2) It is less conservative than the other methods.

Example. Applying the Holm–Bonferroni method to the example finds the same significant genes as the Bonferroni method.

A very different approach to the multiplicity problem in microarray experiments has been taken by Allison et al. (2002). They assess the true positive rate

using the fact that if all the null hypotheses were true (i.e., none of the genes are differentially expressed) and the gene expression levels were independent across all genes, then the distribution of p-values would be uniform on the interval $[0, 1]$, regardless of the statistical test used or the sample size. On the other hand, if some subset of the genes are differentially expressed, the p-values will tend to cluster at low values. This effect can be mirrored by modeling the set of p-values as a random sample from a mixture of beta distributions. Applying Bayes's rule, the posterior probability that a gene is differentially expressed can then be calculated for each gene.

7.9 THE FALSE DISCOVERY RATE

All the FWER-controlling adjustments described in Section 7.8 are very large. This is because controlling the FWER is a stringent criterion that inherently forces large adjustments. When G is large, as in microarray experiments, FWER-controlling adjustments are likely to be too strong and result in far too many false negatives. This is clearly undesirable, particularly when making a large number of inferences. The overall conclusion is not necessarily erroneous as soon as one of them is incorrect. All that one is concerned about is preventing an inordinately large number of false positives from clouding the results.

In such situations Benjamini and Hochberg (1995; see also Yuketieli and Benjamini, 1999) proposed controlling the *false discovery rate* (FDR) instead. The FDR is defined as the expected proportion of false positives among the positive findings:

$$\text{FDR} = \text{E}\left[\frac{V}{R} \mid R > 0\right] \text{Prob}[R > 0].$$

If all the null hypotheses were true, the FDR would equal the FWER and controlling the FDR would be equivalent to controlling the FWER. If not every null hypothesis was true, the FDR maintains some control over the number of false positives in an adaptive fashion, in the sense that the more the number of the hypotheses that are truly false, the smaller is the FDR. Hence procedures that control the FDR tend to be more powerful than procedures that control the FWER at the same level. Benjamini and Hochberg (1995) suggested the following step-up procedure to adjust the ordered p-values so as to control the FDR:

- Benjamini–Hochberg: The *Benjamini–Hochberg adjusted p-values* are

$$\tilde{p}_K^{BH} = p_K,$$

$$\tilde{p}_{K-k}^{BH} = \min\left(\tilde{p}_{K-k+1}^{BH}, \left(\frac{K}{K-k}\right) p_{K-k}\right).$$

These p-values are less conservative than Hochberg's step-up adjustments and are guaranteed to control the FDR when the original p-values are independent and uniformly distributed under their respective null hypotheses.

7.9.1 The Positive False Discovery Rate

Storey and Tibshirani (2001) proposed a modified version of the FDR, called the *positive false discovery rate* (pFDR):

$$\text{pFDR} = \text{E}\left[\frac{V}{R} \mid R > 0\right].$$

The pFDR emphasizes the fact that an adjustment is only necessary when there are positive findings.

Given a decision rule, the pFDR can be estimated via a permutation procedure. Suppose that the decision rule is to reject any test statistics, T, that exceed a specified value t_+ and that, of the G test statistics, h_{obs} exceeded t_+. In B^* permutations suppose that an average number h^* of test statistics exceeded t_+ so that, when there are no true positives, the number of (false) positives observed is h^*. Then a natural estimate of the pFDR is

$$\text{pFDR} = \frac{h^*}{h_{obs}}.$$

If the number G_+ of genes that are truly differentially expressed is not small, then this estimate of pFDR will be too high. The way to improve it is to multiply this crude estimate by an estimate of $\pi_+ = G_+/G$. A somewhat ad hoc estimate of π_+ can be obtained by considering the genes with the smallest values of T (i.e., those such that $T < t_-$, where t_- is some prespecified value) as being truly not differentially expressed, as they are the least likely to be differentially expressed. Suppose that k_{obs} test statistics had $T < t_-$, and that in the B^* permutations, on average, k^* test statistics had $T < t_-$. Then an estimate of π_+ is $\hat{\pi}_+ = k_{obs}/k^*$ and an improved estimate of pFDR is given by

$$\text{pFDR} = \hat{\pi}_+ \left(\frac{h^*}{h_{obs}}\right).$$

Example. In analyzing the data in the example with the two-sample t test, any gene whose t test statistic exceeded $t_+ = 2.447$ (the 97.5th quantile of a t distribution with 6 degrees of freedom) in absolute value was flagged as being significant at the 5% level. Recall that 998 such genes were flagged. In the 34 possible permutations of the data, an average of 138.15 genes are flagged, leading to a simple pFDR estimate of $138.15/998 = 13.8\%$. A total of 1259

genes have t test statistics below $t_- = 0.718$ (the 75th quantile of a t distribution with 6 degrees of freedom) in absolute value, whereas in the 34 possible permutations of the data, an average of 1958.71 genes have t test statistics below t_- in absolute value, so that $\hat{\pi}_+ = 1259/1958.71 = 0.643$. Hence the improved estimate of pFDR is pFDR $= (0.643)(0.138) = 0.089$, which is about 9%.

Unlike the procedures in Section 7.8, the pFDR does not actually provide a p-value adjustment. Recalling that the p-value is the smallest Type I error rate at which the null hypothesis is rejected, an analogue to the p-value associated with a particular test statistic with the pFDR approach is the q-value (Storey, 2001), which is analogously defined as being the smallest pFDR at which that test statistic is declared significant.

Further details of this way of assessing the effect of multiple testing are provided by Storey and Tibshirani (2001, 2002). Theoretical aspects of this procedure have been developed by Storey (2001, 2002) and Efron et al. (2001) by casting it in an Empirical Bayes framework.

7.10 SMALL VARIANCE-ADJUSTED t TESTS AND SAM

Let us now revisit the t test. With small samples the t test statistic tends to be highly correlated with the standard error term that appears in its denominator. As a result the test has a propensity for picking up significant findings at a higher rate from among those genes with low sample variance than from among those genes with high sample variance (as observed in Figure 7.2). This property of the t test is especially troubling because it is difficult to estimate standard errors well when the sample size is low and small standard errors can occur purely by chance. Since the sample sizes used in microarray experiments are typically very small, the small sample effect of the t test tends to manifest itself in such experiments as a high false positive rate for genes whose variability is low and a high false negative rate for genes whose variability is high. This effect is somewhat related to the problem of competition bias in model selection, where when several models compete to be selected, the ones that appear the best with the data at hand get selected. This is clearly an undesirable state of affairs, and proposals to avoid this problem have begun to appear in the microarray data analysis literature.

Example. Figure 7.6 shows scatterplots of (1) the two-sample t test statistics versus the pooled standard errors for the two groups and (2) the absolute value of the two-sample t test statistics versus the pooled standard errors for the two groups. The scatterplot on the left has a rotated volcano shape indicating that genes with small variances have large t test statistics, and vice versa. Figure 7.7 shows the proportion of significant t statistics as a function of the pooled standard errors for α values of 0.05, 0.01, and 0.001. The graphs demonstrate the

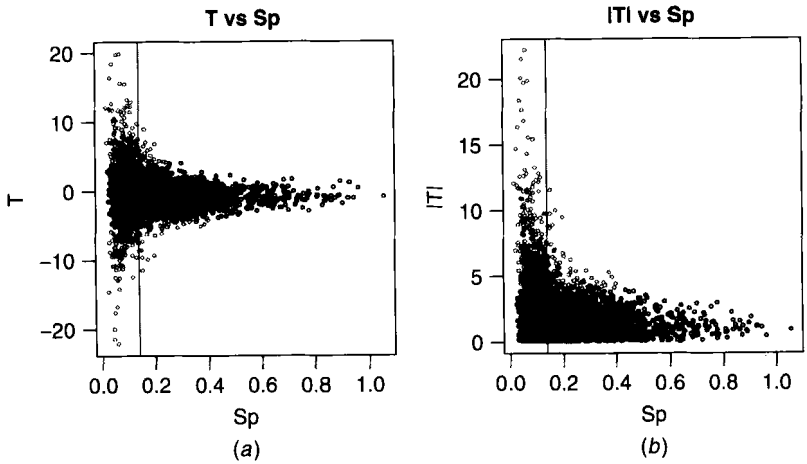


Figure 7.6 Scatterplot of the (*a*) t statistic, (*b*) absolute value of the t statistic versus the pooled standard error. In each graph the gray dots refer to genes declared significant by the two-sample t test, the vertical line is the median pooled standard error.

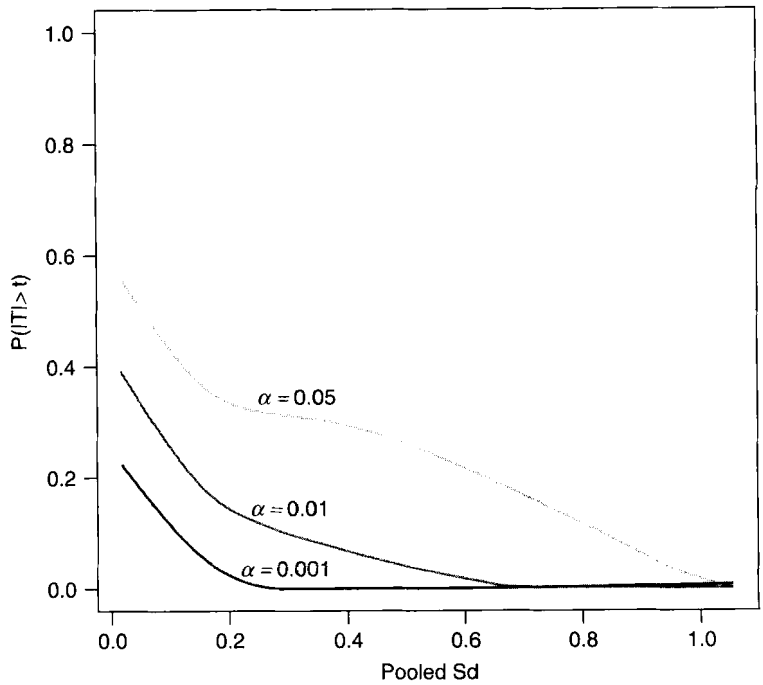


Figure 7.7 Proportion of $|t| > t_{\alpha/2}$, $\alpha = 0.05, 0.01, 0.001$.

problem of obtaining too many significant t-statistics for small values of the pooled standard error.

7.10.1 Modifying the t Statistic

One solution to the problem was suggested by Tusher et al. (2001). They add a carefully chosen constant, a so-called fudge factor, to the denominator of the t statistic. Recall that the t test statistic, T_g, for the gth gene, has the form $T_g = r_g/s_g$, where $r_g = |\bar{x}_g - \bar{y}_g|$ and $s_g = s_{pg}\sqrt{(1/n_1) + (1/n_2)}$ (see Section 7.3). The adjusted t statistic is

$$T_g(c) = \frac{r_g}{s_g + c},$$

where c is the fudge factor. This test statistic is often called the *SAM t statistic*, where SAM stands for "significance analysis of microarrays."

$T_g(0)$ is, of course, the ordinary t statistic, T_g. $T_g(c)$ with a very large value of c is equivalent to the t statistic without its denominator, namely to r_g. The plan is to choose an intermediate positive value of c for which, given c, the dependence of $T_g(c)$ on s_g is as small as possible. The simplest way to do this, in practice, is to study the relationship of $T_g(c)$ versus s_g for a number of different values of c, with the intention of retaining as the fudge factor, c, the one for which the dependence of $T_g(c)$ on s_g is least.

Tusher et al. (2001) (see also the documentation accompanying the software package, SAM) implement this is as follows: Let s^α be the αth percentile of the $\{s_g\}$ values, and let $T_g(s^\alpha) = r_g/(s_g + s^\alpha)$. Compute the percentiles, $q_1 < q_2 \cdots < q_{100}$, of the s_g values. For $\alpha \in \{0, 5, 10, \ldots, 100\}$, compute the mad (median absolute deviation from the median), $v_j(\alpha)$, of the $T_g(s^\alpha)$ values within the interval $[q_j, q_{j+1}]$ for $j = 1, 2, \ldots, n$. Then compute $cv(\alpha)$, the coefficient of variation of the $v_j(\alpha)$ values. Choose as $\hat{\alpha}$ the value of α that minimizes $cv(\alpha)$. Fix as $\hat{c}$ the value $s^{\hat{\alpha}}$.

An alternative proposal for estimating the fudge factor (Broberg, 2002) involves studying the false negative rate versus the false positive rate, a relationship called the *receiver operating characteristic* (ROC) curve, for various values of c, and choosing as the fudge factor the value of c that corresponds to the point on the ROC curve that is nearest the origin.

7.10.2 Assessing Significance with the SAM t Statistic

Once the SAM t statistics, $T_g(\hat{c})$, are calculated, the critical value of $T_g(\hat{c})$ that separates significance from nonsignificance must be set. For the ordinary t statistic this is done by looking up the quantiles of a t-distribution. However, the null distribution of the SAM t statistic, $T_g(\hat{c})$, is not a t-distribution, so this is no longer correct. In fact the null distribution is intractable. Therefore Tusher

et al. (2001) assess the significance of the observed $T_g(\hat{c})$ values via a permutation procedure.

Suppose that a suitable $\hat{c}$ has been identified and that the $T_g(\hat{c})$ values have been calculated and sorted into increasing order: $T_{(1)}(\hat{c}) \leq T_{(2)}(\hat{c}) \leq \cdots \leq T_{(G)}(\hat{c})$. The permutation procedure proceeds by permuting the columns of the data matrix, X, and assigning the first n_1 columns to group 1 and the remaining n_2 columns to group 2. A total of B such permutations will be done. For the bth permutation, compute the statistics, $T_g^{*b}(\hat{c})$, and the corresponding order statistics: $T_{(1)}^{*b}(\hat{c}) \leq T_{(2)}^{*b}(\hat{c}) \leq \cdots \leq T_{(G)}^{*b}(\hat{c})$. From the set of B permutations, the expected order statistics of $T_g(\hat{c})$ can be estimated by $\bar{T}_{(g)}(\hat{c}) = \sum_{b=1}^{B} T_{(g)}^{*b}(\hat{c})/B$. Any gene g that is such that its $T_g(\hat{c})$ value substantially exceeds its $\bar{T}_{(g)}(\hat{c})$ value is possibly differentially expressed.

This can be examined further by plotting the $T_g(\hat{c})$ values versus the $\bar{T}_{(g)}(\hat{c})$ values. The central part of this plot lies along the identity line, where $T_g(\hat{c}) = \bar{T}_{(g)}(\hat{c})$, indicating genes that are not differentially expressed. The ends tail away from this line; the further a gene is located from the identity line, the more likely it is that the gene is significantly differentially expressed.

The procedure to declare significance is as follows: For a fixed threshold, Δ, starting at the origin and moving up to the right, find the first i_1 genes such that $T_g(\hat{c}) - \bar{T}_{(g)}(\hat{c}) > \Delta$ and call all genes past i_1 "significant positive." Similarly, starting at origin and moving down to the left, find the first i_2 genes such that $T_g(\hat{c}) - \bar{T}_{(g)}(\hat{c}) < -\Delta$ and call all genes past i_2 "significant negative." For a given value of Δ, call the smallest value of $T_g(\hat{c})$ among the significant positive genes the "upper cut point," $\text{cut}_{up}(\Delta)$, and the largest value of $T_g(\hat{c})$ among the significant negative genes the "lower cut point," $\text{cut}_{lo}(\Delta)$.

This process can be carried out for a series of Δ values. For each value of Δ,

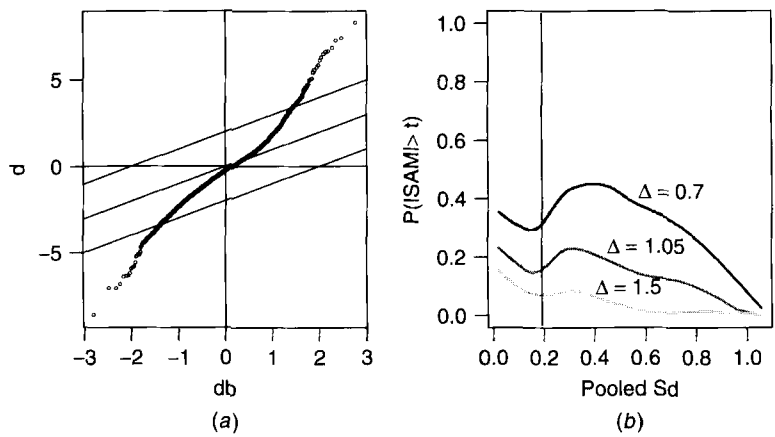


Figure 7.8 Graphs showing the performance of SAM. (*a*) Scatterplot of the sorted SAM t statistics versus their expected values; the oblique lines correspond to $\Delta = 2$. (*b*) Proportion of significant genes produced by SAM versus pooled standard error.

count the total number of significant genes and determine the average number of genes falsely identified as differentially expressed. For the latter, compute the median, k_m, and the 90th percentile, $k_{0.9}$, of the proportions of values among each of the B sets of $T^{*b}_{(g)}(\hat{c})$ values that fall in between the cut points, $\text{cut}_{lo}(\Delta)$ or $\text{cut}_{up}(\Delta)$. The proportion of genes that are truly not differentially expressed (i.e., π_+ in Section 7.9.1) is taken to be twice the proportion of values, $T_g(\hat{c})$, that fall in between the quartiles of all the values of all the $T^{*b}_{(g)}(\hat{c})$ values. The values of k_m and $k_{0.9}$ are multiplied by this proportion and used to calculate the positive false discovery rate, pFDR, as k_m (or $k_{0.9}$) divided by the number of significant genes. By evaluating pFDR for several values of Δ, a suitable strategy can be devised to decide which genes are significantly differentially expressed.

Example. Figure 7.8*a* shows a scatterplot of the sorted SAM t statistics versus their expected values for the example—the oblique lines correspond to $\Delta = 2$. Figure 7.8*b* shows the proportion of significant genes produced by SAM versus the pooled standard error. Table 7.2 gives a list of typical values of Δ, the number of false discoveries, the number of genes declared significant, the pFDR for both 50% and 90% and the FPR (false positive rate) for both 50% and 90%.

Table 7.2 Summary of a SAM analysis

Δ	FP (50%)	FP (90%)	Called	FDR (50%)	FDR (90%)	FPR (50%)	FPR (90%)
0.1	1847	1951	3514	0.526	0.555	0.453	0.479
0.2	1351	1515	2996	0.451	0.506	0.331	0.372
0.3	949	1130	2550	0.372	0.443	0.233	0.277
0.4	645	812	2182	0.296	0.372	0.158	0.199
0.5	400	538	1787	0.224	0.301	0.098	0.132
0.6	249	366	1567	0.159	0.233	0.061	0.090
0.7	143	228	1306	0.109	0.175	0.035	0.056
0.8	76	135	1112	0.068	0.121	0.019	0.033
0.9	39	80	931	0.042	0.086	0.010	0.020
1.0	20	45	746	0.027	0.061	0.005	0.011
1.1	10	28	628	0.017	0.045	0.002	0.007
1.2	6	16	537	0.011	0.030	0.001	0.004
1.3	4	8	446	0.008	0.019	0.001	0.002
1.4	2	5	389	0.005	0.014	0.000	0.001
1.5	1	3	311	0.004	0.010	0.000	0.001
1.6	1	2	269	0.002	0.009	0.000	0.000
1.7	0	2	226	0.000	0.008	0.000	0.000
1.8	0	1	186	0.000	0.003	0.000	0.000
1.9	0	1	154	0.000	0.004	0.000	0.000
2.0	0	1	139	0.000	0.004	0.000	0.000

7.10.3 Strategies for Using SAM

Since the pFDR does not actually provide a p-value adjustment and pFDR is not monotone in Δ, it is sometimes unclear as to how to decide which genes are significantly differentially expressed, that is, essentially, how to set Δ. Some strategies for selecting a suitable value for Δ are as follows:

1. Settle on the highest pFDR the researcher is willing to tolerate (e.g., 5% or 1%). Select the smallest value of Δ that corresponds to that pFDR. In our example in Table 7.2, if we choose pFDR (90%) = 1%, this corresponds to $\Delta = 1.2$.
2. It is sometimes difficult to prespecify a value for pFDR or Δ. In the event it may be more convenient to stay with the more familiar "classical" strategy of choosing a Δ that corresponds to a fixed proportion of false positives, say 0.01. From Table 7.2 this method would produce $\Delta = 1.1$.
3. Begin with strategy 2 to pick a Δ, then check the pFDR for that Δ, and if the pFDR is too high, increase Δ as long as (a) there is a sizeable reduction in the pFDR and (b) the number of genes declared significant does not decrease substantially. For Table 7.2 we may argue that $\Delta = 1.1$ corresponds to a pFDR of 4.5%, which is sufficiently low.
4. Begin with an initial number of genes the researcher would like to follow up on. Calculate the pFDR and FPR for that number. If they are satisfactorily small, stop. Otherwise, adjust the number of genes until both pFDR and FPR are at comfortable levels.

We may still pick up genes exhibiting fold changes that are so small as to be biologically irrelevant. In the event that we want to omit them and pick up only those genes that exhibit at least an h-fold change, then, in addition to being significant positive or significant negative, a gene must also satisfy $|\bar{x} - \bar{y}| > \log(h)$ in order to be declared significantly differentially expressed.

7.10.4 An Empirical Bayes Framework

The theoretical underpinnings of the SAM approach were investigated by Efron et al. (2001) by casting it in an Empirical Bayes framework. This framework is as follows: Suppose that p_E is the probability that a gene is differentially expressed and that $f_E(z)$ and $f_0(z)$ denote the probability density functions of $Z = T(c)$ for genes that are differentially expressed and not differentially expressed respectively. Then

$$f(z) = (1 - p_E)f_0(z) + p_E f_E(z)$$

is the probability density function for the mixture distribution of Z.

Applying Bayes's rule to this model gives the posterior probability that a gene is differentially expressed:

$$p_E(z) = 1 - \left[(1 - p_E)\frac{f_0(z)}{f(z)}\right].$$

The density $f(z)$ can be estimated from the observed $\{Z_g\}$ values. The null density $f_0(z)$ is estimated by permuting the columns of X as with the SAM procedure. Efron et al. (2001) describe how to use logistic regression to estimate $f_0(z)/f(z)$. The probability p_E is set equal to its maximum value: $1 - \min_Z\{f(Z)/f_0(Z)\}$. Based on these estimates, the posterior probability $p_1(z)$ that a gene is differentially expressed can be determined for each gene.

7.10.5 Understanding the SAM Adjustment

In order to understand what SAM does, we present a careful analysis of the original microarray data question and explain the behavior of SAM.

Microarray data typically exhibits a strong dependence relationship between gene effect mean and variance, that is, between μ_g and σ_g (see Fig. 7.9). This

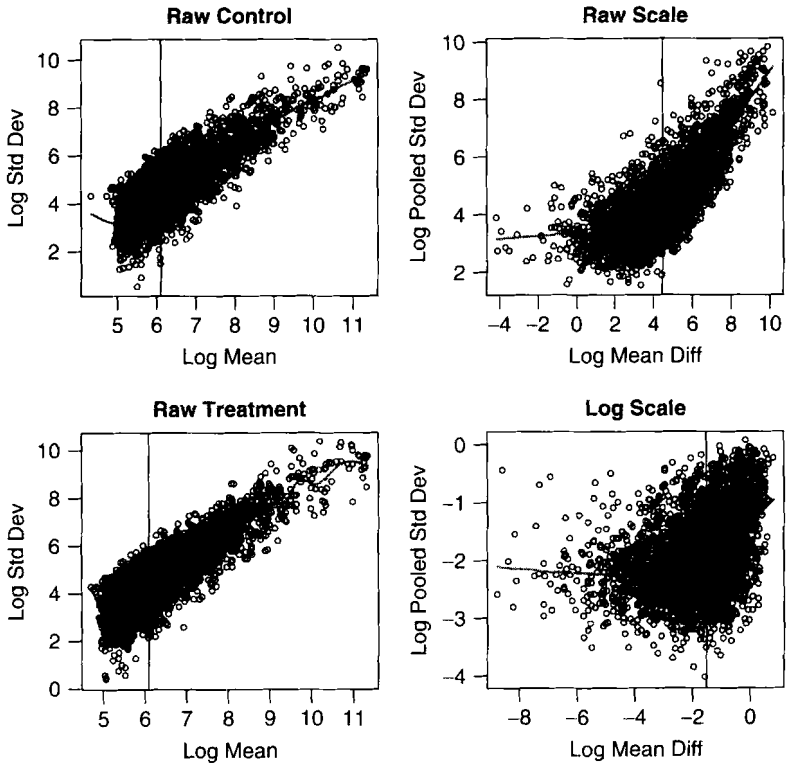


Figure 7.9 The four graphs show the strong dependence between location and scale at the raw scale. But, for the pool standard error and mean differences of the logged data, the dependence is much smaller.

dependence is reflected in the values of $\bar{x}_g$ and s_g in the raw data. This is one reason why a transformation such as log or square root, followed by a normalization step, are applied to the original data as described in Chapter 5. The hope is that by doing these transformations, some or all of the following problems will be resolved for the overwhelming majority of the genes:

1. μ_g and σ_g are dependent.
2. Even if μ_g and σ_g are independent, the σ_g are not homogeneous, because they vary from gene to gene.
3. σ_g varies from group to group.

One cannot expect to change or eliminate all these problems with a simple transformation. It maybe possible to eliminate the first or even the second alone, but in practice, we may expect combinations of the three or all the three problems to exist in one dataset.

Example. Figure 7.9 shows that the dependence has been reduced somewhat after log transformation and normalization.

This background is useful because it lets us study different scenarios that may arise in practical situations and enables us to understand what SAM will do in each of these cases. Here are the scenarios.

Case 1. Assume that σ_g is constant for all the groups and all the genes and σ_g is independent of μ_g for all groups.

In this case $T_g(c)$ and s_g are dependent for all values of c, but the correlation goes to zero as c goes to infinity. SAM will choose a value of α close to 100%. SAM with a large constant is equivalent to using the t statistic without the denominator, and finding the critical value for the t statistic that corresponds to significance.

Case 2. Assume that (a) σ_g is the same for both groups and is distributed as F_σ and (b) σ_g is independent of μ_g for both groups.

In this case $T_g(c)$ and s_g are negatively correlated for small c, but the correlation becomes positive as c goes to infinity. SAM will choose a fudge factor c that makes the correlation more or less zero. Simulation results suggest that when the distribution of σ_g is very skewed to the right, such as a chi-squared distribution with one or two degrees of freedom, the resulting fudge factor c corresponds to very small values of α near 0%. However, when the distribution of σ_g is not heavily skewed, SAM will choose values of α close to 100%. This does not imply that there is no dependence between $T_g(c)$ and s_g, since, for almost all distributions F_σ, there exists no constant, c, that makes the distribution of $T_g(c)$ independent of s_p. However, SAM may produce a reasonable reduction of the dependence.

Case 3. Assume that σ_g is the same for all groups and σ_g is dependent of μ_g for all groups and their joint distribution is $F_{\mu,\sigma}$.

In this case it appears also unlikely that a constant c would totally eliminate the dependence of the distribution of $T_g(c)$ from s_g. However, again, SAM should produce a reasonable reduction of the dependence.

Case 4. In addition to case 3 assume a correlation structure among the genes.

The column permutations in SAM preserves the correlation structure, but the same comments as above apply.

Example. Figures 7.1 and 7.9 indicate that σ_g and μ_g are very positively correlated. On the other hand, this correlation structure is reduced when we subtract the gene means, so it may be just a result of the high variation among gene mean effects. Nevertheless, it is worth noting that this effect should be checked before using the methods for cases 2 or 3 and some modifications may be necessary to account for it.

7.11 CONDITIONAL t

Amaratunga and Cabrera (2003c) propose a novel method of addressing the dependence of T_g from s_g by determining, from the distribution of T_g conditioned on s_g, the critical value of T_g that separates significance from nonsignificance. This method is called the *conditional t* (CT) approach.

The CT procedure provides a solution to the problem of small sample standard deviations by estimating the conditional distribution of T_g given s_g and calculating the critical values $t_\alpha(s)$ that help us decide which genes are up- or down-regulated and which are not according to whether or not $T_g > t_\alpha(s_g)$.

The procedure to calculate the critical values, $t_\alpha(s)$, will depend on which of the above four cases is assumed. We start with the basic method that will be used for handling case 2. Case 1 is difficult to separate from case 2, in practice, because we never know when the variances are constant or variable. Procedures for case 3 and case 4 are extensions of the procedure for case 2.

CT for Case 2. The simplest development of the method is in the situation in which σ_g is a realization from the distribution F_σ, where F_σ is the same for all the groups and all the genes and σ_g is independent of μ_g. The procedure is comprised of two steps:

Step 1. Estimate F_σ.

Step 2. Estimate the conditional distribution of $T_g|s_g$, and as a consequence, estimate the values $t_\alpha(s_g)$ for a few α's.

These steps are now described in more detail.

Step 1. Estimate F_σ. We know that the empirical distribution of s_g, namely $\hat{F}_s$, is a biased estimator of the distribution F_σ. This bias is especially large for very small sample sizes as is typical of many microarray experiments. The reason for the large bias is that the distribution of $s_p^2|\sigma^2$ is approximately $\sigma^2\chi^2_{n-1}/(n-1)$. It follows that, for small n, the marginal distribution of s_p^2 has heavier tails than those of F_σ.

This bias can be corrected by using a simulation method initially proposed in Amaratunga and Cabrera (2001b). This method is itself a version of the *target estimation* procedure of Cabrera and Fernholz (1999) and Cabrera and Watson (1997). The idea is to estimate the function $g : [0:1] \to [0,1]$ defined by $g(F_\sigma(x)) = \hat{F}_s(x)$. Since g is strictly monotonic, it can be inverted in order to obtain an estimate of $F_\sigma(x)$. The steps to estimate $F_\sigma(x)$ are as follows:

a. Generate a null distribution for the data by subtracting the sample means and dividing by the standard deviations.

b. Assume that $\hat{F}_s(x)$ is the true distribution of σ. Then resample from the null distribution of x and multiply each sample by a σ generated from $\hat{F}_s(x)$. Repeat this 10,000 times and, this way, get 10,000 pairs of samples for 10,000.

c. From each pair of samples, calculate a value for the pooled sample standard deviation, namely s_g^*, for $g = 1, \ldots, 10{,}000$. Let $\hat{F}_{s^*}(x)$ be the empirical distribution of the s_g^*'s. Then the estimator of g is obtained by mapping the empirical distribution $\hat{F}_s$ into $\hat{F}_{s^*}$. More precisely

$$\hat{g}(y = \hat{F}_s(x)) = \hat{F}_{s^*}(\hat{F}_s^{-1}(y)) \quad \text{and} \quad \hat{g}^{-1}(y) = \hat{F}_s(\hat{F}_{s^*}^{-1}(y)).$$

Hence the estimator of F_σ is

$$\hat{F}_\sigma(x) = \hat{F}_s(\hat{F}_{s^*}^{-1}(\hat{F}_s(x))).$$

$\hat{F}_\sigma(x)$ will be used in the second part of the method to generate the standard deviations of the gene populations.

Step 2. The second part of the method involves generating the conditional distribution of $t|s_d$, and the first steps are the same as Steps a and b of the algorithm above:

a. Generate a null distribution for the data by subtracting the sample means and dividing by the standard deviations.

b. Resample from the null distribution of x and multiply each sample by a σ generated from $\hat{F}_\sigma(x)$. Repeat this 10,000 times and, in this way, obtain 10,000 pairs of samples for 10,000. From each pair of samples, calculate a value for the pooled sample standard deviation and the two-sample t statistic, namely s_g and t_g for $g = 1, \ldots, 10{,}000$.

c. We estimate $t_\alpha(s_g)$ using a quantile regression estimate for t_g versus s_g and estimate the regression quantile curve for the $1-\alpha$ quantile. A crude

but effective way to estimate the quantile curve is to split the 10,000 points into 100 groups of 100 points sorted by s_g and calculate the $1 - \alpha$ quantile for each group. We call it $t_{(j)}$ and calculate the group medians for s_g, and call it $s_{(j)}$ $j = 1, \ldots, 100$. Then we estimate $t_\alpha(s_g)$ by fitting a smoother such as lowess or a smoothing spline to $t_{(j)}$ versus $s_{(j)}$. To estimate the quantile function, it is recommended to take the log of $t_{(j)}$ and $s_{(j)}$ first and then to estimate the quantile function.

CT for Case 3. Now assume that σ and μ are not independent. The main difficulty is that instead of estimating the distribution of σ alone, we must now obtain and estimate the joint distribution of σ and μ. Conceptually this is not a problem, but computationally it requires using two-dimensional smoothers and inverting two-dimensional functions.

The first part of the procedure is more complicated than for case 2 because we need to invert a function $h : R^2 \to R^2$. We can assume that h is continuous and differentiable, and g has to be one to one in order to have a well-defined inverse h^{-1}.

The second part of the procedure in case 2 can be replicated for case 3, with the exception that σ_g and μ_g are sampled from their joint distribution. As in case 2 we estimate the cutoff, $t_\alpha(s_p)$, from the joint distribution of t_α and s_p by conditioning on s_p. Moreover it is also possible to estimate the cutoff conditioning also on the overall sample mean, $t_\alpha(s_g, \bar{x}_p)$.

It is easy to see that the overall error rate of the CT procedure is α. Let s and t be the random variables representing the pooled variance estimate and the t statistic for a randomly selected gene. Let $f(t, s)$ be the joint probability density function of t and s. This is a mixing distribution, since s has a distribution that depends of the gene. The CT procedure consists of rejecting a null hypothesis if $t > h(s)$ and conditioning on s the probability of type one error is α. The following calculation shows that the overall unconditional probability of type one error is also α:

$$\int_0^\infty \int_{h(s)}^\infty f(t,s)\,dt\,ds = \int_0^\infty \left(\int_{-\infty}^\infty f(t,s)\,dt \right) \frac{\int_{h(s)}^\infty f(t,s)\,dt}{\int_{-\infty}^\infty f(t,s)\,dt} ds$$

$$= \int_0^\infty \int_{-\infty}^\infty f(t,s)\,dt\alpha\,ds = \alpha \int_0^\infty \int_{-\infty}^\infty f(t,s)\,dt\,ds = \alpha.$$

Example. Figure 7.10 shows a comparison between the pooled standard errors of the genes and three distributions: χ distributions with 0.5, 2, and 6 degrees of freedom. If the assumption of equal variances was true (and the genes were all independent of one another), we would expect that the pooled standard errors would be approximately proportional to a χ distribution with 6 degrees of freedom. Instead, they appear to be closer to a χ distribution with 0.5 degrees of freedom. Such a large difference strongly suggests that the gene variances are heterogeneous. Therefore we apply the CT method for case

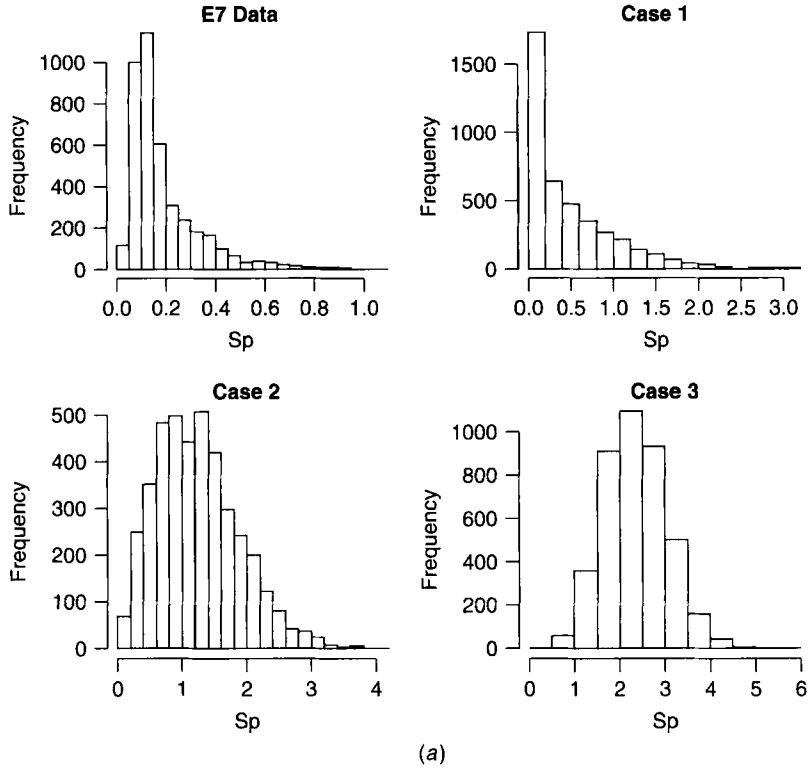
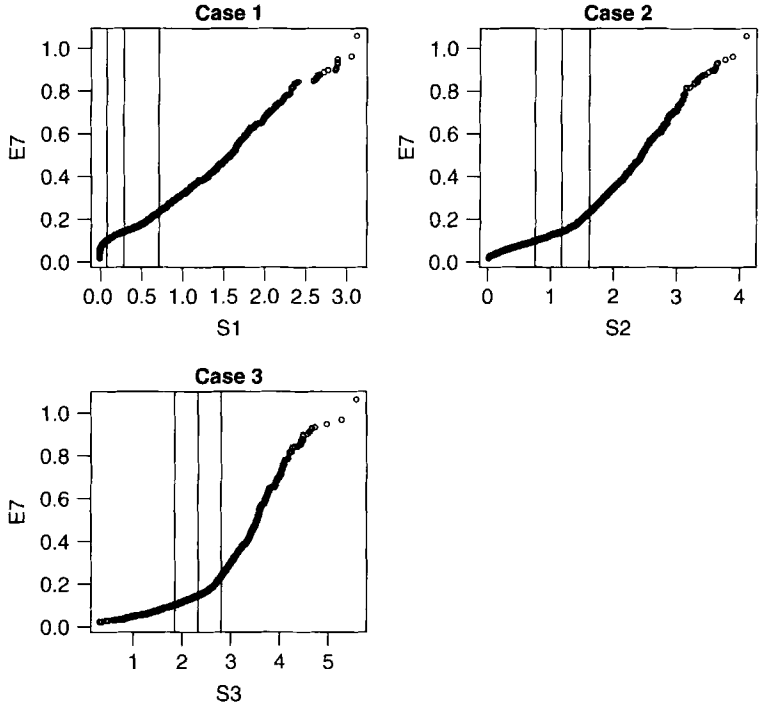


Figure 7.10 (*a*) A histogram of the pooled standard errors for the genes is compared to three distributions. Case 1: Chi-square with 0.5 degrees of freedom. Case 2: Chi-square with 2 degrees of freedom. Case 3: Chi-square with 6 degrees of freedom. (*b*) Quantile-quantile plots of the pooled standard errors for the genes versus three distributions. Case 1: Chi-square with 0.5 degrees of freedom. Case 2: Chi-square with 2 degrees of freedom. Case 3: Chi-square with 6 degrees of freedom.

2. The curves in Figure 7.11 represent the proportion of significant genes reported by the CT procedure using the method for case 2. It is clear that the CT method greatly reduces the dependence of the significance of the t statistic on s_g. Although SAM does an exemplary job of correcting the problem at the low range of s_g, it is not clear that a simple constant correction will produce a homogeneous t across the whole range of s_g values. The CT approach is a more direct means despite its call for more assumptions. It should produce, in general, a more homogenous result.

7.12 BORROWING STRENGTH ACROSS GENES

Inferences drawn from experiments with little replication can be terribly unreliable and nonreproducible. Largely this is because the fewer the number of



(*b*)

Figure 7.10 *(Continued)*

samples used in an experiment, the less precise the variance estimates are. Using the signal-to-noise analogy, without a reasonable estimate of "noise," it becomes difficult to separate the "signal" from the "noise."

This is of particular concern with microarray experiments as they are notorious for having few true replicates, particularly biological replicates. Statistical inferences reached purely on an individual gene basis could be driven by weak variance estimates and may not be particularly trustworthy. On the other hand, even if a microarray experiment has only a few replicates, there is always data on a large number of genes. Thus an appealing idea for improving inferences from microarray experiments is to "borrow strength across genes."

7.12.1 Simple Methods

The simplest approach is to assume that every gene has the same variance and then estimate that variance as the average variance across all the genes:

$$s_{g1}^2 = \frac{\sum_{h=1}^{G} s_h^2}{G}.$$

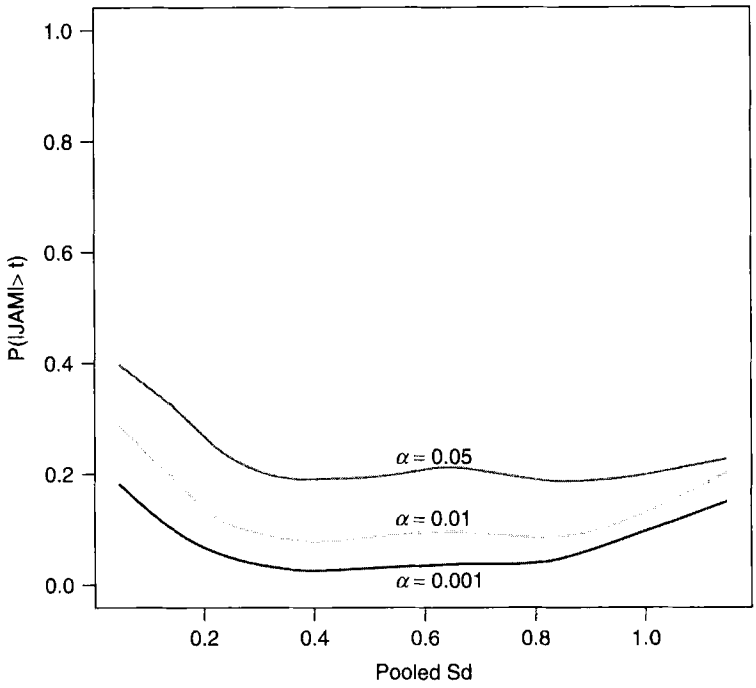


Figure 7.11 Proportion of significant genes producted by the conditional t method for values of $\alpha = 0.05, 0.01, 0.001$ versus the pooled standard errors.

Amaratunga and Cabrera (2001b) describe such a situation, although they use a different approach.

However, rarely is it the case that all genes have the same variance. More often, the variance tends to be high for genes whose expression levels are high, low for genes whose expression levels are low, and variances of genes whose expression levels are similar to one another tend to be closer than genes whose expression levels are very different from one another. In this case it is reasonable to assume that $\sigma_{gt}^2 = f(\mu_{gt})$, where f is a smooth continuous function. So we can fit the model

$$s_g^2 = f(\bar{x}_g) + e_g$$

using a semiparametric smoothing procedure such as lowess or a spline smoother, and take the fitted value as s_g^2,

$$s_{g2}^2 = \hat{f}(\bar{x}_g).$$

Another approach on the same lines is to use as s_{g2}^2 the local average of the standard deviation for genes showing similar expression levels as gene g. To

do this, first order all the genes within a treatment group according to their average expression levels. Then consider the given gene and the k next higher expressing genes and the k next lower expressing genes. Take the average of the standard deviations of these $2k + 1$ genes as s_{g2}^2.

Using either s_{g1}^2 or s_{g2}^2 may produce an overly smooth estimator of σ_g^2. This can be remedied by estimating σ_g^2 by a composite estimator that is a weighted combination of the observed variance of the gth gene and the smoothed variance estimate:

$$\hat{\sigma}_g^2 = \lambda s_g^2 + (1 - \lambda)s_{g1}^2 \quad \text{or} \quad \hat{\sigma}_g^2 = \lambda s_g^2 + (1 - \lambda)s_{g2}^2.$$

Doing this results in increased precision for variance estimates, and thus inferences are made more trustworthy. However, the question remains, how to choose λ?

7.12.2 A Bayesian Model

Baldi and Long (2001) present a formal development of modeling the (μ, σ^2) dependence by casting it in a Bayesian framework. They begin by assuming that the data has a normal distribution, so that for a particular gene in a particular group

$$x_i \sim \mathrm{N}(\mu, \sigma^2),$$

where, for simplicity, we have omitted the subscript g for gene and j for group. Let $\bar{x}$ and s^2 denote the sample mean and sample variance for this gene in this group.

Following fairly standard Bayesian practice, the variance parameter, the prior distribution of μ and σ^2 is defined in two parts as follows: σ_g^2 is taken to follow an inverse gamma distribution while, given σ_g^2, the mean parameter μ_g is taken to follow a normal distribution:

$$\mu|\sigma^2 \sim \mathrm{N}\left(\mu, \frac{\sigma^2}{\lambda}\right),$$

$$\sigma^2 \sim \mathrm{IG}(\nu_0, \sigma_0^2).$$

This formulation corresponds to a conjugate prior. A bonus is that the resulting joint prior distribution of (μ, σ^2) forces μ and σ^2 to be dependent, as can be observed in many microarray experiments.

Applying Bayes's theorem, followed by some algebraic manipulations, we can obtain the posterior distribution:

$$(\mu, \sigma^2|x) \sim \mathrm{N}\left(\mu, \frac{\sigma^2}{\lambda}\right) \mathrm{IG}(\nu_n, \sigma_n^2)$$

with

$$\mu_n = \frac{\lambda_0}{\lambda_0 + n}\mu_0 + \frac{n}{\lambda_0 + n}\bar{x},$$

$$\lambda_n = \lambda_0 + n,$$

$$\nu_n = \nu_0 + n,$$

$$\nu_n\sigma_n^2 = \nu_0\sigma_0^2 + (n-1)s^2 + \frac{\lambda_0 n}{\lambda_0 + n}(\bar{x} - \mu_0)^2.$$

Observe that the parameters of the posterior distribution combine information from the prior and the data. The posterior mean is a weighted average of the prior mean and the sample mean. The posterior degrees of freedom is the prior degrees of freedom plus the sample size. The posterior sum of squares is the prior sum of squares plus the sample sum of squares plus the residual uncertainty due to the discrepancy between the prior mean and the sample mean.

The prior mean is usually set to the sample mean: $\mu_0 = \bar{x}$ so that $\mu_n = \bar{x}$. The mean of the posterior distribution is

$$\mu_P = \mu_n = \bar{x},$$

$$\sigma_P^2 = \frac{\nu_n}{\nu_n - 2}\sigma_n^2 = \frac{\nu_0\sigma_0^2 + (n-1)s^2}{\nu_0 + n - 2},$$

provided that $\nu_0 + n > 2$.

Baldi and Long (2001) use a simple rule of thumb to assign a value to ν_0. They assume that a minimum of K points are necessary to adequately estimate a standard deviation (they use $K = 10$, but Tukey has made persuasive arguments that $K = 30$) and choose ν_0 so that $\nu_0 + n = K$.

These values are then plugged into the t test statistic.

7.13 TWO-CHANNEL EXPERIMENTS

Consider a two-channel microarray experiment whose objective is to compare two types of samples, A_1 and A_2. Suppose that there are K microarrays and that on each microarray there are two channels, one channel corresponding to A_1 and the other channel corresponding to A_2. Let X_{gjk} be the log-transformed and normalized spot intensity level for gth gene and the channel of the jth array that corresponds to sample type A_k; here $g = 1, \ldots, G$, $j = 1, \ldots, n$, and $k = 1, 2$.

The value $Y_{gj} = X_{gj1} - X_{gj2}$ is the difference in the log expression level between the two channels in the jth sample for the gth gene (or equivalently the log of the fold change between the two channels); $\bar{Y}_g = \sum_{j=1}^{n} Y_{gj}/n$ is the mean

and $V_g = \sum_{j=1}^{n}(Y_{gj} - \bar{Y}_g)^2/(n-1)$ is the variance of the $\{Y_{gj}\}$ values for the gth gene.

7.13.1 The Paired Sample *t* Test and SAM

The *paired sample t test* statistic for testing for differential expression is

$$T_g = \frac{\bar{Y}_g}{\sqrt{V_g/n}}.$$

If the data is drawn from a normal distribution, the null distribution of T_g is a t-distribution with degrees of freedom $v = n - 1$. If the observed value of T_g is $T_{g;obs}$, then the p-value is given by the probability $p_g = \text{Prob}(|T_g| > T_{g;obs})$. A gene is declared significantly differentially expressed at level of significance α if $p_g < \alpha$.

As described in Section 7.10 for the two-sample t test, the paired sample t test can also have the problem that, with small samples, that the t test statistic tends to be highly correlated with the standard error term appearing in its denominator. This results in a high false positive rate for genes whose variability is low and a high false negative rate for genes whose variability is high. The SAM modification to the t statistic described in Section 7.10.1 can also be used here. The modified SAM t statistic is

$$T_g = \frac{\bar{Y}_g}{(\sqrt{V_g/n}) + c},$$

where the fudge factor c is estimated as described in Section 7.10.1. The rest of the procedure then proceeds as described there.

7.13.2 Borrowing Strength via Hierarchical Modeling

Several authors (Lee et al., 2000; Efron et al., 2001; Newton et al., 2001; Pan et al., 2002; Lonnstedt and Speed, 2002) have proposed various ways of borrowing strength across genes via Bayesian hierarchical modeling. These constructions begin by assuming that some unknown proportion, p_E, of the G genes are actually differentially expressed. For those genes, the indicator variable $I_g = 1$ while, for the rest, $I_g = 0$. The question is to determine, based on the data, which genes are the most likely to truly have $I_g = 1$.

A mixture model is developed as follows: Suppose that p_E is the probability that gene g is differentially expressed and that $f_E(y)$ and $f_0(y)$ denote the probability density functions of Y_g for genes that are differentially expressed and not differentially expressed respectively. Then

$$f(y) = (1 - p_E)f_0(y) + p_E f_E(y)$$

is the probability density function for the mixture distribution of Y_g.

Lee et al. (2000) assume a normal distribution for Y_g, for each g, the various components of the mixture model can be estimated using, for example, the EM algorithm (Dempster et al., 1977).

Lonnstedt and Speed (2002) similarly assume that $Y_{gj} \sim N(\mu_g, \sigma_g^2)$. In the prior distribution, genes having no previous evidence of effects are considered exchangeable, making the posterior mean effect for each gene borrow strength from the observed effects of the other genes. Thus a large observed effect of a gene will be shrunk toward zero in the posterior mean when the gene is exchangeable with other genes that have mostly small observed effects. Candidate genes, for which there is some prior evidence for an effect, will be treated separately.

Following fairly standard Bayesian practice as before (Section 7.3.2), the variance parameter, σ_g^2, is taken to follow an inverse gamma distribution while, given σ_g^2, the mean parameter μ_g is taken to follow a normal distribution:

$$\tau_g = \frac{na}{2\sigma_g^2} \sim \Gamma(\upsilon, 1),$$

$$\mu_g|\tau_g = \begin{cases} 0 & \text{if } I_g = 0, \\ N\left(0, \dfrac{cna}{2\tau_i}\right) & \text{if } I_g = 1. \end{cases}$$

This formulation implies a correlation between the difference in means and the variance for those genes that are differentially expressed.

Applying Bayes's rule, we can work out the log posterior odds for the gth gene to be differentially expressed:

$$B_g = \log\left[\frac{\text{Prob}(I_g = 1 \mid M_{gj})}{\text{Prob}(I_g = 0 \mid M_{gj})}\right],$$

which works out to

$$B_g = \log\left(\frac{p}{1-p}\right) \frac{1}{\sqrt{1+nc}} \left[\frac{a + s_g^2 + \bar{M}_g^2}{a + s_g^2 + \dfrac{\bar{M}_g^2}{1+nc}}\right]^{\upsilon n/2}.$$

The B_g values provide a ranking of the genes with respect to the posterior probability of each gene being differentially expressed.

Generally, it is impossible to estimate simultaneously all the four parameters p, υ, a, and c. To circumvent this problem, p is fixed at some prespecified value. Then υ and a are estimated by the method of moments, while c is estimated from the top proportion of genes.

Another approach to this problem is given by Newton et al. (2001) based on the model given in Section 6.3.

SUPPLEMENTARY READING

Many basic statistical textbooks (e.g., Triola, 2001, 2002) describe the fundamentals of statistical hypothesis testing. Cox and Hinkley (1974) presents a more advanced philosophical discussion.

EXERCISES

7.1. Consider the following subset of data related to four genes, G1, G2, G3, G4. Their expression levels (log transformed and normalized) in four control tissues C1, C2, C3, C4, and four test tissues, T1, T2, T3, T4, are shown below:

	C1	C2	C3	C4	T1	T2	T3	T4
G1	9.011	9.064	9.067	9.008	8.944	9.087	8.963	9.074
G2	10.556	10.373	10.657	10.336	10.101	10.073	10.095	11.273
G3	11.967	12.014	11.757	12.101	11.604	11.782	11.503	11.861
G4	10.211	10.282	10.284	10.087	10.104	9.981	10.131	10.473

a. Are any of these genes significantly differentially expressed in test tissue versus control tissue if the investigator decides to regard twofold or greater-fold changes as significant.

b. Use a two-sample t test at the 5% level (two sided) to test whether any of the genes are significantly differentially expressed in test tissue versus control tissue.

c. Repeat part b using Welch's test.

d. Repeat part b using the Mann–Whitney–Wilcoxon test.

e. Repeat part b using the robust t test.

f. Repeat part b using a permutation test with the difference in medians as test statistic.

g. For each gene, examine the residuals to check whether any observations can be considered outliers. Observe that there are two extreme outliers. Remove them and repeat parts a to f. Do any conclusions change?

7.2. Golub et al. (1999) (data available online) compared the gene expression profiles of 11 AML patents with that of 27 ALL patients.

a. What method did the authors of this article use to select 50 genes?

b. Determine which genes are statistically significantly differentially expressed in AML versus ALL using **(i)** t test **(ii)** Welch's test **(iii)** robust t test **(iv)** SAM test **(v)** Wilcoxon Mann Whitney rank sum test, with and without a Bonferroni adjustment for multiplicity. Compare these lists with each other and the list obtained by the authors of the article.

c. Determine the pFDR for each of the unadjusted tests in part b.

7.3. The dataset `data7.txt` in the library `DNAMR` contains eight biological samples representing the expression levels for cell tissue for eight rats. Four of the rats are treated with a drug and the other four are treated with a placebo.

a. Calculate the mean, variance, and t test statistics for each gene, and construct the following plots for this data: cube root variance versus mean (average across groups), t test statistics versus mean, t test statistics versus cube root variance. Do you observe any volcano effect in your graph? Produce a list of significantly expressed genes.

b. Use the SAM methodology. Compare the four SAM strategies suggested in Section 7.10.3, and choose an appropriate value for Δ. Produce a list of differentially expressed genes that comes out for your choice of Δ, and report the pFDR and the expected number of false discoveries.

c. Using the functions included in the `DNAMR` library for `R/SPLUS`, determine the significant genes using the conditional t method. Finally, compare this list with the previous lists from parts (a) and (b) and try to reach a conclusion as to which genes are differentially expressed.

Exploration and Analysis of DNA Microarray and Protein Array Data
Edited by DHAMMIKA AMARATUNGA and JAVIER CABRERA

CHAPTER 8

Model-Based Inference and Experimental Design Considerations

Over the years a significant number of applied statistics problems have been successfully solved by, either directly or indirectly, applying statistical linear modeling techniques. Therefore it is not surprising that this workhorse of mainstream applied statistics has been tried and found to be useful for analyzing comparative microarray experiments as well. In addition to being of practical value for analysis, these models also provide a constructive framework upon which to reflect on what experimental designs might be appropriate for a proposed microarray experiment, an essential, but often sadly neglected, aspect of any research endeavor.

The literature on this topic is growing gradually. Kerr and Churchill (2001a, 2001b) and Kerr et al. (2000, 2002) are "early" references on the application of linear modeling techniques for the analysis of data from multichannel cDNA microarrays. They also advocated the application of sound experimental design principles to microarray experimentation and proposed various innovative designs for multichannel cDNA microarray experiments. Churchill (2002) and Yang and Speed (2002) are more recent reviews of this work. Wolfinger et al. (2001) proposed a two-stage approach for fitting linear models, including mixed effect models. Chu et al. (2002) discuss linear models for oligonucleotide array experiments.

Here we will review some of the statistical models that have been used for analyzing microarray data. Obviously the model that one would consider using in a particular circumstance depends entirely on the experimental design of the situation, and because it would be far too space-consuming to cover a large range of situations, we will focus only on some common ones. Experimental design issues will also be addressed.

Exploration and Analysis of DNA Microarray and Protein Array Data
By Dhammika Amaratunga and Javier Cabrera
ISBN 0-471-27398-8

8.1 THE *F* TEST

Consider a simple comparative microarray experiment whose objective is to investigate how genes express differentially across a single factor, V. The factor might represent treatments, tissue types, times, or something else. Following Kerr and Churchill (2001a, b), we will use the generic term "*varieties*" to refer to them. Let Y_{gij} be the suitably transformed and normalized expression level measurement for the gth gene ($g = 1, \ldots, G$) in the jth microarray ($j = 1, \ldots, J_i$) assigned to variety i ($i = 1, \ldots, I$). Let $N = \sum_{i=1}^{I} J_i$ denote the total sample size.

The simplest approach is to model the data for each gene separately as

$$Y_{gij} = \mu_g + V_{ig} + \varepsilon_{gij},$$

where μ_g represents the average signal for the gth gene, V_{ig} represents the additional signal due to the effect of the ith variety on the gth gene, and ε_{gij} represents an *error* term that subsumes all sources of variability not accounted for by the terms in the model, including random noise. The traditional assumption is that the $\{\varepsilon_{gij}\}$ are independently and identically distributed as a normal distribution with mean 0 and variance σ_g^2, which we write as $\varepsilon_{gij} \sim NID(0, \sigma_g^2)$.

This model is fitted for each gene using ordinary least squares, and statistical theory shows that the estimates so obtained have several desirable optimality properties. The primary hypothesis of interest, whether the gth gene is differentially expressed across the varieties (i.e., whether $V_{ig} = 0$ for all i), can be tested for statistical significance via an F test. This type of approach is called *analysis of variance* (ANOVA).

The F test statistic for testing whether the gth gene is differentially expressed across the varieties involves the *mean square among varieties*,

$$MS(V) = \frac{\sum_{i=1}^{I} (\bar{y}_i - \bar{y})^2}{I - 1},$$

and the *mean square error*,

$$MS(E) = \frac{\sum_{i=1}^{I} \sum_{j=1}^{J_i} (y_{ij} - \bar{y}_i)^2}{N - 1},$$

where $\bar{y}_i = \sum_{j=1}^{J_i} y_{ij}/J_i$ is the mean of the ith group and $\bar{y} = \sum_{i=1}^{I} \sum_{j=1}^{J_i} y_{ij}/N$ is the overall mean (the subscript g has been omitted from all the equations for simplicity). The F test statistic,

$$F = \frac{MS(V)}{MS(E)},$$

is the average squared difference in intensities across the varieties, as measured by $MS(V)$, relative to the variability or "noise" in the observations, as measured by $MS(E)$. Under the null hypothesis of no difference in intensities across the varieties, both $MS(V)$ and $MS(E)$ are estimates of the error variance σ_g^2, so that their ratio, F, is close to unity and is distributed as a F distribution with $I-1$ and $N-I$ degrees of freedom. When there are differences among the varieties, $MS(V)$ would generally be substantially larger than $MS(E)$ and F would generally be substantially greater than unity. The larger the value of F, the greater is the weight of evidence against the null hypothesis. If the observed value of F is F_{obs}, then the p-value is given by the probability $p_F = \text{Prob}(F > F_{obs})$. A gene is declared significantly differentially expressed at level of significance α if $p_F < \alpha$.

Example. Experiment E8 was conducted to study the gene expression profiles of mice in response to a particular drug. Nine mice were treated with the drug and 1 hour, 2 hours, and 3 hours after treatment, three mice were randomly sacrificed and mRNA from their liver was harvested. In addition there were three control mice to provide 0 hour information. A dozen microarrays (one for each mouse) containing 2004 genes were challenged with the mRNA and intensity data collected. The F test was applied for each gene separately, and 335 were found to be significantly differentially expressed across the four treatments. When a Bonferroni correction was applied, only two genes were found to be significantly differentially expressed across the four time points.

The F test is an extension of the t test that can be applied when $I \geq 2$. When $I = 2$, the F test is equivalent to the usual two-sample t test based on T_e described in Section 7.3. With very small sample sizes, the F test statistic, like the t test statistic, tends to be highly correlated with the mean square error term that appears in its denominator, causing it to pick up significant findings at a higher rate from amongst those genes with low variance than from among those genes with high variance, resulting in a high false positive rate for genes whose variability is low and a high false negative rate for genes whose variability is high.

In the SAM approach of Tusher et al. (2001), the F test statistic is regularized to adjust for this effect as follows:

$$F(c) = \frac{MS(V)}{MS(E) + c},$$

where c is a fudge factor whose value is estimated as for the t test statistic as described in Section 7.4 to reduce the dependence of $F(c)$ versus c.

8.2 THE BASIC LINEAR MODEL

In a simple comparative microarray experiment whose objective is to investigate how genes express differentially across a single factor, varieties V, there are three *effects* or *factors*—varieties (V), arrays (A), and genes (G)—that could potentially influence the expression level measurements $\{Y_{gij}\}$, where Y_{gij} is the suitably transformed and normalized expression level measurement for the gth gene ($g = 1, \ldots, G$) in the jth microarray ($j = 1, \ldots, J_i$) assigned to variety i ($i = 1, \ldots, I$). Therefore it is reasonable to try to formulate a model that describes the relationship between Y_{gij} and these three effects and their interactions or some subset of them. We can use the platform of this model to estimate the extent of the influence of each effect and to assess how significant it is.

Before we write down a model, it behooves us to think about what effects we ought to include in it and what effects, if any, it may be acceptable, or even necessary, to exclude. The obvious candidates for any model are the *main effects*:

- An array effect, A, would account for overall differences in expression level measurements among the arrays after the effects of all the other factors in the model have been removed. If the normalization effort was successful, the array effect should be fairly small.
- A gene effect, G, would account for differences among the average expression level measurements across the multitude of genes. Such an effect transpires due to many causes. For example, the facts that some genes have higher natural expression levels than others, some sequences tend to be labeled more efficiently than others due to factors such as sequence length and sequence composition, and some genes tend to hybridize more efficiently than others.
- A variety effect, V, would account for differences in expression level measurements if some of the varieties are substantially higher or lower overall than others.

Then there are the various *two-factor interaction effects*:

- A variety–gene interaction effect, VG, would account for how a gene expresses differentially across the varieties. Given a particular gene g, if any one of the $(VG)_{gi}$ terms is larger than the others relative to the underlying variability, it means that that particular variety is inducing a higher level of expression than the other varieties. Clearly, contrasts among the $(VG)_{gi}$, for each g, are the quantities of greatest interest in comparative microarray experiments.
- An array–gene interaction effect, AG, would account for the variability of a spot across the arrays averaged over all the spots. This effect would be

observed if the concentration or amount of DNA spotted on the microarrays varies from array to array.

- A variety-array interaction effect, VA, would account for variability across the varieties for arrays. This effect, however, is not estimable as each array contains only a single variety.

The simplest additive linear model that can be fitted to all the genes simultaneously involves the factors V, A, G, and VG:

$$Y_{gij} = \mu + V_i + A_{j(i)} + G_g + (VG)_{gi} + \varepsilon_{gij}.$$

Here μ represents the average signal across the whole experiment, and the errors ε_{gij} are independently and identically distributed as a normal distribution with mean 0 and variance σ^2: $\varepsilon_{gij} \sim NID(0, \sigma^2)$. The model is fitted using ordinary least squares, and hypotheses of interest, such as whether each gene is differentially expressed across the varieties, can be tested for statistical significance via F tests.

With microarray data, it is possible that none of the assumptions of normality, independence, and homoscedasticity hold:

- *Nonnormality.* Empirical evidence seems to indicate that it is reasonable to assume that the error distribution is symmetric and normal-like (i.e., bell-shaped) in the middle. The problem is in the tails. For one thing they tend to be quite heavy (see Chapter 7), and except for very well-behaved experiments, there tend to be a handful of extreme outliers that could damage some estimates severely. In addition there is a truncation effect at very high gene expression levels as a result of saturation.
- *Lack of independence.* Genes rarely express in isolation but rather along biological pathways. Therefore it would be wrong to assume that the expression levels of the genes in the experiment are totally independent of one another. On the other hand, it is impossible to model any aspect of the gene correlation structure in advance and the size of the samples used in typical microarray experiments just does not permit it to be inferred from the data. Therefore, generally, the best one can do is to assume independence and hope that it does not affect the properties of the test too much. There is good reason to believe that this might be the case. Followup analyses, such as cluster based methods can then address the lack of independence should it remain a concern.
- *Heteroscedasticity.* In some microarray experiments it can be observed that all the genes appear, perhaps after a transformation, to have the same variance; in these instances it is fine to assert that $\varepsilon_{gij} \sim (0, \sigma^2)$. However, in most microarray experiments, it appears to be the case that those genes that exhibit high expression levels also tend to exhibit high variances, and vice versa; in these instances it is more appropriate to write $\varepsilon_{gij} \sim (0, \sigma_g^2)$,

which postulates a gene-specific variance. In fact it may even be appropriate, in some situations, to write $\varepsilon_{gij} \sim (0, \sigma^2_{gi})$ to cover the eventuality that if gene g is differentially expressed across the different varieties i, that it also has different variances across the varieties. In some instances it may be better to model the variance to level relationship explicitly by $\sigma^2_g = f(\mu_g)$ or $\sigma^2_{gi} = f(\mu_{gi})$, where μ_g denotes the true overall mean expression level of the gth gene and μ_{gi} denotes the true mean expression level of the gth gene in the ith variety.

8.3 FITTING THE MODEL IN TWO STAGES

There is a natural categorization of the effects in a microarray experiment into *gene-specific effects* (effects involving G) and *global effects* (effects not involving G). These two sets of effects are *orthogonal* to one another; that is, they are statistically independent of one another and the effect of one does not mask nor interfere with the other.

Motivated by this categorization and the computational and statistical disadvantages associated with fitting a linear model in one giant step, Wolfinger et al. (2001) suggested breaking the fitting down into two stages, essentially fitting two submodels, one submodel to the global effects (they call this the *normalization model*) and one submodel to the gene-specific effects (they call this the *gene model*).

The normalization model

$$Y_{gij} = \mu + V_i + A_{j(i)} + \delta_{gij}$$

is fitted first and serves to adjust the data for the global effects that otherwise could bias the gene-specific inferences. The array effect could be regarded as a random effect. There are no gene-specific effects in the model. The error term $\delta_{gij} \sim (0, \sigma^2_0)$.

In the second stage, the residuals, R_{gij}, from the normalization model are regarded as gene expression level measurements that have been centered and normalized for extraneous effects and used as input to the gene model:

$$R_{gij} = G_g + (VG)_{gi} + \varepsilon_{gij},$$

where the error term $\varepsilon_{gij} \sim (0, \sigma^2)$.

In principle, the two-stage fit and the one-stage fit should produce results that are close, if not identical, to each other, since the effects being fitted in the normalization model are orthogonal to the effects being fitted in the gene model. However, this is not exactly the case as the residuals, R_{gij}, are generally slightly correlated to one another. Nevertheless, this effect should be small, and in practice, there should be little difference between the two sets of results.

There are a few key advantages to the two-stage process:

- It is computationally much less demanding.
- When fitting the gene model, it is possible to accommodate gene-specific heteroscedasticity by letting $\varepsilon_{gij} \sim (0, \sigma_g^2)$.
- The first fit residuals, R_{gij}, can be used as input to clustering.

8.4 MULTICHANNEL EXPERIMENTS

In multichannel experiments, in addition to the effects mentioned in Section 8.1, there is a global effect due to dye (D). Some dyes tend to produce consistently higher fluorescent signals compared to other dyes. Therefore, when modeling such experiments, a dye main effect that measures the overall effect of dye-to-dye variability on expression level measurement should be included in the model. Now let Y_{gijk} denote the suitably transformed and normalized expression level measurement for the gth gene with the kth dye in the jth microarray representing variety i. Including the dye effect in the model, we can model this situation as

$$Y_{gijk} = \mu + V_i + A_j + D_k + G_g + (VG)_{gi} + \varepsilon_{gij}.$$

To account for spot-to-spot variation, we can add a term AG:

$$Y_{gij} = \mu + V_i + A_j + D_k + G_g + (VG)_{gi} + (AG)_{gj} + \varepsilon_{gij}.$$

In addition, to account for the possibility that dyes might be interacting with genes, we can add a dye-gene interaction effect DG:

$$Y_{gij} = \mu + V_i + A_j + D_k + G_g + (VG)_{gi} + (AG)_{gj} + (DG)_{gk} + \varepsilon_{gij}.$$

8.5 EXPERIMENTAL DESIGN CONSIDERATIONS

Most experiments involve studying how a variable of interest is affected by a series of factors. The *design* of such an experiment refers to the assignment of samples over the levels of the various factors. For microarray experiments the variable of interest is the expression level of a gene and the experimental design refers to the assignment of samples over the levels of factors such as variety and dye. The number of replicates to use for the various different types of replication is also an experimental design consideration.

8.5.1 Comparing Two Varieties with Two-Channel Microarrays

We will commence our discussion with multichannel microarray experiments, where the scope for improving inference by applying principles of classical experimental design is most apparent.

The following simple example can be used to illustrate some of the key points to keep in mind when designing a microarray experiment. An experimenter is planning to perform a DNA microarray experiment to compare the effects of two varieties A and B, and intends to use two two-channel cDNA microarrays, A_1 and A_2. We will call the two channels R and G to represent the two dyes, red and green, that are most often used. There are four obvious designs:

DESIGN D_1	ARRAY A_1	ARRAY A_2
Channel R	A	B
Channel G	A	B

DESIGN D_2	ARRAY A_1	ARRAY A_2
Channel R	A	A
Channel G	B	B

DESIGN D_3	ARRAY A_1	ARRAY A_2
Channel R	A	B
Channel G	B	A

DESIGN D_4	ARRAY A_1	ARRAY A_2
Channel R	A	B
Channel G	REF	REF

In design D_1, array specific effects are *confounded* with variety effects in that if a gene is differentially expressed in A_1 versus A_2, it will be impossible to know whether to attribute it to array or to variety. Thus it is better to avoid this design if possible. Of course, with single-channel arrays, this aspect is unavoidable. Incidentally, in experimental design parlance, arrays are essentially experimental blocks with as many levels as there are channels; in two-channel experiments, they are blocks of size two.

In design D_2, dye-specific effects are confounded with variety effects. Since it is known that sizable dye effects are possible, some care must be taken if using this design.

In design D_3, the dyes assigned to the two varieties in the first array are switched in the second array. This modification makes it possible to separate

out both array-specific effects as well as dye specific effects by fitting the model

$$Y_{gij} = \mu + V_i + A_j + D_k + G_g + (VG)_{gi} + (AG)_{gj} + (DG)_{gk} + \varepsilon_{gij}$$

or one of the other smaller models mentioned above. In the two-stage modeling approach, the normalization model would be

$$Y_{gij} = \mu + V_i + A_j + D_k + \delta_{gij1},$$

with each of the effects having one degree of freedom. The gene model would be

$$R_{gij} = G_g + (VG)_{gi} + (AG)_{gj} + (DG)_{gk} + \varepsilon_{gij}.$$

For obvious reasons this type of design is called a *dye-swap design* or *dye-flip design*. While there is a clear advantage to dye-swap designs, they do require some extra effort on the part of the experimenter because each sample has to be labeled with both dyes.

If there are biological replicates, it is advisable for the dye-swap design to be applied to each pair of biological replicates, giving rise to the *replicated dye-swap design*:

DESIGN D_3'	ARRAY A_1	ARRAY A_2	ARRAY A_3	ARRAY A_4
Channel R	A_1	B_1	A_2	B_2
Channel G	B_1	A_1	B_2	A_2

Here A_i refers to the ith biological replicate given variety A and B_i refers to the ith biological replicate given variety B.

Design D_4 includes REF, a reference variety. It is useful to have such a variety to which hybridization results can be referred. However, if the primary objective of the experiment is to compare varieties A and B, it is not advisable to use this design. It is more efficient to make a key comparison directly on one array rather than indirectly via an intermediate comparison.

8.5.2 Comparing Multiple Varieties with Two-Channel Microarrays

Now suppose that an experimenter is planning a microarray experiment to compare the effects of several varieties—for illustration, say three varieties, A, B, and C—and intends to use two-channel microarrays. Suppose that a reference variety REF, of no intrinsic interest, is also available. Two possible designs are the reference sample design and the loop design. The *reference sample design* is as follows:

DESIGN D_1	ARRAY A_1	ARRAY A_2	ARRAY A_3
Channel R	REF	REF	REF
Channel G	A	B	C

In this design, dye effects are confounded with test variety versus reference variety effects, but since these are not of intrinsic interest, this is not a problem.

The *loop design* was proposed by Kerr and Churchill (2001) as a natural extension of the dye-swap design:

DESIGN D_2	ARRAY A_1	ARRAY A_2	ARRAY A_3
Channel R	A	B	C
Channel G	B	C	A

A dye swap could be included in the loop design to yield a *saturated design*:

DESIGN D_3	ARRAY A_1	ARRAY A_2	ARRAY A_3	ARRAY A_4	ARRAY A_5	ARRAY A_6
Channel R	A	B	C	B	C	A
Channel G	B	C	A	A	B	C

The reference variety could be included in the loop design if necessary:

DESIGN D_4	ARRAY A_1	ARRAY A_2	ARRAY A_3	ARRAY A_4
Channel R	REF	A	B	C
Channel G	A	B	C	REF

The loop design has two clear advantages over the reference sample design. One is that the dye effect is estimable in the loop design. The second is that there is essentially double the amount of information in the loop design for the varieties of interest compared to the reference sample design. Loop designs are useful for temporal studies. In this case A, B, and C would be three successive time points.

However, despite their nice properties, there are some drawbacks to loop designs as well. One is that like dye-swap designs, they require some extra effort on the part of the experimenter because each sample has to be labeled with both dyes. However, doing so is also likely to introduce additional variability. Another risk, particularly with large experiments is this: microarray technology

is still fallible, and it is not uncommon to have a problem with an array. If this happens with a loop design and the defective array cannot be salvaged, there may be some difficulty in drawing proper conclusions from the study unless there was adequate replication.

Now consider the situation in which one of the varieties, say A, is a control, and the goal of the experiment is to compare the other two varieties, B and C, to A. As a general rule, if a pairwise comparison is considered important, it is always advisable to have an array that represents that comparison in the design. Thus a sensible design for this situation is the *comparison to control design*:

DESIGN D_5	ARRAY A_1	ARRAY A_2
Channel R	A (control)	A (control)
Channel G	B	C

The general guidelines outlined above are applicable to complex settings as well. An example is Churchill and Oliver (2001), who apply them to propose an alternative design for a complex microarray experiment described by Jin et al. (2001) and involving three factors: strain, sex, and age. To establish a library of gene expression data or to compare many samples to one another, one could use two-channel arrays with a common reference sample or, provided the experiment is well under control, single-channel arrays.

8.5.3 Single-Channel Microarray Experiments

The experimental designs used in single-channel microarray experiments should also require careful consideration. For example, consider an experiment in which four treatments, A, B, C, D, are being compared. Each treatment is to be given to four animals. If the sample from each animal corresponded to one array, there would be 16 arrays in all, which we can refer to as A_1, A_2, A_3, A_4, B_1, B_2, B_3, B_4, C_1, C_2, C_3, C_4, D_1, D_2, D_3, D_4. Suppose that the facility is a small one, so that at most four arrays can be performed in any one day. This means that the experiment has to be run over a period of four days. Assume that the experiment was run as follows:

Day 1	A_1, A_2, A_3, A_4
Day 2	B_1, B_2, B_3, B_4
Day 3	C_1, C_2, C_3, C_4
Day 4	D_1, D_2, D_3, D_4

In this case, if there was a day effect (and such an effect has been observed in practice), then the treatment effect is totally confounded with day effect. Instead it is much better to run:

Day 1	A_1, B_1, C_1, D_1
Day 2	A_2, B_2, C_2, D_2
Day 3	A_3, B_3, C_3, D_3
Day 4	A_4, B_4, C_4, D_4

In this design, day effects can be estimated and these estimates can be used to adjust the treatment-gene effects.

8.6 MISCELLANEOUS ISSUES

In general, it is advisable to adhere, as much as possible, to the fundamental precepts of the theory of *design of experiments* (*DOE*, for short): randomization, replication and balance.

- *Randomization.* Arrays should be assigned to varieties at random.
- *Replication.* Replication was described in Section 6.1. While the examples in this section have been shown with the minimum number of arrays possible for illustrative purposes, it is always advisable to replicate as much as possible.
- *Balance.* Ultimately the power of experimental design lies in been able to study many factors with few arrays, but doing so in such a way as to maximize the information content. One of the keys to this is proper statistical balance. An effect is *balanced* with respect to another if the first effect occurs equally often with the second effect. Balance confers orthogonality on the two effects and prevents an effect of interest being influenced by another effect. For example, by balancing varieties with dyes (i.e., by ensuring that each variety is labeled with each dye an equal number of times), the variety–gene interactions of interest are not biased by dye–gene interactions. This is particularly crucial when genes are not replicated on arrays, as in this case the dye–gene effect interaction would not be estimable and it would not be possible to adjust the variety–gene interaction for it.

The more carefully planned an experiment is, the better the use that can be made of available resources.

SUPPLEMENTARY READING

There is an extensive literature on statistical linear models and statistical experimental designs, dating back to Sir R. A. Fisher, a renowned geneticist and statistician who, motivated largely by problems that arose from agricultural experiments, pioneered work in these areas. Fisher (1951) remains, to this day, the best exposition of the philosophy behind practical experimental design. Cochran and Cox (1992) is another classical textbook on this topic. McCulloch and Searle (2001) is an up-to-date treatment of the theory of linear models.

EXERCISES

8.1. Consider only the 0, 1, and 2 hour data Experiment E8.

- **a.** Carry out gene-specific F tests to determine which genes are significantly differentially expressed across the three groups at the 5% level **(i)** without any adjustment for multiplicity **(ii)** with Holm's adjustment for multiplicity.
- **b.** Calculate the FDR for the results in part a.
- **c.** For the analysis in part a, draw a scatterplot of $\log(MS(V))$ versus $\log(MS(E))$. Comment.
- **d.** Fit the linear models suggested in Section 8.2 to the data. Compare the results here with those in part a.
- **e.** Carry out gene-specific t tests to determine which genes are not significantly differentially expressed at 1 hour, compared to the control, but are significantly differentially expressed at 2 hours, compared to the control. Compare these genes with those picked out in part a.

8.2. Lee et al. (2002) mention a two-channel microarray experiment that was run to compare two types of kidney tissue, wild type (W) and mutant (M). The experiment had the following design:

DESIGN	ARRAY A_1	ARRAY A_2	ARRAY A_3	ARRAY A_4
Channel R	W	W	M	M
Channel G	W	M	W	M

Compare this design with the following alternative design:

DESIGN	ARRAY A_1	ARRAY A_2	ARRAY A_3	ARRAY A_4
Channel R	W	W	M	M
Channel G	M	M	W	W

8.3. Compare the loop design

DESIGN	ARRAY A_1	ARRAY A_2	ARRAY A_3	ARRAY A_4
Channel *R*	*A*	*B*	*C*	*D*
Channel *G*	*B*	*C*	*D*	*A*

to the *modified loop design* also suggested by Kerr and Churchill (2001b):

DESIGN	ARRAY A_1	ARRAY A_2	ARRAY A_3	ARRAY A_4
Channel *R*	*A*	*A*	*B*	*B*
Channel *G*	*C*	*D*	*C*	*D*

Exploration and Analysis of DNA Microarray and Protein Array Data
Edited by DHAMMIKA AMARATUNGA and JAVIER CABRERA

CHAPTER 9

Pattern Discovery

Thus far we have been discussing statistical techniques for identifying those genes that are differentially expressed across a series of conditions. These analyses were essentially all conducted on a gene-by-gene basis. While there is little doubt that these analyses yield useful results, they do suffer from one basic shortcoming: they neither expose nor exploit the correlated patterns of gene expression displayed by genes behaving jointly, such as genes performing similar functions or genes operating along a genetic pathway. As a result they fail to make use of what should ideally be the full potential of multi-gene experiments. This can be resolved by applying multivariate analysis techniques to elicit more complex structures from microarray data.

Multivariate methods can be used both for finding multivariate patterns in data (called *pattern discovery* or *unsupervised classification* or *cluster analysis*) and for predicting classes (called *class prediction* or *supervised classification* or *discriminant analysis*). We will discuss pattern discovery in this chapter and class prediction in the next.

9.1 INITIAL CONSIDERATIONS

When taking a multivariate approach, it is customary in the microarray literature to organize the data as a *gene expression matrix*, a $G \times p$ matrix, $X = \{x_{gi}\}$, whose G rows and p columns represent, respectively, the G genes and p samples. Depending on the experiment, the p samples may correspond to p tissue types, cell lines, times, patients, treatments, experimental conditions, or something else. The values x_{gi} that make up the gene expression matrix could be either the measured gene expression level for the gth gene in the ith sample, suitably transformed and normalized, or, particularly in two-channel experi-

Exploration and Analysis of DNA Microarray and Protein Array Data
By Dhammika Amaratunga and Javier Cabrera
ISBN 0-471-27398-8

ments, the log of the ratio of the normalized gene expression level for the gth gene in the ith sample relative to its corresponding value in a reference sample.

When it comes to analysis, there is a dichotomy of approaches. Depending on the goal of the analysis, the columns may be regarded as the variables and the rows as the observations, as in traditional multivariate statistical analysis, or the roles of the rows and the columns could be reversed. If the objective of the analysis is to identify groups of genes that have similar regulatory mechanisms, the columns (i.e., the samples) are regarded as the variables and the rows (i.e., the genes) are regarded as the observations. However, if the objective of the analysis is to classify the samples on the basis of their gene expression profiles, the rows (i.e., the genes) are regarded as the variables and the columns (i.e., the samples), are regarded as the observations. In this latter case, not only is the notation diametrically opposite to traditional multivariate statistical analysis notation, but also unlike traditional multivariate statistical analysis, the number of variables, G, greatly exceeds the number of observations, p. In contrast, almost all traditional multivariate data analysis methods were developed with the expectation that the number of cases would exceed the number of variables.

We will discuss both multivariate approaches in this chapter. However, to discuss methods of clustering genes in Section 9.2, we will use traditional multivariate statistical analysis notation, namely the columns are the variables and the rows are the observations. In Section 9.3, where the goal is to summarize the information provided by a large pool of genes into a few variables that are more manageable, the genes are treated as variables and the samples as cases, contrary to classical statistical notation.

In many applications, besides the gene expression data, there is also auxiliary information available about the individual rows and/or columns. This information can be stored as *covariates* for the rows and/or columns. For example, we may know that the samples can be categorized as treatment or control, that they come from different patients (perhaps demographic information, such as age and gender, is also available), and that they are from different tissue types. On the other hand, for some, if not all genes, we may have some information regarding their functionality; certainly their sequences will be available. In this chapter we discuss *unsupervised* methods that do not consider the covariate information directly in the analysis, although it may be used for interpreting the findings of the analysis. In the next chapter we will discuss *supervised* methods that do take covariate information into account.

The definitions of sample variance-covariance and sample correlation between two genes were given in Section 5.5. These two definitions are applied for the definitions of the sample variance covariance and correlation matrices:

$$S = \begin{pmatrix} s_1^2, s_{12}, \ldots, s_{1G} \\ s_{21}, s_2^2, \ldots, s_{2G} \\ \ldots\ldots\ldots\ldots \\ s_{G1}, s_{G2}, \ldots, s_G^2 \end{pmatrix} = \frac{X^t X - G^{-1} X^t 1 1^t X}{G - 1}$$

and

$$R = \begin{pmatrix} 1, r_{12}, \ldots, r_{1G} \\ r_{21}, 1, \ldots, r_{2G} \\ \ldots\ldots\ldots\ldots \\ r_{G1}, r_{G2}, \ldots, 1 \end{pmatrix},$$

where s_{ij} and r_{ij} is the sample covariance and correlation coefficients between the ith and jth genes respectively. These two $G \times G$ matrices are of rank $p - 1$. Given the fact that in microarray experiments G is normally in the thousands, the dimension reduction methods described in Section 9.3 are needed. If we consider the gene expression matrix as a dataset where the genes are the observations and the samples are the variables then the dimension of the correlation and covariance matrices is $p \times p$ and, since p is typically much smaller than G, there is less of a problem with the dimensions (unless, of course, when p is very large).

Example. We will use a subset of the data of Khan et al. (2001) to illustrate the methods of this chapter. The dataset contains gene expression measurements, obtained using cDNA microarrays, from four types of pediatric small round blue cell tumors (SRBCT). Here we concentrate on the subset with types: rhabdomyosarcoma (RMS) and the Ewing family of tumors (EWS). This data was filtered to remove any gene that consistently expressed below a certain minimum level of expression, leaving expression data for 2308 genes ($G = 2308$). A subset of 43 cells is considered here (23 EWS, 20 RMS so $p = 43$).

Working with 2308 genes is too many for the purpose of the illustrations that will be presented throughout this chapter. For this reason we will concentrate in only 100 genes that were selected because they produced the highest t-values for comparing the means of the two groups of samples for each individual gene. The complete set will be described more thoroughly in Chapter 10.

9.2 CLUSTER ANALYSIS

Broadly speaking, *cluster analysis*, a significant branch of *unsupervised learning*, refers to a hodgepodge collection of algorithms and procedures. Cluster analysis has been developed over the years, and across a variety of disciplines, for organizing a given multivariate dataset into an assortment of *clusters* in such a way that the observations within each cluster are more or less similar to each other.

As might be expected, cluster analysis was one of the first multivariate data analysis techniques employed for an analysis of microarray data (the seminal

paper in this regard is Eisen et al., 1998). There is a compelling argument for using cluster analysis for analyzing gene expression data. It is reasonable, after all, to expect that a set of genes operating in a particular genetic pathway would behave fairly similarly across a series of conditions. For this reason their expression levels are likely to be relatively highly correlated and, in a cluster analysis, should all fall into a single cluster. A cluster analysis will sort the entirety of genes (or a suitably selected subset of them) into a series of clusters in such a way that those genes that behaved the most similarly in the experiment will be members of the same cluster, while genes that behaved differently will be members of separate clusters. The hope, of course, is that genes performing similar functions or participating in the same genetic pathway would all congregate in the same cluster.

The gene clusters generated by the cluster analysis can then be assessed in the context of known or putative genetic pathways, such as metabolic pathways, gene families, and subcellular components, in order to deduce functional relationships. For example, if a gene is known to code for a particular enzyme, it can be mapped onto the reaction that is catalyzed by that enzyme. By exploring constructs of all qualitatively feasible metabolic pathways from a set of biochemical reactions, inferences can be made regarding the pathway. As another example, in experiments involving normal and diseased subjects, the findings from a cluster analysis could lead to the discovery of a genetic pathway (or the disruption of one) that causes a disease. Of course, cluster analysis cannot reveal functionally related genes if they do not display similar expression patterns or if they express with a time delay. Still, with technology having evolved to such a state that it is possible to array almost an entire genome onto a microarray, cluster analysis has emerged as one of the most valuable tools for gathering information about how genes work in combination.

Both the statistics and data mining literature are replete with clustering methods that are mostly algorithmic in nature. Most clustering algorithms can be classified as being either hierarchical or partitioning. We will discuss these in the following sections. However, all clustering methods depend on either a dissimilarity or similarity measure, which quantifies how far, or how close, two observations (in this case, genes) are from each other. We will discuss such measures first.

For the clustering approach, we treat the gene expression levels from a gene (i.e., the gene's *expression profile* over the samples) as multivariate observations. This does not mean that we cannot use clustering methodology when we believe that genes are treated as variables, but conceptually it is more rigorous to think of genes as multivariate observations in the remainder of this section.

9.2.1 Dissimilarity Measures and Similarity Measures

Given data for two genes, g and h, with corresponding data $x_g = (x_{gj})$ and $x_h = (x_{hj})$ (i.e., the gth and hth rows of X), a *dissimilarity measure* (sometimes referred to as a *distance*), $D(x_g, x_h)$, is a statistic that states quantitatively how

dissimilar x_g and x_h are to each other. There are many choices for D and many of the better choices satisfy the following *dissimilarity axioms*: (1) $D \geq 0$, (2) $D = 0$ if and only if $x_g = x_h$, (3) D gets larger the further x_g and x_h are apart, and (4) $D(x_g, x_h) = D(x_h, x_g)$. Some choices for D also satisfy either (5) the *triangle inequality*, $D(x_g, x_h) \leq D(x_g, x_i) + D(x_i, x_h)$ or (6) the *ultrametric inequality*, $D(x_g, x_h) \leq \max(D(x_g, x_i), D(x_h, x_i))$.

The most widely used dissimilarity measure is the *Euclidean distance*, D_E. $D_E(x_g, x_h)$ is the geometrical distance between x_g and x_h in the p-dimensional space in which they lie:

$$D_E(x_g, x_h) = \sqrt{\sum_{j=1}^{p} (x_{gj} - x_{hj})^2}.$$

D_E satisfies all the dissimilarity axioms above but has the drawback that changing the column variances could substantially change the ordering of the distances between the genes and, as a result, change the clustering. Of course, one could hope that the normalization step would have relegated this to a non-issue by bringing the column variances into close alignment with one another. Otherwise, one way to reduce this effect is to divide each column by its standard deviation or median absolute deviation. This gives the *standardized Euclidean distance*:

$$D_{SE}(x_g, x_h) = \sqrt{\sum_{j=1}^{p} \left(\frac{x_{gj} - x_{hj}}{s_j} \right)^2}.$$

However, some care is necessary when rescaling the data this way as it could also dilute the differences between the clusters with respect to the columns that are intrinsically the best discriminators. Skewness could also exacerbate the effect of scaling on the data.

Two other dissimilarity measures that have been used for clustering are the *Manhattan* or *city block distance*,

$$D_M = \sum_{j=1}^{p} |x_{gj} - x_{hj}|,$$

and the *Canberra distance*,

$$D_{CAN} = \sum_{j=1}^{p} \frac{|x_{gj} - x_{hj}|}{x_{gj} + x_{hj}}.$$

Clustering can be also be based on similarities between pairs of observations rather than dissimilarities between pairs of observations. A measure of similarity, $C(x_g, x_h)$, between two objects, x_g, x_h, must comply with the conditions: (1)

$C(x_g, x_h) = C(x_h, x_g)$, (2) $C(x_g, x_h) \leq C(x_g, x_g)$ for all g, h, and (3) C gets smaller the further x_g and x_h are apart. A similarity measure can be converted to a dissimilarity measure by the standard transformation (see Mardia, Kent, and Bibby, 1979):

$$D_C(x_g, x_h) = \sqrt{C(x_g, x_g) + C(x_h, x_h) - 2C(x_h, x_g)}.$$

One popular example of a similarity measure is Pearson's correlation coefficient, R:

$$R(x_g, x_h) = \frac{\sum_{j=1}^{p}(x_{gj} - \bar{x}_{g\cdot})(x_{hj} - \bar{x}_{h\cdot})}{\sqrt{\sum_{j=1}^{p}(x_{gj} - \bar{x}_{g\cdot})^2 \sum_{j=1}^{p}(x_{hj} - \bar{x}_{h\cdot})^2}}.$$

R measures how linearly correlated $\{x_g\}$ and $\{x_h\}$ are to each other. It lies between -1 and $+1$ and, the closer it is to these values, the more linearly correlated $\{x_g\}$ and $\{x_h\}$ are to each other, with negative values indicating negative association. Values near zero connote the absence of a linear correlation between $\{x_g\}$ and $\{x_h\}$.

R can be converted to a dissimilarity measure using either the standard transformation

$$D_{C2}(x_g, x_h) = \sqrt{1 - R(x_g, x_h)^2},$$

or the transformation

$$D_{C1}(x_g, x_h) = 1 - |R(x_g, x_h)|.$$

Note that neither D_{C1} nor D_{C2} quite satisfies the dissimilarity axioms. For instance, instead of axioms (2) and (3), $D_{C1} = 0$ if and only if x_g and x_h are linearly correlated (rather than if and only if $x_g = x_h$), and D_{C1} increases toward its maximum value of one the less linearly correlated x_g and x_h are. Nevertheless, it is a useful measure to use with microarray data, as co-expressing genes could have expression levels that are highly correlated to each other despite how far apart their expression levels are.

When the observations have a natural reference value, c, the observations may be centered at c rather than at the mean:

$$R_c(x_g, x_h) = \frac{\sum_{j=1}^{p}(x_{gj} - c)(x_{hj} - c)}{\sqrt{\sum_{j=1}^{p}(x_{gj} - c)^2 \sum_{j=1}^{p}(x_{hj} - c)^2}}.$$

For example, when clustering gene expression ratios's rank and the observations are log expression ratios, $c = \log_2(1) = 0$ is a natural reference value.

Spearman's rank correlation coefficient (see Section 5.6), which is the Pearson correlation coefficient calculated on the ranks of the data, measures closeness in terms of whether two observations are monotonically related to each other.

9.2.2 Guilt by Association

When a set of genes is known to be associated with a disease (or other factor), discovering that there is a novel gene whose expression profile closely matches that of one of the known genes could prove to be a very valuable piece of information about the genetic pathway involved in the disease process. Besides assisting in better understanding of pathways, medical applications are also possible; for example, if the novel gene is one that is expressed earlier in the progression of the disease than any of the known genes, it could perhaps be used as a disease marker allowing for earlier diagnosis and treatment of the disease. Walker et al. (1999) call this concept *guilt by association*.

Any of the dissimilarity or similarity measures mentioned in Section 9.2.1 could be used for searching through a database of gene expression profiles that includes data for both known genes and novel genes. Thus, for example, if gene d is a known gene, any novel gene g that is such that $R(x_d, x_g)$ is relatively very high, would be considered "guilty by association" and subjected to closer scrutiny to assess its involvement in the disease process.

On the other hand, since many genes tend to express only under specific circumstances, it is possible that these measures would be dampened by the many genes that are not expressed. Also, when the database has been derived from diverse sources, there may be some doubt as to whether the data are directly comparable. For these reasons it may be preferable to dichotomize the data (i.e., transform them to a binary variable that is set equal to 1 if the gene is expressed and 0 otherwise) and use the log odds ratio or the Fisher exact test (Agresti, 2002) as a measure or test of association.

The premise underlying guilt by association is that functionally related genes would display very similar expression patterns. This has been demonstrated to be true to some extent as, for instance, when they are co-regulated by common transcription factors. However, in other instances they may not necessarily display similar expression patterns, and conversely, genes having quite different functions may exhibit similar expression patterns simply due to chance. Thus some care is necessary, particularly in view of the large number of correlations being estimated, that a novel gene with an expression profile that, just by chance, happens to look correlated to that of a known gene, is not inadvertently found "guilty," and vice versa.

9.2.3 Hierarchical Clustering

Hierarchical clustering (Sokal and Michener, 1958, is an often cited early reference, but not the earliest) is one the most widely used clustering methods. It is not surprising that some of the key developments in this area, such as Eisen et al. (1998) and Alizadeh et al. (2000) utilized hierarchical clustering methodology. Hierarchical clustering methods can themselves be classified as being either bottom up or top down.

Bottom-up clustering (also known as *agglomerative hierarchical clustering*) algorithms are initiated with each gene situated in its own cluster. At the next

and subsequent steps, the closest pair of clusters is agglomerated (i.e., combined). In principal, the process can be continued until all the data falls into one giant cluster.

Whenever two clusters are agglomerated, the distances between the new cluster and all the other clusters are recalculated. Different hierarchical clustering schemes calculate the distance between two clusters differently:

- In *complete linkage hierarchical clustering* (or *farthest-neighbor clustering*), the distance between two clusters is taken to be the largest dissimilarity measure between any two members in different clusters.
- In *single linkage hierarchical clustering* (or *nearest-neighbor clustering*), the distance between two clusters is taken to be the smallest dissimilarity measure between any two members in different clusters.
- In *average linkage hierarchical clustering*, the distance between two clusters is taken to be the arithmetic mean of the dissimilarity measures between all pairs of members in different clusters.
- In *centroid clustering*, the distance between two clusters is taken to be the dissimilarity measure between the cluster centers.
- In *Ward's clustering*, the distance between two clusters is taken to be the sum of squares between clusters divided by the total sum of squares, or equivalently, the change in R^2 when a cluster is split into the two clusters, where the *coefficient of determination*, R^2, is the percent of the variation that can be explained by the clustering.

Despite their apparent similarity these methods have different properties and will generally cluster the data in quite different ways and may even impose a structure of their own. The complete linkage hierarchical clustering algorithm is set up to minimize the maximum within-cluster distance, and hence it tends to find compact clusters but may overemphasize small differences between clusters. The single linkage hierarchical clustering algorithm is set up to maximize the connectedness of a cluster, and hence it exhibits a highly undesirable tendency to find chainlike clusters; by creating chains, two dissimilar observations may be placed in the same cluster merely because they are linked via a few intermediate observations. The average linkage hierarchical clustering algorithm and the centroid clustering algorithm are compromises between the above two; note, however, that unlike the other methods, they are not invariant to monotone transformations of the distances. Nevertheless, the number of small tight clusters they usually produce can be useful for the discovery process.

Eisen et al. (1998) applied an average linkage hierarchical clustering procedure with dissimilarity measure D_c and $c = 0$ to a dataset consisting of gene expression ratios generated from an experiment in the budding yeast *Saccharomyces cerevisiae*. The data was a combination of time course data from separate experiments involving the diauxic shift (DeRisi et al., 1997), the mitotic cell division cycle (Spellman et al., 1998), sporulation (Chu et al., 1998), and temperature and reducing shocks. The goal of the exercise was to understand the genetic processes taking place during the life cycle of the yeast. The cluster

analysis successfully identified patterns of genomic expression correlated with the status of cellular processes within the yeast during diauxic shift, mitosis, sporulation, and heat shock disruption. In another experiment Alizadeh et al. (1999) applied hierarchical clustering to separate diffuse B-cell lymphomas, an often fatal type of non-Hodgkins lymphoma, into two subtypes, which corresponded to distinct stages in the differentiation of B-cells and showed substantial survival differences.

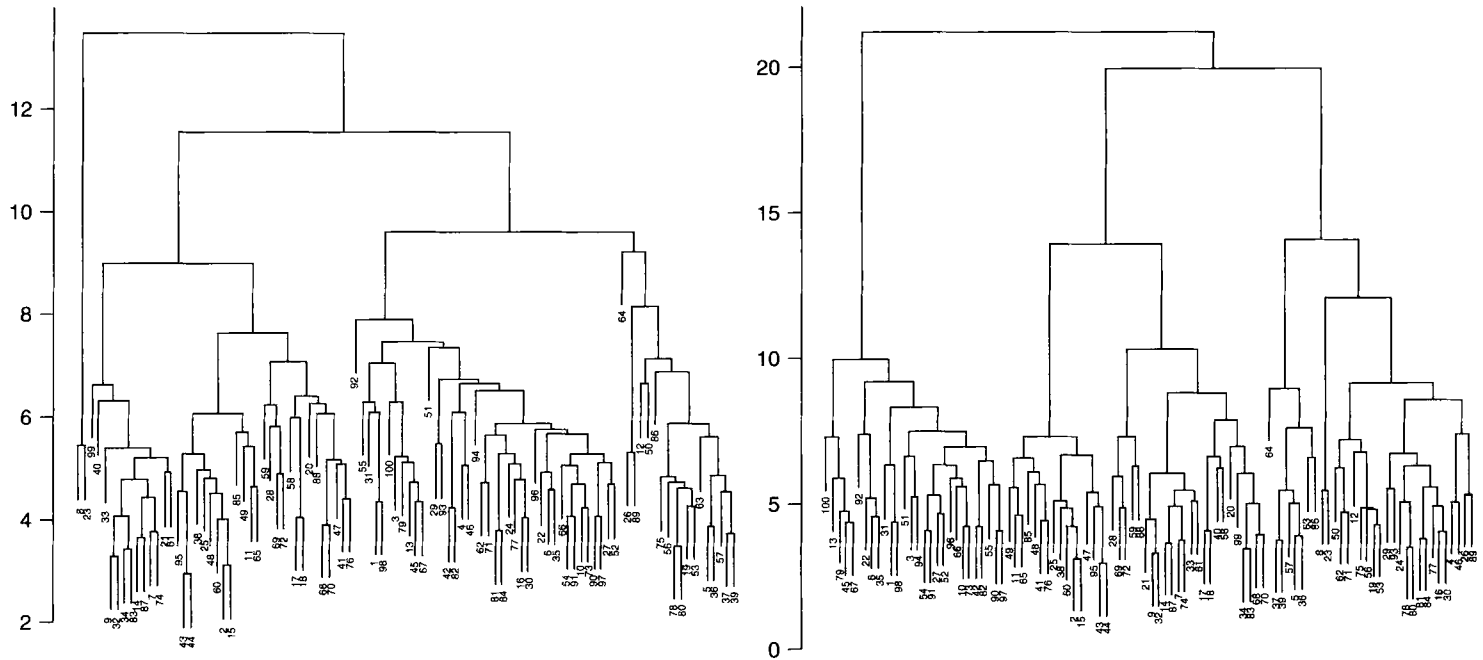
Top-down clustering (also known as *divisive hierarchical clustering*) algorithms are initiated with all the genes placed together in one cluster. At the next and subsequent steps, the loosest cluster is split into two. In principal, the process can be continued until each gene is alone in its own cluster. A serious computational issue that sometimes hinders the use of top-down clustering methods is that at the early stages there are a huge number of ways (e.g., $2^{G-1} - 1$, in the first stage) of splitting even the initial cluster. Divisive algorithms are rarely used in practice.

Typically the hierarchical clustering process is terminated either once a specified number of clusters has been reached or a criterion has been optimized or has converged. Several criteria for choosing an appropriate number of clusters have been proposed, none entirely satisfactory. Some criteria are as follows:

- Ward's (1963) statistic, which is R^2 of the entire configuration. An adequate clustering is gauged by graphing the change in R^2 against the number of clusters.
- The *gap statistic* (Tibshirani et al., 2000), which is the change in within cluster dispersion compared to its expected value.
- A normalized ratio of between- and within-cluster distances (Calinski and Harabasz, 1974).
- Difference of weighted within cluster sum of squares (Krzanowski and Lai, 1985).
- A prediction-based resampling method for classifying microarray data (Dudoit and Fridlyand, 2002).
- A stability-based resampling method (Ben-Hur et al., 2002), where a stable clustering pattern is characterized as a high degree of similarity between a reference clustering and clusterings obtained from subsamples of the data.

The hierarchy of fusions in which the clusters are formed either by a bottom-up clustering algorithm or by the hierarchy of divisions in which the clusters are divided by a top-down clustering algorithm can be displayed diagrammatically as a hierarchical tree called a *dendrogram*. Each node of the dendrogram represents a cluster and its "children" are the subclusters. One reason for the popularity of hierarchical clustering is the ease with which dendrograms can be interpreted.

Example. Figure 9.1 shows dendrograms for the hierarchical decompositions obtained by applying (1) average linkage hierarchical clustering and (2) com-



(*a*) Average Linkage (*b*) Complete Linkage

Figure 9.1 Tree dendograms using the (*a*) average linkage, (*b*) complete linkage criteria for a dataset with two tumors, (*c*) single linkage, and (*d*) centroid linkage.

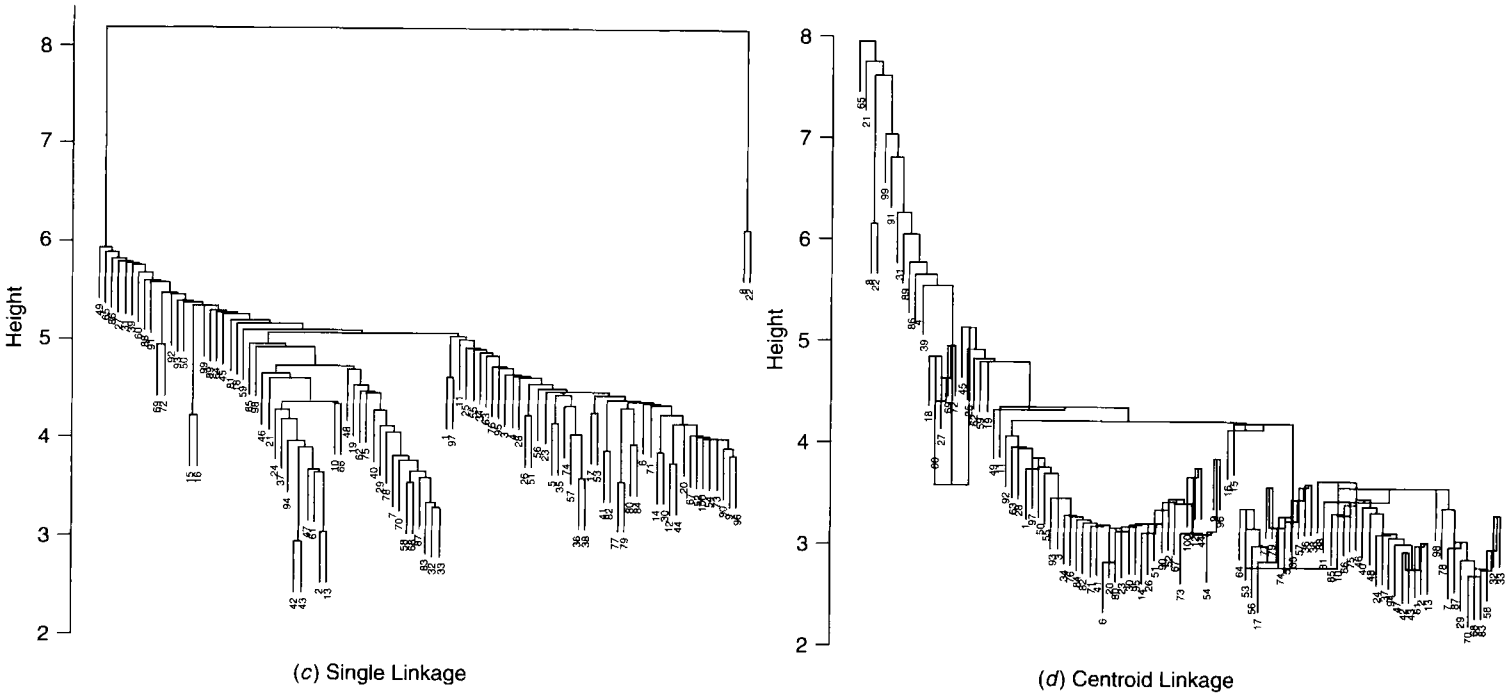


Figure 9.1 *(Continued)*

plete linkage hierarchical clustering with dissimilarity measure D_C to the tumor example data described above. It is easy to observe that these two methods produce slightly skewed trees, whereas Ward's method, shown in Figure 9.2, produces a more balanced and clear tree.

In general, it is known that all hierarchical clustering methods may produce unbalanced trees, and in many cases some of the clusters can consist of single observations. Another undesirable pattern that one observes when using these methods is a big cluster with most of the data and a few small clusters around it. Nevertheless, hierarchical clustering remains a popular clustering tool.

Example. On the question of selecting the number of clusters, Figure 9.2 shows the dendrogram generated by the Ward method and next to it is a graph of Ward's statistic versus the corresponding number of clusters. From the dip in this second graph at 12, it was decided to select 12 as the number of clusters. The average profiles of these 12 clusters are displayed in Figure 9.3. The profiles clearly show that it is very easy to differentiate between the two groups of tumors (samples).

We can also apply the clustering methods to the samples. In this case the genes will act as the variables.

Example. For simplicity we continue with the top 100 genes selected using the t statistic. Figure 9.4 displays the dendrogram produced by Ward's hierarchical clustering procedure that clearly separates the samples into two groups. The two groups correspond to the two tumor groups. To complete the analysis, we may draw a microarray image graph combining the elements that we have seen here in Figures 9.2, 9.3, and 9.4. The graph is shown in Figure 9.5. The main panel represents the image graph of the intensities for the 100×43 array. A horizontal and a vertical bar on the top and left side of the main image indicate the clustering of genes and samples respectively. The right panel shows the 12 cluster profiles on a normalized scale from zero to one. Finally, the lower panel shows the color scale for the main image.

Friedman and Meulman (2002) present a distance-based clustering approach called COSA (which stands for "clustering objects on subsets of attributes") that attempts to identify groups of samples that exhibit preferentially close values in different, possibly overlapping subsets of genes.

9.2.4 Partitioning Methods

Partitioning methods split the data up into a specified number of non-overlapping clusters. The general idea behind most partitioning methods is to cluster the genes so that the sum of squared dissimilarities of each gene from the closest of a set of representative central genes is minimized. Clearly, this

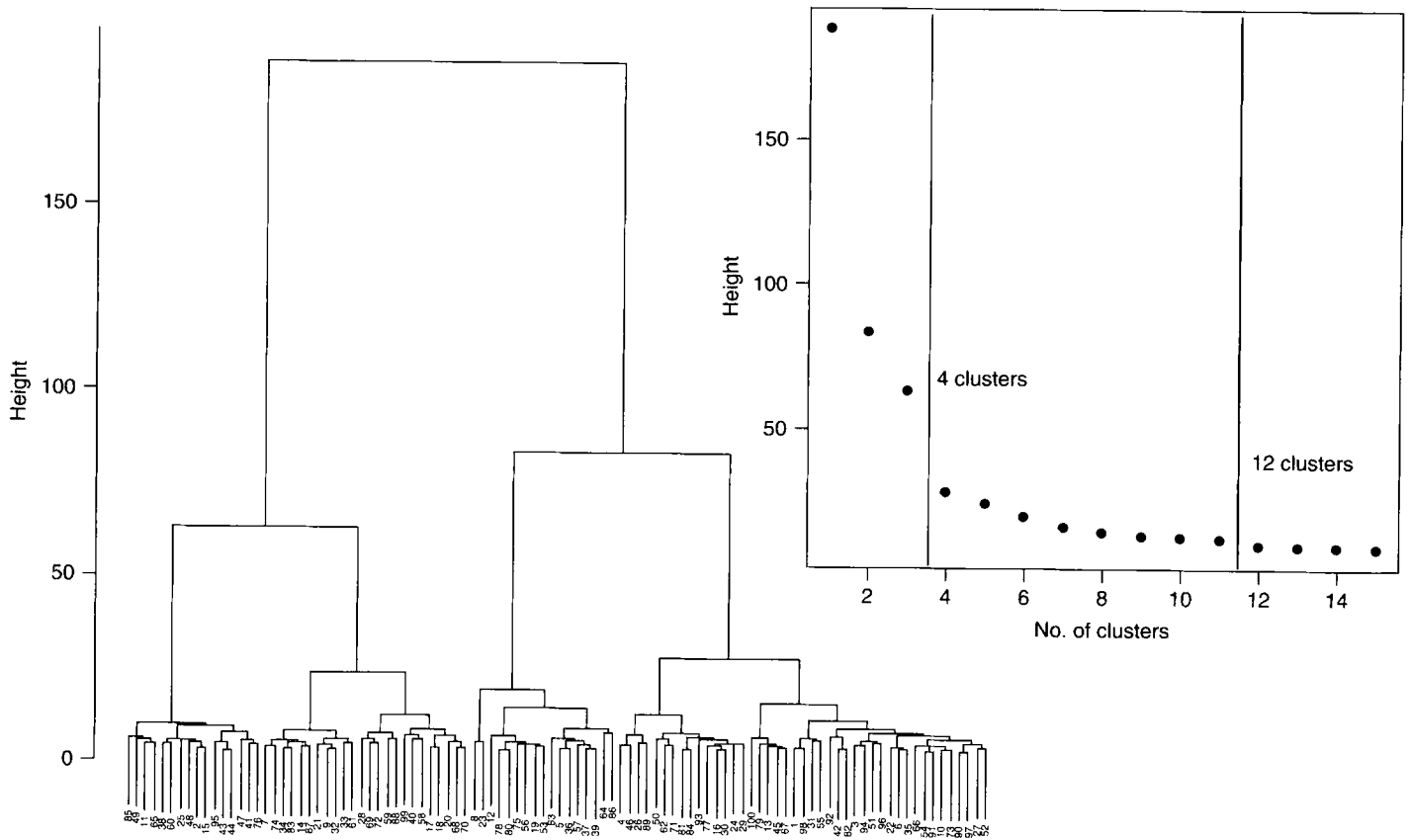


Figure 9.2 Clusters of samples for a dataset with two tumors.

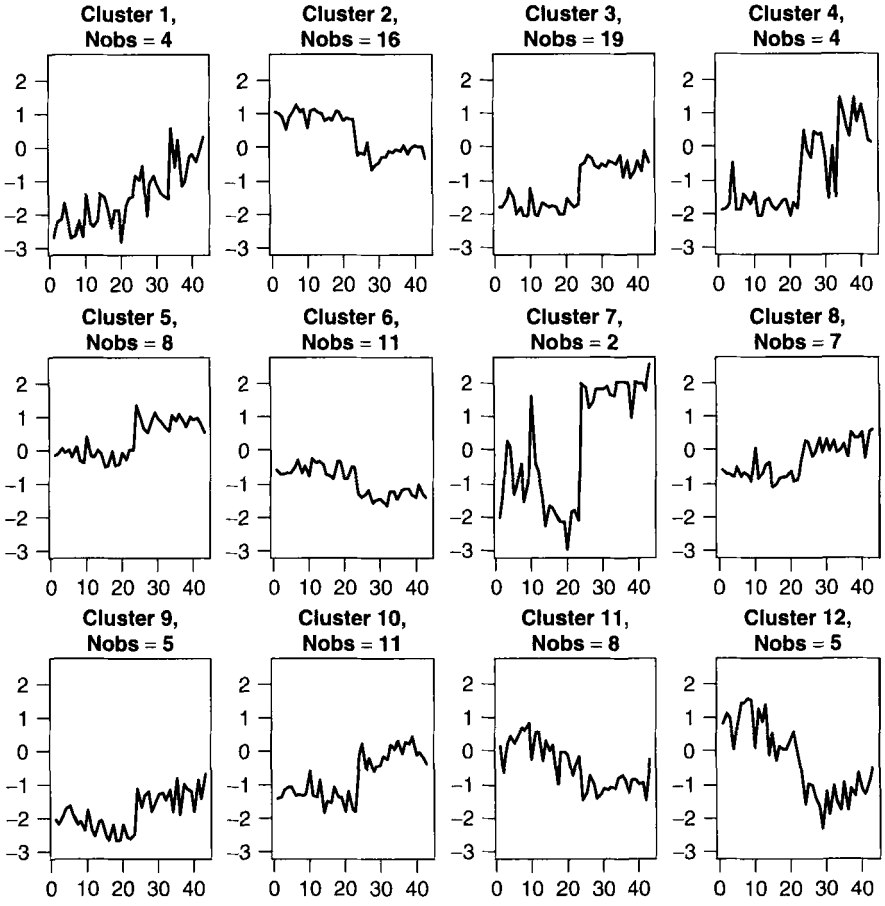


Figure 9.3 Average profiles of the 12 clusters obtained using Ward's method. Nobs is the number of observations within the cluster.

problem cannot be solved in real time and several algorithms, three of which are k-means clustering, k-medoids clustering, and self-organizing maps, have been developed to produce approximations.

The k-means clustering algorithm (an early version was described by MacQueen, 1967) is a procedure that clusters the G genes around k cluster centers. It is an iterative procedure that is begun with a set of k initial cluster centers. Each gene is then placed in the cluster whose center is closest in distance to the gene. The genes in each cluster are then averaged to produce a new cluster center. The procedure is repeated with the repositioned cluster centers. This process is continued until no gene is reallocated to a new cluster or a criterion function has been optimized.

At each stage, cluster statistics can be computed to assess the strength of the clusters. One such statistic is $\bar{D}$, the average intracluster distance across

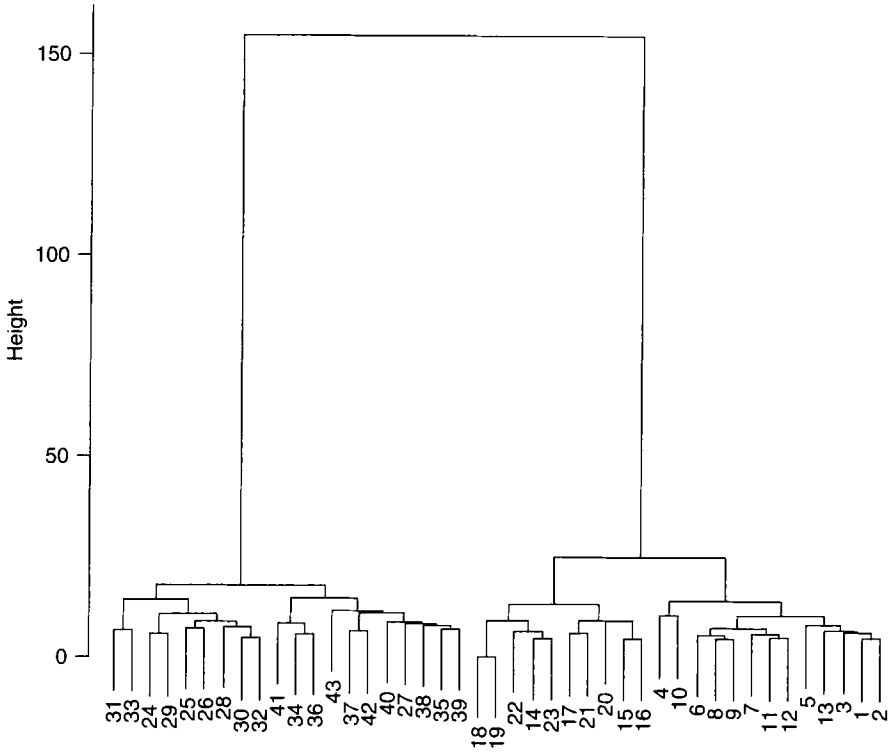


Figure 9.4 Hierarchical tree of a cluster analysis of 43 samples using the 100 genes dataset. The clustering method applied was Ward's.

clusters:

$$\bar{D} = \frac{1}{G} \sum_{r=1}^{k} \sum_{s=1}^{n_r} D(x_{rs}, \bar{x}_r),$$

where n_r is the number of members of the rth cluster, x_{rs} is the sth member of the rth cluster and $\bar{x}_r$ is mean of the rth cluster. $\bar{D}$ indicates the tightness of the clusters.

Another cluster statistic is S, the *total within cluster sum of squares*:

$$S = \sum_{r=1}^{k} \sum_{s=1}^{n_r} \sum_{j=1}^{p} (x_{rsj} - \bar{x}_{rj})^2,$$

where n_r is the number of members of the rth cluster, x_{rsj} is the jth coordinate of the sth member in the rth cluster, and $\bar{x}_{rj}$ is the jth coordinate of the mean of

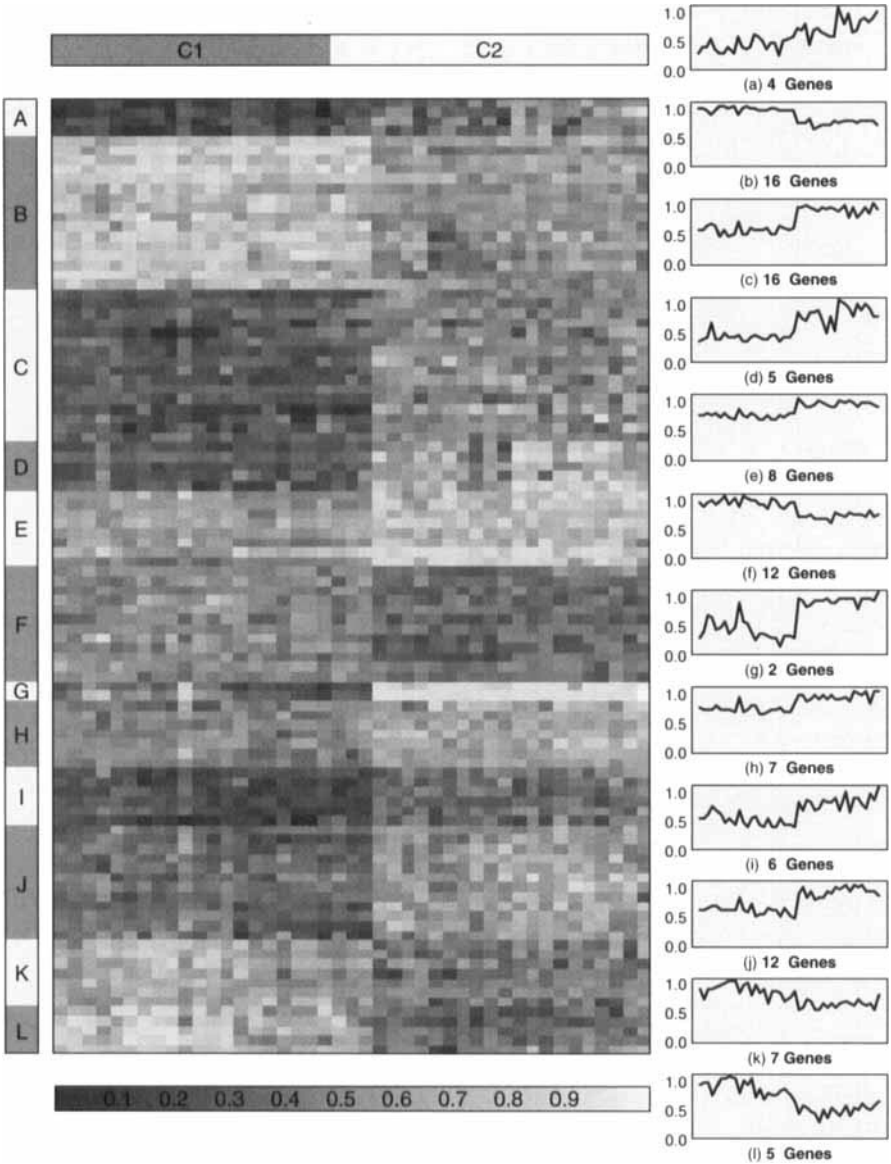


Figure 9.5 Microarray graph summarizing the gene clusters and the sample clusters. The main panel is the image plot of the intensities for the 100×43 array. The horizontal and vertical bars on the top and left side of the main image indicate the clustering of genes and samples respectively. The right panel shows the 12 cluster profiles on a normalized scale from zero to one. The lower panel shows the color scale for the main image.

the rth cluster. A stopping rule that is sometimes used is to stop the iteration once the total within-cluster sum of squares stops reducing by appreciable amounts; in other words, until the process converges to a local minimum of the total within-cluster sum of squares.

Typically the entire procedure is repeated with a set of different randomly generated initial cluster centers and the best solution, the one that has the smallest total within-cluster sum of squares overall, is chosen as the final partition for that value of k. Since it also impossible to know in advance how many clusters there are in the data, the procedure is also generally repeated with several different values of k. For instance, Brazma and Vilo (2000) applied k-means clustering to a 6221×80 gene expression matrix, in which k was varied from 2 to 1000 and, for each k, the process was run 10 times with different random initial cluster centers. Tavazoie et al. (2000) also applied k-means clustering to gene expression data.

The k-medoids clustering algorithm (Kaufman and Rousseeuw, 1990) is identical to the k-means clustering algorithm, except that the cluster centers are taken to be the p-dimensional medians (which are sometimes called *medoids*) rather than the means. Dudoit and Fridyland (2002) use k-medoids for clustering microarray data because medoids, like medians, are less affected by outliers than means.

The *self-organizing map* (SOM) (Kohonen, 1995) is a neural network procedure that is also similar to k-means clustering. It imposes a constraint, however, that forces the cluster centers to lie in a discrete two-dimensional space. Thus it produces a mapping of the data from a multidimensional space to a two-dimensional space in which the clusters are sorted according to their degree of similarity. As a result neighboring clusters are interpreted as being similar, while clusters that appear more distant in the two-dimensional space are more diverse. Tamayo et al. (1999) and Toronen et al. (1999) used SOM for clustering microarray data.

Partitioning methods are inherently nonhierarchical. In partitioning methods, unlike in hierarchical methods, the clusters obtained when the data are partitioned into k clusters cannot be constructed as a merger of the clusters obtained when the data is partitioned into $k + 1$ clusters. Generally, partitioning methods will produce spherical clusters. Tibshirani et al. (2000) report finding that k-means clustering produces tighter clusters than hierarchical clustering.

Example. Consider the dataset consisting of the top 100 genes, Table 9.1 shows the number of membership discrepancies between the groups produced by the various clustering methods when choosing twelve clusters. For example, there are 17 discrepancies between Ward's and k-means groupings because 17 observations appeared in different clusters. The k-means method was started at the grouping resulting from Ward's method because it achieved the lowest value of the within clusters sum of squares, compared with the other two possible methods. The single linkage and centroid methods produce very skewed

Table 9.1 Discrepancies for four clustering procedures applied to a subset of 100 genes

	K-Means	Ward	Average	Single	Centroid
Ward	17				
Average	39	39			
Complete	27	23	30		
Single	79	78	64	73	
Centroid	80	77	64	72	8

trees, possibly caused by the correlations among the genes, and as a result there are many small clusters and high discrepancies with the other methods.

9.2.5 Model-Based Clustering

Model-based clustering is a partitioning method in which a probability framework is posited for the clusters. The model states that (1) the genes fall into k clusters, (2) a proportion, p_r (where $\sum_{r=1}^{k} p_r = 1$), of the genes belong to the rth cluster, (3) the genes that belong to the rth cluster were all generated from a distribution, $f(x;\theta_r)$, and (4) the parameter θ_r is different from cluster to cluster. This then implies that any observation x is a realization from the *mixture model*:

$$f(x) = \sum_{r=1}^{k} p_r f(x;\theta_r)$$

Usually $f(x;\theta_r)$ is taken to be a p-variate Gaussian distribution. In this case θ_r has two components: $\theta_r = (\mu_r, \sum_r)$, where μ_r is the mean and $\sum_r$ is the variance-covariance matrix for the Gaussian distribution in the rth cluster. Banfield and Raftery (1993) point out that a geometrical structure can be imposed on the clusters by specifying a format for the variance-covariance matrices, $\sum_r$. This has the advantage of reducing the otherwise large number of parameters that have to be estimated. Four structures worth considering for $\sum_r$ are (1) $\sum_r = \lambda I$ (where I is the identity matrix), which forces spherical clusters of equal volume, (2) $\sum_r = \lambda_r I$, which forces spherical clusters of possibly unequal volume, (3) $\sum_r = \lambda DAD'$, where A is a diagonal matrix and D is an orthogonal matrix, which forces elliptical clusters having equal volume, shape and orientation across the clusters, and (4) not imposing any structure on $\sum_r$. The unconstrained model 4 is, of course, the most general, but it requires a large number of observations per cluster in order to be fitted adequately. The model 1 appears to be closely related to k-means clustering.

Given a value for the desired number of clusters, k, the parameters, θ_r, of the individual clusters, and the mixing proportions, p_r, are estimated using the EM algorithm. The EM algorithm involves alternating through a series of expecta-

tion (E) and maximization (M) steps. In the E step, the probability of each observation belonging to the each cluster is estimated conditionally on the current values of θ_r. In the M step, the values for θ_r are estimated based on the current cluster membership probabilities. Once the algorithm ultimately converges, each observation becomes a member of the cluster in which it has the largest conditional probability. Banfield and Raftery (1993) provide additional details of the procedure.

One advantage of model-based clustering is that, instead of having to heuristically judge which clustering result seems best, as has to be done with most other clustering procedures, with model-based clustering one has recourse to a probabilistic framework that can be used to compare across competing clustering results. Thus, in practice, one would fit the model with different values of k and different structures of $\sum_r$, and then, for each model fitted, a criterion function that judges how well the model fits the data, without overfitting, can be used to pick the best model and, thereby, the best clustering result. The two criteria that are generally used are the *Akaike information criterion* (AIC),

$$\text{AIC}_m = 2 \log p(X|\hat{\theta}_m, M_m) - 2v_m,$$

and the *Bayesian information criterion* (BIC),

$$\text{BIC}_m = 2 \log p(X|\hat{\theta}_m, M_m) - v_m \log(G).$$

Here M_m refers to the mth model, v_m is the number of parameters in θ_m, and $\hat{\theta}_m$ is the maximum likelihood estimator of θ_m. Large values indicate better models.

Model-based clustering will find spherical or elliptical clusters, depending on how $\sum_r$ is specified, but will not find nonconvex structures. Model-based clustering has been applied to microarray data by McLachlan et al. (2002), Pan et al. (2002), and Yeung et al. (2001). Somewhat similar approaches are described by Holmes and Bruno (2000) and Barash and Friedman (2002).

9.2.6 Chinese Restaurant Clustering

Chinese restaurant clustering, proposed by Lo et al. (1995), is a Bayesian approach to model-based clustering. The idea is to construct a Dirichlet prior distribution over the space of partitions and build a likelihood using the mixture model given in Section 9.2.5. Bayes's theorem is then applied to determine the posterior distribution, and its mode over the space of partitions is calculated.

In practice, this involves the construction of a Gibbs sampler that produces a sequence of partitions. The iteration is continued until the mode partition of the posterior distribution is clearly identified. This algorithm has the drawback that it is computationally highly intensive; that is, it may take a long time to produce the desired partition. However, it is a very interesting approach

because it does not separate the selection of the number of clusters from the assignment of the genes to the clusters. In that sense it is more natural than many of the more conventional clustering algorithms. The "Chinese restaurant" label comes from the practice of some Chinese restaurants of sitting the entering customers at a vacant or new table according to a decision of the restaurant host. This phenomenon resembles the initial steps of the algorithm, where the genes are assigned to their respective initial clusters. Cabrera and Lo (2003) developed a blocking algorithm that greatly improved the computational performance of Chinese restaurant clustering. When applied to zebrafish microarray data it produced a clear separation, which was not evident from traditional methods such as k-means and Ward's method.

9.2.7 Discussion

It is highly unlikely that gene expression data can be clearly and unambiguously separated into a set of well-defined clusters. Consequently different clustering algorithms will generally produce different, even conflicting, results. Loosely, a good clustering method will produce clusters whose within-cluster similarity is high and between-cluster similarity is low. However, the kinds of clusters found will vary according to the clustering method used, and they may not be directly comparable. The best method, if one even exists, would be data dependent. It is impossible to assert therefore that any one clustering method, or, for that matter, any one of the seemingly endless variations on the basic algorithms, is uniformly better than any other method. Hence, in practice, it is best to run more than one clustering method on any given dataset.

9.3 SEEKING PATTERNS VISUALLY

Clusters and other patterns in multivariate data can also be captured by representing the data visually. To discuss this, we will treat the G genes as G variables and the samples as cases. Of course, the dimensionality of microarray data (i.e., the number of variables [genes] in microarray data) precludes displaying the data as is. Instead, the data must be *projected* onto a lower (e.g., k) dimensional space (usually $k = 2$, maybe $k = 3$) and plotted in this latter space. Projecting G-dimensional observations into k-dimensional observations essentially involves fashioning k new variables out of the G original ones. The process of projecting the data this way is called *dimension reduction.*

There are a number of ways to reduce the dimensionality of a dataset. For many of the simpler methods, like principal component analysis and factor analysis, the k new variables are k linear combinations of the G original variables; these methods are called *linear reduction techniques.* On the other hand, methods like multidimensional scaling are *nonlinear reduction techniques.* Another method, projection pursuit, was proposed by Friedman and Tukey (1974) as a way to explicitly "pursue" projections that have interesting structure.

9.3.1 Principal Components Analysis

We begin with *principal components analysis* (PCA), a method of classical multivariate analysis that is the most commonly used technique for dimension reduction. Several data analysts have used PCA for working with microarray data (e.g., on PCA for analyzing a temporal microarray dataset, see Raychaudhuri et al., 2000; Yeung and Ruzzo, 2001).

PCA is particularly useful in situations where we are dealing with many correlated, and therefore redundant, variables and we want to reduce them to a few new uncorrelated variables, constructed as *projections*, linear combinations of the original variables, without losing too much information. The first new variable, namely the first principal component, is the linear combination of expression patterns that explains the greatest amount of variability in the data. The second principal component is the linear combination of expression patterns that explains the greatest amount of variability remaining in the data after accounting for the first principal component. Each succeeding principal component is similarly obtained.

Projecting the data into the dimensions spanned by the leading principal components will reveal data structures, such as clusters, that stretch the data point cloud out. This is why PCA is often used as a way of examining the data for clusters. When there are a few well-separated clusters, it is possible that PCA will find projections that separate the clusters. However, in other cases, it may not work so well: for instance, when there are a large number of noisy variables that do not contribute much information regarding the clusters, or when the clusters themselves are located in such a way that they do not stretch the point cloud out, or a rather extreme example when the clusters are centered at the corners of a high dimensional configuration, such as a simplex.

Example. We now return to the top 100 gene data described in Section 9.1. Figure 9.6*a* shows a two-dimensional view of this 100-dimensional data, while Figure 9.6*b* shows a two-dimensional view of the 43-dimensional space. The two dimensions plotted are the first two principal components, denoted PC1 and PC2 (each is a linear combination of the 100 "variables"), obtained by PCA. In both views it can easily be perceived that the data has two groups. Note, however, that this structure would not have been evident from a one-dimensional view in the PC1 direction in the second figure.

The projecion of X in the direction l is Xl, the variance of which is $l'Sl$. Thus the first principal component is the projection l that maximizes $l'Sl$. Generally, principal components are calculated directly from the eigenvalues and eigenvectors of either the variance-covariance matrix of X or the correlation matrix of X. However, because in microarray experiments these matrices are of a very high dimension, $G \times G$, it is computationally much more efficient to use the singular value decomposition, in which the largest matrix that needs to be computed is of size $G \times p$, where p is considerably smaller than G.

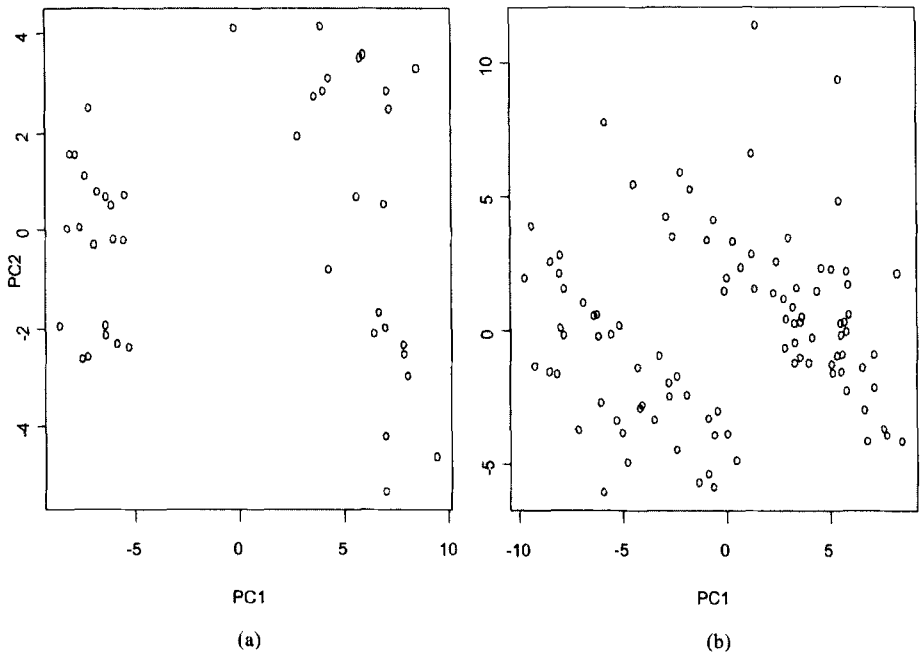


Figure 9.6 Principal components of the top 100 genes for 43 tumor cells: (*a*) Cells are the observations and genes are the variables. (*b*) Genes are the observations and cells are the variables.

Suppose that the rows of X are centered; that is, the average of x_g is zero for all g. The *singular value decomposition* (SVD) of X is defined as

$$X = UDV^T,$$

where U is a $G \times p$ orthogonal matrix ($U^tU = I_r$), which projects the G-dimensional samples into p-dimensional samples, V is a $p \times p$ orthogonal matrix ($V^tV = I_r$), which projects the p-dimensional genes into other p-dimensional genes, and D is a $p \times p$ diagonal matrix, whose diagonal elements, s_h, are called *singular values*. We will assume that $s_1 \leq s_2 \leq \cdots \leq s_p$. Alter et al. (2000) describe an analysis of microarray data using SVD.

From the SVD, it follows that the sample variance-covariance matrix of X with the genes as variables is

$$S = XX^t = UD^2U^t.$$

Hence the column vectors of U are the principal components of S and the square of the diagonal elements of D are their respective variances:

$$D^2 = \begin{pmatrix} s_1^2 \cdots 0 \\ \cdots \\ 0 \cdots s_p^2 \end{pmatrix}, \quad U = (u_1, \ldots, u_p).$$

We denote the eigenvalues of S: $\lambda_1 = s_1^2, \ldots, \lambda_p = s_p^2$.

We can also obtain the eigenvalues and eigenvectors of the sample correlation matrix, R, using the same procedure if we standardize the vector x_g by subtracting the mean and dividing by the standard deviation.

As we mentioned in Section 9.1, our notation in this section differs from the standard SVD notation in classical multivariate analysis because in microarray data the multivariate observations are the columns and the variables (i.e., the genes) are the rows of the data matrix. In classical multivariate notation the reverse is true: the observations are the rows and the variables are the columns of the data matrix. For this reason our formulas for the singular value decomposition and principal components are slightly different than the ones found in any classical multivariate analysis text.

The objective now is to select a subset of k principal components containing most of the information in the original data. There are several ways to select k. The proportion of the variance explained by the k components is $p_k = (\lambda_1 + \cdots + \lambda_k)/(\lambda_1 + \cdots + \lambda_p)$. The number of principal components could be selected by one of the following criteria:

1. k components explain some fixed percentage of the variance (70%, 80%, etc.).
2. k eigenvalues are greater than the average of the eigenvalues (for the correlation matrix the average is 1).
3. *Scree plot.* Graph the eigenvalues and look for the last sharp decline and choose k as the number of points above the cutoff.
4. *Null hypothesis test that the last m eigenvalues are equal.* This is tantamount to testing that they are all essentially close to zero. Use as test statistic

$$u = \left(G - \frac{2m+11}{6}\right)\left(m \times \log \bar{\lambda} - \sum_{i=p-m+1}^{p} \log \lambda_i\right)$$

where $\bar{\lambda} = \sum_{i=p-m+1}^{p} \lambda_i/m$. The null distribution of the test statistic, u, is approximately a chi-squared distribution with $(m-1)(m+2)/2$ degrees of freedom. In many microarray experiments this method will eliminate only a few components because asymptotic results do not hold for cases with large number of variables and relatively few observations. However, this result is also true if we concentrate only on a range of components, say the first m, as long as certain assumptions about the multiplicity of eigenvalues are true. This topic is the subject of further research.

Example. Returning to the example in Section 9.1, Table 9.2 shows the summary of the 43 principal components of the 100 genes. This table is used to decide how many components are needed when following the methods above for principal components selection.

Table 9.2 Principal components summary for all the 43 principal components

Component	Variance	Cumulative Variance	Percent	Cumulative Percent
1	26.845	26.845	50.38	50.378
2	11.732	38.577	22.07	72.394
3	1.846	40.422	3.463	75.858
4	1.534	41.957	2.880	78.737
5	1.227	43.184	2.303	81.040
6	1.047	44.231	1.965	83.005
7	0.892	45.123	1.674	84.679
8	0.803	45.926	1.508	86.187
9	0.763	46.689	1.431	87.618
10	0.615	47.304	1.155	88.773
11	0.539	47.843	1.011	89.784
12	0.471	48.314	0.884	90.668
13	0.434	48.748	0.814	91.482
14	0.395	49.143	0.742	92.223
15	0.343	49.486	0.644	92.867
16	0.333	49.819	0.624	93.491
17	0.308	50.127	0.579	94.070
18	0.303	50.430	0.568	94.638
19	0.277	50.707	0.519	95.158
20	0.261	50.967	0.489	95.647
21	0.216	51.184	0.406	96.053
22	0.209	51.392	0.392	96.445
23	0.190	51.582	0.357	96.801
24	0.181	51.763	0.339	97.141
25	0.171	51.934	0.321	97.462
26	0.146	52.080	0.274	97.735
27	0.138	52.218	0.260	97.995
28	0.123	52.342	0.231	98.226
29	0.116	52.457	0.217	98.443
30	0.110	52.567	0.206	98.649
31	0.106	52.674	0.199	98.849
32	0.098	52.771	0.183	99.032
33	0.081	52.852	0.151	99.183
34	0.075	52.927	0.141	99.324
35	0.068	52.994	0.127	99.451
36	0.061	53.056	0.115	99.566
37	0.054	53.110	0.101	99.667
38	0.048	53.157	0.090	99.757
39	0.039	53.197	0.074	99.831
40	0.036	53.233	0.068	99.899
41	0.030	53.263	0.057	99.956
42	0.022	53.286	0.042	99.998
43	0.001	53.287	0.002	100.00

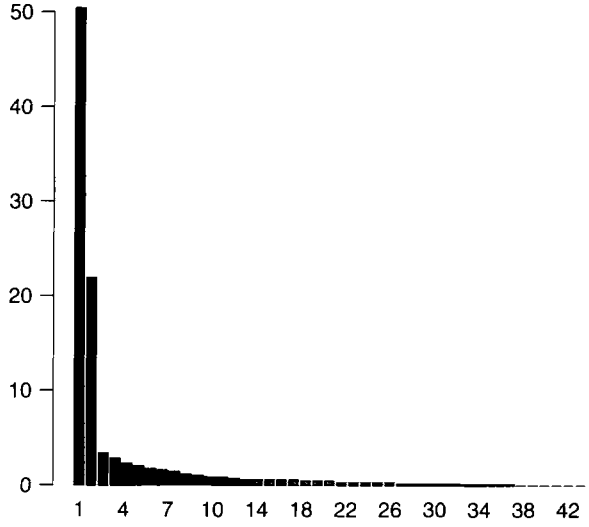


Figure 9.7 Scree plot.

1. Using criterion 1, it is appropriate to choose k between 2 and 12, since they determine the range of 70 to 90% variability. More than 12 components would mean very small increments in the variance.
2. The average of the eigenvalues is 1.24, which suggests that we should keep no more than 5 components.
3. The Scree plot is shown in Figure 9.7, suggesting that the number of components would be either 2 or 6 or 9.
4. There is a group of very small eigenvalues that we are going to discard, or otherwise the method produces negligible results. We concentrate on the last 25 principal components. Then the test becomes significant for 6 components or less and it becomes very significant for 2 or less.

In conclusion, it appears that two is the best number of PCs because it satisfies criteria 1 through 3, and we saw in Figure 9.6 that the two-dimensional view in the plane spanned by the first two PCs does indeed show the primary structure in the data, which is the separation of the samples into two groups. If 72% of the variability is not regarded as high enough, either six or nine principal components could be chosen:

$p-m$	24	20	15	9	8	7	6	5	4	3	2	1
U	0.1	5	32	146	182	222	279	340	425	554	1632	3260
χ^2	9.2	37	94	195	215	237	259	282	307	332	358	386

9.3.2 Factor Analysis

Factor analysis (FA) assumes the existence of a few latent variables that define the phenomena under study. The observations are functions of these unknown latent variables, or more specifically are linear combinations of the latent variables.

The objective of factor analysis is to estimate the latent variables and to try to express them in a form as related as possible to the original observations. The statistical model for factor analysis is

$$x = \Lambda f + \varepsilon,$$

where x is a column vector of G components, and it is assumed that the vector f has k components and $E(f) = 0$, $\text{cov}(f) = I$. The matrix $\Lambda = \{\lambda_{ij}\}$ is a $G \times k$ matrix of the coefficients of the linear combinations of the factors that compose the observed variables. The error term, ε, is a G-dimensional vector satisfying $E(\varepsilon) = 0$ and $\text{cov}(\varepsilon) = \psi = \text{diag}(\psi_1, \ldots, \psi_G)$. In addition f and ε are assumed independent of each other in the sense that $\text{cov}(f, \varepsilon) = 0$. Next we enumerate some of the important elements of the factor model:

1. *Covariance matrix.* The factor model expresses the $G \times (G-1)/2$ covariances among the G coordinates of x in terms of $G \times k$ loadings $\{\lambda_{ij}\}$ and G variances $\{\psi_i\}$:

$$\Sigma = \text{cov}(\Lambda f + \varepsilon) = \text{cov}(\Lambda f) + \text{cov}(\varepsilon) = \Lambda \, \text{cov}(f) \Lambda' + \psi = \Lambda\Lambda' + \psi.$$

2. *Factor loadings.* The factor loadings λ_{ij} represent the covariances of the variables with the factors. For example, the loading of variable x_1 on factor f_2 is

$$\text{cov}(x_1, f_2) = \text{cov}(\lambda_{12} f_2, f_2) = \lambda_{12} \, \text{var}(f_2) = \lambda_{12}.$$

3. *Communality.* We break down the variance of a variable between a component due to the common factors and a variable specific component.

$$\sigma_{ii} = \text{var}(x_i) = (\lambda_{i1}^2 + \cdots + \lambda_{im}^2) + \psi_i = h_i^2 + \psi_i.$$

   The communality component is h_i^2 and the component specific to the variable is ψ_i.

4. *Nonuniqueness.* The factors are identifiable only up to an orthogonal transformation. Let T be an orthogonal transformation, namely $TT' = I$. Then

$$\Lambda f + \varepsilon = \Lambda T T' f + \varepsilon = \Lambda^* f^* + \varepsilon,$$

where $\Lambda^* = \Lambda T$ and $f^* = T'f$. In addition, the properties of f are preserved by f^*. $E(f^*) = 0$, $\text{cov}(f^*) = I$, and $\text{cov}(f^*, \varepsilon) = 0$. In terms of the decomposition of the covariance matrix $\Sigma = \Lambda^*\Lambda^{*\prime} + \psi = \Lambda\Lambda' + \psi$, the communalities do not change since $h_i^{*2} = h_i^2$.

Estimation. We assume the factor model with m factors. There are two basic methods for estimating the factors:

1. *Principal components method.*

$$S = CDC' \cong C_1 D_1 C_1' + \hat{\Psi} = \hat{\Lambda}\hat{\Lambda}' + \hat{\Psi},$$

where $\hat{\Psi}_i = s_{ii} - \sum_1^m \hat{\lambda}_{ij}^2$. This decomposition is iterated a few times.

2. *Maximum likelihood method.* Assume that the observations are $N(\mu, \Sigma)$, and obtain the maximum likelihood estimators of Λ and ψ.

In choosing the number of factors, we have ideas parallel those of PCA:

1. m factors explain some fixed percentage of the variance (70% or 80%).
2. m eigenvalues are greater than the average of the eigenvalues (for the correlation matrix the average is 1).
3. *Scree plot.* Graph the eigenvalues and look for the last sharp decline and choose m as the number of points above the cut off.
4. *Null hypothesis test that there are m factors.* The test statistic is

$$u = \left(G - \frac{2p + 4m + 11}{6}\right) \ln\left(\frac{|\hat{\Lambda}\hat{\Lambda}' + \hat{\Psi}|}{|S|}\right).$$

The null distribution of the test statistic u is approximately a chi-squared distribution with $((p-m)^2 - p - m)/2$ degrees of freedom.

Rotations. Since factors are identifiable only up to an orthogonal transformation, it is convenient to choose the orthogonal transformation or rotation that produces a set of factors that are the easiest to interpret. Since it is likely that the number of factors m is much smaller than the number of variables G, the rotation is to be chosen so each variable contributes mainly to one or few factors. In some cases rotations produce a grouping of G variables into m subgroups represented by the factors. There are several methods for obtaining the appropriate rotation.

1. *Graphical approach.* When m is 2 or 3, it is easier to just do a graph and find the best rotation by eye. Software such as Ggobi or JMP provide tools for performing the rotation with the aid of a mouse.

2. *Varimax.* This method computes the rotation that maximizes the variance of the square loadings in each column of $\hat{\Lambda}$.
3. *Quartimax.* Maximizes the variance of the square loadings of each row of $\hat{\Lambda}$.
4. *Promax.* Power transformation plus rotation, so it is a transformation.
5. *Procustes.* Rotation to match a canonical configuration.
6. *General oblique.* $\hat{\Lambda}$ is now not necessarily orthogonal, but it is a nonsingular matrix.

Factor analysis is sometimes criticized because the assumptions of the underlying factors may be unrealistic. Many phenomena are very complex in nature and may not fit into the FA framework. In addition the application of a rotation may produce overoptimistic results just by chance. For these two reasons we recommend that FA be used only as an exploratory data analysis technique that is helpful for summarizing the variables in problems, such as the microarray data analysis, where the number of variables (genes) requires simplification and, hopefully, meaningful simplification.

Example. Factor analysis seems to suggest that nine factors or two are a good number. This is similar to the PCA conclusions in the sense of the dimensionality of the data. In Table 9.3 we give a table obtained from the R software with the factor loadings, and it shows that the most substantial changes indicate that we should select either two or nine factors.

9.3.3 Biplots

The *biplot* (Gabriel, 1971; Gabriel and Odoroff, 1990) is a graphical display of X in which two sets of markers are plotted simultaneously. One set of markers $a_1, \ldots, a_G$ represents the rows of X, and the other set of markers, $b_1, \ldots, b_p$, represents the columns of X. The basis of the biplot is that any matrix, X, can be approximated by a rank two matrix, X_2, of the same size as X and that this latter matrix, X_2, can be factored as $X_2 = AB'$, where A is a $G \times 2$ matrix, whose ith row is a_i, and B is a $p \times 2$ matrix, whose jth row is b_j, so that $X \approx AB'$.

Such an approximation can be obtained in several ways. For example, in the SVD, $X = UDV'$, if only the two largest singular values, s_1 and s_2, are

Table 9.3 Factor analysis results using 10 factors

	Fact1	Fact2	Fact3	Fact4	Fact5	Fact6	Fact7	Fact8	Fact9	Fact10
Ssloadings	17.73	11.42	1.187	1.055	1.015	0.891	0.822	0.632	0.560	0.317
Proportional variance	0.412	0.266	0.028	0.025	0.024	0.021	0.019	0.015	0.013	0.007
Cumulative variance	0.412	0.678	0.706	0.730	0.754	0.775	0.794	0.809	0.822	0.829

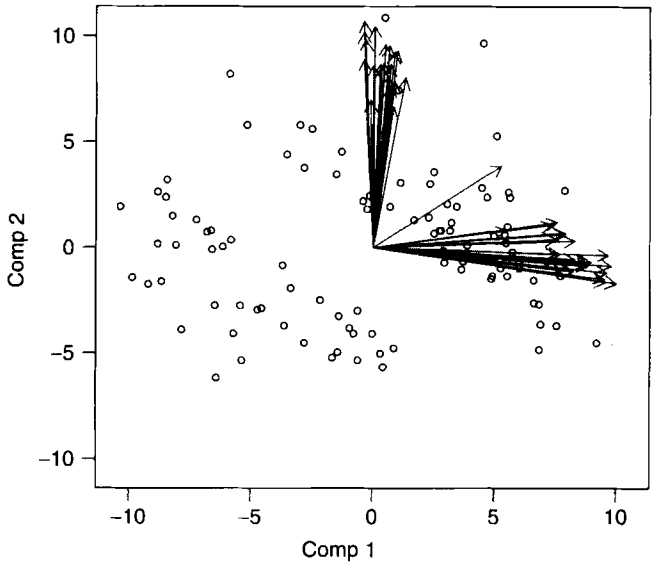


Figure 9.8 Biplot of the first two principal components.

retained, a rank two decomposition $X_2 = U_2 D_2 V_2^t$ can be obtained and factored as $A = U_2 D_2^a$ and $B = U_2 D_2^b$ with $a + b = 1$, for example, $(a, b) = (0, 1)$, $(a, b) = (1, 0)$, or $(a, b) = (0.5, 0.5)$. Chapman et al. (2002) use biplots for visually exploring microarray data from plant pathlogy experiments.

Example. Figures 9.8 and 9.9 show two biplots. The first biplot is for the first two principal components. The second biplot is for the first two factors after Varimax rotation. The points represent the 100 genes, and they display a pattern of two clear clusters. The reason for these two clusters is that these 100 genes were selected as having a highly significant t statistic for differentiating between two types of tumors. Hence there are two types of genes: (1) those that are differentially upregulated for the first tumor group and (2) those that are differentially upregulated for the second group. The PCA and FA methods capture this fact automatically, and it shows as clusters in the biplots. In addition the biplot shows that the cells are also split into two groups that also correspond to the two groups of tumors. The biplot graph for the FA is better for this separation because it does a *Varimax* rotation in the factor space.

9.3.4 Spectral Map Analysis

Wouters et al. (2002) found the *spectral map*, an extension of the SVD-based biplot originally developed by Lewi (1976) for displaying activity spectra of

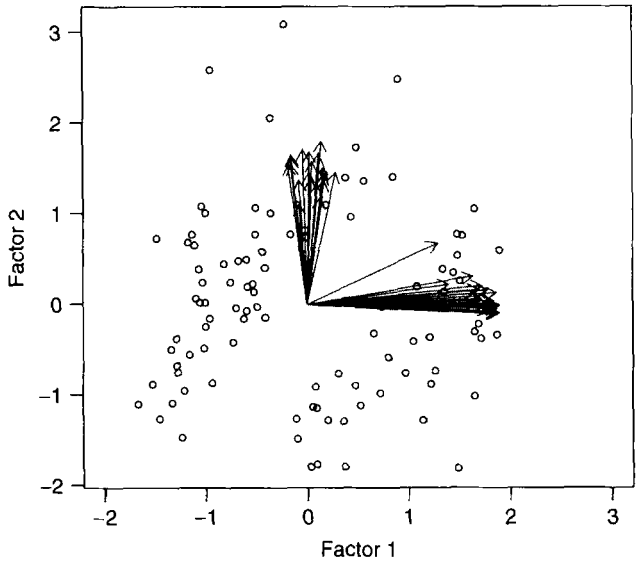


Figure 9.9 Biplot of the first two factors after Varimax rotation.

chemical compounds useful as a means of uncovering patterns in microarray data.

Spectral map analysis proceeds as follows: First, the data are transformed into relative values such that they sum to unity along the rows and along the columns. The row weighting has the effect that genes that have lower, and therefore generally unreliable, intensity measurements get lower weights than genes with higher intensity measurements. If x_{i+} is the ith row sum, x_{+j} is the jth column sum, and x_{++} is the overall total, this operation is

$$x_{ij} \rightarrow \frac{x_{ij}x_{++}}{x_{i+}x_{+j}}.$$

Next the transformed data are doubly centered. This removes the "size" effect, leaving only contrasts between the different rows and contrasts between the different columns. If $\bar{x}_{i+}$ is the ith row mean, $\bar{x}_{+j}$ is the jth column mean, and $\bar{x}_{++}$ is the overall total, this operation is

$$x_{ij} \rightarrow x_{ij} - \bar{x}_{i+} - \bar{x}_{+j} + \bar{x}_{++}.$$

The centered data is now globally standardized. If $W_n = \text{diag}_n(1/n)$, $W_p = \text{diag}_p(1/p)$ and $d = 1_n W_n Y^2 W_p I_p$, this operation is

$$x_{ij} \rightarrow \frac{x_{ij}}{d}.$$

Let the resulting matrix be denoted Z. A generalized version of the singular value decomposition is used in spectral map analysis:

$$W_n^{1/2} Z W_p^{1/2} = UDV^T,$$

where again U and V are orthogonal matrices and D is a diagonal matrix of singular values.

The factor scores, $S = W_n^{-1/2} UD^{\alpha}$, and the factor loadings, $L = W_p^{-1/2} VD^{\beta}$ are plotted on a biplot for the first few singular values. Different values for α and β produce biplots with different characteristics. In drawing spectral maps, one generally sets both to 0.5, which produces a distortion of the interpoint distances.

9.3.5 Multidimensional Scaling

In Section 9.2 we described the methods for constructing similarity or dissimilarity measures among observations and how to use them for obtaining clusters. The reverse problem is also interesting. Suppose that we have obtained a measure of similarity or dissimilarity between a set of objects. Can we produce a set of points that represent the objects in some low-dimensional Euclidean space?

In some microarray experiments we may want to assign more importance to certain genes, truncate some low values, in essence define a complicated measure of dissimilarity or similarity. In these cases it may be useful to be able to represent the data in a low-dimensional space based on the dissimilarity or similarity measure.

One method of doing this is *multidimensional scaling*. We begin with a matrix of dissimilarities $D = (d_{ij})$ among n objects (objects can be subjects, genes, etc.). If the information available consists of a similarity matrix S, then we proceed to obtain D as shown in 9.2.

Let $A = (a_{ij}) = \left(-\frac{1}{2} d_{ij}\right)$, and let the matrix $B = (I - G^{-1} \; 1 \; 1')A \cdot (I - G^{-1} \; 1 \; 1')$, where I is the $G \times G$ identity matrix and 1 is a vector of length G with all its values equal to one. Then, if B is positive semidefinite, let Y denote the first k eigenvectors of B, standardized so their length squared is the corresponding eigenvalue. The matrix Y gives a configuration of points in the k-dimensional real space with a distance matrix that is closest to D. The representation may not always be adequate if the dimension k is not sufficiently large for our data.

9.3.6 Projection Pursuit

Data of three or more dimensions are difficult to visualize. On the other hand, two-dimensional, or even three-dimensional, views (i.e., two- or three-dimensional projections) of the data are easy to visualize. In a two-dimensional graph, it is not hard to make out clusters or any other data structures. In a

three-dimensional graph, we can use rotation software that enables us to visualize the data. However, as the dimension gets higher, visualization becomes difficult at best.

One solution is to look at low-dimensional projections of high-dimensional data, but again, we encounter a problem because, as the dimension gets higher, the number of views becomes far too large. The motivation behind *projection pursuit* (PP) methodology (Friedman and Tukey, 1974; further developed by Friedman, 1987; see also Barnett, 1981) is to find a few low-dimensional views of the data that describe the structure of the high-dimensional dataset, such as clusters, outliers, or subspaces containing the data, as they may provide interesting information about the scientific questions motivating the data analysis.

A projection is considered interesting if it shows a nonrandom or nonnormally distributed point cloud. Projections showing a pattern of clusters or showing outliers are considered "interesting" since they differ markedly from a normal distribution. However, projections chosen at random are likely to be close to a normal distribution. This is a consequence of the central limit theorem because projections are linear combinations of variables.

The method of projection pursuit finds the projections that optimize a criterion called the *projection pursuit index* that measures how interesting a structure is within a view. The most common indexes are the Legendre index and the Hermite index.

Let $Y = PX$ be a one dimensional projection of our data. The *Hermite index* measures the distance from the empirical distribution of Y to a normal distribution. It was proposed by Hall (1989); Cook et al. (1993, 1995) recommended using just two of the Hermite polynomial expansions of this distance resulting in a very simple expression that it is easily computable and hence not difficult to optimize. The two term Hermite index is

$$I_H(P) = a_1^2 - 2^{1/2}\pi^{-1/4}a_0 + \tfrac{1}{2}\pi^{-1/2},$$

where $a_0 = \text{ave}(\pi^{-1/4}e^{-Y^2/2})$ and $a_1 = \text{ave}(\pi^{-1/4}e^{-Y^2/2} \times Y)$. The function "ave" represents the average of the expression over the sample points.

In order to define the Legendre index, we transform the projection Y into a variable U in the interval $[-1, 1]$ by the function $U = 2\Phi(Y) - 1$, where $\Phi(t)$ is the normal distribution function. The *Legendre index* measures the L_2 distance between the distribution of U and a uniform distribution on the interval $[-1, 1]$. This index was proposed by Friedman (1987). Again, we use a two-term approximation based on Legendre polynomial expansion:

$$I_L(P) = a_1^2 + a_2^2,$$

where $a_1 = \sqrt{\frac{3}{2}}\,\text{ave}(U)$ and $a_2 = \sqrt{\frac{5}{8}}\,\text{ave}(3U^2 - 1)$.

The method of projection pursuit consists of selecting projections that optimize a projection pursuit index and examining these projections graphically for

interesting structures. Cook et al. (1993, 1995) provide a detailed assessment of these and other indexes.

9.3.7 Data Visualization with the Grand Tour and Projection Pursuit

Cook et al. (1993, 1995) describe Xgobi/Ggobi, a fascinating computer implementation of these ideas that combines the idea of a Grand Tour (essentially, a movie of data projections, a continuous sequence of two-dimensional projections of multidimensional data; Asimov, 1985) with that of projection pursuit.

Example. Figure 9.10 shows a screen of the software Ggobi in action. The dataset is the same tumor data except that now 63 patients are included, corresponding to four types of tumors. The projection pursuit method succeeds in identifying four clusters corresponding to the four types of tumors without using the tumor information. The main panel in Figure 9.10 shows a two-dimensional projection selected by the PP index with the four clusters in different colors and glyphs. The top left panel shows the main controls, and the left bottom panel displays the controls and the line graph of the PP index that is being optimized. The graph shows a deep valley at whose bottom the optimization algorithm was turned on. The index value corresponds to a sequence

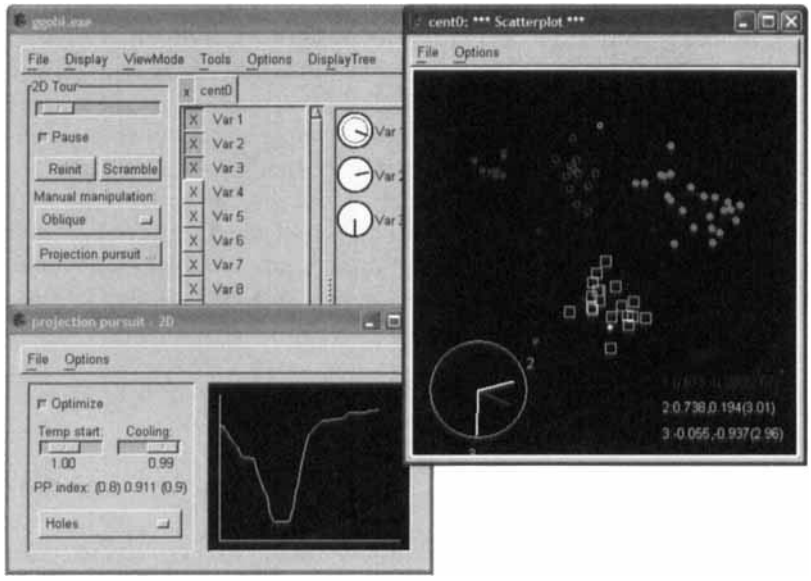


Figure 9.10 Ggobi display finding four clusters of tumors using the PP index on the set of 63 cases. The main panel shows the two-dimensional projection selected by the PP index with the four clusters in different colors and glyphs. The top left panel shows the main controls, and the left bottom panel displays the controls and the line graph of the PP index that has been optimized. The line graph shows the index value for a sequence of projections ending at the current one.

of projections, ending at the current one, which is the optimum reached by the algorithm. This is the projection shown in the central panel.

9.4 TWO-WAY CLUSTERING

Section 9.2 dealt with methods for clustering genes (these same algorithms can also be applied to cluster samples), while Section 9.3 dealt with methods for spotting clustering samples (these same methods can also be applied to spot genes that cluster together). In fact it makes sense to consider *two-way clustering*, in which both genes and samples (i.e., both the rows and the columns of the expression matrix) are clustered simultaneously. The goal of such an analysis is to identify groups of genes that participate in a biological activity taking place in only a subset of the samples.

Two-way clustering is somewhat more challenging than one-way clustering and new tools have been developed for this purpose. A straightforward approach, however, is to apply one-way clustering procedures separately to the rows and the columns and then to reorder the rows and the columns in such a way as to produce a two-way clustering.

9.4.1 Block Clustering

Block clustering (Hartigan, 1972; used by Tibshirani et al., 2000, for gene expression data) reorders the rows and columns of X to produce a matrix with homogeneous blocks of gene expression. The algorithm is started off with all the data in one block. At the next and each subsequent stage, the row or column split of all existing blocks that reduces total within-block variance the most is used to create new blocks. If an existing row or column split intersect a block, the block has to be split accordingly. Otherwise, all split points are tried. The process is continued until a large number of blocks are obtained. Then some blocks are recombined until an optimum number of blocks is obtained.

A form of block clustering called *coupled two-clustering* was applied to colon cancer data by Alon et al. (1999) and to colon cancer and leukemia data by Getz et al. (2000).

9.4.2 Gene Shaving

Gene shaving (Hastie et al., 2000) is a two-way clustering procedure that finds possibly overlapping clusters.

Initially each row of the gene expression matrix, X, is centered to have zero mean. Then a linear combination of rows (i.e., genes) having maximal variation in the column space is found. This is the first principal component of the rows of X. A specified proportion (typically 10%) of the genes having the lowest correlation with this linear combination is removed ("shaved") from the data. This process is repeated until only a single gene remains. This process

generates a nested sequence of gene blocks, one of which is selected as the first cluster by optimizing a criterion, usually the gap statistic (Tibshirani et al., 2001; also see Section 9.2.3).

At the next and each subsequent step, the rows of the gene expression matrix are orthogonalized with respect to the average gene in the cluster and the above steps are repeated with the orthogonalized data to find more clusters.

9.4.3 The Plaid Model

The *plaid model* was proposed by Lazzeroni and Owen (2002) as a method for identifying K, possibly overlapping, clusters of genes in which similarity within a cluster may extend across only a subset of the p samples.

The rough idea is that each cluster has its own mean expression level. Thus the genes belonging to the kth cluster have mean $\mu_0 + \mu_k$ across the samples in that cluster, where μ_0 refers to a background expression level and μ_k refers to the average expression level unique to the kth cluster. This is equivalent to writing

$$X_{ij} = \mu_0 + \sum_{k=1}^{K} \mu_k \rho_{ik} \kappa_{jk} + \varepsilon_{ij},$$

where $\rho_{ik} = 1$ if and only if the ith gene belongs to the kth cluster and is zero otherwise, $\kappa_{jk} = 1$ if and only if the jth sample belongs to the kth cluster and is zero otherwise, and ε_{ij} is a zero-mean error term.

The formulation can be set to be highly flexible. The constraint $\sum_{k=1}^{K} \rho_{ik} = 1$, that insists that a gene belongs to one and only one cluster, is not imposed. Instead, some genes may appear in more than one cluster, so that $\sum_{k=1}^{K} \rho_{ik} \geq 2$ for those genes, and some genes may not appear in any of the clusters, so that $\sum_{k=1}^{K} \rho_{ik} = 0$ for those genes. The constraint $\sum_{k=1}^{K} \kappa_{jk} = 1$, which insists that a sample belongs to only one cluster, is also not imposed. Instead, some samples may appear in more than one cluster, so that $\sum_{k=1}^{K} \kappa_{jk} \geq 2$ for those samples, and some samples may not appear in any of the clusters, so that $\sum_{k=1}^{K} \kappa_{jk} = 0$ for those samples.

It is useful to add more structure to this basic model by extending it to allow for a distinct regulatory effect for each gene and each sample within a cluster:

$$X_{ij} = \mu_0 + \sum_{k=1}^{K} (\mu_k + \alpha_{ik} + \beta_{jk}) \rho_{ik} \kappa_{jk} + \varepsilon_{ij}.$$

Observe that this model is essentially a superposition of K two-way ANOVA models. The constraints, $\sum_{k=1}^{K} \rho_{ik} \alpha_{ik} = 0$ and $\sum_{k=1}^{K} \kappa_{jk} \beta_{jk} = 0$, are imposed to avoid over parameterization of the model.

Somewhat more generality is obtained by writing

$$X_{ij} = \sum_{k=0}^{K} \theta_{ijk} \rho_{ik} \kappa_{jk} + \varepsilon_{ij}.$$

Fitting this model is nontrivial. Lazzeroni and Owen (2002) present an algorithm for doing so.

SOFTWARE NOTES

Software for commonly used clustering techniques, such as the various hierarchical clustering procedures and k-means, has been implemented in a number of different forms and are available in all statistical packages. Some implementations are better than others at handling large datasets and applications with large numbers of variables.

In `R` and `SPLUS`, the relevant functions are `hclust` for hierarchical clustering and `kmeans` for k-means. The methods described in Kaufman and Rousseeuw (1990) are also available in `R` and `SPLUS`: `agnes` for agglomerative hierarchical clustering, `pam` (which stands for "partitioning around medoids") and `clara` (which stands for "clustering large applications") for k-medoids clustering, `diana` for divisive hierarchical clustering, and `fanny` for fuzzy clustering. Software for model-based clustering is available in `R` and `SPLUS` as the function `mclust`.

EisenLab's `cluster` is a popular tool for clustering large microarray datasets via hierarchical clustering, self-organizing maps, k-means and principal components analysis. `TreeView` is an equally popular tool for displaying hierarchical cluster analysis results as dendrograms. There are a number of other such programs developed especially for clustering microarray data, such as `GeneClust` (which does hierarchical clustering and gene shaving and uses a simulation-based procedure to evaluate the results), `GeneCluster` (which does self-organizing maps), `GeneSpring` (which does hierarchical clustering, self-organizing maps, k-means, and principal components analysis), and `Partek` (which does self-organizing maps and k-means).

SUPPLEMENTARY READING

There has been such an extensive body of work on the subject of cluster analysis that there are entire books devoted just to this topic, including Aldenderfer and Blashfield (1984), Everitt (1993), Gordon (1999), Hartigan (1972) and Kaufman and Rousseeuw (1990). The latter emphasizes outlier-resistant techniques. Surveys are provided by Cormack (1979) and Gnanadesikan and Kettenring (1989). A number of multivariate analysis textbooks also provide

detailed accounts of clustering techniques, including Gnanadesikan (1997), Krzanowski (2000), Mardia, Kent, and Bibby (1979), and Seber (1984). The books by Hastie, Tibshirani, and Friedman (2001) and Ripley (1996) lie at the interface of statistics and data mining.

EXERCISES

9.1. Verify that D_E, D_M, D_∞, and D_{CAN} satisfy the dissimilarity axioms 1 through 5.

9.2. Use the full Khan et al. (2001) dataset of 88 samples that is included in the DNAMR package.

- **a.** Perform the principal components analysis to reduce the dimension of the gene set.
- **b.** Select the number of principal components that appears to represent the entire dataset using the four criteria that are given for this purpose.
- **c.** Graph the principal components using the **(i)** biplot and **(ii)** spectral map analysis, and try to identify the clusters.

9.3. Continue with the full Khan et al. (2001) dataset of 88 samples that is included in the DNAMR package.

- **a.** Perform the factor analysis using the varimax rotation procedure.
- **b.** Select the number of factors that appears to represent the entire dataset using the four criteria that are given for this purpose.
- **c.** Graph the main factors using the biplot and try to interpret the factors in the graph and try to identify clusters.
- **d.** Compare the results with those of Problem 9.3.

9.4. Continue with the full Khan et al. (2001) dataset of 88 samples that is included in the DNAMR package. Use a few principal components selected in Problem 9.2, or otherwise, use the principal components data set provided in the book's Web page.

- **a.** Perform a cluster analysis to search for patterns among the genes using single-linkage hierarchical clustering.
- **b.** Compare the result from part a with other methods such as Ward's method and average linkage.
- **c.** Use the results of the previous part b as the initial configuration for the k-means clustering procedure and compare the results.

9.5. Graph the results from Problem 9.4 using the microarray plot in DNAMR.

Exploration and Analysis of DNA Microarray and Protein Array Data
Edited by DHAMMIKA AMARATUNGA and JAVIER CABRERA

CHAPTER 10

Class Prediction

Microarray experiments can be used to classify mRNA samples on the basis of the type of mRNA that is present in them. For example, suppose that mRNA samples are available from several tumors. The tumors are known to be of various different classes, but for this set of tumors, we know to which class each tumor belongs. Now it is likely that different genes are expressed in the cells of different tumor classes. Therefore it can be conjectured that it ought to be possible to differentiate among the tumor classes by studying and contrasting their gene expression profiles, that is, by studying how the types and amounts of mRNA present in them vary from class to class and applying *class prediction* or *supervised classification* techniques to develop a classification rule to discriminate them. The knowledge gleaned from this exercise can be used not only to gather valuable information regarding the gene expression pattern of the underlying disease process, but also to predict the class of a new tumor of unknown class based on its gene expression profile.

This paradigm has tremendous potential as it is sometimes difficult to distinguish among certain tumor/cancer subtypes by clinical and histopathological means, as is current practice, but yet it is possible to discriminate among them by studying their gene expression data. The path breaking paper in this regard was by Golub et al. (1999), who demonstrated its feasibility by separating out two different but clinically indistinguishable types of leukemia, ALL (acute lymphocytic leukemia) and AML (acute myelocytic leukemia), based on gene expression information. Incidentally this data has been subsequently reanalyzed many times using various different methods (the book by Lin and Johnson, 2002, has several re-analyses).

Since then many applications of this idea have been reported. For instance, Hedenfalk et al. (2001) compared gene expression profiles for two types of hereditary breast cancer (breast cancer with BRCA1 mutation and breast cancer with BRCA2 mutation) and found that distinctly different groups of genes

Exploration and Analysis of DNA Microarray and Protein Array Data
By Dhammika Amaratunga and Javier Cabrera
ISBN 0-471-27398-8

are expressed by the two types, suggesting that a heritable mutation affects the gene expression profile of the cancer.

In addition quite a few applications have used cluster analysis to analyze microarray data even when class information was available. The better the resulting clusters matched the known classes, the better the clustering was deemed to be performing and the more informative the data structures that produced those clusters. This, however, is an indirect and thereby highly inefficient approach to the problem of classification.

Example. We will use the data of Khan et al. (2001) to illustrate the methods of this chapter. The dataset contains gene expression measurements, obtained using cDNA microarrays, from four types of pediatric small round blue cell tumors (SRBCT): neuroblastoma (NB), rhabdomyosarcoma (RMS), the Ewing family of tumors (EWS), and Burkitt lymphomas (BL), a subtype of non-Hodgkins lymphoma. The four cancer types are clinically and histologically similar, yet their response to treatment is markedly different, making accurate diagnosis essential for proper therapy. The purpose of the study was to classify, as accurately as possible, a cell as being one of these four types using gene expression information. The microarrays measured the expression levels of 6567 genes. This data was filtered to remove any gene that consistently expressed below a certain minimum level of expression, leaving expression data for 2308 genes. A total of 88 cells were analyzed. Data for 63 of these cells (23 EWS, 20 RMS, 12 NB, 8 BL) was used as a training set, while the data for the remaining 25 cells (6 EWS, 5 RMS, 6 NB, 3 BL, 5 non-SRBCT) was set aside to make up a blind test set.

10.1 INITIAL CONSIDERATIONS

As in the previous chapter, the data in this chapter will be set up as a *gene expression matrix*, a $G \times p$ matrix, $X = \{x_{ij}\}$, whose G rows and p columns represent, respectively, the G genes and p samples. Depending on the experiment, the p samples may correspond to p tissues, cell lines, tumors, or something else. The p samples belong to k different classes. It is known a priori to which class each sample belongs and there are n_j samples from the jth class, $\sum_{j=1}^{k} n_j = p$. The p-vector, $y = \{y_j\}$, indicates to which class each sample belongs: thus $y_j = s$ if the jth sample belongs to the sth class.

The values x_{ij} that make up the gene expression matrix could be either the measured gene expression level for the ith gene in the jth sample, suitably transformed and normalized, or, particularly in two-channel experiments, the log of the ratio of the normalized gene expression level for the ith gene in the jth sample relative to its corresponding value in a reference sample. A generic G-vector, $x = (x_1, \ldots, x_G)$ will denote a *gene expression profile*, a vector of gene expression data for the G genes.

The gene expression matrix, X, functions as a *training set* (also called a

learning set or *design set*) for classification as it is a set of samples for which the classes are known and gene expression data is available. The objective of supervised classification is to use the training set data for "training" purposes, that is, to develop a *classification rule*. The idea is that given a new sample with gene expression profile x, the classification rule can be used to predict, as accurately as possible, the true class of the new sample (assuming that its true class is one of the k classes) based on its gene expression profile, x. Generally, the classification rule will be based on a *classifier* that partitions the space of all possible x's into k disjoint subsets, $A_1, \ldots, A_k$, such that if x falls into A_s, then x is predicted to belong to class s.

10.1.1 Misclassification Rates

If a classification rule predicts that x belongs to class s when the truth is that x belongs to some other class t ($t \neq s$), then a *misclassification* is deemed to have occurred. The proportion of misclassifications in the training set, called the *misclassification rate*, is the most natural measure for evaluating the performance a classification procedure.

However, since the classification rule would have been optimized, in some sense, for the training set, this raw misclassification rate tends to seriously underestimate the true error rate of the procedure.

One way to circumvent this problem is *test set cross-validation* which consists of setting aside a portion of the samples as a *validation set* or *test set*, then construct the classification rule based on the training set, which is now all the samples other than the test set, and use the proportion of misclassifications in the validation set as an assessment of the performance of the procedure.

Another strategy that is on the same lines, but is a less wasteful use of resources, is *leave-out cross-validation*. In *leave-out-one cross-validation*, each sample in turn is set aside as a one-sample validation set and its class is predicted by constructing the classification rule based on the rest of the samples as the training set. The proportion of misclassifications is an indication of the performance of the classification procedure. A variant of this strategy is *leave-out-k cross-validation* in which the number of samples set aside as the validation set at each step is k rather than one.

Bootstrap methodology (Efron and Tibshirani, 1993, is a good general reference) can be used to improve the behavior of the raw misclassification rate, E_{obs}, as a measure of the true misclassification rate, E, of a classification procedure. A set of p samples, chosen at random with replacement from the original p samples, is used as the training set (called the *bootstrap training set*), with which a classification rule is constructed. The original p samples are used as the validation set, from which a misclassification rate can be determined. This is repeated several times and the average misclassification rate, E^*, is determined. The misclassification rate of the procedure is taken to be $E_{boot} = E_{obs} + (E^* - E_{obs}) = E^*$. A slightly modified version of this, called the

0.632 bootstrap, that has been shown to be an improvement is $E_{0.632} = 0.368E_{obs} + 0.632E^{**}$, where E^{**} is the average bootstrap misclassification rate for those samples that are not included in the bootstrap training set.

10.1.2 Reducing the Number of Classifiers

The idea is to use either the genes or certain combinations of the genes as classifiers for classifying the samples into classes. Since the number of genes generally greatly exceeds the number of samples, if we were to treat all the genes as classifiers, there will a great deal more classifiers than samples. By retaining such a large number of classifiers, it is incredibly easy for a classification rule to find good-looking but irreproducible and meaningless separation. This will result in a spuriously low misclassification rate in the training set, but a high misclassification rate in a test set.

There is an intuitive geometrical argument that illustrates this fact. Suppose that we have a dataset of three samples and two genes, where the three samples are members of two classes. If we represent the three samples as three points in the two-dimensional plane, it is easy to see that, if the points are not aligned, there is always a line in the plane that splits the three samples into the two classes. The same is true if we have four samples and three genes in three dimensions divided into two classes, as there is a two-dimensional plane that produces the correct classification. The argument generalizes to a dataset with $p + 1$ samples and p genes that spans the p-dimensional space and the samples are members of two classes. Then there is a $(p - 1)$-dimensional plane that produces the correct classification. The good behavior of this linear classification is just a geometrical artifact and should not be taken to imply that the genes in the dataset are good classifiers. However, when the number of samples is much greater than the number of genes, the geometrical artifact disappears, and any good classification will be a consequence of the relationship between the gene expression levels and the classes. This means that in order to demonstrate that a set of genes are genuinely good classifiers, we need to have many more samples than classifiers.

Besides this issue, biological considerations make it highly likely that, among the thousands of genes printed on an array, only a handful are really useful as classifiers in any situation, with the retention of the rest merely contributing noise and obfuscating the separation between classes. Thus the performance of a classification procedure would be vastly improved by reducing this number beforehand to a much smaller number of relevant genes, a process known as *gene filtering*. It is best that this reduction be carried out independently of the classification rule as; otherwise, there will be a risk of overfitting, but this is rarely possible.

Besides the use of prior knowledge, the simplest strategy to reduce the number of genes is to argue that the genes that are likely to be the best classifiers will express differentially across the different classes. If this is the case, they

can be identified using the tests for differential expression (e.g., the t test, SAM, CT, or F test, with a sufficiently low FDR) that were discussed in Chapters 7 and 8.

While it surely makes sense to eliminate as a potential classifier any gene that expresses at about the same level in all the samples, there are some problems with this approach: (1) filtering out too many genes may lead to loss of most of the classification information, particularly with the risk that some of those genes are just false discoveries, while retaining too few genes may not reduce enough noise, (2) there may exist a set of genes that together acts as a classifier, but each individual gene in the set does not, (3) there could be a redundancy of information as many genes may be picking up the same pattern of differential expression, and (4) since, in microarray experiments, the gene pool is very large, there will be individual genes that, just by chance, may appear to be good classifiers in the dataset at hand but this result may not be reproducible. In fact it is possible that some of the extraordinarily good-looking results that were reported in the early days of microarray research but that subsequently failed to reproduce could be due to this phenomenon.

Another way to filter genes that overcomes many of the previous objections is to consider multiple genes simultaneously. Bo and Jonassen (2002) show that a pair of genes in combination separates two classes better than doing the filtering gene by gene. Gene pairs (or other multiples) can be selected using Hotelling's t test (e.g., see Mardia et al., 1979), the multivariate form of the t test.

Multiple genes can also be considered simultaneously by constructing linear combinations of genes. This can be done using dimension reduction methods, such as principal components analysis or factor analysis, which were introduced in Section 9.3. By dropping all but the most important linear combinations, these methods will produce a small set of classifiers made up of *features*, linear combinations of the genes, in such a way that they preserve, in some sense, almost all the information contained in the original genes. The classifiers are then features rather than individual genes.

Khan et al. (2001) use PCA to generate features for class prediction with microarray data. However, one nagging concern with using PCA this way is that the information we are most interested in, that is, information related to class differences, could be overwhelmed by other aspects of the data that are irrelevant to the class prediction problem. Preceding PCA or FA by a gene-filtering step should mitigate this concern somewhat.

Partial least squares (PLS) is another method for defining features. Whereas PCA sequentially constructs orthogonal linear combinations, Xl, of the G genes that maximize the variance, $\text{var}(Xl)$, without paying any heed to the classes, y, PLS sequentially constructs orthogonal linear combinations, Xl, of the G genes that maximize the covariance, $\text{cov}(Xl, y)$ between Xl and the classes, y. This seems to address the concern with PCA mentioned above, but, if PLS was preceded by a gene filtering step, it is unlikely to produce results that are substantially different from PCA. However, Nguyen and Rocke (2002) do report an

improvement in classification by using PLS rather than PCA for feature selection in a microarray context.

An alternative is to run a PCA as above and then rank the principal components in the order suggested by the ratio of between class variance to within class variance. This strategy was originally proposed by Krzanowski (1992) for a chemometrics problem and then was implemented and extended by Landgrebe et al. (2002) and Coombes (2002) for classifying microarray data. An alternative that may be able to do this more directly is *projection pursuit regression* (Friedman and Stuetzle, 1981), which tries to find smooth functions of the linear combinations Xl that correlate with y.

Yet another way to reduce the effective number of classifiers is to run a clustering procedure over the set of genes or a large interesting subset of them, and form features that are the averages of the clusters or, alternatively, one or two principal components.

Many researchers have recognized the importance of classifier selection for microarray classification problems. Golub et al. (1999) and Dudoit et al. (2002) discuss individual gene selection, and the latter, in particular, shows how the performance of many classification rules can be improved by reducing the number of classifiers. Several papers in the book edited by Lin and Johnson (2002) discuss the general issue.

For simplicity, in the remainder of this chapter, we will refer to G classifiers even though the actual number of classifiers may have been reduced to a smaller number.

Example. We will illustrate some of the proposed dimension reduction techniques with the example of the four types of SRBCT tumors.

Method 1. Take the first 10 principal components for the entire set of 2308 genes. Since the sample covariance matrix is a 2308 by 2308 singular matrix of rank 62, we use the singular value decomposition described in Chapter 9. Figure 10.1 shows the scatter plot of the 63 training samples and the 25 test samples in the coordinates of the first two principal components (i.e., the two first PCA basis). The training samples are represented by small filled symbols, while the testing samples are the unfilled larger symbols. The test set symbols in the graph show a nearly random pattern, indicating that two principal components are not enough to produce a good classification rule.

Method 2. Select the genes that have a significant F statistic at $\alpha = 0.001$ the level and take the top 10 principal components. This subset contains a total of 450 significant genes. As in the previous case, the sample covariance matrix is high dimensional and highly singular and hence the principal components are calculated with the singular value decomposition method. Figure 10.2 shows the scatter plot of the 63 training samples and the 25 test samples in the two first PCA basis with the same style as Figure 10.1. The data shows a clear separation between the four classes for both training and testing sets

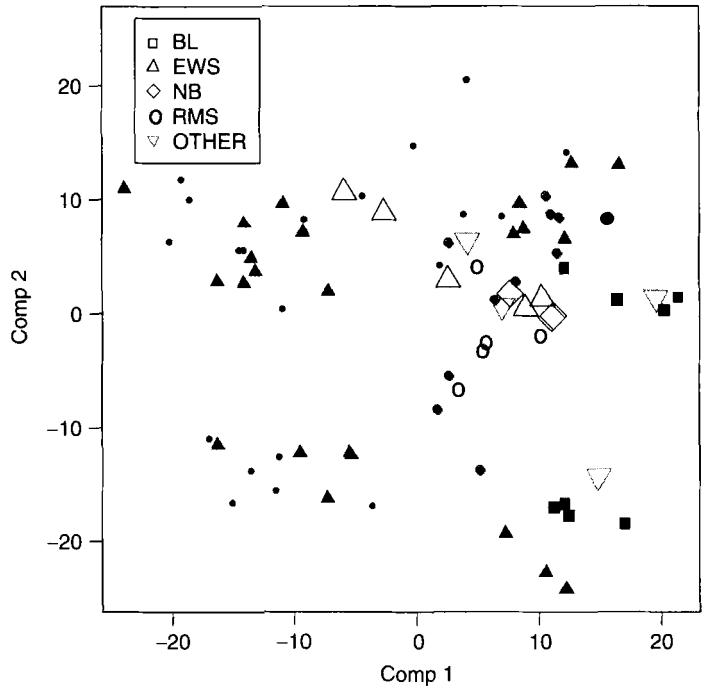


Figure 10.1 This graph shows the 63 training samples and the 25 test samples in the coordinates of the first two PCA basis. The training samples are represented by small filled symbols while the testing samples are the unfilled larger symbols. The data shows a nearly random pattern that illustrates the poor performance of this classification rule.

that suggest that this dimension reduction method will produce good classification rules.

Method 3a. Select the 50 most significant genes for the F statistic and cluster the genes into 10 clusters. Take the average of the genes in each cluster to produce a set of 10 classification variables. The variables can be sorted in order of significance using the values of the F statistic. Figure 10.3 shows the scatter plot of the 63 training samples and the 25 test samples in the two first PCA basis with the same style as Figure 10.1 and 10.2. The data shows a clear separation between the four classes for both training and testing sets that is almost as good as the one in Figure 10.2. This suggests that this dimension reduction method will produce good classification rules.

Method 3b. This is a variant of the previous method. Take the 10 classifiers in method 3a, and calculate the 10 principal components. The 10 principal components will produce the same classification rule that the previous 10 cluster mean variables, but there maybe subsets of principal components that perform better than equal size sets of cluster means. Figure 10.4 shows a clear separation between the four classes for both training and testing sets

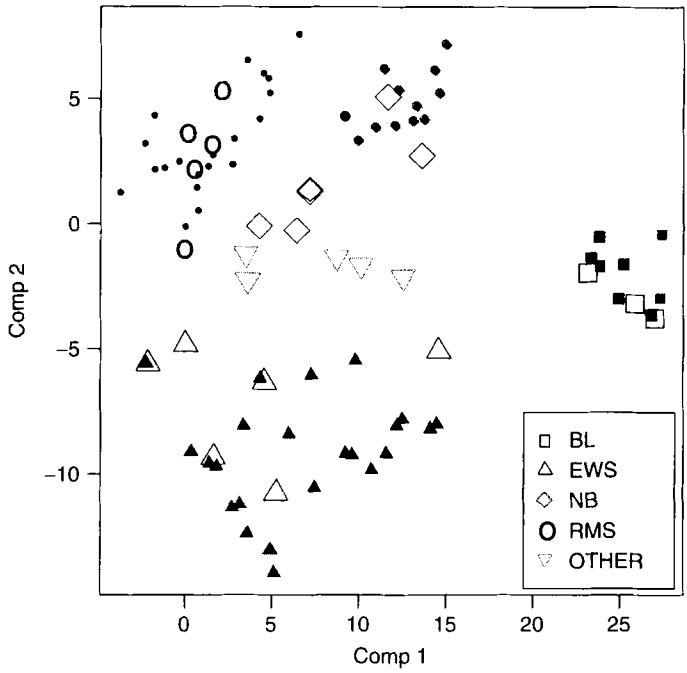


Figure 10.2 Principal components for 450 significant genes. This graph shows the 63 training samples and the 25 test samples in the two first PCA basis. The data shows a strong clustering pattern that explains the excellent performance of the classification rules.

that is as good as the one in Figure 10.1. This suggests that this dimension reduction method will produce good classification rules.

Method 4. Select the 30 most significant genes for the F statistic and take the top 10 principal components. Figure 10.5 shows the scatterplot of the 63 training samples and the 25 test samples in the two first PCA basis. The data shows a clear separation between the four classes for both training and testing sets that suggest that this dimension reduction method will produce good classification rules.

10.2 LINEAR DISCRIMINANT ANALYSIS

The oldest and one of the simplest methods of supervised classification, *linear discriminant analysis* (LDA) (Fisher, 1936) endures to this day as one of most popular classification techniques. It is based on finding the linear projections (views) of the data that most effectively separates out the k classes.

The most common situation is the case $k = 2$, where there are just two classes of samples. In this case classification can be based on the projection $w'x$: the projection is made in the direction w where the classes are most

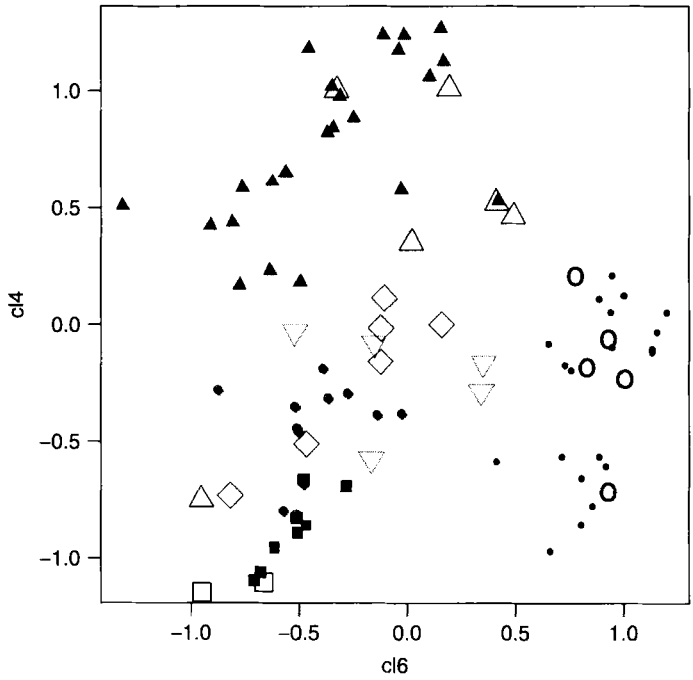


Figure 10.3 Two cluster means of 50 significant genes. This graph shows the 63 training samples and the 25 test samples in the two first PCA basis. The data shows a strong clustering pattern that explains the excellent performance of the classification rules.

widely separated in the training set. Letting n_s, the G-vector $\bar{x}_s$ and the $G \times G$ matrix S_s denote, respectively, the sample size, the mean and the variance-covariance matrix of the sth class in the training set, and letting $S = ((n_1 - 1)S_1 + (n_2 - 1)S_2)/(n_1 + n_2 - 2)$ denote the pooled variance-covariance matrix, a standardized measure of the separation between the two samples in the training set in the direction w is given by

$$\lambda = \frac{(w'\bar{x}_1 - w'\bar{x}_2)^2}{w'Sw}.$$

The direction w that maximizes λ is

$$w = S^{-}(\bar{x}_1 - \bar{x}_2),$$

where S^{-} denotes the generalized inverse of S, because, with microarray data, S will almost always be singular. The classification rule then is based on the *linear classifier*:

$$w'x = (\bar{x}_1 - \bar{x}_2)'S^{-}x.$$

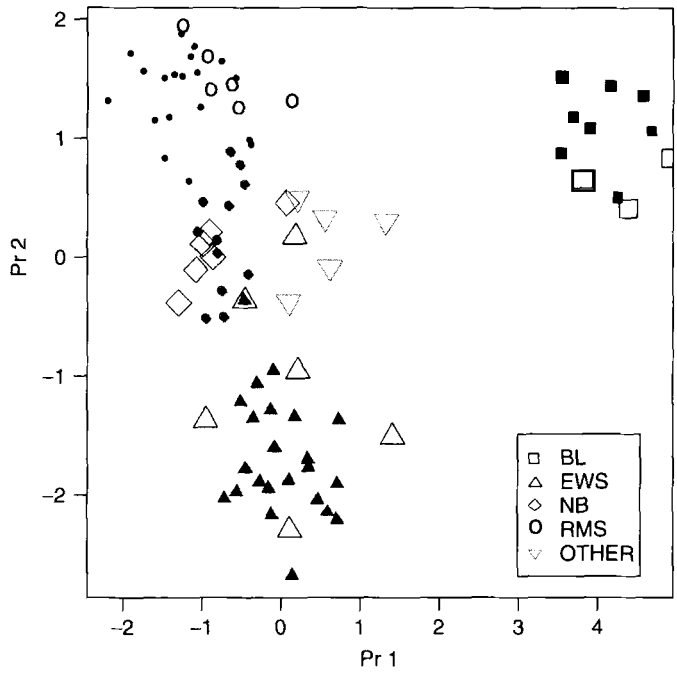


Figure 10.4 The first two principal components for the means of 10 clusters obtained from a subset of the top 50 significant genes. This graph shows the 63 training samples and the 25 test samples in the two first PCA basis. The graph shows that for the EWS tumors two observations in the testing and one in the training are in the boundary with the group of NB tumors.

The classification rule is

If $w'x > w'(\bar{x}_1 + \bar{x}_2)'/2$, then x is classified as belonging to class 1.
Otherwise x is classified as belonging to class 2.

While no distributional assumptions were made in the derivation above, this rule can also be obtained under the assumption that the data is normally distributed with the same variance covariance matrix, in which case the rule can also be shown to possess various optimality properties.

This method can be readily generalized to the case of more than two classifiers (Rao, 1948). Again, let n_s, the G-vector $\bar{x}_s$ and the $G \times G$ matrix S_s denote, respectively, the sample size, the mean and the variance-covariance matrix of the sth class in the training set. Let

$$\bar{x} = \frac{\sum_{s=1}^{k} n_s \bar{x}_s}{\sum_{s=1}^{k} n_s} \quad \text{and} \quad S = \frac{\sum_{s=1}^{k} (n_s - 1) S_s}{\sum_{s=1}^{k} (n_s - 1)}$$

denote the overall mean and pooled variance-covariance matrix. Then a stan-

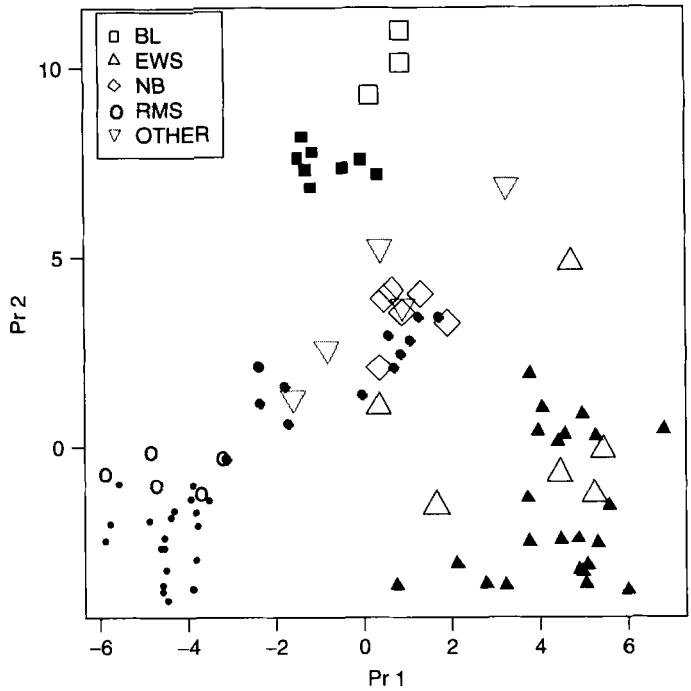


Figure 10.5 Principal components for the top 30 significant genes. This graph shows the 63 training samples and the 25 test samples in the two first PCA basis. The data shows a strong clustering pattern that explains the excellent performance of the classification rules.

dardized measure of the separation between the k sample means in the training set in the direction w is given by

$$\lambda = \frac{w'Bw}{w'Sw},$$

where $B = \sum_{s=1}^{k} (\bar{x}_s - \bar{x})(\bar{x}_s - \bar{x})'$ is the between class matrix.

The extreme values of λ correspond to the eigenvalues, $\lambda_1, \ldots, \lambda_t$, and eigenvectors, $w_1, \ldots, w_t$ of $S^- B$, where S^- denote the generalized inverse of S. There are at most $t = \min(k - 1, G)$ distinct eigenvalues. The eigenvector that corresponds to the largest of these eigenvalues is the direction in which the classes are maximally separated in the training set and provides the view of most interest. On the other hand, the eigenvectors corresponding to the smallest eigenvalues tend to obscure the separation of the samples into classes.

Let $D_h(x) = \sum_{s=1}^{t} [(x - \bar{x}_h)' w_s]^2$ denote the squared distance (in terms of the t eigenvectors) of x from the hth class mean. Then x is predicted to belong to the class whose mean is closest to x, namely to the class h that has the smallest

value of $D_h(x)$. This is equivalent to using the linear discriminant function

$$L_h(x) = \bar{x}_h' S^- x - \tfrac{1}{2}\bar{x}_h' S^- \bar{x}_h$$

and assigning x to the class with the largest $L_h(x)$.

It has been found that LDA with no initial gene filtering does not perform well. Indeed, in the comparison study of Dudoit et al. (2002), it was one of the worst performers. Besides the issues discussed in Section 10.1.2, the difficulty of estimating S efficiently with a very small p, a common problem in microarray studies where replication is limited, is at least partly responsible for this phenomenon. In fact the performance of LDA improves significantly with aggressive gene filtering.

Example. In order to illustrate the use of the LDA method and the improvement garnered by gene filtering, we apply it to the four variable reduction methods in Section 10.2. Since each reduction method produces 10 classifiers in a given order, we apply LDA to a few subsets in the given order. The results are shown in Table 10.1. The message is very clear, since methods 2, 3, and 4 did reasonably well and much better than method 1. In particular, methods 3 and 4 used very few genes. Method 3 with four classifiers used only twenty genes, which was the lowest in terms of number of genes.

10.3 EXTENSIONS OF FISHER'S LDA

Fisher's basic method has, over the years, been extended in various ways.

***Quadratic Discriminant Analysis* (QDA).** Dropping LDA's assumption that the true variance covariance matrices of the classes are the same produces the QDA classification rule: assign x to the class with the largest value of

$$Q_h(x) = (x - \bar{x}_h)S_h^-(x - \bar{x}_h)' + \log(S_h).$$

Table 10.1 Results of Fisher LDA for the four rules for selecting classifiers giving the number of misclassifications for both training and testing samples

	10 PC of 2308 Genes		10 PC of 450 Genes		10 Cluster Means of 50 Genes		10 PC of 30 Genes	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
2 Classifiers	35	13	0	2	4	2	3	2
3 Classifiers	5	5	0	0	0	1	1	1
4 Classifiers	0	3	0	0	0	0	0	0
10 Classifiers	0	3	0	0	0	0	0	0

Table 10.2 Results of QDA for the four rules for selecting classifiers giving the number of misclassifications for both training and testing samples

	10 PC of 2308 Genes		10 PC of 450 Genes		10 Cluster Means of 50 Genes		10 PC of 30 Genes	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
2 Classifiers	26	14	0	2	3	2	4	6
3 Classifiers	0	6	0	1	1	1	0	5
4 Classifiers	0	6	0	1	0	0	0	0
7 Classifiers	0	3	0	0	0	0	0	0

Note: The lowest row uses 7 classifiers because one of the classes has only 8 samples.

Example. We repeated the LDA analysis using the QDA method. Since each reduction method produces 10 classifiers in a given order, we apply LDA to a few subsets in the given order. The results are shown in Table 10.2 are more or less the same as those in table 10.1, perhaps slightly worse in the sense that they vary a bit more among the last three variable reduction procedure. The explanation is that some of the class sizes are very small (8 is the smallest one so we may use a maximum of 7 classifiers) and in such cases using a QDA rule means increases the numbers of parameters in the model to a point that it may produce a worse fit. In any case the differences between QDA and LDA are small.

***Diagonal Linear Discriminant Analysis* (DLDA).** A special case of QDA occurs when S_h is set to be a diagonal matrix that is the same for every class. Thus correlations between genes and variance differences between samples are ignored. It turns out that, like LDA, DLDA is a linear classification rule. Dudoit et al. (2002) found that DLDA to be one of the strongest performers in their comparison study.

***Diagonal Quadratic Discriminant Analysis* (DQDA).** Another special case of QDA is when S_h is set to be a diagonal matrix that is different across classes. This is a quadratic classification rule.

***Regularized Discriminant Analysis* (RDA).** When the sample sizes in the training set are small, as is typically the case with microarray data, it is likely to be difficult to estimate S_h efficiently. Consequently the performance of QDA may not offer an improvement over LDA, even though the covariance matrices of the classes are moderately different (as we saw in the example above). A compromise is to use a weighted average of S_h and S:

$$S_h^* = \frac{(1-\alpha)S_h + \alpha S}{(1-\alpha)n_h + \alpha n}$$

as an estimate of the covariance matrix of the hth class. QDA is then applied as above with S_h^* in place of S_h, with the value of α estimated to maximize per-

formance via cross-validation. This compromise was reached by *shrinking* S_h toward a common value, S, along the lines of the concept of *regularization*, which suggests shrinking a highly parametrized model toward one that is less highly parametrized. It has been demonstrated that regularization improves certain properties of the estimation, at the cost, however, of a slightly biased fit. Another mode of regularization is to shrink S towards a diagonal matrix. It has being argued (e.g., see Friedman, 1989) that one advantage of this double regularization for the classification problem is that it can be used to reduce the emphasis of the smaller eigenvalues.

***Prediction Analysis for Microarrays* (PAM).** The plan behind this method proposed by Tibshirani et al. (2002) is to use a simple centroid distance classification rule but to regularize by shrinking the centroids in a similar way as was done in the SAM method described in Section 7.3. Define the G-dimensional vector, $d_j = (\bar{x}_j - \bar{x})/[m_j(s + s_0)]$, where $m_j = \sqrt{1/n_j + 1/n}$, s is a vector of length G containing the within-class standard deviations for each of the G genes and s_0 is a fixed constant equal to the median of the components of s. We define the new shrunken centroids as

$$\bar{x}_j = \bar{x} + m_j(s + s_0)d_j',$$

where $d_j' = \text{sign}(d_j)(|d_j| - \Delta)_+$ and Δ is a fixed scalar. The sign function gives a vector of the signs of the components of d_j, that is, it returns a vector of $+1$'s, -1's, and 0's for positive, negative and zero components respectively. The value of Δ is chosen according to the method of cross-validation described in Section 10.5 below. The procedure has an in-built gene selection mechanism as the vector d_j' may have some zero components, which would imply that those genes are automatically excluded as classifiers.

***Flexible Discriminant Analysis* (FDA).** This method consists of two steps. First, a nonlinear model is fitted to the data using a binary numeric representation of the response. For this step, any nonlinear nonparametric regression estimator, such as generalized additive models, lowess, projection pursuit regression, or MARS, can be used. Second, a linear discriminant classification rule is applied to the fitted values from the first step as predictor variable and the same response variable. Hastie et al. (1994) provide further details.

Bayes's Rule. In the event that prior probabilities $(\pi_1, \ldots, \pi_k)$ can be assigned to the k classes, classification by Bayes's rule is to assign x to the class with the largest value of

$$L_h^B(x) = \bar{x}_h' S^- x - \tfrac{1}{2}\bar{x}_h' S^- \bar{x}_h + \log(\pi_h).$$

For a toxicogenomics problem in which the classes were not well defined, Raghavan et al. (2003) apply a modified version called *fuzzy class prediction*.

10.4 NEAREST NEIGHBORS

Nearest-neighbor methods for classification (first proposed by Fix and Hodges, 1951) are among the oldest and more successful classification methods.

Following the notation in Section 10.1, let x_j represent the jth sample and y_j give the class number of the jth sample. Let x be the candidate sample for classification, and let $S_{k,x}$ be the set of the k nearest neighbors of x in the training set. The simplest k-nearest neighbor (kNN, for short) method consists of estimating the probability that x belongs to the ith class $p(i|x)$ by the proportion of the k nearest neighbors that belong to the ith class:

$$\hat{p}(i|x) = \frac{\#\{g_j = i \mid x_j \in S_{k,x}\}}{k}.$$

The classification rule is based on a "majority vote": x is assigned to the ith class if i maximizes the probability $\hat{p}(i|x)$.

This method assumes that the $p(i|x)$ is approximately constant in the region containing the k nearest neighbors of x, for all i. In practice, this means that the number, p, of samples should be large and the number, G, of genes should be small compared to p, which is atypical of microarray experiments. Since this is never the case, a few of the principal components or factors should be taken as classifiers as explained above in Section 10.1.1.

Dudoit et al. (2002) found that along with DLDA, kNN had very good performance in their comparison study, particularly when it was preceded by gene filtering. Pomeroy et al. (2002) analyzed microarray data from 99 patients using kNN after gene filtering and demonstrated that medulloblastomas, the most common malignant brain tumors of childhood, were molecularly distinct from other brain tumors.

More sophisticated implementations of the kNN method are readily available, for example:

1. The decision rule can be rendered more complex by introducing the idea of a loss function. Let l_{ij} represent the loss that is sustained by stating that x belongs to the ith class when, in reality, x belongs to the jth class. Then we calculate the risks

$$r_i = \sum_{j=1}^{p} l_{ij}\hat{p}(j|x).$$

   The decision rule will assign x to the class that gives the minimum of the risks r_i.
2. By rescaling the variables by dividing them by their corresponding standard deviations before computing the k nearest neighbors, the distance between the samples will be scale independent.

3. Friedman (1994) introduced a combination of nearest neighbors and recursive partitioning that is very successful.

Example. We redo the classification using the kNN method. The results are shown in Table 10.3 are more or less the same as Table 10.1, perhaps slightly worse in the sense that only procedure 3 achieves 0 misclassifications.

10.5 RECURSIVE PARTITIONING

A classification rule induces a partition in the space of all possible samples that assigns each possible sample to a class. The best partition is selected by optimizing a classification criterion. In reality the number of possible partitions produced by a method depends only in the configuration of the observed samples, and as a result the number of possible partitions grows exponentially with the number of samples in our dataset. In most practical cases it is not possible to find the globally optimal partition and we resort to methods that produce nearly optimal partitions. One such method is *recursive partitioning* that is used to generate classification trees.

Trees are the obvious method for displaying the results of recursive partitioning, and consequently statisticians have been growing them at least since Morgan and Sonquist (1963). Many methods for growing classification trees have been proposed by statisticians (e.g., CHAID—Hartigan, 1975; FIRM—Hawkins and Kass, 1982; CART—Breiman et al., 1984; SPlus TREE—Clark and Pregibon, 1992) and computer scientists (e.g., C4.5—Quinlan, 1993). These conventional recursive partitioning methods generate partitions of the samples with the goal of reaching a partition that generates a good prediction rule.

In Figure 10.6, on the left panel, we show a simple classification function of two variables $f(X, Y)$, which is expressed in the form of a tree rule on the right panel. The way to read the tree is "If $X \geq 2$, then class 1; if $X < 2$ and $Y \geq 2$, then class 2;...." One of the nice properties of such trees is that they produce classification rules that have a wide appeal because they resemble decision rules that are easier to understand compared to most other competitive methods.

10.5.1 Classification Trees

A *classification tree* is an easily understandable way of graphically displaying the results of a recursive partitioning procedure. For gene expression data the inputs to a classifier are gene expressions, ratios of gene expressions, or linear combinations thereof, obtained from principal components or factor analyses. The example in Figure 10.7 shows a classification tree. The process of building a classification tree has two stages: (1) building the tree, and (2) pruning the tree.

Building the Tree. A binary tree begins at a root node where the data is split into two buckets using one of the classification variables from the set. The split

Table 10.3 Results of kNN for the four rules for selecting classifiers giving the number of misclassifications for the testing samples

	10 PC of 2308 Genes		10 PC of 450 Genes		10 Cluster Means of 50 Genes		10 PC of 30 Genes	
	$K = 1, 2$	$K = 10$	$K = 1, 4, \ldots, 10$	$K = 2, 3$	$K = 1, 2$	$K = 10$	$K = 1, 2, 3$	$K = 10$
2 Classifiers	14	10	2	1	2	2	2	2
3 Classifiers	10	8	1	0	0	0	1	1
4 Classifiers	3	2	1	1	0	0	1	1
10 Classifiers	6	3	1	1	0	0	1	1

Note: The two columns show differences for using different numbers of nearest neighbors.

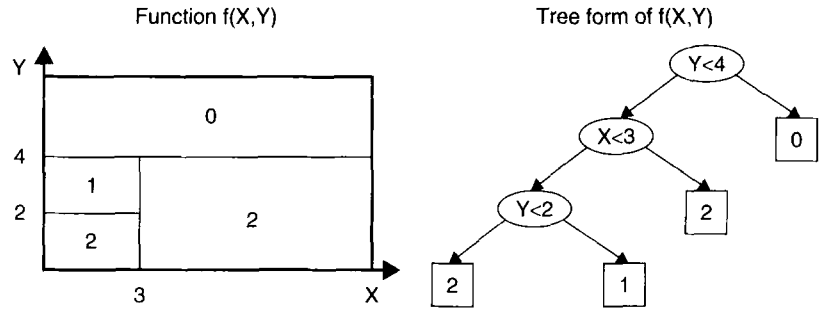


Figure 10.6 Classification tree for a function $f(X, Y)$.

is performed by optimizing one of the criteria below over all possible partitions generated by some logical condition in any of the classification variables. The conditions are of the form: $x_i > c$ goes to the right bucket and $x_i \leq c$ goes to the left bucket. This produces two new nodes, right and left (R, L), which are split into two buckets each by the same process that took place at the root node. If the size of a node is less than a predetermined constant m, then the

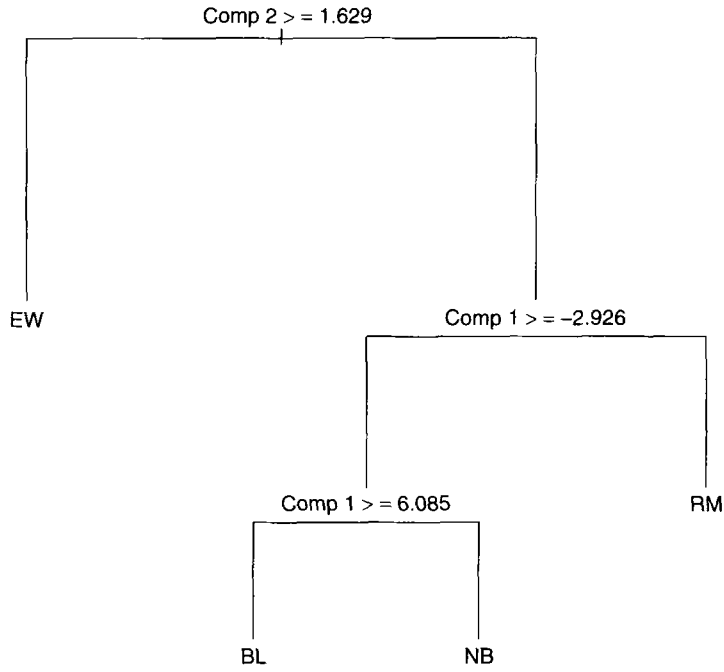


Figure 10.7 Classification tree for the cancer groups using 10 principal components of the top 100 cancer genes. The classification rule produces zero mistakes in the training set and five mistakes in the training set.

node is labeled a terminal node, and it is not split. The process continues until all nodes have either been split or are terminal nodes. The result is a binary tree that produces a partition (made up of terminal nodes).

In some microarray data analysis classification problems, we may use categorical classifiers. For example, instead of using the spot intensity X as a classifier variable, we find that the spot intensities have been categorized into variables Z reflecting that a group of one or more genes all highly express ($Z = 1$) or not all highly express ($Z = 0$). These classifiers may generate categorical splits where the categories of Z are split into two groups such that the splitting criterion is minimized.

Let us consider the case when the objective is to classify the samples into only two classes, class 0 and class 1. The objective of the tree method is to produce a partition of terminal nodes that are relatively pure in the sense that almost all the observations at a terminal node belong to only one class. The most popular criteria that have been propose to achieve this goal are as follows:

- *Gini index.* This is the criterion used in the original version of Breiman et al. (1984)'s *classification and regression tree* (CART) methodology. The objective function is

$$C_g = qq_R p_R + pq_L p_L,$$

  where p and q are the proportions of observations going to the left and right buckets, p_L and q_L are the proportions of 0's and 1's in the left-side bucket, and p_R and q_R are the proportion of 0's and 1's in the right-side bucket respectively.
- *Entropy.* This criterion is used by the C4.5/C5 algorithms by Quinlan (1993) that is very widely used in computer science applications:

$$C_e = q(-q_R \log q_R - p_R \log p_R) + p(-q_L \log q_L - p_L \log p_L)$$

- *Tree.* A deviance-based criterion proposed by Clark and Pregibon (1992) has been implemented in statistical software such as SPLUS and R:

$$C_t = q \min(q_R, p_R) + p \min(q_L, p_L).$$

A less well-known, but interesting, criterion is Lee and Buja's (1999) data-mining criterion,

$$C_{LB} = \min(p_L, p_R).$$

Many the criteria above are widely used by practitioners and the choice of a "best" criterion remains an open question.

The tree method can also be applied for regression analysis that is when there is a continuous response variable that is analyzed by a tree model. The

splitting criterion is

$$C_r = \frac{n_L \hat{\sigma}_L^2 + n_R \hat{\sigma}_R^2}{n_L + n_R},$$

where $n_R, \hat{\sigma}_R^2$, and $n_L, \hat{\sigma}_L^2$ are the number of observations and sample variance in the right and left bucket respectively.

Pruning the Tree. The tree-building step is likely to experience two kinds of problems:

1. An overfitted tree, with small buckets at the terminal nodes.
2. An oversized tree, which is hard for the practitioner to interpret.

A way to correct these problems is by pruning the tree. Since step one of the tree-building procedure produces an ordered sequence of trees, the question becomes where to stop.

One procedure to do this is *test set cross validation*, which consists of separating a portion of the data (25%–50%) that we will call *testing set* and leave the remaining part of the dataset for training. The tree is built using the training set alone and results in an ordered sequence of trees. For each tree in the sequence, a misclassification rate is estimated using the testing set alone, and the tree with the lowest misclassification rate is chosen. This procedure requires a large initial dataset, and even then, it seems unnatural to discard a part of the dataset for training since it will reduce the performance of the method. An alternative technique that improves on this is called *V-fold cross-validation*; details can be found in Breiman et al. (1984).

One weakness of classification trees is that some nodes may have few observations, making the predictions at those nodes unreliable. Techniques have been proposed to improve the predictive ability at those nodes and therefore of the tree, in general. Two such procedures are *bootstrap aggregating* or *bagging* (Breiman, 1996) and *boosting* (Freund and Schapire, 1997). The idea behind both these methods is to produce slightly perturbed classifiers from the training data by resampling. Bagging generates replicate training sets by sampling with replacement from the training set. Boosting retains all the samples but weights each sample differently, and generates different classifiers by adjusting the weights. Each method generates multiple classifiers that are combined by voting to form a composite classifier. In bagging, each component classifier has the same vote, while in boosting, each component classifier has a different vote based on an assessment of its accuracy. Dudoit et al. (2002) found that boosting and bagging improves the performance of CART-like procedures with microarray data.

Example. In Figure 10.7 we show a tree graph generated using the tree method with the four-tumor data example. The input classifiers are the two top principal components of the best 100 genes and the node splitting criterion used

was the Gini index. The method produced a tree with three splits and four terminal nodes that fits the training data perfectly and has 5 mistakes on the 20 testing samples. This performance is typical of trees with small training sets.

10.5.2 Activity Region Finding

The *activity region finding* (ARF) method for growing classification trees was proposed by Amaratunga and Cabrera (2003a). ARF trees are trees that exclude large parts of the data where no information is available and concentrate only in subsets that contain the important information. The advantage of ARF over recursive partitioning (RP) methods is that it produces simpler more condensed trees in cases where RP methods give very complex and elaborated hard to interpret trees.

Building the ARF Tree. The ARF tree is a ternary tree; that is, each node is split into three groups. The splits are of the form $c_1 \leq x_i \leq c_2$ so there are three subgroups.

The basis of the ARF approach is the H criterion. For data of the form $D = \{(Y_i, x_i)\}$, where $i = 1, \ldots, N$, Y_i is a Bernoulli variable, which is either 0 ("failure") or 1 ("success"), and $x_i = (x_{1i}, \ldots, x_{ri})'$ is an r-vector of predictor variables. The objective is to discover *high-activity regions* (HARs) ranges of values of x (i.e., subsets of the form $S = \Pi I_k$, as described above) associated with high values of $\text{Prob}(Y = 1 \mid S)$ (i.e., with high-success probabilities).

It is natural to consider that, for the kth predictor variable, x_k, an "interesting" interval, $I_k = \{a_k \leq x_k \leq b_k\}$, is one that has a substantially higher proportion of successes compared to D, meaning one such that $p(I_k) = \text{Prob}(Y = 1 \mid I_k)$ is substantially larger than $p(D) = \text{Prob}(Y = 1 \mid D)$. In order to compare $p(I_k)$ across subsets, I_k, of different sizes on an equal footing, we need a statistic that is not much dependent on $n(I_k)$, the number of observations in I_k. Such a statistic is

$$z(I_k; D) = \frac{p(I_k) - p(D)}{\sigma_p},$$

where $\sigma_p^2 = p(D)(1 - p(D))/n(I_k)$, as $z(I_k; D)$ is approximately $N(0, 1)$, regardless of sample size, except for very small samples, for a random binary series of length N with success probability $p(D)$. The larger the value of $z(I_k; D)$ is, the more interesting is I_k.

10.6 NEURAL NETWORKS

Inspired by the way the brain supposedly processes information, *neural networks* are a class of highly flexible nonlinear models for studying complex patterns in data and for predicting new observations from existing ones. Use

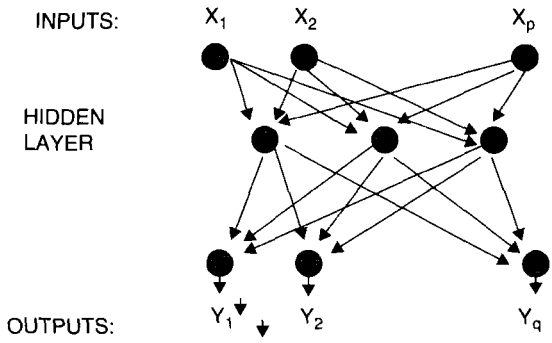


Figure 10.8 Graph of an ANN with one hidden layer.

of these models is very popular in data mining, machine learning, and other application areas related to computer science, although they are significantly less favored by statisticians. One reason is that neural networks intentionally tend to be heavily overparametrized, which goes totally against conventional statistical wisdom that identifies overparametrization with overfitting.

However, in many modern classification problems this overparametrization may not be a serious concern because the data may have the following characteristics: there is no overlap between the classes, the surface that separates the classes is highly nonlinear, and there are massive amounts of data so that the separation can be identified by the procedure. For problems with such characteristics, the risk of overfitting is less of a concern than the ability to capture highly nonlinear separations.

A basic model for neural networks appears under the name *feed-forward single hidden layer neural nets*, which consist of an input layer, an output layer and a hidden layer in between. Figure 10.8 gives a visual scheme of the structure of such neural nets. Each node has one or more inputs and one output. The neural net model is structured as follows:

1. The input layer (or first layer) consists of as many nodes as classifiers are available for the fit. The output of each node is the corresponding value of the classifiers assigned to it (see Fig. 10.8).
2. Each node inputs the outputs of the nodes of the prior layer and it outputs a fixed function of the linear combination. The function, sometimes called a *transfer function*, is usually a logistic function or any other sigmoidal shaped function for classification problems and the identity function for regression problems. The sigmoidal function is

$$h(x, y) = \tanh(\alpha_0 + \alpha_1(x'y)^2)$$

3. The output layer (or last layer) has as many nodes as responses are available for the fit.

Table 10.4 Results of ANN for the four rules for selecting classifiers giving the number of misclassifications for the testing samples

	10 PC of 2308 Genes		10 PC of 450 Genes		10 Cluster Means of 50 Genes		10 PC of 30 Genes	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
2 Classifiers	18	14	0	2	0.5	1.5	0	3.5
3 Classifiers	8	14	0	1	0	0.5	0	1.5
4 Classifiers	0	3	0	0.5	0	0.5	0	1.5
10 Classifiers	0	8	0	0.5	0	0.5	0	1.5

Note: The two columns show differences for using different numbers of nearest neighbors.

The process of estimating the parameters of the neural net ("learning") uses a very complicated "backfitting" algorithm that does not always find the optimal parameter values. However, a nice feature of the neural net model is that it can be implemented in computer hardware. It would be a mistake for statisticians to ignore neural net methodology because it does produce excellent results in many applications. The difficulties mentioned above will likely be overcome with new research and greater computational resources.

Example. We redo the classification using the `ann` (which stands for *artificial neural network*) function in the R software. We used one hidden layer with $K = 0, \ldots, 20$ nodes and found that the number of nodes made little effect. We allowed direct links between the input layer and the output layer. The results are shown in Table 10.4 are more or less the same as Table 10.3. The 0.5 in the table means that different trainings produced different results in the classification because of the differences in the initial conditions. The ANN was trained one thousand times, and the results are reported as 0.5 when about half of the time we got zero misclassifications; the other times we have 1 or more misclassifications. It appears that procedures 2 and 3 performed slightly better than procedure 4.

10.7 SUPPORT VECTOR MACHINES

Support vector machines (SVM) are generalizations of the linear classifier methods (e.g., LDA) that have become very popular in the machine learning literature.

Suppose that the training set is classified into two classes $\{+1, -1\}$, then the SVM classification rule is of the form $r(x) = \text{sign}(\beta' x - \beta_0)$. This function defines a separation hyperplane between the two classes. Some of the features that make SVM popular are as follows:

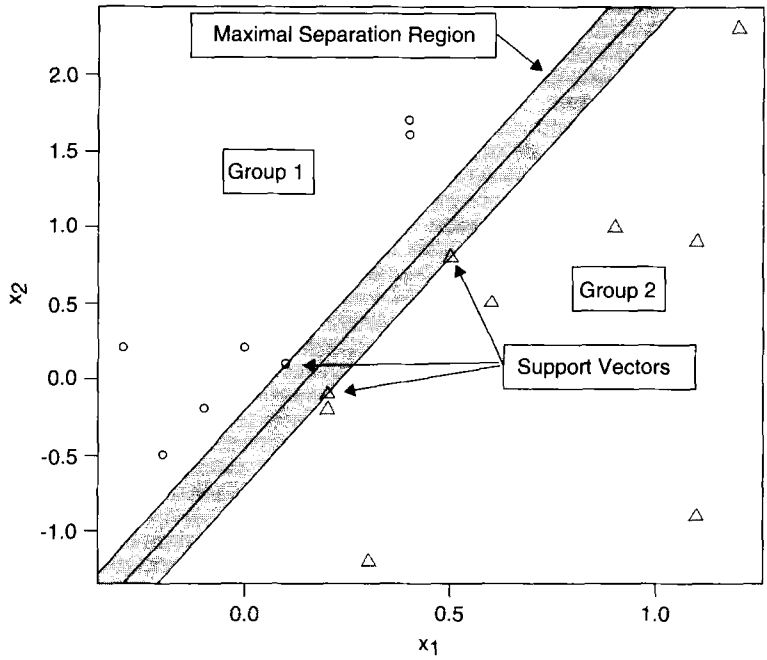


Figure 10.9 SVM example with two groups of points. The shaded area represents the separation region. The arrows indicate the location of the support vectors.

1. The criterion for estimating the hyperplane is to maximize the margin of separation between the classes, as shown in Figure 10.9, by the shaded region in the graph. This is an interesting idea, but it is not affine invariant and it may not be optimal in situations when the scales of the different classifiers may not be very similar.

2. The linear classifiers can be extended to nonlinear ones by augmenting the set of classifiers to the sometimes-called "feature" space, which includes the classifiers plus nonlinear functions of them. The form of the nonlinear classifier is

$$r(x) = \text{sign}\left(\sum_{i=1}^{p} \beta_i h(x_i, x) - \beta_0\right),$$

where the most popular forms for h are the following:

a. *Radial basis functions*,

$$h(x, y) = \exp\left(\frac{-\|x - y\|^2}{2\sigma^2}\right).$$

b. *Sigmoidal functions*,

$$h(x, y) = \tanh(\alpha_0 + \alpha_1(x'y)^2).$$

3. The estimation of the classifier rule parameters is performed using a quadratic programming algorithm. The solution can be expressed as a function of a few of the samples that are called support vectors that gives the name to the method. Figure 10.9 shows a graph of an example of SVM in two dimensions, indicating the support vectors and the maximal separation region.

SVMs have been used for analyzing microarray data. For example, Brown et al. (2000) use SVMs to predict functional roles for uncharacterized yeast ORFs.

A good general introduction to SVM can be found in a collection of four papers in Hearst (1998), where the authors give high praise to the theoretical simplicity of SVM compared to neural nets, and the reasons why SVM are becoming very popular in the area of machine learning. On the other hand, they point out that there are still no applications where these methods have been shown to be significantly superior to other nonlinear classification techniques. Our intuition of SVM is that they are different than other methods in the sense that they pay special attention to the boundary of separation between the regions corresponding to each class, and this may yield small improvements of the classification prediction rate.

Incidentally, computer-intensive nonlinear classification methods are becoming more and more popular because of the widespread availability of very fast computers, and there are many implementations of them in modern software.

10.8 INTEGRATION OF GENOMIC INFORMATION

Gene expression information from microarray studies can be integrated with information from other sources, such as annotation and partial information regarding genetic pathways, to develop more complete views of biological processes. We outline briefly some efforts in this regard.

10.8.1 Integration of Gene Expression Data and Molecular Structure Data

Blower et al. (2002) describe a method called *SAT analysis* for systematically associating molecular features of compounds with gene expression patterns, with the objective of predicting which molecular substructures would be present in drugs that are active in cells whose genes are expressed according to a specific pattern. SAT analysis links together three databases of information on cells and chemical compounds. The three databases are A, a $k \times l$ "activity matrix" consisting of experimental measures of the inhibitory effect of each of the k compounds against each of l cell lines; T, a $G \times l$ "target matrix" of gene expression patterns measured by DNA microarrays for G genes in the l cell

lines; and S, a $h \times k$ "structure matrix" of 0's and 1's that identifies which of a very large number, h, of molecular structural features are present in each of the k compounds.

If A and T are properly standardized, the $k \times G$ matrix AT^t consists of Pearson correlation coefficients, and the $k \times G$ matrix SAT^t consists of association measures whose (i, j)th value measures the tendency of the ith structural feature to occur in cell lines in which the jth gene is expressed. Mining this latter matrix via a series of targeted subsetting strategies leads to insights regarding these associations.

10.8.2 Pathway Inference

The study of co-expression of genes across a series of experimental conditions, such as time, provides an assortment of clues from which it is hoped that the genetic pathways involved in a biological process could be reconstructed. The simplest way of assessing these clues is by determining the functional classes to which co-expressing genes belong and using any knowledge gained from doing this to "fill in the blanks" whenever partial information about a biological pathway is available. More complicated assessments involve the use of Bayesian network models.

A Bayesian network model can be graphically represented as a directed graph. The nodes of the graph represent genes. Arrows connect those nodes where the expression of one gene regulates the expression of another, either directly or due to an external stimulus. The state of a daughter node conditional on the state of its parent nodes is modeled by a probability distribution. The whole model is fitted either by using a scoring function to evaluate how well the network matches the observed data or by performing tests for conditional independence on the observations. However, definitive conclusions are difficult to reach from fitting these models due to various reasons such as (1) the lack of availability of sufficient data to adequately and reliably fit the model and (2) the nonuniqueness of the fit, indicating the statistical equivalence of totally different gene regulation pathways.

As functional genomics develops, research that involves integrating genomic data from diverse sources will lead to a better understanding of complex biological processes.

SOFTWARE NOTES

Software for commonly used supervised classification techniques has been implemented in a number of different platforms and are available in all statistical packages. As with clustering algorithms, some implementations are better than others at handling large datasets and applications with large numbers of variables.

In `R` and `SPLUS` and its associated libraries, some relevant functions are `lda`

for Fisher's linear discriminant analysis, `qda` for quadratic discriminant analysis, `knn` for the nearest-neighbors method, `rpart` (`tree` in `SPLUS`) for classification trees, and `nnet` for neural networks.

Some programs specially designed for microarray data are `GeneCluster` (which does kNN) and `Partek` (which does gene filtering, discriminant analysis and neural networks):

`PAM` is available online at `www-stat.stanford.edu/~tibs/PAM`.

`ARF` is available online at `www.rci.rutgers.edu/~cabrera/DM`.

SUPPLEMENTARY READING

As with cluster analysis, the literature on class prediction is very broad. The Hand (1997) book is a useful general reference. A number of multivariate analysis textbooks also provide detailed accounts of supervised classification methodologies, including Gnanadesikan (1997), Krzanowski (2000), Mardia, Kent, and Bibby (1979), and Seber (1984). The books by Hastie, Tibshirani, and Friedman (2001) and McLachlan (1992) and Ripley (1996) lie at the interface of statistics and data mining.

EXERCISES

10.1. Golub et al. (1999) performed a supervised classification on oligonucleotide microarray data (the data is available online) related to two types of leukemia: AML and ALL. Read the article and answer the following questions:

- **a.** Briefly outline the authors' goals and the analysis they performed and the conclusion that they reached.
- **b.** Summarize the data using PCA or FA on all the genes.
- **c.** Perform a classification using (i) LDA, (ii) DLDA, (iii) QDA, and (iv) kNN using all the genes in the data. Report the misclassification rate in the training set and the test set.
- **d.** Repeat parts b and c using only the 50 genes identified in the paper as being the strongest classifiers. Report the misclassification rate in the training set and the test set.
- **e.** Compare the results of the various analyses that you performed, and draw your conclusions.

10.2. Use the Khan et al. (2001) data that was covered in this chapter and from the training set select the top 5 genes that produce the most significant F statistic for comparing the mean expression between the four tumor classes.

a. Use the LDA procedure to produce a classification rule for the four tumor classes and estimate the misclassification rate for the training and testing sets.

b. Repeat the procedure for the top 10 genes, 15 genes, 20 genes, and 25 genes. Make a table showing the misclassification rates of the 5 classification rules and comment on the table.

c. Repeat the procedure for all the previous cases, but instead of using the raw gene data, take the top two principal components as the classifiers. Make a new table of the performance of all the previous classifiers, and summarize your findings.

10.3. For the Khan et al. (2001) data in the previous problem:

a. Draw a biplot using the top 100 genes according to the F ratio criterion.

b. Once this is done, draw a biplot of the training sets and try to find clusters by hand; that is, print the biplot graph, and with a pencil draw the regions corresponding to each class.

c. Proceed by graphing the dots corresponding to the testing set, and check the misclassification rate of your clusters.

Exploration and Analysis of DNA Microarray and Protein Array Data
Edited by DHAMMIKA AMARATUNGA and JAVIER CABRERA

CHAPTER 11

Protein Arrays

Protein array experiments display strong similarities to their DNA microarray counterparts. Protein arrays are rapidly becoming established as a powerful means to detect proteins, to monitor their expression levels, to ascertain their functions, and to investigate how they interact with each other and with external effects. Although the protein array field is still in its infancy, it is gaining momentum and is fast becoming one of the most promising areas of biomedical research.

11.1 INTRODUCTION

Proteins are the workhorses that regulate and perform the main functions of the cell (see Chapter 2). They carry out all kinds of housekeeping activities, they are catalysts of chemical reactions, they act as channels and pumps, and they perform motor functions. Some of the proteins involved in protein array experiments are as follows:

- *Antibodies.* Antibodies are proteins produced by B-lymphocyte cells, which are a certain type of white blood cell. As part of the immune system, the function of an antibody is to bind with a specific protein (antigen) lying on the surface of a foreign cell. This protein-binding property plays an important role in the technology for the realization of protein array experiments. There are five classes of antibodies that are also called *immunoglobulins*: lgA, lgD, lgE, lgG, and lgM.
- *Antigens.* Antigens are proteins that lie on the surface of foreign cells and are detected by specific antibodies. Antibodies will bind with antigens in order to neutralize them and to help other parts of an organism's immune system recognize foreign cells such as bacteria or viruses.

Exploration and Analysis of DNA Microarray and Protein Array Data
By Dhammika Amaratunga and Javier Cabrera
ISBN 0-471-27398-8

- *Enzymes.* These are proteins that perform catalytic functions; that is, they accelerate a chemical reaction without been consumed by it. In particular, enzymes are involved in the synthesis of DNA and proteins. Enzymes are involved in the synthesis of proteins from RNA code by *translation*. The RNA code is subdivided into triplets of ordered nucleotides that are called *codons*. Proteins are formed of chains of amino acid molecules. There are 20 possible amino acids, and each codon codes for one specific amino acid—but more than one codon may code for the same amino acid. The process of protein formation consists of translating the RNA code into a chain of amino acids bonded together to form the protein molecule. The enzyme's role in the protein formation is similar to the role of an assembly line in the making of a product.

Although some of the basic concepts of protein arrays are covered in this chapter, some more complete general references in the genomics literature are Kodadek (2001), MacBeath (2002), and Angenendt et al. (2002).

11.2 PROTEIN ARRAY EXPERIMENTS

Theoretically a protein array experiment can be fashioned to follow a path similar to a DNA microarray experiment. A protein sample is extracted from cells or whole tissues and labeled with dye, the labeled protein sample is incubated with a prefabricated array consisting of a large number of proteins printed in high density on a glass slide, any unbound labeled protein sample is removed via a filtration process, and the array is scanned to measure the amount of bound sample protein.

However, there are some crucial differences between a protein array experiment and a DNA microarray experiment. In addition to its amino acid sequence, the three-dimensional structure a protein folds into is an essential determinant of its function. Thus the protein on the array must be folded appropriately but in such a way that the recognition sites on the protein are not obscured. Clearly, proteins cannot simply be printed onto a two-dimensional glass surface to study function as is done with DNA. Thus the technology of DNA microarrays described in Chapter 3 has to be modified considerably.

Several methods for fabricating protein arrays have been proposed. The basic idea is to bind the protein to the glass slide with some agent and to label it with a fluorescent dye that can be detected by a scanner. Three methods have been proposed for constructing protein arrays:

1. *Sandwich immunoassays.* The microarray spots are made of packed antibodies that will bind with specific proteins and then a second set of antibodies that have been labeled with a fluorescent dye will bind with the captured proteins (Fig. 11.1*a*). The scanner will read the fluorescence signal and assign a measure of protein abundance.

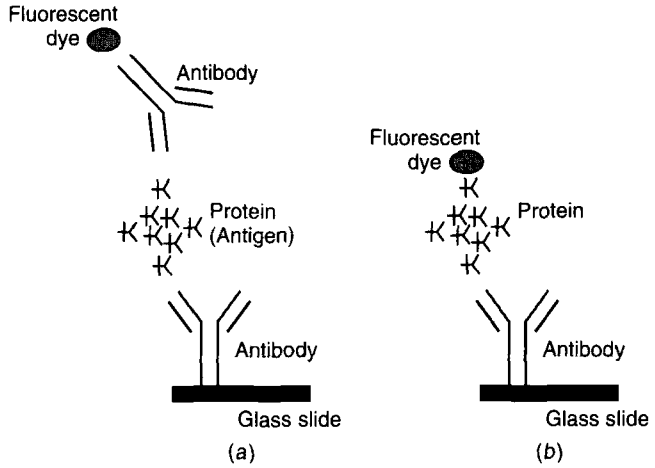


Figure 11.1 Protein array types: (*a*) sandwich method, (*b*) antigen method.

2. *Antigen capture immunoassays.* This method is similar to sandwich immunoassays except that the second antibody with the fluorescent tag is not used (Fig. 11.1*b*). Instead, it requires chemical labeling of the proteins (or some alternative method of measuring protein abundance on each spot).
3. *Antigen capture immunoassays.* This approach consists of immobilizing the protein molecules in the sample directly. An antibody labeled with fluorescent dye is used to detect any particular protein.

When two dyes are used, dye-swap designs (see Section 8.5.1) may be used as one dye may bind more efficiently with certain proteins than the other dye.

11.3 SPECIAL ISSUES WITH PROTEIN ARRAYS

Although there are many similarities between the images scanned from protein arrays and the images scanned from DNA microarrays, the processes that generated them (described in the previous section and in Chapter 3) are quite different. Some of the issues that differentiate protein arrays from their DNA siblings that affect the data analysis are as follows:

1. The objective of protein arrays is not only detection of protein but also measurement of protein abundance, whereas the objective of most DNA microarray experiments is focused on which genes are expressed or differentially expressed.
2. In DNA microarray experiments there is a PCR step that amplifies the sample. In protein arrays there is no such amplification step. For single

dye experiments, it is possible to amplify the signal by three orders of magnitude using enzyme catalyzers (Knezevic et al., 2001), but for two dye experiments, this technology has not been developed as yet. Consequently the detection level is an important issue because a protein that is present in the sample at a low concentration may not be detected by the protein array experiment.

3. Cross-detection is also an issue because some antigens may bind to more than one protein.
4. The protein population is much more diverse than the gene population and involves many more interactions. For example, there are more than two thousand proteins in the human cell controlling gene expression only. Therefore there is great potential for much larger microarrays and more complex experiments than for the DNA case. The technology for protein array spotting is advancing rapidly and will be a useful means of analyzing patterns of variation in hundreds of thousands of proteins.

Besides this application of protein detection, another role of protein arrays is to study the functions of proteins. The advantage of protein arrays for this purpose is that they are well-suited to the control conditions of experiments. A typical experiment consists of studying the interaction between two proteins. The aim is to be able to study the functionality of many proteins at once in one experiment.

11.4 ANALYSIS

Fluorescence data from protein arrays is analyzed using an analogous approach to that used with DNA microarray data. In the schematic display shown in Figure 11.2, we outline a series of steps that need to be followed for analyzing protein array data. Observe that although Figure 11.2 is a modification of the DNA microarray analysis schematic display shown in Figure 1.1, the data analysis part is essentially similar. We now review a few of the main steps here:

Step 1. *Spotting of microarray and background array.* The raw image produced by the scanner the is input data; spotted intensities and spotted background the are output data.

Step 2. *Log (or similar) transformation.* This is to remove, totally or partially, the heavy skewness of the spotted intensities.

Step 3. *Quality control.* The procedures described in Chapter 4 can be used to check the quality of the spotted arrays and the spots themselves.

Step 4. *Normalization.* Global or intensity dependent normalization among a group of arrays is applied to correct for any systematic biases in the measurement scales. With protein arrays this step should be applied with caution because it could lower the signal of some high signal spots.

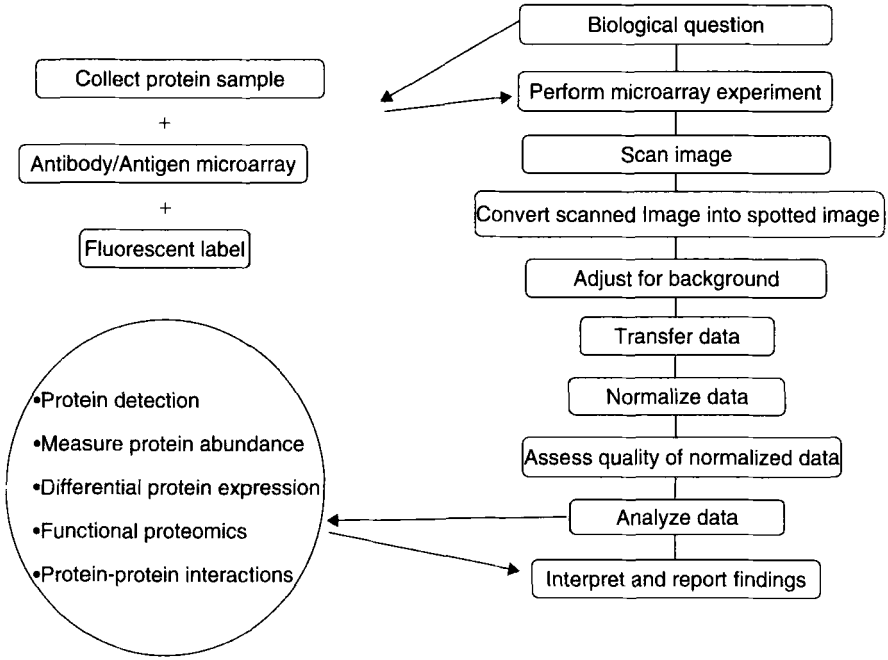


Figure 11.2 Protein array analysis schematic.

Step 5. *Outliers among the proteins.* These can be identified using the methods outlined in Chapter 5.

Step 6. *Outliers among the arrays.* These can be identified using concordance correlation coefficients and the other methods outlined in Chapter 5.

Step 7. *Analysis of the corresponding biological problem.*

11.5 USING ANTIBODY ANTIGEN ARRAYS TO MEASURE PROTEIN CONCENTRATIONS

In these early days of protein array experiments, some researchers are exploring groups of antibody/antigen pairs to show that it is possible to estimate protein concentrations using antibody/antigen microarrays. Haab et al. (2001) developed a method for protein array printing and used the arrays to measure the quantities of many specific proteins in complex solutions. They conducted a comparative fluorescence assay with two dyes, using 115 antibody/antigen pairs, with 6 to 12 replicates per pair, comprising a total of 1188 spots per microarray. In one group of 6 arrays antibodies were employed to detect their corresponding antigen pair and in another group of 6 microarrays the reverse experiment was performed, that is, using antigens to detect antibodies. The

researchers reported that 50% of the antigen arrayed and 20% of the antibody array allowed detection of proteins at some of the antibody-antigen pairs and allowed the detection of proteins at concentrations of 1 ng/ml (nanogram/milliliter). Haab et al. (2001) indicated that these sensitivities are great enough for measuring many clinically important proteins in patient blood samples.

The method used to determine when a protein was detected relayed on using 6 arrays at six different concentrations of the sample. Recall that each individual array contains six replicate spots (except for a few with up to 12) of the same antibody/antigen so for each protein we observe a 6×6 array of logged ratios. A threshold value was assigned for each protein by calculating the mean of the six replicates at the lowest concentration plus two times their standard deviation. For each of the remaining five concentrations, if all the six spots gave ratios above the threshold value, then the protein was detected at that concentration. The results of the experiment showed a 50% detection rate of arrayed antigens and a 20% of detection of arrayed antibodies at the highest concentration and lesser values for the lower concentrations.

This method of calculating the threshold is highly variable. Suppose that for simplicity, the ratios for the low concentration sample for a particular protein have a normal distribution with zero mean and standard deviation one. Then the ideal threshold would be equal to 1.96, but because we are calculating the threshold with only six values, the resulting threshold would range between 0.5 and 3.5 approximately 95% of the time.

An alternative way to check if a protein is detected by the microarray is to consider the concordance correlation between the observed values and the true protein concentration values. The concordance correlation coefficient (see Section 5.6) measures the agreement between two sets of paired numbers.

Figure 11.3 shows the histogram of the observed concordance correlation coefficient of the observed ratios and their corresponding ideal values for both

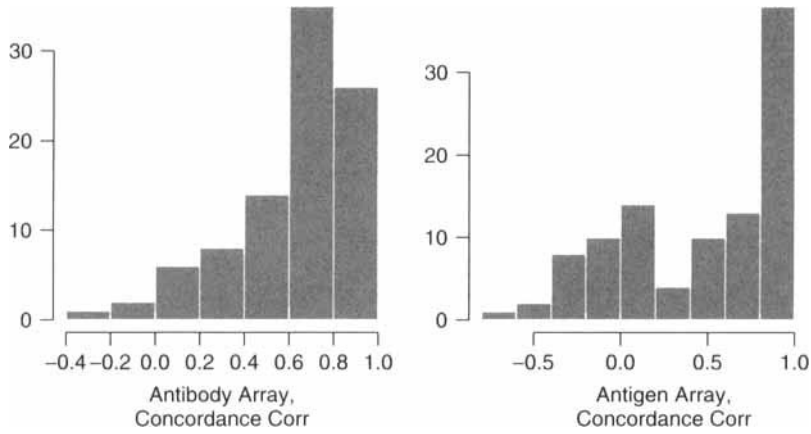


Figure 11.3 Concordance correlations for antigen and antibody arrays.

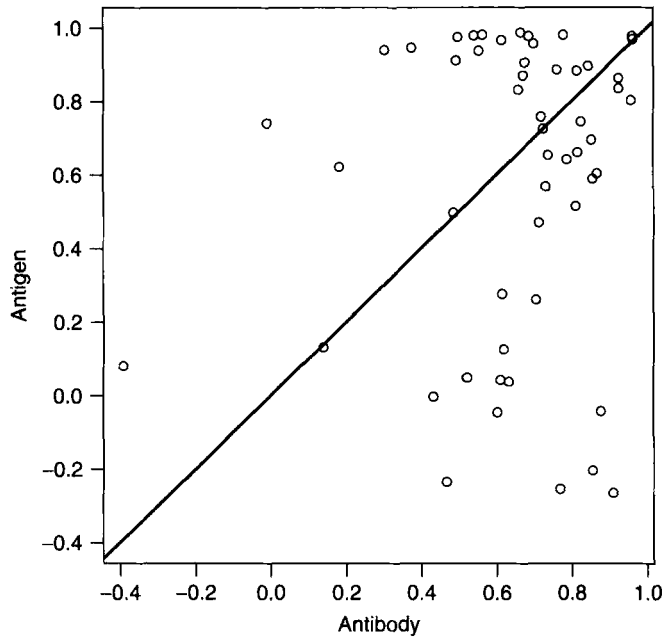


Figure 11.4 Concordance correlations for antigen arrays versus antibody arrays (with the identity line).

sets of antibody and antigen microarrays. A simple way to compare these two sets of concordance correlation coefficients is by drawing their scatterplot as in Figure 11.4. In Figure 11.4 there appear to be two distinct groups of proteins on the top right corner, some are above the diagonal line indicting that the concordance correlation coefficients are higher for the arrayed antigens and some are below the line indicting that the concordance correlation coefficients are higher for the arrayed antibodies. In order to estimate a threshold for the concordance correlation, we permuted the samples and calculated a null distribution for the concordance correlation coefficients. It turned out that the 95th percentile of the null distribution corresponded to, approximately, a 0.75 concordance, although the values differ from protein to protein. The number of detected proteins with arrayed antibodies was 59, or approximately 50%, and the number of detected proteins with arrayed antigens was 31, which is approximately 27%. These numbers appear to be different enough to suggest that the detection rates for antibody arrays are slightly higher than the detection rates for the antigen arrays.

This kind of study is only the beginning in a new period of biological research. As advances in technology propel genomics and proteomics forward, novel technologies will emerge generating fresh challenges for the sophisticated data analyst.

EXERCISES

11.1. In the analysis in Section 11.4, in the third paragraph, it was suggested that if the low concentration sample for a particular protein has a normal distribution with zero mean and standard deviation one, the ideal threshold would be equal to 1.96. However, because we are calculating the threshold with only 6 values, the resulting threshold would range between 0.5 and 3.5 approximately 95% of the time.

a. Perform a small simulation to verify this result.

b. Repeat the procedure assuming that the distribution of the low concentration ratios is a chi-squared distribution with 2 degrees of freedom.

11.2. The dataset E11 in the DNAMR library consists of 12 samples containing 1200 spots corresponding to 200 proteins with 6 replicates each. The first 6 samples are technical replicates at a concentration of 1 ng/ml. The second set of 6 samples are also technical replicates but spotted at a concentration of 10 ng/ml. The objective is to determine which proteins are detected in the sense of being differentially expressed between both groups. Carry out this analysis making sure that you follow the basic analysis steps and use the quantile normalization option.

References

Agresti, A. (2002). *Categorical Data Analysis*, 2nd ed. New York: Wiley.

Alberts, B., D. Bray, J. Lewis, M. Raff, K. Roberts, and J. Watson (1994). *Molecular Biology of the Cell*. Reading, MA: Addison-Wesley.

Aldenderfer, M. S., and R. K. Blashfield (1984). *Cluster Analysis*. London: Sage.

Allison, D. B., G. L. Gadbury, M. Heo, J. R. Fernández, C. K. Lee, T. A. Prolla, and R. Weindruch (2002). A mixture model approach for the analysis of microarray gene expression data. *Comput. Stat. Data Anal.*, **39**, 1–20.

Alizadeh, A., A. Eisen, M. B. Davis, R. E. Ma, C. Lossos, I. S. Rosenwald, A. Boldrick, J. C. Sabet, H. Tran, T. Yu, X. Powell, J. I. Yang, L. Marti, G. E. Moore, T. Hudson, J. Lu Jr., L. Lewis, D. B. Tibshirani, R. Sherock, G. Chan, W. C. Greiner, T. C. Weisenburger, D. D. Armitage, J. O. Warnke, R. Levy, R. Wilson, W. Grever, M. R. Byrd, J. C. Botstein, D. P. O. Brown, and L. M. Staudt (2000). Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, **403**, 503–511.

Alon, U., N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Nat. Acad. Sci.*, **96**, 6745–6750.

Alter, O., P. O. Brown, and D. Botstein (2000). Singular value decomposition for genome-wide expression data processing and modeling. *Proc. Nat. Acad. Sci.*, **97**, 10101–10106.

Amaratunga, D., and J. Cabrera (2001a). Outlier resistance, standardization and modeling issues for DNA microarray data. In L. T. Fernholz, S. Morgenthaler, and W. Stahel, eds., *Statistics and Genetics for the Envirnmental Sciences*, Basel: Birkhauser Verlag.

Amaratunga, D., and J. Cabrera (2001b). Statistical analysis of viral microchip data. *J. Am. Stat. Assoc.*, **96**, 1161–1170.

Amaratunga, D., and J. Cabrera (2003a). Mining data to find subsets of high activity. *J. Stat. Plan. Infer.*, forthcoming.

Exploration and Analysis of DNA Microarray and Protein Array Data
By Dhammika Amaratunga and Javier Cabrera
ISBN 0-471-27398-8

Amaratunga, D., and J. Cabrera (2003b). Methods for assessing the quality of DNA microarrays. Unpublished manuscript.

Amaratunga, D., and J. Cabrera (2003c). Conditional *t*. Unpublished manuscript.

Amaratunga, D., and J. Cabrera (2003d). A robust Bayes analysis of DNA microarray data. Unpublished manuscript.

Ambroise, C., and G. J. McLachlan (2002). Selection bias in gene extraction on basis of microarray gene expression data. *Proc. Nat. Acad. Sci.*, **99**, 6562–6566.

Angenendt, P., J. Glokler, D. Murphy, H. Lehrach, and D. J. Cahill (2002). Toward optimized antibody microarrays: A comparison of current microarray support materials. *Anal. Biochem.*, **309**, 253–260.

Anscombe, F., and J. W. Tukey (1963). The examination and analysis of residuals. *Technomet.*, **5**, 141–160.

Asimov, D. (1985). The grand tour: A tool for viewing multidimensional data. *SIAM J. Sci. Stat. Comput.*, **6**, 128–143.

Astrand, M. (2001). Normalizing oligonucleotide arrays. Unpublished manuscript.

Baldi, P., and G. W. Hatfield (2002). *DNA Microarrays and Gene Expression.* Cambridge University Press.

Baldi, P., and A. D. Long (2001). A Bayesian framework for the analysis of microarray expression data: Regularized *t*-test and statistical inferences of gene changes. *Bioinform.*, **7**, 509–519.

Banfield, J. D., and A. E. Raftery (1993). Model-based Gaussian and non-Gaussian clustering. *Biomet.*, **49**, 803–821.

Barash, Y., and N. Friedman (2002). Context-specific Bayesian clustering for gene expression data. *J. Comput. Biol.*, **9**, 169–191.

Barnett, V., ed. (1981). *Interpreting Multivariate Data.* New York: Wiley.

Barnett, V., and T. Lewis (1994). *Outliers in Statistical Data*, 3rd ed. New York: Wiley.

Bassett, D. E., M. B. Eisen, and M. S. Boguski (1999). Gene expression informatics—It's all in your mine. *Nature Genet. Suppl.*, **21**, 51–55.

Ben-Dor, A., R. Shamir, and Z. Yakhini (1999). Clustering gene expression patterns. *J. Comput. Biol.*, **6**, 281–297.

Ben-Hur, A., A. Elisseeff, and I. Guyon (2002). A stability-based method for discovering structure in clustered data. *Pacific Symp. Biocomputing*, 6–17.

Benjamini, Y., and Y. Hochberg (1995). Controlling the False Discovery Rate: A practical and powerful approach to multiple testing. *J. Roy. Stat. Soc.*, **B57**, 289–300.

Bittner, M., P. Meltzer, Y. Chen, Y. Jiang, E. Seftor, M. Hendrix, M. Radmacher, R. Simon, Z. Yakhini, A. Ben-Dor, N. Sampas, E. Dougherty, E. Wang, F. Marincola, C. Gooden, J. Lueders, A. Glatfelter, P. Pollock, J. Carpten, E. Gillanders, D. Leja, K. Dietrich, C. Beaudry, M. Berens, D. Alberts, V. Sondak, N. Hayward, and J. Trent (2000). Molecular classification of cutaneous malignant melanoma by gene expression profiling. *Nature*, **406**, 536–540.

Blower, P. E., C. Yang, M. A. Fligner, J. S. Verducci, L. Yu, S. Richman, and J. N. Weinstein (2002). Pharmacogenomic analysis: Correlating molecular substructure classes with microarray gene expression data. *Pharmacogenom. J.*, **2**, 259–271.

Bo, T. H., and I. Jonassen (2002). New feature subset selection procedures for classification of expression profiles. *Genome Biol.*, **3**, research 0017.1–0017.11.

Bolstad, B. M., R. A. Irizzary, M. Astrand, and T. P. Speed (2002). A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. Unpublished manuscript.

Bouton, C. M., and J. Pevsner (2000). DRAGON: Database referencing of array genes online. *Bioinform.*, **16**, 1038–1039.

Bouton, C. M., and J. Pevsner (2002). DRAGON View: Information visualization for annotated microarray data, *Bioinform.*, **18**, 323–324.

Brazma, A., and J. Vilo (2000). Gene expression data analysis. *FEBS Lett.*, **480**, 17–24.

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone (1984). *Classification and Regression Trees.* Monterey, CA: Wadsworth.

Brillinger, D. R., L. T. Fernholz, and S. Morgenthaler (eds.). (1997). *The Practice of Data Analysis.* Princeton: Princeton University Press.

Broberg, P. (2002). Ranking genes with respect to differential expression. *Genome Biol.*, **3**, preprint 0007.1–preprint 0007.23.

Brown, C. S., P. C. Goodwin, and P. K. Sorger (2001). Image metrics in the statistical analysis of DNA microarray data. *Proc. Nat. Acad. Sci.*, **98**, 8944–8949.

Brown, M. P., W. N. Grundy, D. Lin, N. Cristianini, C. W. Sugnet, T. S. Furey, M. Ares Jr., and D. Haussler (2000). Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc. Nat. Acad. Sci.*, **97**, 262–267.

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

Bryan, J., K. S. Pollard, and M. J. van der Laan (2002). Paired and unpaired comparison and clustering with gene expression data. *Stat. Sinica*, **12**, 87–110.

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining Knowledge Disc.*, **2**, 121–167.

Cabrera, J., and L. T. Fernholz (1999). Target estimation for bias and mean square reduction. *An. Stat.*, **27**, 1080–1104.

Cabrera, J. and A. Lo (2003). Multivariate Chinese restaurant clustering. Unpublished manuscript.

Cabrera, J. and A. McDougall (2002). *Statistical Consulting.* New York: Springer.

Cabrera, J. and G. S. Watson (1997). Simulation methods for mean and median bias reduction. *J. Stat. Plan. Infer.*, **57**, 143–152.

Calinski, T., and J. Harabasz (1974). A dendrite method for cluster analysis. *Commun. Stat.*, **3**, 1–27.

Causton, H. C., J. Quackenbush, and A. Brazma (2003). *Microarray Gene Expression Data Analysis: A Beginner's Guide.* Blackwell Publishing.

Chambers, J., A. Angulo, D. Amaratunga, H. Guo, Y. Jiang, J. S. Wan, A. Bittner, K. Frueh, M. R. Jackson, P. A. Peterson, M. G. Erlander, and P. Ghazal (1999). DNA microarrays of the complex human cytomegalovirus genome: Profiling kinetic class with drug sensitivviral gene expression. *J. Virol.*, **73**, 5757–5766.

Chambers, J. M., W. S. Cleveland, B. Kleiner, and P. A. Tukey (1983). *Graphical Methods for Data Analysis.* Boston: Duxbury Press.

Chapman, S., P. Schenk, K. Kazan, and J. Manners (2002). Using biplots to interpret gene expression patterns in plants. *Bioinform.*, **18**, 202–204.

Chen, Y., E. D. Dougherty, and M. L. Bittner (1997). Ratio-based decisions and the quantitative analysis of cDNA microarray images. *J. Biomed. Opt.*, **2**, 364–374.

Chu, S., J. DeRisi, M. Eisen, J. Mulholland, D. Botstein, P. O. Brown, and I. Herskowitz (1998). The transcriptional program of sporulation in budding yeast. *Science*, **282**, 699–705.

Chu, T.-M., B. Weir, and R. Wolfinger (2002a). A systematic statistical linear modeling approach to oligonucleotide array experiments. *Math. Biosci.*, **176**, 35–51.

Chu, T.-M., B. Weir, and R. Wolfinger (2002b). Comparison of Li-Wong and loglinear mixed models for the statistical analysis of oligonucleotide arrays. Unpublished manuscript.

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

Churchill, G. A., and B. Oliver (2001). Sex, flies, and microarrays. *Nature Genet.*, **29**, 355–356.

Clark, D., and L. Russell (1997). *Molecular Biology Made Simple and Fun*. Vienna, IL: Cache River Press.

Clark, L. A., and D. Pregibon (1992). Tree-based models. In J. Chambers and T. J. Hastie, eds., *Statistical Models in S*. Monterey, CA: Wadsworth.

Clark, P. J., and F. C. Evans (1954). Distance to nearest neighbor as a measure of spatial relationships in populations. *Ecology*, **35**, 445–453.

Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots. *J. Am. Stat. Assoc.*, **74**, 829–836.

Cochran, W. G., and G. M. Cox (1992). *Experimental Designs*. New York: Wiley.

Colantuoni, C., S. Zeger, and J. Pevsner (2002). Local mean normalization of microarray element signal intensities across an array surface: Quality control and correction of spatically systematic hybridization artifacts. *Biotechn.*, **32**, 1316–1320.

Cook, D., A. Buja, and J. Cabrera (1993). Projection pursuit indices based on orthogonal function expansions. *J. Comput. Graph. Stat.*, **2**, 225–250.

Cook, D., A. Buja, J. Cabrera, and C. Hurley (1995). Grand tour and projection pursuit. *J. Comput. Graph. Stat.*, **4**, 155–172.

Coombes, K. R. (2002). PCANOVA: Combining principal components with analysis of variance to assess group structure. Unpublished manuscript.

Cormack, R. M. (1971). A review of classification. *J. Roy. Stat. Soc.*, **A134**, 321–367.

Cox, D. R., and D. Hinkley (1974). *Theoretical Statistics*. London: Chapman and Hall.

Cui, X., M. K. Kerr, and G. A. Churchill (2002). Data transformation for cDNA microarray data. Unpublished manuscript.

D'haeseleer, P., S. Liang, and R. Somogyi (2000). Genetic network inference: From co-expression clustering to reverse engineer. *Bioinform.*, **16**, 707–726.

Daniel, C., and F. S. Wood (1971). *Fitting Equations to Data*. New York: Wiley.

Debouck, C., and P. N. Goodfellow (1999). DNA microarrays in drug discovery and development. *Nature Genet. Suppl.*, **21**, 48–50.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Stat. Soc.*, **B39**, 1–38.

DeRisi, J. L., V. R. Iyer, and P. O. Brown (1997). Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, **278**, 680–686.

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

Draghici, S., A. Kuklin, B. Hoff, and S. Shams (2001). Experimental design, analysis of variance and slide quality assessment in gene expression arrays. *Curr. Opin. Drug Disc. Devel.*, **4**, 332–337.

Dudoit, S., and J. Fridlyand (2002). A prediction-based resampling method to estimate the number of clusters in a dataset. *Genome Biol.*, **3**, 0036.1–0036.21.

Dudoit, S., J. Fridlyand, and T. Speed (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. *J. Am. Stat. Assoc.*, **97**, 77–87.

Dudoit, S., Y. H. Yang, M. C. Callow, and T. P. Speed (2002). Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. *Stat. Sinica*, **12**, 111–140.

Duggan, D. J., M. Bittner, Y. Chen, P. Meltzer, and J. M. Trent (1999). Expression profiling using cDNA microarrays. *Nature Genet. Suppl.*, **21**, 10–14.

Durbin, B., J. Hardin, D. Hawkins, and D. M. Rocke (2002). A variance-stabilizing transformation for gene expression microarray data. Unpublished manuscript.

Efron, B. (2001). Robbins, empirical Bayes, and microarrays. *Technical Report of the Stanford University Department of Statistics.*

Efron, B., and R. Tibshirani (1991). Statistical analysis in the computer age. *Science*, **253**, 390–395.

Efron, B., and R. Tibshirani (1993). *An Introduction to the Bootstrap*. London: Chapman and Hall.

Efron, B., and R. Tibshirani (2002). Empirical Bayes methods and false discovery rates for microarrays. *Genet. Epidemiol.*, **23**, 70–86.

Efron, B., R. Tibshirani, J. D. Storey, and V. Tusher (2001). Empirical Bayes analysis of a microarray experiment. *J. Am. Stat. Assoc.*, **96**, 1151–1160.

Efron, B., J. D. Storey, and R. Tibshirani (2001). Microarrays, empirical Bayes methods, and false discovery rates. *Technical Report of the Stanford University Department of Statistics.*

Eisen, M. B., P. T. Spellman, P. O. Brown, and D. Botstein (1998). Cluster analysis and display of genome-wide expression patterns. *Proc. Nat. Acad. Sci.*, **95**, 14863–14868.

Everitt, B. S. (1993). *Cluster Analysis*, 3rd ed. London: Halsted Press.

Ewens, W. J., and G. R. Grant (2001). *Statistical Methods in Bioinformatics: An Introduction*. New York: Springer Verlag.

Fayyad, U. M., G. Piatetsky-Shapiro, and P. Smyth, eds. (1996). *Advances in Knowledge Discovery and Data Mining*. Cambridge: MIT Press.

Fellenberg, K., N. Hauser, B. Brors, A. Neutzner, J. Hoheisel, and M. Vingron (2001). Correspondence analysis applied to microarray data. *Proc. Nat. Acad. Sci.*, **98**, 10781–10786.

Fernholz, L. T., S. Morgenthaler, and W. Stahel, eds. (2001). *Statistics in Genetics and in the Environmental Sciences*. Basel: Birkhauser-Verlag.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *An. Eugen.*, **7**, 179–188.

Fisher, R. A. (1951). *The Design of Experiments*, 6th ed. London: Oliver and Boyd.

Fix, E., and J. Hodges (1951). Discriminatory analysis. Nonparametric discrimination: Consistency properties. *Technical Report of the USAF School of Aviation Medicine*, Randolph Field, TX.

Fraley, C., and A. E. Raftery (1998). How many clusters? Which clustering method? Answers via model-based cluster analysis. *Comput. J.*, **41**, 578–588.

Freund, Y., and R. E. Schapire (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, **55**, 119–139.

Friedman, J. H. (1987). Exploratory projection pursuit. *J. Am. Stat. Assoc.*, **82**, 249–266.

Friedman, J. H. (1989). Regularized discriminant analysis. *J. Am. Stat. Assoc.*, **84**, 165–175.

Friedman, J. H. (1994). Flexible metric nearest neighbor classification. Unpublished manuscript.

Friedman, J. H., and J. J. Meulman (2002). Clustering objects on subsets of attributes. Unpublished manuscript.

Friedman, J. H., and W. Stuetzle (1981). Projection pursuit regression. *J. Am. Stat. Assoc.*, **76**, 817–823.

Friedman, J. H., and J. W. Tukey (1974). A projection pursuit algorithm for exploratory data analysis. *IEEE Trans. Comput.*, **23**, 881–890.

Gabriel, K. R. (1971). The biplot graphical display of matrices with applications to principal component analysis. *Biometrika*, **58**, 453–467.

Gabriel, K. R., and C. L. Odoroff (1990). Biplots in biomedical research. *Stat. Med.*, **9**, 469–485.

Getz, G., E. Levine, and E. Domany (2000). Coupled two-way clustering analysis of gene microarray data. *Proc. Nat. Acad. Sci.*, **97**, 12079–12084.

Ghosh, D. (2002). Singular value decomposition regression models for classification of tumors from microarray experiments. *Pacific Symp. Biocomput.*, **7**, 18–29.

Gibson, G. (2002). Microarrays in ecology and evolution: A preview. *Ecology*, **11**, 17–24.

Glasbey, C. A., and P. Ghazal (2002). Combinatorial image analysis of DNA microarray features. Unpublished manuscript.

Gnanadesikan, R. (1997). *Statistical Data Analysis of Multivariate Observations*, 2nd ed. New York: Wiley.

Gnanadesikan, R., and J. R. Kettenring (1989). Discriminant analysis and clustering. *Stat. Sci.*, **4**, 34–69.

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

Gonick, L., and M. Wheelis (1991). *A Cartoon Guide to Genetics*. New York: Harper Collins.

Gordon, A. D. (1999). *Classification*. Boca Raton, FL: Chapman and Hall/CRC.

Haab, B., M. Dunham, and P. Brown (2001). Protein microarrays for highly parallel detection and quantitation of specific proteins and antibodies in complex solutions. *Genome Biology*, **2**, research 00004.1–00004.13.

Hall, P. (1989). Polynomial projection pursuit. *An. Stat.*, **17**, 589–605.

Hamadeh, H., and C. A. Afshari (2000). Gene chips and functional genomics. *Am. Scientist*, **88**, 508–515.

Hand, D. J. (1997). *Construction and Assessment of Classification Rules.* New York: Wiley.

Hartemink, A. J., D. K. Gifford, T. S. Jaakkola, and R. A. Young (2001). Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. *Pacific Symp. Biocomput.*

Hartigan, J. A. (1972). Direct clustering of a data matrix. *J. Am. Stat. Assoc.*, **67**, 123–129.

Hartigan, J. A. (1975). *Clustering Algorithms.* New York: Wiley.

Hastie, T., R. Tibshirani, and A. Buja (1994). Flexible discriminant analysis, *J. Am. Stat. Assoc.*, **89**, 1255–1270.

Hastie, T., R. Tibshirani, M. B. Eisen, A. Alizadeh, R. Levy, L. Staudt, W. C. Chan, D. Botstein, and P. Brown (2000). "Gene shaving" as a method for identifying distinct sets of genes with similar expression patterns. *Genome Biol.*, **1**, research 0003.1–0003.21.

Hastie, T., R. Tibshirani, and J. Friedman (2001). *Elements of Statistical Learning: Data Mining, Inference and Prediction.* New York: Springer Verlag.

Hawkins, D. M., and G. V. Kass (1982). Automatic interaction detection. In D. M. Hawkins, ed., *Topics in Multivariate Analysis.* Cambridge: Cambridge University Press.

Hearst, M. (1998). SVM—Trends and controversies. *IEEE Intell. Syst.* 13, **18.**

Hedenfalk, I., D. Duggan, Y. Chen, M. Radmacher, M. Bittner, R. Simon, P. Meltzer, B. Gusterson, M. Esteller, O. P. Kallioniemi, B. Wilfond, A. Borg, and J. Trent (2001). Gene-expression profiles in hereditary breast cancer. *New Eng. J. Med.*, **344**, 539–548.

Herrero, J., A. Valencia, and J. Dopazo (2001). A hierarchical unsupervised growing neural network for clustering gene expression patterns. *Bioinform.*, **17**, 126–136.

Hill, A. A., E. L. Brown, M. Z. Whitley, G. Tucker-Kellogg, C. P. Hunter, and D. K. Slonim (2001). Evaluation of normalization procedures for oligonucleotide array data based on spiked cRNA controls. *Genome Biol.*, **2(12)**, research 0055.1–0055.13.

Hoaglin, D. C. (1982). Exploratory data analysis. In Kotz, S., N. L. Johnson, and C. B. Read, eds., *Encyclopedia of Statistical Sciences, Vol. 2*, 579–583. New York: Wiley.

Hoaglin, D. C., F. Mosteller, and J. W. Tukey (1983). *Understanding Robust and Exploratory Data Analysis.* New York: Wiley.

Hochberg, Y. (1988). A sharper Bonferroni procedure for multiple tests of significance. *Biometrika*, **75**, 800–803.

Hoffmann, R., T. Seidl, and M. Dugas (2002). Profound effect of normalization on detection of differentially expressed genes in oligonucleotide microarray data analysis. *Genome Biol.*, **3**, research 0033.1–0033.11.

Holder, D., V. Pikounis, R. Raubertas, V. Svetnik, and K. Soper (2001). Statistical analysis of high density oligonucleotide arrays: A SAFER approach. Unpublished manuscript.

Hollander, M., and D. A. Wolfe (1999). *Nonparametric Statistical Methods*, 2nd ed. New York: Wiley.

Holloway, A. J., R. K. Van Laar, R. W. Tothill, and D. D. Bowtell (2002). Options available—from start to finish—for obtaining data from DNA microarrays II. *Nature Genet.*, **32**, 481–489.

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scan. J. Stat.*, **6**, 65–70.

Holmes, I., and W. J. Bruno (2000). Finding regulatory elements using joint likelihoods for sequence and expression profile data. In R. Altman et al., eds., *Proc. Eighth An. Int. Conf. Intelligent Systems for Molecular Biology*, La Jolla, CA. 202–210. AAAI Press.

Huber, W., A. V. Heydebreck, H. Sültmann, A. Poustka, and M. Vingron (2002). Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinform.*, **18**, 1–9.

Ibrahim, J. G., M. H. Chen, and R. J. Gray (2002). Bayesian models for gene expression with DNA microarray data. *J. Am. Stat. Assoc.*, **97**, 88–99.

Ihaka, R., and R. Gentleman (1996). R: A language for data analysis and graphics. *J. Comput. Graph. Stat.*, **5**, 299–314.

Irizarry, R. A., B. Hobbs, F. Collin, Y. D. Beazer-Barclay, K. J. Antonellis, U. Scherf, and T. P. Speed (2002). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. Unpublished manuscript.

Iyer, V. R., M. B. Eisen, D. T. Ross, G. Schuler, T. Moore, J. C. F. Lee, J. M. Trent, L. M. Staudt, D. Shalon, D. Botstein, and P. O. Brown (1999). The transcriptional program in the response of human fibroblasts to serum. *Science*, **283**, 83–87.

Jain, A. N., T. A. Tokuyasu, A. M. Snijders, R. Segraves, D. G. Albertson, and D. Pinkel (2002). Fully automatic quantification of microarray image data. *Genome Res.*, **12**, 325–332.

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

Kaufman, L., and P. J. Rousseeuw (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. New York: Wiley.

Kerr, K., C. A. Afshari, B. Lee, P. Bushel, J. Martinez, N. J. Walker, and G. A. Churchill (2002). Statistical analysis of a gene expression microarray experiment with replication. *Stat. Sinica*, **12**, 203–218.

Kerr, M. K., and G. A. Churchill (2000). Bootstrapping cluster analysis: Assessing the reliability of conclusions from microarray experiments. *Proc. Nat. Acad. Sci.*, **98**, 8961–8965.

Kerr, M. K., and G. A. Churchill (2001a). Statistical design and the analysis of gene expression microarray data. *Genetics Res. Cambridge*, **77**, 123–128.

Kerr, M. K., and G. A. Churchill (2001b). Experimental design for gene expression microarrays. *Biostat.*, **2**, 183–202.

Kerr, M. K., M. Martin, and G. A. Churchill (2000). Analysis of variance for gene expression microarray data. *J. Comput. Biol.*, **7**, 819–837.

Khan, J., J. S. Wei, M. Ringner, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson, and P. S. Meltzer (2001). Classification

and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Med.*, **7**, 673–679.

Knezevic, V., C. Leethanakul, V. E. Bichsel, J. M. Worth, V. V. Prabhu, J. S. Gutkind, L. A. Liotta, P. J. Munson, E. F. Petricoin, and D. B. Krizman (2001). Proteomic profiling of the cancer microenvironment by antibody arrays. *Proteomics*, **1**, 1271–1278.

Knudsen, S. (2002). *A Biologist's Guide to Analysis of DNA Microarray Data.* New York: Wiley.

Kodadek, T. (2001). Protein microarrays: Prospects and problems. *Chem. Biol.*, **8**, 105–115.

Kohonen, T. (1995). *Self-Organizing Maps.* Berlin: Springer Verlag.

Kothapalli, R., S. J. Yoder, S. Mane, and T. P. Loughran Jr. (2002). Microarray results: How accurate are they? *BMC Bioinform.*, **3**, 22.

Krzanowski, W. J. (1992). Ranking principal components to reflect group structure. *J. Chemomet.*, **6**, 97–102.

Krzanowski, W. J. (2000). *Principles of Multivariate Analysis: A User's Perspective*, 2nd ed. Oxford: Oxford University Press.

Krzanowski, W. J., and Y. T. Lai (1985). A criterion for determining the number of groups in a data set using sum of squares clustering. *Biometrics*, **44**, 23–34.

Kuklin, A., A. Petrov, and S. Shams (2001). Quality control in microarray image analysis. *GIT Imag. Microscopy*, **1**, 2–3.

Lander, E. S. (1999). Array of hope. *Nature Genet. Suppl.*, **21**, 3–4.

Landgrebe, J., W. Wurst, and G. Welzl (2002). Permutation validated principal components analysis of microarray data. *Genome Biol.*, **3**, 1–11.

Lashkari, D. A., J. L. DeRisi, J. H. McCusker, A. F. Namath, C., Gentile, S. Y. Hwang, Y., P. O. Brown, and R. W. Davis (1997). Yeast microarrays for genome wide parallel genetic and gene expression analysis. *Proc. Nat. Acad. Sci.*, **94**, 13057–13062.

Lazzeroni, L., and A. B. Owen (2002). Plaid models for gene expression data. *Stat. Sinica*, **12**, 61–86.

Lee, M. L. T., W. Lu, G. A. Whitmore, and D. Beier (2002). Models for microarray gene expression data. *J. Biopharm. Stat.*, **12**, 1–19.

Lee, M. L. T., F. C. Kuo, G. A. Whitmore, and J. Sklar (2000). Importance of replication in microarray gene expression studies: Statistical methods and evidence from repetitive cDNA hybridizations. *Proc. Nat. Acad. Sci.*, **97**, 9834–9839.

Lee, Y. S., and A. Buja (1999). Data mining criteria for tree-based regression and classification. Unpublished manuscript.

Lemon, W. J., J. J. T. Palatini, R. Krahe, and F. A. Wright (2001). Theoretical and experimental comparisons of gene expression indexes for oligonucleotide arrays. Unpublished manuscript.

Lennon, G. G. (2000). High-throughput gene expression analysis for drug discovery. *Drug Disc. Today*, **5**, 59–66.

Lewi, P. J. (1976). Spectral mapping, a technique for classifying biological activity profiles of chemical compounds. *Arzneimttel Forsch. (Drug Res.)*, **26**, 1295–1300.

Li, C., and W. H. Wong (2001a). Model-based analysis of oligonucleotide arrays: model validation, design issues and standard error application. *Genome Biol.*, **2(8)**, research 0032.1–0032.11.

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

Li, W., and M. Xiong (2002). Tclass: Tumor classification system based on gene expression profile. *Bioinform.*, **18**, 325–326.

Liang, S., S. Fuhrman, and R. Somogyi (1998). REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. *Pacific Symp. Biocomputing*, **98**, 18–29.

Lin, L. I.-K. (1989). A concordance correlation coefficient to evaluate reproducibility. *Biometrics*, **45**, 255–268.

Lin, S. M., and K. F. Johnson, eds. (2002). *Methods of Microarray Data Analysis: Papers from CAMDA 2000.* Dordrecht: Kluwer Academic.

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

Lo, A. Y., L. J. Brunner, and A. T. Chan (2000). Weighted Chinese restaurant processes and bayesian mixture models. Unpublished manuscript.

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

Lockhart, D., H. Dong, M. Byrne, M. Follettie, M. Gallo, M. Chee, M. Mittmann, C. Wang, M. Kobayashi, H. Horton, and E. Brown (1996). Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nature Biotechn.*, **14**, 1675–1680.

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

Lukashin, A. V., and R. Fuchs (2001). Analysis of temporal gene expression profiles: Clustering by simulated annealing and determining the optimal number of clusters. *Bioinform.*, **17**, 405–414.

MacBeath, G. (2002). Protein microarrays and proteomics. *Nature Genet.*, **32**, 526–532.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. *Proc. Fifth Berkeley Symp. Mathematical Statistics and Probability*, **1**, 281–297.

Manly, B. F. J. (1992). *Randomization and Monte Carlo Methods in Biology.* New York: Chapman and Hall.

Mardia, K. V., J. T. Kent, and J. M. Bibby (1979). *Multivariate Analysis.* London: Academic Press.

McCulloch, C. E., and S. R. Searle (2001). *Generalized, Linear and Mixed Models.* New York: Wiley.

McLachlan, G. J. (1992). *Discriminant Analysis and Statistical Pattern Recognition.* New York: Wiley.

McLachlan, G. J., R. W. Bean, and D. Peel (2002). A mixture model-based approach to clustering of microarray expression data. *Bioinform.*, **18**, 413–422.

Miller, R. (1986). *Beyond ANOVA, Basics of Applied Statistics.* New York: Wiley.

Morgan, J. N., and J. A. Sonquist (1963). Problems in the analysis of survey data and a proposal. *J. Am. Stat. Assoc.*, **58**, 415–434.

Mosteller, F., and J. W. Tukey (1977). *Data Analysis and Regression.* Reading, MA: Addison-Wesley.

Nadon, R., and J. Shoemaker (2002). Statistical issues with microarrays: Processing and analysis. *Trends Genet.*, **18**, 265–271.

Naef, F., D. A. Lim, N. Patil, and M. Magnasco (2001). From features to expression: High-density oligonucleotide arrays analysis revisited. *Proc. DIMACS Workshop on Analysis of Gene Expression Data.*

Newton, M. A., C. M. Kendziorski, C. S. Richmond, F. R. Blattner, and K. W. Tsui (2001). On differential variability of expression ratios: Improving statistical inference about gene expression changes from microarray data. *J. Comput. Biol.*, **8**, 37–52.

Nguyen, D. V., A. B. Arpat, N. Wang, and R. J. Carroll (2002). DNA microarray experiments: Biological and technological aspects. *Biometrics*, **58**, 701–717.

Nguyen, D. V., and D. M. Rocke (2002). Tumor classification by partial least squares using microarray gene expression data. *Bioinform.*, **18**, 39–50.

Oliver, S. (2000). Guilt-by-association goes global. *Nature*, **403**, 601–603.

Pan, W., J. Lin, and C. Le (2002). Model-based cluster analysis of microarray gene-expression data. *Genome Biol.*, **3(2)**, research 0009.1–0009.8.

Parmigiani, G., E. S. Garrett, R. Irizarry, and S. L. Zeger (2003). *The Analysis of Gene Expression Data: Methods and Software.* New York: Springer.

Perou, C. M., T. Sørlie, M. B. Eisen, M. van de Rijn, S. S. Jeffrey, C. A. Rees, J. R. Pollack, D. T. Ross, H. Johnsen, L. A. Akslen, Ø. Fluge, A. Pergamenschikov, C. Williams, S. X. Zhu, P. E. Lønning, A. L. Børresen-Dale, P. O. Brown, and D. Botstein (2000). Molecular portraits of human breast tumours. *Nature*, **406**, 747–752.

Pomeroy, S. L., P. Tamayo, M. Gaasenbeek, L. M. Sturla, M. Angelo, M. E. McLaughlin, J. Y. Kim, L. C. Goumnerova, P. M. Black, C. Lau, J. C. Allen, D. Zagzag, J. M. Olson, T. Curran, C. Wetmore, J. A. Biegel, T. Poggio, S. Mukherjee, R. Rifkin, A. Califano, G. Stolovitzky, D. N. Louis, J. P. Mesirov, E. S. Lander, and T. R. Golub (2002). Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, **415**, 436–442.

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

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning.* San Mateo, CA: Morgan Kauffman.

Raghavan, N., D. Amaratunga, A. Nie, and M. McMillian (2003). Fuzzy class prediction in toxicogenomics and other microarray applications. Unpublished manuscript.

Ramoni, M., P. Sebastiani, and I. S. Kohane (2002). Cluster analysis of gene expression dynamics. *Proc. Nat. Acad. Sciences*, **99**, 9121–9126.

Rao, C. R. (1948). The utilization of multiple measurements in problems of biological classifications. *J. Roy. Stat. Assoc.*, **B10**, 159–203.

Raychaudhuri, S., J. M. Stuart, and R. B. Altman (2000). Principal components analysis to summarize microarray experiments: Application to sporulation time series. *Pacific Symp. Biocomputing*, **5**, 452–463.

Ripley, B. D. (1996). *Pattern Recognition and Neural Networks.* Cambridge: Cambridge University Press.

Rocke, D. M., and B. Durbin (2001). A model for measurement error for gene expression arrays. *J. Comput. Biol.*, **8**, 557–569.

Rocke, D. M., and B. Durbin (2002). Approximate variance-stabilizing transformations for gene expression microarrys. Unpublished manuscript.

Sapir, M., and G. A. Churchill (2000). Estimating the posterior probability of differential gene expression from microarray data. Unpublished manuscript.

Schadt, E. E., C. Li, B. Ellis, and W. H. Wong (2001). Feature extraction and normalization algorithms for high-density oligonucleotide gene expression array data. *J. Cell. Biochem. 84*, **S37**, 120–125.

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

Schena, M. (1999). *DNA Microarrays: A Practical Approach.* Oxford: Oxford University Press.

Schena, M., D. Shalon, R. W. Davis, and P. O. Brown (1995). Quantitative monitoring of geneexpression patterns with a complementary DNA microarray. *Science*, **270**, 467–470.

Schölkopf, B. (1998). SVMs—a practical consequence of learning theory. *IEEE Intell. Syst.*, **13**, 18–21.

Schuchhardt, J., D. Beule, A. Malik, E. Wolski, H. Eickhoff, H. Lehrach, and H. Herzel (2000). Normalization strategies for cDNA microarrays. *Nucleic Acids Res.*, **28**, e47.

Seaman, M. A., K. R. Levin, and R. C. Serlin (1991). New developments in pairwise multiple comparisons: Some powerful and practicable procedures. *Psycholog. Bull.*, **110**, 577–586.

Seber, G. A. F. (1984). *Multivariate Observations.* New York: Wiley.

Seungchan, K., E. K. Dougherty, Y. Chen, S. Krishnamoorthy, P. Meltzer, J. M. Trent, and M. Bittner (2000). Multivariate measurement of gene expression relationships. *Genomics*, **67**, 201–209.

Sidak, Z. (1967). Rectangular confidence regions for the means of multivariate normal distributions. *J. Am. Stat. Assoc.*, **62**, 626–633.

Slonim, D. K. (2002). From patterns to pathways: Gene expression data analysis comes of age. *Nature Genet.*, **32**, 502–508.

Slonim, D. K., P. Tamayo, J. P. Mesirov, T. R. Golub, and E. S. Lander (2000). Class prediction and discovery using gene expression data. *Proc. RECOMB IV*: 263–271.

Smyth, G. K., Y. H. Yang, and T. P. Speed (2002). Statistical issues in cDNA microarray data analysis, *Technical report of the Department of Statistics at the University of California, Berkeley.*

Sokal, R. R., and C. D. Michener (1958). A statistical method for evaluating systematic relationships. *Univ. Kansas Sci. Bull.*, **38**, 1409–1438.

Sorlie, T., C. M. Perou, R. Tibshirani, T. Aas, S. Geisler, H. Johnsen, T. Hastie, M. B. Eisen, M. van De Rijn, S. S. Jeffrey, T. Thorsen, H. Quist, J. C. Matese, P. O. Brown, D. Botstein, P. E. Lonning, and A. L. Borresen-Dale (2001). Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proc. Nat. Acad. Sci.*, **98**, 10869–10874.

Speed, T. P. (2003). *Statistical Analysis of Gene Expression Microarray Data.* Chapman and Hall/CRC Press.

Spellman, P. T., G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher (1998). Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Mol. Biol. Cell*, **9**, 3273–3297.

Storey, J. D. (2001). The positive False Discovery Rate: A Bayesian interpretation and the *q*-value. *Technical Report of the Stanford University Department of Statistics.*

Storey, J. D. (2002). A direct approach to false discovery rates. *J. Roy. Stat. Soc.*, **B64**, 479–498.

Storey, J. D., and R. Tibshirani (2001). Estimating false discovery rates under dependence, with applications to DNA microarrays. *Technical Report of the Stanford University Department of Statistics.*

Storey, J. D., and R. Tibshirani (2002). Statistical methods for detecting differential gene expression. *J. Mol. Biol.*, forthcoming.

Strachan, T., and A. P. Read (1999). *Human Molecular Genetics*, 2nd ed. New York: Wiley.

Tamayo, P., D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. S. Lander, and T. R. Golub (1999). Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proc. Nat. Acad. Sci.*, **96**, 2907–2912.

Tavazoie, S., J. D. Hughes, M. J. Campbell, R. J. Cho, and G. M. Church (1999). Systematic determination of genetic network architecture. *Nature Genet.*, **22**, 281–285.

Therneau, T., R. C. Tschumper, and D. Jelinek (2002). Sharpening spots: Correcting for bleedover in cDNA array images. *Math. Biosci.*, **176**, 1–15.

Tibshirani, R., and B. Efron (2002). Pre-validation and inference in microarrays. *Stat. Appl. Genet. Mol. Biol.*, **1**.

Tibshirani, R., T. Hastie, M. Eisen, D. Ross, D. Botstein, and P. Brown (1999). Clustering methods for the analysis of DNA microarray data. *Technical Report of the Stanford University Department of Statistics.*

Tibshirani, R., T. Hastie, B. Narashiman, and G. Chu (2002). Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proc. Nat. Acad. Sci.*, **99**, 6567–6572.

Tibshirani, R., T. Hastie, B. Narashiman, and G. Chu (2002). Class prediction by nearest shrunken centroids, with applications to DNA microarrays. Unpublished manuscript.

Tibshirani, R., T. Hastie, B. Narasimhan, M. Eisen, G. Sherlock, P. Brown, and D. Botstein (2002). Exploratory screening of genes and clusters from microarray experiments. *Stat. Sinica*, **12**, 47–60.

Tibshirani, R., G. Walther, D. Botstein, and P. Brown (2001). Cluster validation by prediction strength. *Technical Report of the Stanford University Department of Statistics.*

Tibshirani, R., G. Walther, and T. Hastie (2001). Estimating the number of clusters in a dataset via the gap statistic. *J. Roy. Stat. Soc.*, **B64**, 411–423.

Toronen, P., M. Kolehmainen, G. Wong, and E. Castren (1999). Analysis of gene expression data using self-organizing maps. *FEBS Lett.*, **451**, 142–146.

Triola, M. F. (2001). *Elementary Statistics using Excel.* Reading, MA: Addison-Wesley.

Triola, M. F. (2002). *Elementary Statistics*, 8th ed. Reading, MA: Addison-Wesley.

Tseng, G. C., M. K. Oh, L. Rohlin, J. C. Liao, and W. H. Wong (2001). Issues in cDNA microarray analysis: Quality filtering, channel normalization, models of variations and assessment of gene effects. *Nucl. Acids Res.*, **29**, 2549–2557.

Tukey, J. W. (1962). The future of data analysis. *An. Math. Stat.*, **33**, 1–67.

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

Tukey, J. W. (1980). We need both exploratory and confirmatory. *Am. Statist.*, **34**, 23–25.

Tukey, J. W. (1986). In L. V. Jones, ed., *The Collected Works of John W. Tukey, Vol 3: Philosophy and Principles of Data Analysis 1949–1964.* Pacific Grove, CA: Wadsworth and Brooks/Cole.

Tusher, V. G., R. Tibshirani, and G. Chu (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proc. Nat. Acad. Sci.*, **98**, 5116–5121.

Van Der Laan, M. J., and J. Bryan (2001). Gene expression analysis with the parametric bootstrap. *Biostat.*, **2**, 445–461.

Velleman, P. F., and D. C. Hoaglin (1981). *Applications, Basics and Computing of Exploratory Data Analysis.* Boston: Duxbury Press.

Vingron, M. (2001). Bioinformatics needs to adopt statistical thinking. *Bioinform.*, **17**, 389–390.

Walker, M. G., W. Volkmuth, E. Sprinzak, D. Hodgson, and T. Klingler (1999). Prediction of gene function by genome-scale expression analysis: Prostate cancer-associated genes. *Genome Res.*, **9**, 1198–1203.

Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. *J. Am. Stat. Assoc.*, **58**, 236–244.

Wang, X., S. Ghosh, and S. W. Guo (2001). Quantitative quality control in microarray image processing and data acquisition. *Nucl. Acids Res.*, **29**, e75.

Weinstein, J. N., T. G. Myers, P. M. O'Connor, S. H. Friend, A. J. Fornace, K. W. Kohn, T. Fojo, S. E. Bates, L. V. Rubinstein, N. L. Anderson, J. K. Buolamwini, W. W. van Osdol, A. P. Monks, D. A. Scudiero, E. A. Sausville, D. W. Zaharevitz, V. Viswanadhan, B. Bunow, G. S. Johnson, R. E. Wittes, and K. D. Paull (1997). An information-intensive approach to the molecular pharmacology of cancer. *Science*, **275**, 343–349.

Westfall, P. H., and S. S. Young (1993). *Resampling-Based Multiple Testing: Examples and Methods for P-Value Adjustment.* New York: Wiley.

Wilson, E. B., and M. M. Hilferty (1931). The distribution of chi-square. *Proc. Nat. Acad. Sci.*, **17**, 694.

Wolfinger, R. D., G. Gibson, E. D. Wolfinger, L. Bennett, H. Hamadeh, P. Bushel, C. Afshari, and R. S. Paules (2001). Assessing gene significance from cDNA microarray expression data via mixed models. *J. Comput. Biol.*, **8**, 625–637.

Wouters, L., H. W. Gohlmann, L. Bijnens, G. Molenberghs, and P. J. Lewi (2002). Graphical exploration of gene expression data: A comparative study of three multivariate methods. Unpublished manuscript.

Xing, E. P., and R. M. Karp (2001). CLIFF: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. *Bioinform.*, **17**, S306–S315.

Xiong, M., L. Jin, W. Li, and E. Boerwinkle (2000). Computational methods for gene expression-based tumor classification. *Biotechniques*, **29**, 1264–1268.

Xiong, M., W. Li, J. Zhao, L. Jin, and E. Boerwinkle (2001). Feature (gene) selection in gene expression-based tumor classification. *Mol. Genet. Metabol.*, **73**, 239–247.

Xiong, M., X. Fang, and J. Zhao (2001). Biomarker identification by feature wrappers. *Genome Res.*, **11**, 1878–1887.

Yang, Y. H., M. J. Buckley, S. Dudoit, and T. P. Speed (2000). Comparison of methods for image analysis on cDNA microarray data. *Technical Report of the Department of Statistics, University of California at Berkeley*.

Yang, Y. H., M. J. Buckley, and T. P. Speed (2001). Analysis of microarray images. *Brief. Bioinform.*, **2**, 341–349.

Yang, Y. H., S. Dudoit, P. Lu, and T. P. Speed (2001). Normalization for cDNA microarray data. In M. L. Bittner, Y. Chen, A. N. Dorsel, and E. R. Dougherty, eds., *Microarrays: Optical Technologies and Informatics*, Vol. 4266 of Proceedings of SPIE.

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

Yang, Y. H., and T. Speed (2002). Design issues for cDNA microarray experiments. *Nature Rev. Genet.*, **3**, 579–588.

Yeang, C. H., S. Ramaswamy, P. Tamayo, S. Mukherjee, R. M. Rifkin, M. Angelo, M. Reich, E. S. Lander, J. Mesirov, and T. Golub (2001). Molecular classification of multiple tumor types. *Bioinform.*, **17**, S316–S322.

Yeung, K. Y., C. Fraley, A. Murua, A. E. Raftery, and W. L. Ruzzo (2001). Model-based clustering and data transformations for gene expression data. *Bioinform.*, **17**, 977–987.

Yeung, K. Y., D. R. Haynor, and W. L. Ruzzo (2001). Validating clustering for gene expression data. *Bioinform.*, **17**, 309–318.

Yeung, K. Y., and W. L. Ruzzo (2001). Principal component analysis for clustering gene expression data. *Bioinform.*, **17**, 763–774.

Yuketieli, D., and Y. Benjamini (1999). Resampling based false discovery rate controlling multiple test procedures for correlated test statistics. *J. Stat. Plan. Infer.*, **82**, 171–196.

Zhang, H., C. Y. Yu, B. Singer, and M. Xiong (2001). Recursive partitioning for tumor classification with gene expression microarray data. *Proc. Nat. Acad. Sci.*, **98**, 6730–6735.

Zhang, X., D. Amaratunga, and K. Roeder (2002). Identifying differentially expressed genes for class prediction using classification error and gene clustering. Unpublished manuscript.

Author Index

Exploration and Analysis of DNA Microarray and Protein Array Data
By Dhammika Amaratunga and Javier Cabrera
ISBN 0-471-27398-8

Exploration and Analysis of DNA Microarray and Protein Array Data
Edited by DHAMMIKA AMARATUNGA and JAVIER CABRERA

Subject Index

Exploration and Analysis of DNA Microarray and Protein Array Data
By Dhammika Amaratunga and Javier Cabrera
ISBN 0-471-27398-8

Exploration and Analysis of DNA Microarray and Protein Array Data
Edited by DHAMMIKA AMARATUNGA and JAVIER CABRERA

WILEY SERIES IN PROBABILITY AND STATISTICS

ESTABLISHED BY WALTER A. SHEWHART AND SAMUEL S. WILKS

The ***Wiley Series in Probability and Statistics*** is well established and authoritative. It covers many topics of current research interest in both pure and applied statistics and probability theory. Written by leading statisticians and institutions, the titles span both state-of-the-art developments in the field and classical methods.

Reflecting the wide range of current research in statistics, the series encompasses applied, methodological and theoretical statistics, ranging from applications and new techniques made possible by advances in computerized practice to rigorous treatment of theoretical approaches.

This series provides essential and invaluable reading for all statisticians, whether in academia, industry, government, or research.

ABRAHAM and LEDOLTER · Statistical Methods for Forecasting
AGRESTI · Analysis of Ordinal Categorical Data
AGRESTI · An Introduction to Categorical Data Analysis
AGRESTI · Categorical Data Analysis, *Second Edition*
ALTMAN, GILL, and McDONALD · Numerical Issues in Statistical Computing for the Social Scientist
AMARATUNGA and CABRERA · Exploration and Analysis of DNA Microarray and Protein Array Data
ANDĚL · Mathematics of Chance
ANDERSON · An Introduction to Multivariate Statistical Analysis, *Third Edition*
*ANDERSON · The Statistical Analysis of Time Series
ANDERSON, AUQUIER, HAUCK, OAKES, VANDAELE, and WEISBERG · Statistical Methods for Comparative Studies
ANDERSON and LOYNES · The Teaching of Practical Statistics
ARMITAGE and DAVID (editors) · Advances in Biometry
ARNOLD, BALAKRISHNAN, and NAGARAJA · Records
*ARTHANARI and DODGE · Mathematical Programming in Statistics
*BAILEY · The Elements of Stochastic Processes with Applications to the Natural Sciences
BALAKRISHNAN and KOUTRAS · Runs and Scans with Applications
BARNETT · Comparative Statistical Inference, *Third Edition*
BARNETT and LEWIS · Outliers in Statistical Data, *Third Edition*
BARTOSZYNSKI and NIEWIADOMSKA-BUGAJ · Probability and Statistical Inference
BASILEVSKY · Statistical Factor Analysis and Related Methods: Theory and Applications
BASU and RIGDON · Statistical Methods for the Reliability of Repairable Systems
BATES and WATTS · Nonlinear Regression Analysis and Its Applications
BECHHOFER, SANTNER, and GOLDSMAN · Design and Analysis of Experiments for Statistical Selection, Screening, and Multiple Comparisons
BELSLEY · Conditioning Diagnostics: Collinearity and Weak Data in Regression

*Now available in a lower priced paperback edition in the Wiley Classics Library.

BELSLEY, KUH, and WELSCH · Regression Diagnostics: Identifying Influential Data and Sources of Collinearity
BENDAT and PIERSOL · Random Data: Analysis and Measurement Procedures, *Third Edition*
BERRY, CHALONER, and GEWEKE · Bayesian Analysis in Statistics and Econometrics: Essays in Honor of Arnold Zellner
BERNARDO and SMITH · Bayesian Theory
BHAT and MILLER · Elements of Applied Stochastic Processes, *Third Edition*
BHATTACHARYA and JOHNSON · Statistical Concepts and Methods
BHATTACHARYA and WAYMIRE · Stochastic Processes with Applications
BILLINGSLEY · Convergence of Probability Measures, *Second Edition*
BILLINGSLEY · Probability and Measure, *Third Edition*
BIRKES and DODGE · Alternative Methods of Regression
BLISCHKE AND MURTHY (editors) · Case Studies in Reliability and Maintenance
BLISCHKE AND MURTHY · Reliability: Modeling, Prediction, and Optimization
BLOOMFIELD · Fourier Analysis of Time Series: An Introduction, *Second Edition*
BOLLEN · Structural Equations with Latent Variables
BOROVKOV · Ergodicity and Stability of Stochastic Processes
BOULEAU · Numerical Methods for Stochastic Processes
BOX · Bayesian Inference in Statistical Analysis
BOX · R. A. Fisher, the Life of a Scientist
BOX and DRAPER · Empirical Model-Building and Response Surfaces
*BOX and DRAPER · Evolutionary Operation: A Statistical Method for Process Improvement
BOX, HUNTER, and HUNTER · Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building
BOX and LUCEÑO · Statistical Control by Monitoring and Feedback Adjustment
BRANDIMARTE · Numerical Methods in Finance: A MATLAB-Based Introduction
BROWN and HOLLANDER · Statistics: A Biomedical Introduction
BRUNNER, DOMHOF, and LANGER · Nonparametric Analysis of Longitudinal Data in Factorial Experiments
BUCKLEW · Large Deviation Techniques in Decision, Simulation, and Estimation
CAIROLI and DALANG · Sequential Stochastic Optimization
CHAN · Time Series: Applications to Finance
CHATTERJEE and HADI · Sensitivity Analysis in Linear Regression
CHATTERJEE and PRICE · Regression Analysis by Example, *Third Edition*
CHERNICK · Bootstrap Methods: A Practitioner's Guide
CHERNICK and FRIIS · Introductory Biostatistics for the Health Sciences
CHILÈS and DELFINER · Geostatistics: Modeling Spatial Uncertainty
CHOW and LIU · Design and Analysis of Clinical Trials: Concepts and Methodologies, *Second Edition*
CLARKE and DISNEY · Probability and Random Processes: A First Course with Applications, *Second Edition*
*COCHRAN and COX · Experimental Designs, *Second Edition*
CONGDON · Bayesian Statistical Modelling
CONOVER · Practical Nonparametric Statistics, *Second Edition*
COOK · Regression Graphics
COOK and WEISBERG · Applied Regression Including Computing and Graphics
COOK and WEISBERG · An Introduction to Regression Graphics
CORNELL · Experiments with Mixtures, Designs, Models, and the Analysis of Mixture Data, *Third Edition*
COVER and THOMAS · Elements of Information Theory
COX · A Handbook of Introductory Statistical Methods

*Now available in a lower priced paperback edition in the Wiley Classics Library.

*COX · Planning of Experiments
CRESSIE · Statistics for Spatial Data, *Revised Edition*
CSÖRGŐ and HORVÁTH · Limit Theorems in Change Point Analysis
DANIEL · Applications of Statistics to Industrial Experimentation
DANIEL · Biostatistics: A Foundation for Analysis in the Health Sciences, *Sixth Edition*
*DANIEL · Fitting Equations to Data: Computer Analysis of Multifactor Data, *Second Edition*
DASU and JOHNSON · Exploratory Data Mining and Data Cleaning
DAVID and NAGARAJA · Order Statistics, *Third Edition*
*DEGROOT, FIENBERG, and KADANE · Statistics and the Law
DEL CASTILLO · Statistical Process Adjustment for Quality Control
DETTE and STUDDEN · The Theory of Canonical Moments with Applications in Statistics, Probability, and Analysis
DEY and MUKERJEE · Fractional Factorial Plans
DILLON and GOLDSTEIN · Multivariate Analysis: Methods and Applications
DODGE · Alternative Methods of Regression
*DODGE and ROMIG · Sampling Inspection Tables, *Second Edition*
*DOOB · Stochastic Processes
DOWDY, WEARDEN, and CHILKO · Statistics for Research, *Third Edition*
DRAPER and SMITH · Applied Regression Analysis, *Third Edition*
DRYDEN and MARDIA · Statistical Shape Analysis
DUDEWICZ and MISHRA · Modern Mathematical Statistics
DUNN and CLARK · Applied Statistics: Analysis of Variance and Regression, *Second Edition*
DUNN and CLARK · Basic Statistics: A Primer for the Biomedical Sciences, *Third Edition*
DUPUIS and ELLIS · A Weak Convergence Approach to the Theory of Large Deviations
*ELANDT-JOHNSON and JOHNSON · Survival Models and Data Analysis
ENDERS · Applied Econometric Time Series
ETHIER and KURTZ · Markov Processes: Characterization and Convergence
EVANS, HASTINGS, and PEACOCK · Statistical Distributions, *Third Edition*
FELLER · An Introduction to Probability Theory and Its Applications, Volume I, *Third Edition,* Revised; Volume II, *Second Edition*
FISHER and VAN BELLE · Biostatistics: A Methodology for the Health Sciences
*FLEISS · The Design and Analysis of Clinical Experiments
FLEISS · Statistical Methods for Rates and Proportions, *Third Edition*
FLEMING and HARRINGTON · Counting Processes and Survival Analysis
FULLER · Introduction to Statistical Time Series, *Second Edition*
FULLER · Measurement Error Models
GALLANT · Nonlinear Statistical Models
GIESBRECHT and GUMPERTZ · Planning, Construction, and Statistical Analysis of Comparative Experiments
GIFI · Nonlinear Multivariate Analysis
GHOSH, MUKHOPADHYAY, and SEN · Sequential Estimation
GIFI · Nonlinear Multivariate Analysis
GLASSERMAN and YAO · Monotone Structure in Discrete-Event Systems
GNANADESIKAN · Methods for Statistical Data Analysis of Multivariate Observations, *Second Edition*
GOLDSTEIN and LEWIS · Assessment: Problems, Development, and Statistical Issues
GREENWOOD and NIKULIN · A Guide to Chi-Squared Testing
GROSS and HARRIS · Fundamentals of Queueing Theory, *Third Edition*
*HAHN and SHAPIRO · Statistical Models in Engineering
HAHN and MEEKER · Statistical Intervals: A Guide for Practitioners

*Now available in a lower priced paperback edition in the Wiley Classics Library.

HALD · A History of Probability and Statistics and their Applications Before 1750
HALD · A History of Mathematical Statistics from 1750 to 1930
HAMPEL · Robust Statistics: The Approach Based on Influence Functions
HANNAN and DEISTLER · The Statistical Theory of Linear Systems
HEIBERGER · Computation for the Analysis of Designed Experiments
HEDAYAT and SINHA · Design and Inference in Finite Population Sampling
HELLER · MACSYMA for Statisticians
HINKELMAN and KEMPTHORNE: · Design and Analysis of Experiments, Volume 1: Introduction to Experimental Design
HOAGLIN, MOSTELLER, and TUKEY · Exploratory Approach to Analysis of Variance
HOAGLIN, MOSTELLER, and TUKEY · Exploring Data Tables, Trends and Shapes
*HOAGLIN, MOSTELLER, and TUKEY · Understanding Robust and Exploratory Data Analysis
HOCHBERG and TAMHANE · Multiple Comparison Procedures
HOCKING · Methods and Applications of Linear Models: Regression and the Analysis of Variance, *Second Edition*
HOEL · Introduction to Mathematical Statistics, *Fifth Edition*
HOGG and KLUGMAN · Loss Distributions
HOLLANDER and WOLFE · Nonparametric Statistical Methods, *Second Edition*
HOSMER and LEMESHOW · Applied Logistic Regression, *Second Edition*
HOSMER and LEMESHOW · Applied Survival Analysis: Regression Modeling of Time to Event Data
HØYLAND and RAUSAND · System Reliability Theory: Models and Statistical Methods
HUBER · Robust Statistics
HUBERTY · Applied Discriminant Analysis
HUNT and KENNEDY · Financial Derivatives in Theory and Practice
HUSKOVA, BERAN, and DUPAC · Collected Works of Jaroslav Hajek—with Commentary
IMAN and CONOVER · A Modern Approach to Statistics
JACKSON · A User's Guide to Principle Components
JOHN · Statistical Methods in Engineering and Quality Assurance
JOHNSON · Multivariate Statistical Simulation
JOHNSON and BALAKRISHNAN · Advances in the Theory and Practice of Statistics: A Volume in Honor of Samuel Kotz
JUDGE, GRIFFITHS, HILL, LÜTKEPOHL, and LEE · The Theory and Practice of Econometrics, *Second Edition*
JOHNSON and KOTZ · Distributions in Statistics
JOHNSON and KOTZ (editors) · Leading Personalities in Statistical Sciences: From the Seventeenth Century to the Present
JOHNSON, KOTZ, and BALAKRISHNAN · Continuous Univariate Distributions, Volume 1, *Second Edition*
JOHNSON, KOTZ, and BALAKRISHNAN · Continuous Univariate Distributions, Volume 2, *Second Edition*
JOHNSON, KOTZ, and BALAKRISHNAN · Discrete Multivariate Distributions
JOHNSON, KOTZ, and KEMP · Univariate Discrete Distributions, *Second Edition*
JUREČKOVÁ and SEN · Robust Statistical Procedures: Aymptotics and Interrelations
JUREK and MASON · Operator-Limit Distributions in Probability Theory
KADANE · Bayesian Methods and Ethics in a Clinical Trial Design
KADANE AND SCHUM · A Probabilistic Analysis of the Sacco and Vanzetti Evidence
KALBFLEISCH and PRENTICE · The Statistical Analysis of Failure Time Data, *Second Edition*
KASS and VOS · Geometrical Foundations of Asymptotic Inference

*Now available in a lower priced paperback edition in the Wiley Classics Library.

KAUFMAN and ROUSSEEUW · Finding Groups in Data: An Introduction to Cluster Analysis
KEDEM and FOKIANOS · Regression Models for Time Series Analysis
KENDALL, BARDEN, CARNE, and LE · Shape and Shape Theory
KHURI · Advanced Calculus with Applications in Statistics, *Second Edition*
KHURI, MATHEW, and SINHA · Statistical Tests for Mixed Linear Models
KLEIBER and KOTZ · Statistical Size Distributions in Economics and Actuarial Sciences
KLUGMAN, PANJER, and WILLMOT · Loss Models: From Data to Decisions
KLUGMAN, PANJER, and WILLMOT · Solutions Manual to Accompany Loss Models: From Data to Decisions
KOTZ, BALAKRISHNAN, and JOHNSON · Continuous Multivariate Distributions, Volume 1, *Second Edition*
KOTZ and JOHNSON (editors) · Encyclopedia of Statistical Sciences: Volumes 1 to 9 with Index
KOTZ and JOHNSON (editors) · Encyclopedia of Statistical Sciences: Supplement Volume
KOTZ, READ, and BANKS (editors) · Encyclopedia of Statistical Sciences: Update Volume 1
KOTZ, READ, and BANKS (editors) · Encyclopedia of Statistical Sciences: Update Volume 2
KOVALENKO, KUZNETZOV, and PEGG · Mathematical Theory of Reliability of Time-Dependent Systems with Practical Applications
LACHIN · Biostatistical Methods: The Assessment of Relative Risks
LAD · Operational Subjective Statistical Methods: A Mathematical, Philosophical, and Historical Introduction
LAMPERTI · Probability: A Survey of the Mathematical Theory, *Second Edition*
LANGE, RYAN, BILLARD, BRILLINGER, CONQUEST, and GREENHOUSE · Case Studies in Biometry
LARSON · Introduction to Probability Theory and Statistical Inference, *Third Edition*
LAWLESS · Statistical Models and Methods for Lifetime Data, *Second Edition*
LAWSON · Statistical Methods in Spatial Epidemiology
LE · Applied Categorical Data Analysis
LE · Applied Survival Analysis
LEE and WANG · Statistical Methods for Survival Data Analysis, *Third Edition*
LEPAGE and BILLARD · Exploring the Limits of Bootstrap
LEYLAND and GOLDSTEIN (editors) · Multilevel Modelling of Health Statistics
LIAO · Statistical Group Comparison
LINDVALL · Lectures on the Coupling Method
LINHART and ZUCCHINI · Model Selection
LITTLE and RUBIN · Statistical Analysis with Missing Data, *Second Edition*
LLOYD · The Statistical Analysis of Categorical Data
MAGNUS and NEUDECKER · Matrix Differential Calculus with Applications in Statistics and Econometrics, *Revised Edition*
MALLER and ZHOU · Survival Analysis with Long Term Survivors
MALLOWS · Design, Data, and Analysis by Some Friends of Cuthbert Daniel
MANN, SCHAFER, and SINGPURWALLA · Methods for Statistical Analysis of Reliability and Life Data
MANTON, WOODBURY, and TOLLEY · Statistical Applications Using Fuzzy Sets
MARDIA and JUPP · Directional Statistics
MASON, GUNST, and HESS · Statistical Design and Analysis of Experiments with Applications to Engineering and Science, *Second Edition*
McCULLOCH and SEARLE · Generalized, Linear, and Mixed Models
McFADDEN · Management of Data in Clinical Trials

*Now available in a lower priced paperback edition in the Wiley Classics Library.

McLACHLAN · Discriminant Analysis and Statistical Pattern Recognition
McLACHLAN and KRISHNAN · The EM Algorithm and Extensions
McLACHLAN and PEEL · Finite Mixture Models
McNEIL · Epidemiological Research Methods
MEEKER and ESCOBAR · Statistical Methods for Reliability Data
MEERSCHAERT and SCHEFFLER · Limit Distributions for Sums of Independent Random Vectors: Heavy Tails in Theory and Practice
*MILLER · Survival Analysis, *Second Edition*
MONTGOMERY, PECK, and VINING · Introduction to Linear Regression Analysis, *Third Edition*
MORGENTHALER and TUKEY · Configural Polysampling: A Route to Practical Robustness
MUIRHEAD · Aspects of Multivariate Statistical Theory
MURRAY · X-STAT 2.0 Statistical Experimentation, Design Data Analysis, and Nonlinear Optimization
MURTHY, XIE, and JIANG · Weibull Models
MYERS and MONTGOMERY · Response Surface Methodology: Process and Product Optimization Using Designed Experiments, *Second Edition*
MYERS, MONTGOMERY, and VINING · Generalized Linear Models. With Applications in Engineering and the Sciences
NELSON · Accelerated Testing, Statistical Models, Test Plans, and Data Analyses
NELSON · Applied Life Data Analysis
NEWMAN · Biostatistical Methods in Epidemiology
OCHI · Applied Probability and Stochastic Processes in Engineering and Physical Sciences
OKABE, BOOTS, SUGIHARA, and CHIU · Spatial Tesselations: Concepts and Applications of Voronoi Diagrams, *Second Edition*
OLIVER and SMITH · Influence Diagrams, Belief Nets and Decision Analysis
PALTA · Quantitative Methods in Population Health: Extensions of Ordinary Regressions
PANKRATZ · Forecasting with Dynamic Regression Models
PANKRATZ · Forecasting with Univariate Box-Jenkins Models: Concepts and Cases
*PARZEN · Modern Probability Theory and Its Applications
PEÑA, TIAO, and TSAY · A Course in Time Series Analysis
PIANTADOSI · Clinical Trials: A Methodologic Perspective
PORT · Theoretical Probability for Applications
POURAHMADI · Foundations of Time Series Analysis and Prediction Theory
PRESS · Bayesian Statistics: Principles, Models, and Applications
PRESS · Subjective and Objective Bayesian Statistics, *Second Edition*
PRESS and TANUR · The Subjectivity of Scientists and the Bayesian Approach
PUKELSHEIM · Optimal Experimental Design
PURI, VILAPLANA, and WERTZ · New Perspectives in Theoretical and Applied Statistics
PUTERMAN · Markov Decision Processes: Discrete Stochastic Dynamic Programming
*RAO · Linear Statistical Inference and Its Applications, *Second Edition*
RENCHER · Linear Models in Statistics
RENCHER · Methods of Multivariate Analysis, *Second Edition*
RENCHER · Multivariate Statistical Inference with Applications
RIPLEY · Spatial Statistics
RIPLEY · Stochastic Simulation
ROBINSON · Practical Strategies for Experimenting
ROHATGI and SALEH · An Introduction to Probability and Statistics, *Second Edition*
ROLSKI, SCHMIDLI, SCHMIDT, and TEUGELS · Stochastic Processes for Insurance and Finance
ROSENBERGER and LACHIN · Randomization in Clinical Trials: Theory and Practice

*Now available in a lower priced paperback edition in the Wiley Classics Library.

ROSS · Introduction to Probability and Statistics for Engineers and Scientists
ROUSSEEUW and LEROY · Robust Regression and Outlier Detection
RUBIN · Multiple Imputation for Nonresponse in Surveys
RUBINSTEIN · Simulation and the Monte Carlo Method
RUBINSTEIN and MELAMED · Modern Simulation and Modeling
RYAN · Modern Regression Methods
RYAN · Statistical Methods for Quality Improvement, *Second Edition*
SALTELLI, CHAN, and SCOTT (editors) · Sensitivity Analysis
*SCHEFFE · The Analysis of Variance
SCHIMEK · Smoothing and Regression: Approaches, Computation, and Application
SCHOTT · Matrix Analysis for Statistics
SCHUSS · Theory and Applications of Stochastic Differential Equations
SCOTT · Multivariate Density Estimation: Theory, Practice, and Visualization
*SEARLE · Linear Models
SEARLE · Linear Models for Unbalanced Data
SEARLE · Matrix Algebra Useful for Statistics
SEARLE, CASELLA, and McCULLOCH · Variance Components
SEARLE and WILLETT · Matrix Algebra for Applied Economics
SEBER and LEE · Linear Regression Analysis, *Second Edition*
SEBER · Multivariate Observations
SEBER and WILD · Nonlinear Regression
SENNOTT · Stochastic Dynamic Programming and the Control of Queueing Systems
*SERFLING · Approximation Theorems of Mathematical Statistics
SHAFER and VOVK · Probability and Finance: It's Only a Game!
SMALL and McLEISH · Hilbert Space Methods in Probability and Statistical Inference
SRIVASTAVA · Methods of Multivariate Statistics
STAPLETON · Linear Statistical Models
STAUDTE and SHEATHER · Robust Estimation and Testing
STOYAN, KENDALL, and MECKE · Stochastic Geometry and Its Applications, *Second Edition*
STOYAN and STOYAN · Fractals, Random Shapes and Point Fields: Methods of Geometrical Statistics
STYAN · The Collected Papers of T. W. Anderson: 1943–1985
SUTTON, ABRAMS, JONES, SHELDON, and SONG · Methods for Meta-Analysis in Medical Research
TANAKA · Time Series Analysis: Nonstationary and Noninvertible Distribution Theory
THOMPSON · Empirical Model Building
THOMPSON · Sampling, *Second Edition*
THOMPSON · Simulation: A Modeler's Approach
THOMPSON and SEBER · Adaptive Sampling
THOMPSON, WILLIAMS, and FINDLAY · Models for Investors in Real World Markets
TIAO, BISGAARD, HILL, PEÑA, and STIGLER (editors) · Box on Quality and Discovery: with Design, Control, and Robustness
TIERNEY · LISP-STAT: An Object-Oriented Environment for Statistical Computing and Dynamic Graphics
TSAY · Analysis of Financial Time Series
UPTON and FINGLETON · Spatial Data Analysis by Example, Volume II: Categorical and Directional Data
VAN BELLE · Statistical Rules of Thumb
VESTRUP · The Theory of Measures and Integration
VIDAKOVIC · Statistical Modeling by Wavelets
WEISBERG · Applied Linear Regression, *Second Edition*
WELSH · Aspects of Statistical Inference
WESTFALL and YOUNG · Resampling-Based Multiple Testing: Examples and Methods for p-Value Adjustment

*Now available in a lower priced paperback edition in the Wiley Classics Library.

WHITTAKER · Graphical Models in Applied Multivariate Statistics
WINKER · Optimization Heuristics in Economics: Applications of Threshold Accepting
WONNACOTT and WONNACOTT · Econometrics, *Second Edition*
WOODING · Planning Pharmaceutical Clinical Trials: Basic Statistical Principles
WOOLSON and CLARKE · Statistical Methods for the Analysis of Biomedical Data, *Second Edition*
WU and HAMADA · Experiments: Planning, Analysis, and Parameter Design Optimization
YANG · The Construction Theory of Denumerable Markov Processes
*ZELLNER · An Introduction to Bayesian Inference in Econometrics
ZHOU, OBUCHOWSKI, and McCLISH · Statistical Methods in Diagnostic Medicine

*Now available in a lower priced paperback edition in the Wiley Classics Library.