2010

Improving statistical inference for gene expression profiling data by borrowing information

Long Qu
Iowa State University

Follow this and additional works at: http://lib.dr.iastate.edu/etd

Part of the Statistics and Probability Commons

Recommended Citation

This Dissertation is brought to you for free and open access by the Graduate College at Digital Repository @ Iowa State University. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Digital Repository @ Iowa State University. For more information, please contact digirep@iastate.edu.
Improving statistical inference for gene expression profiling data
by borrowing information
by

Long Qu

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Co-majors: Bioinformatics and Computational Biology;
Statistics

Program of Study Committee:
Jack C.M. Dekkers, Co-major Professor
Dan Nettleton, Co-major Professor
Christopher K. Tuggle
Philip Dixon
Jin Tian

Iowa State University
Ames, Iowa
2010

Copyright © Long Qu, 2010. All rights reserved.
Dedicated to Wanping Li, Wanru Li,
and the memory of Wanling Li (1955–2009).
# TABLE OF CONTENTS

LIST OF TABLES ................................................................. vi

LIST OF FIGURES ............................................................... viii

LIST OF ALGORITHMS .......................................................... ix

ACKNOWLEDGEMENTS .......................................................... x

ABSTRACT ................................................................. xi

CHAPTER 1. GENERAL INTRODUCTION ................................. 1

  1.1 High-throughput technologies and a fasting experiment in pigs 2

  1.2 Statistical issues in the design and preprocessing of expression profiling experiments 5

  1.3 Issues in statistical inference for expression profiling data ................................. 8

    1.3.1 Statistical problems in detecting differentially expressed genes ........... 9

    1.3.2 Statistical problems in gene set analysis ................................. 12

    1.3.3 Statistical problems in clustering and classifications of microarray data . 14

  1.4 Thesis organization ................................................................. 16

CHAPTER 2. VARIANCE MODEL SELECTION WITH APPLICATION TO JOINT ANALYSIS OF MULTIPLE MICROARRAY DATASETS UNDER FALSE DISCOVERY RATE CONTROL ....................... 30

Abstract ................................................................. 30

  2.1 Introduction ................................................................. 31

    2.1.1 Joint analysis of from multiple microarray datasets ................................. 31

    2.1.2 Model selection ................................................................. 32

  2.2 Methods ................................................................. 35
### Chapter 2. Simulation Results and Discussion

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2.1 Modified MRPP for testing homogeneity of variances</td>
<td>35</td>
</tr>
<tr>
<td>2.2.2 Cross-validation for selecting variance models</td>
<td>38</td>
</tr>
<tr>
<td>2.2.3 AICc under heteroscedasticity</td>
<td>40</td>
</tr>
<tr>
<td>2.2.4 Combining model selection criteria from multiple genes</td>
<td>41</td>
</tr>
<tr>
<td>2.2.5 Simulation based on real microarray data</td>
<td>43</td>
</tr>
<tr>
<td>2.3 Simulation results and discussion</td>
<td>47</td>
</tr>
<tr>
<td>2.3.1 Proportion of correctly selected models</td>
<td>47</td>
</tr>
<tr>
<td>2.3.2 Ranking genes in terms of differential expression</td>
<td>50</td>
</tr>
<tr>
<td>2.3.3 Control of FDR</td>
<td>55</td>
</tr>
<tr>
<td>2.3.4 Size and power of modified MRPP compared with alternative univariate tests</td>
<td>60</td>
</tr>
<tr>
<td>2.4 Summary</td>
<td>63</td>
</tr>
</tbody>
</table>

### Chapter 3. Improved Estimation of the Noncentrality Parameter Distribution from a Large Number of t-Statistics, with Applications to False Discovery Rate Estimation in Microarray Data Analysis

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>69</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>70</td>
</tr>
<tr>
<td>3.2 A parametric model for the t-statistics</td>
<td>72</td>
</tr>
<tr>
<td>3.3 A nonparametric model for the t-statistics</td>
<td>74</td>
</tr>
<tr>
<td>3.3.1 Penalized likelihood</td>
<td>75</td>
</tr>
<tr>
<td>3.3.2 Sequential quadratic programming</td>
<td>76</td>
</tr>
<tr>
<td>3.3.3 Tuning parameter selection</td>
<td>77</td>
</tr>
<tr>
<td>3.4 A semiparametric approach</td>
<td>78</td>
</tr>
<tr>
<td>3.5 False discovery rate estimation</td>
<td>80</td>
</tr>
<tr>
<td>3.6 Simulation study</td>
<td>81</td>
</tr>
<tr>
<td>3.7 Summary</td>
<td>93</td>
</tr>
</tbody>
</table>
CHAPTER 4. A HIERARCHICAL SEMIPARAMETRIC MODEL FOR INCORPORATING INTER-GENE INFORMATION FOR ANALYSIS OF GENOMIC DATA

Abstract 98

4.1 Introduction to the naïve two-component mixture model 98

4.2 Summarizing a priori inter-gene information 101
  4.2.1 Dissimilarity matrices across genes 102
  4.2.2 Feature generation through multidimensional scaling 105

4.3 Hierarchical semiparametric model 106

4.4 Parameter estimation 108
  4.4.1 Tuning parameter selection 109
  4.4.2 Interval estimation 111

4.5 Simulation study 113

4.6 Analysis of pig fasting data 117

4.7 Discussion 120

4.A A generalized Expectation-Maximization algorithm 122

4.B The gradient and Hessian for use in Newtonian algorithms 125

CHAPTER 5. GENERAL CONCLUSIONS 133

CUMULATIVE BIBLIOGRAPHY 135
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Proportion of studies with total sample size no larger than 10 in the GEO database (April 22, 2010)</td>
<td>9</td>
</tr>
<tr>
<td>2.1</td>
<td>Simulation setting for the ratio of variances.</td>
<td>44</td>
</tr>
<tr>
<td>2.2</td>
<td>Proportion ($\times 100$) of correctly selected models from dataset GSE5406 when sample size $n_1 = n_2 = 3$</td>
<td>48</td>
</tr>
<tr>
<td>2.3</td>
<td>Proportion ($\times 100$) of correctly selected models from dataset GSE5406 when sample size $n_1 = n_2 = 7$</td>
<td>48</td>
</tr>
<tr>
<td>2.4</td>
<td>Area ($\times 1000$) under the ROC curve for selecting individual gene model from dataset GSE5406 and $n_1 = n_2 = 7$</td>
<td>52</td>
</tr>
<tr>
<td>2.5</td>
<td>Area ($\times 1000$) under the ROC curve for selecting individual gene model from dataset GSE5406 and $n_1 = n_2 = 3$</td>
<td>53</td>
</tr>
<tr>
<td>2.6</td>
<td>False discovery proportions and number of rejections for contrast 1 from individual gene model selection on dataset GSE5406 using the BH method when $n_1 = n_2 = 3$ and $r \equiv 1$ for all genes</td>
<td>56</td>
</tr>
<tr>
<td>2.7</td>
<td>False discovery proportions and number of rejections for contrast 1 from individual gene model selection on dataset GSE5406 using the BH method when $n_1 = n_2 = 7$ and $r \equiv 1$ for all genes</td>
<td>57</td>
</tr>
<tr>
<td>2.8</td>
<td>False discovery proportions and number of rejections for contrast 2 from individual gene model selection on dataset GSE5406 using the BH method when $n_1 = n_2 = 3$ and $\mathbb{E}(r_g) = 10$ for all genes</td>
<td>58</td>
</tr>
<tr>
<td>2.9</td>
<td>Estimated Kullback-Leibler divergence of nominal to theoretical $p$-value distributions</td>
<td>61</td>
</tr>
</tbody>
</table>
Table 2.10  Probability that the test statistic under the alternative is at least as extreme as the test statistic under the null when $\mathbb{E}(r_g) = 1$  
61
Table 2.11  Probability that the test statistic under the alternative is at least as extreme as the test statistic under the null when $\mathbb{E}(r_g) = 10$  
61
Table 4.1  AUROC comparison  
116
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Simulation results when $\pi_0 = 0.39$, $\nu_i = 4$, and $\delta_i \sim N(1.5, 1.5^2)$ under ideal error distributions.</td>
<td>84</td>
</tr>
<tr>
<td>3.2</td>
<td>Simulation results when $\pi_0 = 0.39$, $\nu_i = 4$, and $\delta_i \sim N(1.5, 1.5^2)$ under data-based error distributions.</td>
<td>85</td>
</tr>
<tr>
<td>3.3</td>
<td>Simulation results when $\pi_0 = 0.39$, $\nu_i = 4$, and $\delta_i \sim Cauchy(0, 1)$ under ideal error distributions.</td>
<td>87</td>
</tr>
<tr>
<td>3.4</td>
<td>Simulation results when $\pi_0 = 0.39$, $\nu_i = 4$, and $\delta_i \sim N(0, 3^2)$ under ideal error distributions.</td>
<td>88</td>
</tr>
<tr>
<td>3.5</td>
<td>Simulation results when $\pi_0 = 0.39$, $\nu_i = 4$, and $\delta_i \sim Weibull(1.6, 2.6)^{−2.33}$ under ideal error distributions.</td>
<td>89</td>
</tr>
<tr>
<td>3.6</td>
<td>Simulation results when $\pi_0 = 0.39$, $\nu_i = 10$, and $\delta_i \sim Weibull(1.6, 2.6)^{−2.33}$ under ideal error distributions.</td>
<td>90</td>
</tr>
<tr>
<td>3.7</td>
<td>Simulation results when $\pi_0 = 0.99$, $\nu_i = 4$, and $\delta_i \sim N(0, 1.5^2)$ under ideal error distributions.</td>
<td>91</td>
</tr>
<tr>
<td>3.8</td>
<td>Simulation results when $\pi_0 = 0.99$, $\nu_i = 4$, and $\delta_i \sim N(0, 1.5^2)$ under data-based error distributions.</td>
<td>92</td>
</tr>
<tr>
<td>4.1</td>
<td>Flow chart of analysis</td>
<td>103</td>
</tr>
<tr>
<td>4.2</td>
<td>NIC comparison for different $\lambda$</td>
<td>113</td>
</tr>
<tr>
<td>4.3</td>
<td>Model fit with 1D true covariate</td>
<td>114</td>
</tr>
<tr>
<td>4.4</td>
<td>Lfdr vs. $t$</td>
<td>115</td>
</tr>
<tr>
<td>4.5</td>
<td>Eigenvalues and NIC for pig fasting data</td>
<td>118</td>
</tr>
<tr>
<td>4.6</td>
<td>Correlatedness of gene lists</td>
<td>118</td>
</tr>
</tbody>
</table>
LIST OF ALGORITHMS

Algorithm 3.1 Pseudo-code for the SQP algorithm . . . . . . . . . . . . . . . . . . . . . . . . 77
Algorithm 4.1 Pseudo-code for the EM algorithm . . . . . . . . . . . . . . . . . . . . . . . . 123
ACKNOWLEDGEMENTS

I am deeply indebted to my co-major professors, Dr. Jack C.M. Dekkers and Dr. Dan Nettleton, for their extremely helpful guidance, patience, inspiration, and support during my Ph.D. study. I am also thankful for the excellent teaching by Dr. Dan Nettleton, Dr. Philip Dixon, Dr. Mark Kaiser, Dr. Stephen B. Vardeman, Dr. Song-Xi Chen, and other ingenious teachers at Iowa State University. The multi-disciplinary environment of the Bioinformatics and Computational Biology program provided the great opportunity to interact with Dr. Christopher K. Tuggle and other biologists on interesting real-world problems. Finally, I would like to thank my parents, my aunts, and other members of my family for their endless love and support.
Gene expression profiling experiments, in particular, microarray experiments, are popular in genomics research. However, in addition to the great opportunities provided by such experiments, statistical challenges also arise in the analysis of expression profiling data. The current thesis discusses statistical issues associated with gene expression profiling experiments and develops new statistical methods to tackle some of these problems.

In Chapter 2, we consider the insufficient sample size problem in detecting differential gene expression. We address the problem by developing and evaluating methods for variance model selection. The idea is that information about error variances might be learned from related datasets to improve the estimation of error variances. We develop a modified multiresponse permutation procedure (MRPP), modified cross-validation procedures, and the right AICc (corrected Akaike’s information criterion) for choosing a variance model. Through realistic simulations based on three real microarray studies, we evaluate the proposed methods and suggest practical recommendations for data analysis.

In Chapter 3, we address the multiple testing problem by improving the estimation of the distribution of noncentrality parameters given a large number of two-sample t-tests. We provide parametric, nonparametric and semiparametric estimators for the distribution of noncentrality parameters, as well as false discovery rates (FDR) and local FDR. Simulations show that our density estimates are closer to the underlying truth and that our estimates of FDR are also improved relative to competing methods under a variety of situations.

In Chapter 4, we develop a novel combination of two statistical techniques with the aim to by-pass the curse of dimensionality problem in detecting differential expression of genes. We accept the fact that, in “small N, large p” situations, the data are not sufficient to provide
enough information about dependency across genes. Hence, we suggest using a priori biological knowledge to assist statistical inference. We first use multidimensional scaling (MDS) methods to summarize prior knowledge about inter-gene relationships into a set of pseudo-covariates. Then, we develop a hierarchical additive logistic regression model conditional upon the generated pseudo-covariates. Simulations and analysis of real microarray data suggest that our strategy is more powerful than methods that do not use a priori information.

Future research directions are discussed at the end of the thesis.
CHAPTER 1. GENERAL INTRODUCTION

The introduction of “the central dogma of molecular biology” by Crick [1958, 1970] was a great success in outlining genetic information flow in a simple framework. Except for some contradictory examples, which always exist in any real science, and particularly in biology, the central dogma states that, for a vast majority of organisms, the genetic information is transferred from deoxyribonucleic acid (DNA) molecules to messenger ribonucleic acid (mRNA) molecules through transcription, and further to proteins through translation.

The coding in DNAs is completely, or nearly completely known, for a number of interesting organisms, in particular, humans [Venter et al., 2001; International Human Genome Sequencing Consortium, 2001]. However, the actual information contained in the DNA code largely remains a myth. In order to decipher the information stored in the DNA molecules, it is natural to consider its manifestations as mRNAs and proteins.

Recent advancement of genomics research is backed by novel high-throughput technologies. Although these new technologies target on various kinds of molecules, the most mature ones specifically measure, directly or indirectly, the number (or concentration) of mRNA molecules in a biological sample. Such gene expression profiling experiments have become more and more popular using these high-throughput technologies, especially the microarray technology.

The current thesis discusses novel statistical issues associated with gene expression profiling experiments, and develops new statistical methods to tackle some of these problems. In this introductory chapter, we briefly review (i) the microarray technology, as an example of high-throughput technologies, and a pig fasting experiment, (ii) statistical problems associated with the design of such experiments and the preprocessing of data, and (iii) statistical issues in the inference from such data. We end this chapter with a description of the organization of the
following chapters in this thesis.

1.1 High-throughput technologies and a fasting experiment in pigs

Microarrays are a range of devices that fix a large number of different probing chemicals onto a solid substrate at known positions. Under experimental conditions, a mixture of target chemicals can proportionally bind onto different probing chemicals on the solid substrate (nearly) specifically, such that a subsequent visualization of the solid substrate can be used to backtrack the presence and/or the amount of chemicals of interest in the mixture. Depending on the nature of probing chemicals and target chemicals, microarray technologies include nucleotide microarrays (complementary DNA [cDNA] arrays [Schena et al., 1995], short oligo arrays [Lockhart et al., 1996], long oligo arrays [Barczak et al., 2003]), protein microarrays [MacBeath and Schreiber, 2000] and antibody microarrays [Rivas et al., 2008], tissue microarrays [Kononen et al., 1998], etc. Although the underlying principles are straightforward, the real advantage of microarrays is their throughput—the number of different probing chemicals that can be attached onto the solid substrate. Traditionally, experiments based on similar chemical reactions can be performed on a small scale, often one at a time. Microarrays allow thousands to hundreds of thousands of such experiments to be conducted simultaneously.

In this section, we briefly review a gene expression profiling experiment [Lkhagvadorj et al., 2009] that studied the effects of fasting in pigs, using one of the most popular commercial microarray products, Affymetrix GeneChip® technology. The goal is not to review all technical details, but to allow the readers to better understand potential statistical problems discussed in Sections 1.2 and 1.3.

Lkhagvadorj et al. [2009] conducted an experiment using a total of 24 pigs. The biological treatments followed a $2 \times 2$ full factorial structure. Twelve of the pigs had genotype D298 for the melanocortin-4 receptor gene, and twelve had genotype N298. The pigs were either fed ad libitum or fasted for 3 days. The animals were treated in 4 groups of size 6, and the group served as the blocking factor. The primary interest of the researchers was to find the genes or sets of related genes that were transcribed differentially across genotypes and/or fasting. Also
of interest was to find or refine the relationship among the genes and to find a subset of genes that can be used to predict future observations.

The liver samples from the 24 pigs, one sample for each pig, were collected. Total RNAs were then extracted from each sample. The mRNAs in the extract were reversely transcribed to their corresponding cDNAs, and the cDNAs were replicated and in vitro transcribed to the complementary RNAs (cRNAs). The cRNAs have the same sequence as the mRNAs, except that they have incorporated fluorophores. The purpose of this process is to amplify the mRNAs proportionally to an amount above the detection limit, to label them with dyes, and to remove the large amount of RNA species other than mRNAs.

The microarray used in this experiment was the GeneChip® Porcine Genome Array (Affymetrix, Inc., Santa Clara, CA). Porcine gene sequences for the array were selected from UniGene and GenBank databases in August, 2004. A total of 23,256 transcripts were selected. Eleven, possibly overlapping, 25-nt (nucleotides) long regions of each transcript were then chosen to design the “probe set” for the transcript. Usually, only one probe set was designed for each transcript, but there were a small number of transcripts that corresponded to multiple probe sets. Each probe set consists of two 25-nt DNA oligos. One is called the perfect match (PM) probe, which is exactly complementary to the region of the mRNA transcript. The other is called the mismatch (MM) probe, which is the same as the PM probe except that its 13th nucleotide is complementary to that of the PM probe. Millions of each probe oligo are then chemically synthesized onto one of the probe cells on a glass slide, where probe cells are arranged in a 712 × 712 array format. All oligos within the same cell are identical. Rows of cells containing PM probes are intervened with those containing MM probes, and cells containing MM probes are always adjacent to the corresponding PM probe cells, but in different rows. The positions of the eleven pairs of PM and MM probes corresponding to the same mRNA transcript are randomized on the array, but their exact locations are known.

The prepared cRNAs from each pig were then broken up into shorter oligos and incubated with the microarrays overnight. Again, the sample from one pig always corresponds to one array. During incubation, the cRNAs hybridize with the corresponding probes on the microarray. For
an mRNA transcript, if there are more copies of the transcript in the original sample, there will be more corresponding cRNAs that are hybridized to the eleven pairs of probe cells on the microarray. After incubation, each microarray was washed and stained in one of the four modules of a fluidics station. Finally, each microarray was scanned under laser excitation. The probe cells with more cRNAs hybridized will contain more fluorophores, and will emit more light under laser excitation. Hence, the strength of light signal is indicating the amount of corresponding mRNAs in the original liver sample, and the resulting microarray images were used for further statistical analysis.

This is a typical gene expression profiling experiment for finding differentially expressed genes. There are various alternative choices that can be made by the researchers. For example, it is possible to split the biological samples from each pig into several subsamples and hybridize each subsample to a different microarray. It is also possible to use, instead of short oligo arrays, cDNA microarrays, where the biological samples are often labeled with multiple dyes and samples with different dyes can be hybridized to a single glass slide.

Of particular notice is the recent development of sequencing technologies that allow new types of differential expression experiments [Metzker, 2010]. These sequencing-based experiments are not unlike the earlier serial analysis of gene expression (SAGE) [Velculescu et al., 1995], except that the assay sensitivity is greatly increased. Although there are many variants that are still being invented and/or improved, they share a common feature that the nucleotide molecules of interest are broken up into very short sequences, and the number of each short sequence is counted, so that a larger count corresponds to a larger amount of nucleotides in the sample. Compared to microarray experiments, where the end results are array images, these counting experiments return counts and the corresponding short sequences as the results for further analysis.

A final note about the high-throughput technologies is that, based on similar chemical reactions, it is often possible to use these technologies for very different purposes other than expression profiling. Examples are array-based comparative genomic hybridization (aCGH) [Pollack et al., 1999] experiments for detecting copy-number variations, chromatin immuno-
precipitation followed by microarray hybridization (ChIP-chip) [Ren et al., 2000] for finding protein binding sites, fast genotyping of single nucleotide polymorphisms (SNPs) [Hacia et al., 1999], and alternative splicing detection via tiling arrays [Shoemaker et al., 2001].

### 1.2 Statistical issues in the design and preprocessing of expression profiling experiments

Despite the wide range of high-throughput technologies and applications thereof, we provide a schematic overview of the general procedures of statistical design and preprocessing of data from gene expression profiling experiments, which is currently the most common type of applications of these technologies.

In the design stage of the experiment, statistics plays at least two roles. One role is to assist the design of the technology. For example, for the aforementioned Affymetrix technology, choosing the 11 regions in the transcript could make use of probability models. The goal is to design probes that are both specific and sensitive. By specificity, we mean a probe should ideally only hybridize to mRNAs from a single gene, so that transcripts from different genes are not confounded. For example, the randomization of the positions of the 11 probe pairs from the same gene is a simple way to deal with non-specific signals. By sensitivity, we mean the probe should combine to the target strongly and be able to emit enough light when excited by laser. Sensitivity is especially important for reliable detection of lowly expressed genes. However, our knowledge about the thermodynamics of nucleotide hybridization at the interface between solid and liquid phase is rather limited. Much of the research is based on the thermodynamics in solutions, *e.g.*, on secondary structure models of RNAs [Mathews et al., 1999; Zhang et al., 2003].

The other role of statistics is to assist the design of treatments, identifying blocking factors, allocating experimental units to treatments, and sample size determination, among other tasks. The general goal is to reduce bias and variance. Considering the pig fasting experiment as an example, there were 4 treatments and 4 modules in a fluidics station. If a careless researcher used one module for all 6 microarrays from the same biological treatment, module would be
completely confounded with treatments, which could seriously bias the results. On the other hand, since the original experiment followed a block design with 4 blocks, if we intentionally use one module for all of the 6 microarrays coming from the same block, then we only need to use one combined factor to represent the variations caused by both block and module. This will save more error degrees of freedom and can reduce estimation variance [Qu et al., 2007]. For experiments using multi-color microarrays, the design is more complicated and is reviewed nicely in Churchill [2002].

After the experiment has been performed and data have been collected, the statistical analysis can be generally performed in two steps. The first step is the preprocessing of data, whose task includes image analysis [Kamberova and Shah, 2002], quality control [Bolstad et al., 2005; Brettschneider et al., 2008; Frueh, 2006], removing unnecessary technical variations not related with the underlying biological question [Yang et al., 2002], summarization of redundant data [Irizarry et al., 2003a,b; Bolstad et al., 2003], transformation of data [Durbin et al., 2002], imputation of missing data [Jörnsten et al., 2005], etc. The second step of the analysis is statistical inference, which we discuss in the next section.

The design of the experiment and the preprocessing of data is often highly dependent on the technology. Thus, it is more reasonable to develop a set of statistical techniques for each technology. Given the results from preprocessing, the statistical inference step is usually less dependent on the particular technology, and most of the inference tools share great similarities across technologies.

For the pig fasting experiment, Affymetrix GeneChip® technology was used. Because there are positive and negative control probes on the chip that are not part of the porcine genome, and because the in situ synthesis of oligos is rather precise, the image analysis step is relatively easy, compared to custom made cDNA microarrays. The GeneChip® Operating Software (GCOS) developed by Affymetrix, Inc. is almost always used in practice. For quality control, there are a range of parameters reported by GCOS. However, these parameters are designed to be useful for a single microarray and are largely experience-based rule-of-thumbs. They are not very useful for finding outlier arrays among all arrays in an experiment. Of notice
is the R package **affyPLM** [Brettschneider et al., 2008] provided by the Bioconductor project [Gentleman et al., 2004, http://www.bioconductor.org], which fits a linear model robustly to the probes and examines the residuals through boxplots or re-constructed pseudo-images. From our experience, it is usually very powerful in detecting anomalies.

Removing non-biological variations during preprocessing is often called *normalization*. Examples are matching the center of probe signals on each array to a common value, rescaling of the probe signals so that microarrays are more comparable to each other, and matching all empirical quantiles of probe signals across all microarrays, *etc.* [Yang et al., 2002]. These techniques are often found to be useful and are almost always applied. Unfortunately, unless we have technical replicates at every stage of the experiments, there is little justification from the data to guide the normalization process. What normalization has removed from or introduced to the data is largely dependent on the assumptions made about the underlying biology, which makes the normalization a rather subjective choice.

Summarization of signals from different probes corresponding to the same transcript is often applied for the Affymetrix technology. There are a large number of methods available. Irizarry et al. [2006] empirically found that the major factor causing differences among these methods was the way the MM probes are used. Methods that remove the MM signals from the PM signals fall in a large category, whereas methods that ignore the MM signals fall into another large category. The MM probes were originally designed to provide a local background estimate for the PM probes. However, because of the high similarity between MM probe sequences and PM probe sequences, MM probes end up capturing a lot of transcript signal, in addition to the local background. Irizarry et al. [2006] found that removing MM signals from PM signals often makes the final estimate of transcript abundance less biased, but also greatly increases the variances. Bolstad et al. [2003] argued that it is better to live with the bias for finding differentially expressed genes.

Sophisticated data transformation other than taking logarithms and imputation of missing data is sometimes needed for a particular data set, but is not part of routine practice and was not used for the pig fasting experiment. For more on transformation and imputation, please
see Durbin et al. [2002] and Jörnsten et al. [2005], among others.

1.3 Issues in statistical inference for expression profiling data

Once the data are preprocessed, the second step of the analysis is statistical inference, which is dependent on the goal of the experiment, but less dependent on the technology. Statistical inference for expression profiling data may include any of the following tasks: detection of differentially expressed genes [Kerr et al., 2000; Wolfinger et al., 2001; Cui et al., 2005; Tusher et al., 2001; Efron et al., 2001; Lönnstedt and Speed, 2002; Smyth, 2004], detection of differentially expressed gene categories [Dennis et al., 2003; Subramanian et al., 2005; Efron and Tibshirani, 2007; Nettleton et al., 2008], organization of genes and/or biological samples based on their similarities [Eisen et al., 1998; Getz et al., 2000; Kerr and Churchill, 2001] and construction of relationship networks [Stuart et al., 2003; Kim et al., 2003; Freeman et al., 2007], and building predictive models based on expression profiles [Ramaswamy et al., 2001; Tibshirani et al., 2002; Zhu and Hastie, 2004].

The major difficulty underlying almost all these inference tasks is the so-called “small N, large p” problem. That is, the available sample size for expression profiling experiments is often rather small, but the dimensionality is huge due to the high-throughput parallel measurements. For the pig fasting data, only a total of 24 pigs are available. However, 24 should be considered as a “large” experiment compared to others. On April 22, 2010, we queried the major public expression data database, Gene Expression Omnibus (GEO) [Barrett et al., 2005], and found that more than 40% of the datasets had a total sample size no larger than 10, and that this percentage did not change significantly over the past decade (Table 1.1). On the other hand, the dimensionality of a typical microarray dataset is often over tens of thousands, which is much larger than the sample size. Hence, almost all datasets are subject to the curse of dimensionality [Bellman, 1961].
Table 1.1  Proportion of studies with total sample size no larger than 10 in the GEO database (April 22, 2010)

<table>
<thead>
<tr>
<th>Publication year</th>
<th>No. of studies with $n \leq 10$</th>
<th>Total No. of studies</th>
<th>Proportion (se) with $n \leq 10$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010~</td>
<td>927</td>
<td>2071</td>
<td>0.45 (0.01)</td>
</tr>
<tr>
<td>2009</td>
<td>2563</td>
<td>5738</td>
<td>0.45 (0.01)</td>
</tr>
<tr>
<td>2008</td>
<td>1958</td>
<td>4430</td>
<td>0.44 (0.01)</td>
</tr>
<tr>
<td>2007</td>
<td>1938</td>
<td>4293</td>
<td>0.45 (0.01)</td>
</tr>
<tr>
<td>2006</td>
<td>1410</td>
<td>3335</td>
<td>0.42 (0.01)</td>
</tr>
<tr>
<td>2005</td>
<td>1093</td>
<td>2714</td>
<td>0.40 (0.01)</td>
</tr>
<tr>
<td>2004</td>
<td>716</td>
<td>1662</td>
<td>0.43 (0.01)</td>
</tr>
<tr>
<td>2003</td>
<td>530</td>
<td>1031</td>
<td>0.51 (0.02)</td>
</tr>
<tr>
<td>2002</td>
<td>117</td>
<td>288</td>
<td>0.41 (0.03)</td>
</tr>
<tr>
<td>2001</td>
<td>16</td>
<td>47</td>
<td>0.34 (0.07)</td>
</tr>
<tr>
<td>~2000</td>
<td>1</td>
<td>2</td>
<td>0.50 (0.35)</td>
</tr>
</tbody>
</table>

1.3.1 Statistical problems in detecting differentially expressed genes

For the task of finding differentially expressed genes, the common practice is to perform a statistical hypothesis test for each gene [Kerr et al., 2000; Wolfinger et al., 2001]. The problem of insufficient sample size greatly affects the estimation of error variability. For example, a two-sample comparison experiment with $n = 3$ in each treatment group has only 4 error degrees of freedom available for estimating the error variance. If we assume that the estimates of error variances follow a scaled $\chi^2$ distribution with $\nu = 4$ degrees of freedom, the skewness of the distribution is $\sqrt{8/\nu}$, which is strictly positive and decreasing over $\mathbb{R}^+$. Hence, most of the probability is allocated to the region below the expectation. In other words, with very small sample sizes, it is very likely to get an estimate of the error variance that is close to zero, even though the estimator might be unbiased. Moreover, because the number of tests is huge, the probability of getting a nearly zero estimate is also non-ignorable. If we use such an estimate in the two-sample $t$-test, the absolute $t$-statistic will be very large due to a nearly zero denominator. Hence, no matter if the gene is differentially expressed, the $t$-test will report this gene as highly significant.

A frequently used strategy for this problem is to regularize (stabilize) the estimates of
error variances. There are a large number of such methods in the literature. They ameliorate
the problem by improving gene-wise variance estimation through borrowing information from
other genes within a single dataset. For example, Tusher et al. [2001] and Efron et al. [2001]
employed ad hoc modifications to the variance estimator, Kamb and Ramaswami [2001], Jain
et al. [2003], Geller et al. [2003], Durbin et al. [2002] and Huang and Pan [2002] estimated
the error variances by considering its relationship with the means, and Baldi and Long [2001],
Lönnstedt and Speed [2002], Smyth [2004], Cui et al. [2005] and Hwang and Liu [2007] assumed
hierarchical models over the error variances and developed empirical Bayes estimators. Tong
and Wang [2007] also considered the optimality problem of shrinkage variance estimation.

The other problem for a dataset with small sample sizes is the degree of approximation
of the central limit theorem, which is the basis for the use of normality-based tests, e.g., the
F-test in analysis of variance (ANOVA), in large samples. Unfortunately, with microarray
data, it is difficult to assess whether the error distribution is sufficiently close to normality and
whether the central limit theorem provides a good approximation. In practice, non-asymptotic,
distribution-free, permutation tests are often preferred for detecting differentially expressed
genes. Permutation tests actively make use of the symmetry structure in the data under
the null hypothesis [Efron and Tibshirani, 1993; Lehmann and Romano, 2005b] and provide
control of false positives (type I errors) irrespective the underlying error distribution. However,
permutation tests also suffer from small sample sizes. If the sample size is too small, the
cardinality of the reference set of the permutation test statistic is rather small, which greatly
reduces the power of the test due to the discrete nature of permutation p-values. For example,
for a two-treatment permutation t-test with \( n = 3 \) samples for each treatment, the smallest
possible permutation \( p \)-value for a two-sided test is 0.1. That is, the power of the test is zero if
we do not tolerate a type I error rate at least 0.1. The lack-of-continuity of permutation \( p \)-values
is often alleviated by pooling (taking the union of) gene-wise reference sets of test statistics
[Tusher et al., 2001; Efron et al., 2001; Pan, 2003]. Such practice allows smaller permutation
“\( p \)-values” to be reported. However, strictly speaking, these are no longer valid \( p \)-values, unless
strong assumptions about the relationship among the genes are made.
The other difficulty in using the permutation test for analysis of microarray data is that the range of applications is rather limited. Permutation tests are only possible when the null hypothesis exhibits symmetry structures. Examples are $K$-treatment comparison designs, blocked designs, factorial designs without interactions, etc. For more complicated designs, e.g., in the presence of continuous covariates, non-trivial permutation tests are often not possible.

Another class of problems associated with the detection of differential gene expression is the multiplicity of tests. Before the availability of high-throughput technologies, multiple testing problems had been studied for decades, but at a scale of at most tens of tests [Miller, 1981; Westfall and Young, 1993; Hsu, 1996]. Their aim was to control the type I error rate for a family of tests to be below a pre-specified limit. Unfortunately, if the family is too large, e.g., a family defined as all gene-wise tests for differential expression, the power of these correction procedures is often too low to be useful in practice. Usually, only hypotheses with a $p$-value no larger than the order of $10^{-6}$ can be rejected under family-wise type I error rate (FWER) control for microarray data.

A remedy to the large-scale multiple testing problem is to control a less restrictive error rate while still guarding against too many uninteresting discoveries. The most commonly used error rates for expression profiling data are all of the form of controlling some variant of the proportion of false rejections. These include the nominal work of false discovery rates (FDR) [Benjamini and Hochberg, 1995], positive false discovery rates (pFDR) [Storey, 2003], marginal FDR (mFDR) [Tsai et al., 2003] or proportion of false positives [Fernando et al., 2004], local false discovery rates (LFDR) [Efron, 2005], etc. Other choices of the error rates are possible, e.g., generalized family-wise error rates ($k$-FWER) and false discovery proportions (FDP) [Lehmann and Romano, 2005a], but currently they are not widely applied.

A particular problem associated with controlling these less restrictive error rates is that they all require knowledge of the proportion $\pi_0$ of true null hypotheses among the family of tests under consideration. Unfortunately, without enough parametric assumptions, $\pi_0$ is often not identifiable [Genovese and Wasserman, 2002]. Most of the estimators of $\pi_0$ are actually estimating an upper bound of $\pi_0$, e.g., those of Storey and Tibshirani [2003] and Mosig et al.
Such procedures provide a wide safety margin, at the cost of loss of power. Two exceptions are Lai [2007] and Ruppert et al. [2007]. Lai [2007] assumed the availability of two independent tests for each hypothesis and developed moment estimators for $\pi_0$. Although the idea is clever, our experience is that its performance in estimating $\pi_0$ and improving power is often not superior to estimators of the upper bound. On the other hand, Ruppert et al. [2007] avoided the unidentifiability problem by assuming a parametric model on the alternative hypotheses, but their parametric model is so flexible that it can be regarded as nearly nonparametric for practical purposes. Through simulations and analysis of real data, Ruppert et al. [2007] demonstrated very promising results using their estimators.

Another difficulty for the multiple testing problem is the complicated dependence among the tests. The most commonly used procedures often either rely on the independence (or, at least, exchangeability) assumption across genes or ignore such dependencies. Procedures that allow dependence are available [Benjamini and Yekutieli, 2001], but they often suffer from lack of power. The dependency problem is also related with the curse of dimensionality, because the sample size is often insufficient for learning the dependency structure across the large number of genes. However, when permutation tests are applicable, dependence information can often be actively utilized in controlling error rates, so that the power could theoretically be improved compared with ignoring the dependency [Westfall and Young, 1993; Yekutieli and Benjamini, 1999; Nettleton et al., 2008].

### 1.3.2 Statistical problems in gene set analysis

Besides finding differentially expressed genes, biologists often need to find which categories of genes are differentially expressed under different experimental conditions. The categories are often defined *a priori*, using gene ontologies (GO) [The Gene Ontology Consortium, 2000], biochemical pathways [Kanehisa and Goto, 2000], protein interactions [Xenarios and Eisenberg, 2001], *etc.* Many researchers conducted such analysis in two steps. First find a list of genes declared as differentially expressed. Then perform a test of independence using, *e.g.*, a Fisher’s exact test or a $\chi^2$ test, to find the gene categories that are over-represented in the list. This
practice assumes independence across genes and has received many critics [Khatri and Drághici, 2005; Allison et al., 2006; Goeman and Bühlmann, 2007]. However, due to the simplicity and the availability of easy to use software and/or web services [Dennis et al., 2003], such problematic practices are still prevalent in biological literature.

An alternative strategy of gene set analysis is to use a correct permutation test that accounts for the dependency across genes. Barry et al. [2005] developed such a general framework for performing hypothesis tests on the gene list by combining gene-wise statistics in the list. The framework includes many implementations and variants [Efron and Tibshirani, 2007], including the popular GSEA (gene set enrichment analysis) method [Subramanian et al., 2005]. This is not unlike the p-value pooling methods proposed by Fisher [1970] and discussed in detail by Pesarin [2001], except that the test statistic to be combined is not necessarily a p-value.

Another promising approach is to by-pass the construction of gene-wise test statistics and to directly test differential expression in multiple dimensions. The multi-response permutation procedure (MRPP) [Mielke and Berry, 2007] has been applied to this problem by Nettleton et al. [2008]. We can consider this approach as a nonparametric, robust, multivariate analysis of variance (MANOVA). But the procedure has a great advantage over the MANOVA—not requiring the sample size to be larger than the number of dimensions. At first glance, this seems to say that MRPP is free from the curse of dimensionality problem. However, this is not correct. Although MRPP can be applied to data in any dimensions, it does not have nontrivial power against any high-dimensional departures from the null hypothesis. However, empirical evidence suggests that MRPP is usually more powerful for typical departures from the null than alternative methods, especially if the departure only involves changes in correlations among genes but not necessarily changes in means. One difficulty for the MRPP approach that by-passes a gene-wise statistic is that once a gene set has been declared as differentially expressed, it is not clear which subset of genes contributes to the declared difference. Univariate statistics are not suitable for this purpose because they can only measure marginal differences, instead of measuring importance within the context of the multivariate MRPP.

Finally, the multiplicity of tests is still a problem for detection of differentially expressed gene
sets, because the number of gene sets is still quite large. Although the problem of dependence across the test statistics is more severe in this case, there usually exist a priori structures that can provide information about dependence. For example, the gene ontology project defines a directed acyclic graph (DAG) structure of the gene sets, where genes in a parent node of the graph always include the union of genes of its child nodes [The Gene Ontology Consortium, 2000]. Several methods have been proposed to actively utilize such structures in multiple testing corrections [Goeman and Mansmann, 2008; Liang and Nettleton, 2010].

1.3.3 Statistical problems in clustering and classifications of microarray data

Compared with statistical testing and/or estimation, clustering and classification problems are sometimes considered as less formal, black-box techniques, whose usefulness in scientific research is subject to debate [Breiman, 2001]. Nevertheless, such techniques remain popular in the analysis of expression data since early applications of microarrays [Eisen et al., 1998; Ramaswamy et al., 2001].

Clustering can be performed to group similar genes, expression profiles, or both [Getz et al., 2000]. Many clustering algorithms have been proposed in the literature. Most are based on inter-object distances (here the object could refer to either genes or biological samples). Examples are hierarchical clustering, $k$-means clustering, $k$-medoids clustering, etc. [Everitt et al., 2001]. Other related clustering techniques are based on density estimation. Examples are finite mixture models [McLachlan and Peel, 2000], infinite (but discrete) mixture models [Medvedovic et al., 2004], bump hunting [Friedman and Fisher, 1999], DBSCAN [Ester et al., 1996], etc.

There are several major issues associated with cluster analysis. The first is that, frequently, it is not very clear what biologists expect from cluster analysis. The specification of the goals is often rather vague. And it is not clear what cluster membership means for expression data, except for a few special cases where searching for co-regulated genes is the goal. This problem is in contrast to traditional applications of clustering algorithms to evolutionary problems, where there exist underlying genetic models that support the use of such techniques. The
second problem with cluster analysis is associated with the curse of dimensionality. That is, for
distance-based clustering methods, when there are too many pairwise distances to be computed
from a data set with very small sample sizes, the noise will essentially make up most of the
“interesting” inter-object relationships. Similarly, for density based clustering methods, accurate
estimation of density in high dimensions is extremely difficult, because most of the volume in
high dimensions is near the boundary. The third problem with cluster analysis for expression
profiling data is computational efficiency. Because the amount of data is huge, algorithms
with high complexity in either time or space requirements are not useful. However, there have
been some advances in this aspect [Ester et al., 1996; Zhang et al., 1996]. The fourth problem,
validation of clustering results, is associated with the first problem. Often resampling based
methods, \textit{e.g.}, jackknife or bootstrap [Kerr and Churchill, 2001; Shimodaira, 2004], are used to
assess the stability of clustering results. However, without a consensus on what a \textit{true} cluster
means, such efforts are only of theoretical interest.

Clustering methods are often called unsupervised learning in the machine learning literature,
whereas classification methods are called supervised learning. Classification can be thought as a
generalization of clustering that has an additional response variable of interest. The underlying
response variable for a clustering method can be thought as the density. When the response
variable is continuous, classification is a synonym of regression. However, the major goal for
classification is prediction or forecasting. There are many applications of classification methods
using gene expression profiling data [Ramaswamy et al., 2001; Tibshirani et al., 2002; Zhu and
Hastie, 2004]. A similar problem is the genomic selection method for prediction of breeding
values based on a large number of marker genotypes in animal/plant breeding [Meuwissen et al.,
2001], and genome-wise association studies (GWAS) in medicine [McCarthy et al., 2008].

There are at least two major problems for these predictive modeling techniques. The first
one is, again, the curse of dimensionality [Hastie et al., 2009]. In a small dataset with a huge
number of dimensions, even if all predictors are purely random noise, it is still possible to
perfectly classify the data. The generalization error will, of course, be very large. However,
with insufficient sample sizes, the estimation of generalization error is a difficult problem by
itself. Thus, avoiding overfitting in high dimensions is rather difficult. The second problem with classification techniques is the dilemma of prediction ability and interpretability. Ideally, a good learning algorithm should both predict well and be able to be translated into human understandable knowledge. However, for a given dataset, good prediction algorithms are often very hard for biologists to interpret, whereas easy to understand methods are often too simplistic to predict future observations well. The associated problem is that prediction ability is relatively easier to measure objectively using data, but interpretability is often subjective and lacks numerical measures. Hence, which aspect is more important has to be determined case by case.

In summary, our view of these informal, black-box type statistical techniques is that they may be useful for generation of good hypotheses if they are used in conjunction with good, subjective, human knowledge. However, for an objective confirmation of scientific discovery, it is better to avoid these techniques.

1.4 Thesis organization

Despite the wide range of statistical problems that need to be solved for analysis of gene expression profiling data, the current thesis only considers a selected subset of the topics. The topics in the following chapters are related to borrowing information to detect differential expression of genes, where Chapter 2 considers borrowing information from other datasets, Chapter 3 considers borrowing information across genes within a single dataset, and Chapter 4 considers borrowing information from prior knowledge about inter-gene relationships.

In Chapter 2, we consider the insufficient sample size problem in detecting differential gene expression. We address the problem by developing and evaluating methods for variance model selection. The idea is that information about error variances might be learned from related datasets to improve the estimation of error variances. We develop a modified multiresponse permutation procedure (MRPP), modified cross-validation procedures, and the right AICc (corrected Akaike’s information criterion) for choosing a variance model. Through realistic simulations based on three real microarray studies, we evaluate the proposed methods and
suggest practical recommendations for data analysis.

In Chapter 3, we address the multiple testing problem by improving the method of Ruppert et al. [2007]. Given a large number of two-sample t-tests, we provide parametric, nonparametric and semiparametric estimators for the distribution of noncentrality parameters, as well as FDR and local FDR. Simulations show that our density estimates are closer to the underlying truth and that our estimates of FDR are also improved relative to those of Ruppert et al. [2007] under a variety of situations.

In Chapter 4, we develop a novel combination of two statistical techniques with the aim to by-pass the curse of dimensionality problem in detecting differential expression of genes. We accept the fact that, in “small N, large p” situations, the data are not sufficient to provide enough information about the dependency across genes. Hence, we suggest using a priori biological knowledge to assist statistical inference. We first use multidimensional scaling (MDS) methods to summarize prior knowledge about inter-gene relationships into a set of pseudo-covariates. Then, we develop a hierarchical additive logistic regression model conditional upon the generated pseudo-covariates. Simulations and analysis of real microarray data suggest that our strategy is more powerful than methods that do not use a priori information.

Chapter 5 discusses future research directions and concludes the thesis.

Throughout Chapters 2 to 4, Long Qu developed the proposed methodologies, conducted evaluations, and wrote the initial manuscript of each chapter; Dan Nettleton suggested revisions to the proposed methods and/or evaluations and contributed significantly to the writing; Jack C.M. Dekkers suggested further questions to be discussed in each chapter and contributed to the writing; both Jack C.M. Dekkers and Dan Nettleton provided funding to the research in each chapter. In addition, for Chapter 2, Jack C.M. Dekkers provided the initial problem to be solved and Nicola Bacciu participated in the initial explorations to the problem. For Chapter 3, the research was based on an earlier study [Ruppert et al., 2007] co-authored by Dan Nettleton. For Chapter 4, the real microarray dataset analyzed came from a project supervised by Jack C.M. Dekkers.

A final note is that each chapter is self-contained and can be read independently. Similarly,
the scope of mathematical notations is specific to its chapter. The same notation could have
different meanings in different chapters. However, because chapter topics are not overlapping,
this re-use of notations should not cause confusion.

Bibliography


expression data: regularized t-test and statistical inferences of gene changes. Bioinformatics
17, 509–519.

Barczak, A., Rodriguez, M. W., Hanspers, K., Koth, L. L., Tai, Y. C., Bolstad, B. M., Speed,
analysis. Genome Research 13, 1775–85.

Barrett, T., Suzek, T. O., Troup, D. B., Wilhite, S. E., Ngau, W.-C., Ledoux, P., Rudnev, D.,

categories in gene expression studies: a structured permutation approach. Bioinformatics

University Press.

powerful approach to multiple testing. Journal of the Royal Statistical Society. Series B
(Methodological) 57, 289–300.


*Biostatistics* 5, 427–443.
CHAPTER 2. VARIANCE MODEL SELECTION WITH APPLICATION TO JOINT ANALYSIS OF MULTIPLE MICROARRAY DATASETS UNDER FALSE DISCOVERY RATE CONTROL

A paper accepted by Statistics and Its Interface
Long Qu, Dan Nettleton, Jack C.M. Dekkers, Nicola Bacciu

Abstract

We study the problem of selecting homogeneous variance models vs. heterogeneous variance models in the context of joint analysis of multiple microarray datasets. We provide a modified multiresponse permutation procedure (MRPP), modified cross-validation procedures, and the right AICc (corrected Akaike’s information criterion) for choosing a variance model. In a simple univariate setting, our modified MRPP outperforms commonly used competitors. For microarray data analysis, we suggest using the sum of gene-specific selection criteria to choose one best gene-specific model for use with all genes. Through realistic simulations based on three real microarray studies, we evaluated the proposed methods and found that using the correct model does not necessarily provide the best separation between differentially and equivalently expressed genes, but it does control false discovery rates (FDR) at desired levels. A hybrid procedure to decouple FDR control and differential expression detection is recommended.
2.1 Introduction

2.1.1 Joint analysis of from multiple microarray datasets

Microarrays are a popular tool in genomic expression profiling studies for discovering genes that respond to treatments of interest. The measurements from each experimental unit in such studies are very high dimensional expression vectors, ranging from thousands to tens or hundreds of thousands of genes, far exceeding the available sample sizes. This creates a hurdle to analyzing the data from all genes simultaneously using traditional multivariate methods.

A popular and practical choice is to fit a univariate model, e.g., a linear model or a linear mixed model [Wolfinger et al., 2001], to each gene separately. For each contrast of interest, the analysis usually results in a \( p \)-value for each gene. This set of \( p \)-values is then summarized so that a certain error rate is controlled at a pre-specified level. False discovery rate (FDR) and variants thereof have become the de facto standard choices for this purpose [Benjamini and Hochberg, 1995; Storey and Tibshirani, 2003] due to the exploratory nature of microarray expression studies.

A major problem of this approach is that the power for detecting interesting genes is usually very low because 1) the cost of microarrays hinders most researchers from using moderate to large sample sizes, 2) the measurements produced by the current technology are rather noisy, and 3) the pre-existing biological variation among experimental units is often large. Although it may be reasonable to expect sample sizes to increase as the cost of technology decreases, an examination of the microarray datasets hosted in the Gene Expression Omnibus (GEO) database [Barrett et al., 2005] indicates that the percentage of datasets with 10 or fewer experimental units has remained relatively steady at above 40% nearly every year over the past decade. Combining data from multiple datasets collected by a single lab or by several labs offers one way to address the persistent problem of insufficient sample size.

In this paper, we explore the variance model selection problem for joint analysis of multiple microarray datasets to improve the detection of differentially expressed genes using gene-wise linear models. In particular, we focus on how to determine if gene-specific estimates of error
variance should be pooled across datasets and used for testing linear contrasts of means within a dataset. We assume that we have data from several similar datasets using the same type of microarray and biological samples from the same or very similar populations. Our main interest lies in testing linear contrasts within some datasets. This differs from the usual meta-analysis in that different datasets do not necessarily involve the same sets of treatments. Multiple datasets are used only to provide better variance estimates, instead of better estimates of means.

More specifically, the problem we consider can be described as follows. Suppose we have $K$ independent data matrices, $Y_1, Y_2, \ldots, Y_K$, which correspond to $K$ independent sets of high-dimensional vector-valued observations. We will refer to $Y_k$ as a dataset and assume that $Y_k$ is of dimension $G \times n_k$, with rows corresponding to genes and independent columns corresponding to vector-valued observations for all $k = 1, 2, \ldots, K$. Further, we assume that $E(Y_k) = B_k'X_k'$, where $X_k$ is an $n_k \times p_k$ design matrix with full column rank and $B_k$ is a $p_k \times G$ matrix of unknown parameters. For example, suppose the $k$th dataset follows a commonly used (unpaired) two-treatment design with 3 biological samples for each treatment. $X_k$ could be $I_2 \otimes 1_3$, where $I_2$ is a $2 \times 2$ identity matrix, $\otimes$ stands for Kronecker product, and $1_3$ is a vector of 3 ones. The corresponding $B_k$ will be two rows of unknown mean parameters. Each row represents a treatment, and each column represents a gene. Finally, we assume that the covariance matrix of any column of $Y_k$ is $\Sigma_k$ for all $k = 1, 2, \ldots, K$. In this paper, we develop procedures for assessing whether a model assuming $\text{diag}(\Sigma_1) = \text{diag}(\Sigma_2) = \cdots = \text{diag}(\Sigma_K)$ should be selected for making inference about $B_1, B_2, \ldots, B_K$.

2.1.2 Model selection

The aforementioned question is a model selection problem with respect to variances. In general, model selection has long been studied, and there are at least three major approaches. The first is to perform hypothesis testing, e.g., a likelihood ratio test, and to choose a relatively simple model as long as there is no obvious evidence showing that the more complicated alternative models fit the data much better. In our problem, we could begin by testing for equality of variance across all $K$ datasets within all genes. If the null hypothesis is not rejected,
then pooling variance estimates across all datasets within each gene might be a good choice. If
the null hypothesis is rejected, then a refined form of the variance model could be formulated
and tested. For example, if one dataset seems to have increased variation relative to the rest, a
model that allows the variances for that dataset to differ from the variances for other datasets
could be proposed and tested. This procedure can be repeated until a sufficiently rich variance
model is judged to adequately fit the data according to the testing procedure. The second
approach is to order the candidate models according to some criterion and to choose the model
that optimizes the criterion, e.g., Akaike’s information criterion (AIC) [Akaike, 1973, 1974],
Schwarz Bayesian information criterion (BIC) [Schwarz, 1978], or prediction sums of squares
(PRESS) [Allen, 1974]. The third approach is to employ Bayesian techniques and to choose
models by summarizing posterior distributions [Spiegelhalter et al., 2002]. Because Bayesian
methods often rely heavily upon Monte Carlo simulations that are very computationally
intensive for large microarray datasets, we only discuss the first two approaches in this study.

In the context of choosing an appropriate variance model for joint analysis of multiple
datasets, the hypothesis testing approach requires a good test for heterogeneity in variances.
For univariate data, the likelihood ratio tests, i.e., the F-test for two-sample comparison and
Bartlett’s test [Bartlett, 1937; Snedecor and Cochran, 1989] for general one-way designs, are
known to be sensitive to departures from normality. If the normality assumption holds, they
have very good theoretical properties, but in practice, robust tests for variances are often
recommended. For example, Levene’s test performs one-way ANOVA on absolute residuals
from a least squares fit [Levene, 1960]. The Brown-Forsythe test is similar but computes
residuals from a least absolute deviations fit [Brown and Forsythe, 1974]. These tests are only
approximate, and, as will be seen in our simulations, the approximation is often very poor
under typical small sample sizes in microarray datasets. Thus, the resulting nominal p-value
distribution deviates far from the theoretical uniform distribution on the unit interval. In
this study, we propose an alternative permutation based procedure that better controls type I
error and has good power. Moreover, it can be automatically applied to any high-dimensional
dataset.
Commonly used model selection criteria can also be classified into three categories. Methods in the first category seek models that minimize some estimate of the prediction sums of squares. In linear models, Mallows’ $C_P$ [Mallows, 1973] and PRESS are widely used methods for such a purpose. However, these methods are designed for selecting mean structures, i.e., regressors of the model. In this study, we propose two new cross-validation measures designed specifically for differentiating alternative variance models.

The second category of model selection criteria, exemplified by AIC, Takeuchi’s information criterion (TIC) [Takeuchi, 1976], and corrected AIC (AICc) [Hurvich and Tsai, 1989], includes methods that approximate expected estimated Kullback-Leibler divergence as a criterion to rank candidate models. These methods have firm information-theoretic justifications [Akaike, 1973, 1974] and are known to be asymptotically efficient [Shibata, 1980, 1981]. They do not intend to choose a smallest true model asymptotically but to choose a good approximate model based on the available amount of data, because the true model may be infinite dimensional and fall outside the set of candidate models. Among the three methods, AIC is a special case of TIC, but TIC is difficult to estimate and is rarely used. The AICc is a bias-corrected version of AIC, but its derivation is model dependent. Unfortunately, common software implementations often ignore this fact and compute a panacea version of AICc that assumes homogeneity in variances. For example, as of version 9.2 of SAS/STAT [SAS Institute, Inc., 2008], neither the PROC MIXED procedure nor the PROC GLIMMIX procedure reports the correct AICc when a GROUP option is used in the REPEATED or RANDOM statement to specify heterogeneity of variances. Hence, in this study, we will provide the correct AICc formula for linear models with heterogeneous variances.

The third category of model selection criteria includes model dimension consistent criteria, e.g., BIC, Hannan and Quinn’s information criterion (HQIC) [Hannan and Quinn, 1979] and Bozdogan’s consistent AIC (CAIC) [Bozdogan, 1987]. These methods have the property that when the true models are indeed in the set of candidate models, then as sample size increases to infinity, a true model with the smallest model dimension will be selected. However, when the candidate set does not include any true models, these criteria asymptotically choose a small approximate model based on Kullback-Leibler divergence. For the subtle difference compared
with the second category of criteria, see [Burnham and Anderson, 2002].

A completely different strategy to model selection is regularizing parameter estimates through various shrinkage methods. In the context of microarray data analysis, many such methods have been proposed in the literature to combat the insufficient sample size problem in estimating variances. Most of these methods do not require the use of datasets from other studies. They aim to improve gene-wise variance estimation by pooling information from other genes within a single dataset. Examples of such methods are ad hoc modifications [Tusher et al., 2001; Efron et al., 2001], estimates based on mean-variance relationship [Kamb and Ramaswami, 2001; Jain et al., 2003; Geller et al., 2003; Durbin et al., 2002; Huang and Pan, 2002], and hierarchical model based estimates [Baldi and Long, 2001; Lönnstedt and Speed, 2002; Smyth, 2004; Cui et al., 2005]. In this paper, we pick the very popular limma method [Smyth, 2004] as a representative of such shrinkage estimation methods and compare its performance with our model selection approaches.

2.2 Methods

In subsections 2.2.1 through 2.2.3, we propose three independent variance model selection approaches. We develop an approximate permutation test for testing homogeneity of high dimensional spread in subsection 2.2.1. In subsection 2.2.2, we propose cross-validation methods for choosing between variance models. Under normal theory linear model assumptions, the correct AICc formula for selection of the variance model is derived in subsection 2.2.3. Together with other information criteria, we suggest in subsection 2.2.4 to use the sum of information criteria across genes as a means to select a common variance model for all genes in the analysis of multiple microarray datasets. The performance of these methods is assessed through simulations based on real microarray data in subsection 2.2.5 and section 2.3.

2.2.1 Modified MRPP for testing homogeneity of variances

The multi-response permutation procedure (MRPP) [Mielke and Berry, 2007] is a multivariate permutation test that has been successfully applied to the analysis of gene sets for microarray
data [Nettleton et al., 2008], with the advantage that the dimension of the response variable needs not be less than the sample size. For a $K$-treatment design, the usual MRPP statistic is constructed in two steps. First, average pairwise Euclidean distances across observations within each treatment group are calculated as a measure of spread within the treatment group. Next, the test statistic is constructed by a weighted sum of the $K$ average pairwise distances, which is largely motivated by the decomposition of sums of squares in the usual analysis of variance (ANOVA). The treatment labels are then randomly shuffled a large number of times, and the test statistic is re-computed for each shuffling. The $p$-value is reported as the proportion of shufflings that result in a test statistic no larger than the one observed for the original data before shuffling.

There are two obstacles in directly applying the usual MRPP to test the equality of variances across multiple microarray datasets. First, the usual MRPP test statistic is rather insensitive to changes in spread of the multivariate distribution because it is mainly designed for detecting mean differences, as in ANOVA. Second, within each microarray dataset, observations are not exchangeable due to the differences in means across treatments within datasets and the MRPP statistic does not account for this more complicated mean structure.

To overcome these problems, we first conduct a complete QR decomposition for each $X_k$,

$$X_k = Q_k R_k = [Q_{k1}, Q_{k2}] [R_{k1}', 0]'$$

where $Q_k$ is an $n_k \times n_k$ orthonormal matrix with the first $p_k$ columns being $Q_{k1}$ and the remaining columns being $Q_{k2}$, and $R_k$ is an $n_k \times p_k$ matrix with the first $p_k$ rows being an upper triangular matrix $R_{k1}$ and the remaining rows being zero.

To remove the mean structure in $Y_k$, it is trivial to check that the orthogonal projection matrix (the hat matrix) that projects onto the column space of $X_k$ is $Q_{k1} Q_{k1}'$, so that the residual matrix $Y_k (I_k - Q_{k1} Q_{k1}')$ has mean zero, where $I_k$ is the $n_k \times n_k$ identity matrix. Mielke and Berry [Mielke and Berry, 2007] directly used the columns of residual matrices similar to these to perform permutation tests. However, this is problematic because 1) the columns within the same residual matrix are correlated, whereas columns from different residual matrices are independent, and 2) the covariance matrices corresponding to columns within any residual
matrix are not necessarily identical. Hence, even if the null hypothesis is true, the fundamental assumption of exchangeability for the permutation test is violated.

To improve exchangeability, we suggest using a transformation that simultaneously removes the means, decorrelates the columns, and standardizes the columns. Specifically, we define $Z_k = Y_k Q_{k2}$ for all $k = 1, 2, \ldots, K$ and propose to use the columns of $Z_1, Z_2, \ldots, Z_K$ as data for an MRPP-based (approximate) test of variance equality. Note that each $Z_k$ is $G \times d_k$ instead of $G \times n_k$, where $d_k = n_k - p_k$ is the error degrees of freedom for dataset $k$. This reduction makes sense since after accounting for the means, the effective sample size for the variances is actually $d_k$ for the $k$th dataset. A similar reduction is used for residual maximum likelihood [Patterson and Thompson, 1971, REML] estimation of variance components in mixed linear models. Tests not accounting for the loss of information in estimating the location parameters, e.g., the commonly used Levene’s test and the Brown-Forsythe test, will likely lead to poorly controlled type I errors, as verified through simulations in the next section. Also, by using the orthogonality of the columns of $Q_k$, it can be easily checked that the columns of $Z_k$ are uncorrelated and the covariance matrix of any column of $Z_k$ is identically $\Sigma_k$, the same as the covariance matrix of any column in $Y_k$. Although uncorrelatedness implies neither independence nor exchangeability among the columns, unless normality is further assumed, $Z_k$ does offer a better candidate than the raw residuals for approximate permutation tests, by alleviating the problem up to the second order moments.

Lastly, we need a test statistic based on these $Z$ matrices. Both steps in constructing the usual MRPP statistic need some modification. First, the measure of within-treatment spread no longer needs the “pairwise” concept, because we now know the location of $Z_k$, i.e., $E(Z_k) = 0$. Actively using this location information will lead to more powerful tests. To keep our approach closely related to the well established MRPP statistic, we choose the spread measure to be the average Euclidean distance of each column of $Z_k$ to the origin and denote this as $\delta_k$. Second, for testing the equality of spreads across the $K$ datasets, the weighted sum of $\delta_k$ used in the
usual MRPP formulation is no longer sensible. Instead, we propose to use
\[
\min_{k=1,2,\ldots,K} \frac{\delta_k}{\max_{k=1,2,\ldots,K} \delta_k}
\]
as the test statistic, which is similar to the \(F_{\text{max}}\) statistic of Hartley [1950]. Analogous to standard permutation procedures, the dataset labels for columns of \(Z_k\)'s are then shuffled a large number of times and the \(p\)-value is reported as the proportion of shufflings that results in a test statistic smaller or equal to the one observed without shuffling.

This test will be more sensitive when only one or a few of the \(K\) datasets have very different spreads. Other statistics can be constructed to be more sensitive when most of the \(K\) datasets have slightly different spreads, but we believe the latter case is less important for the purpose of joint analysis of multiple datasets because most tests of mean contrasts can tolerate mild departures from the equal variance assumption, especially under balanced designs. Second, this statistic is more sensitive to differences in marginal variability than to changes in correlations among the \(G\) dimensions. This is advantageous for joint analysis of multiple microarray datasets because, in the standard gene-by-gene analysis, pooling error variance estimates across datasets is justified whenever marginal variances are approximately constant across datasets. Thus for the microarray application, it is more important to be able to detect departures from \(\text{diag}(\Sigma_1) = \text{diag}(\Sigma_2) = \cdots = \text{diag}(\Sigma_K)\) than to be able to detect departures from \(\Sigma_1 = \Sigma_2 = \cdots = \Sigma_K\). Furthermore, our experience with real microarray datasets suggests that an expansion of gene-specific error variances by a multiplicative factor from one dataset to another is the most common type of departure from \(\text{diag}(\Sigma_1) = \text{diag}(\Sigma_2) = \cdots = \text{diag}(\Sigma_K)\). Thus, the power of our test is focused on detecting such alternatives.

### 2.2.2 Cross-validation for selecting variance models

Cross-validation (CV), as a data based method for estimating prediction ability, is a powerful tool for model selection. However, the most commonly used CV method in linear models, PRESS, and the closely related Mallows’ \(C_P\) criterion, are not able to identify heterogeneity in variance. Furthermore, the usual application of CV only selects one common model that has
good prediction ability over all $K$ datasets. But when our interest primarily lies in inference within a single dataset, the selected model might not be optimal because the prediction ability on other datasets biases our choice. Hence, it might be more interesting to select different models for inference problems within different datasets.

Here we propose CV based procedures that can solve the above problems. A key point is to consider element-wise squared $Z_k$, denoted as $Z_k^{(2)}$, as the raw data to perform prediction, such that the selection of variance models on $Z_k$ can be roughly treated as the selection of mean models on $Z_k^{(2)}$. For ease of discussion, let $z_k^{(2)}$ denote an arbitrary row of $Z_k^{(2)}$. The same procedure to be discussed can be applied to each row of $Z_k^{(2)}$. In the case of normality based linear models for $Y_k$, the elements of $z_k^{(2)}$ are independently and identically distributed as scaled $\chi^2_1$ random variables with mean $\sigma_k^2 > 0$. Hence the average of all elements of $z_k^{(2)}$ provides a natural estimator of its mean, i.e., the variance of each element in the corresponding row in $Y_k$. A variance model can be specified by a function $M$ that maps dataset $k \in \{1, 2, \ldots, K\}$ onto $\{1, 2, \ldots, J\}$, such that $\sigma_k^2 = \sigma_{k'}^2$ iff $M(k) = M(k')$ for $J \in \{1, 2, \ldots, K\}$. For a full model, $J = K$; and for a reduced model, $J < K$. Once the $z_k^{(2)}$, $z_2^{(2)}$, $\ldots$, $z_K^{(2)}$ are treated as the data, leave-one-out cross-validation can be done as usual. That is, we delete one data point and use the average of other data points that share the same mean according to the model specified by $M$ as a predictor for the deleted data point, and the procedure is repeated for each data point.

Suppose the $i$th element of $z_k^{(2)}$, $z_{ki}^{(2)}$, is deleted, and the prediction based on the remaining data according to the model specified by $M$ for $z_{ki}^{(2)}$ is

$$z_{k(-i)}^{(2)} = \frac{1}{D_k - 1} \left( -z_{ki}^{(2)} + \sum_{\{k': M(k') = M(k)\}} \sum_{j=1}^{d_{k'}} z_{k'j}^{(2)} \right),$$

where $D_k = \sum_{\{k': M(k') = M(k)\}} d_{k'}$ is the total degrees of freedom used for estimating the mean of $z_{ki}^{(2)}$ according to model $M$ without deleting any data. We define the raw PRESS residual as $e_{ki} = z_{ki}^{(2)} - z_{k(-i)}^{(2)}$ and the corrected PRESS residual as $e'_{ki} = e_{ki} \sqrt{(D_k - 1)/(2D_k)}$. It can be easily checked that $E(e_{ki}) = E(e'_{ki}) = 0$ and $\text{Var}(e'_{ki}) = \sigma_k^4$. The multiplicative correction in $e'_{ki}$ removes the dependence of $\text{Var}(e_{ki})$ upon $D_k$, so that $e'_{ki}$ are comparable to each other. Although this correction assumes normality of $Y_k$, it is still reasonable in other cases, especially for
reasonable sample sizes. The final PRESS measure for model selection is then \[ \sum_{k=1}^{K} \sum_{i=1}^{d_k} e_{ki}'^2. \]

The model with the smallest PRESS measure is preferred.

Alternative to the squared error loss above, we may choose the prediction loss to be the \((-2\times)\) prediction log likelihood for each \(z_{ki}^{(2)}\), which relates to the Kullback-Leibler divergence. Let the log likelihood based on a single data point \(z_{ki}^{(2)}\) be \(\ell_{ki}(\sigma_k^2) = \log f_{\chi^2_1}(z_{ki}^{(2)}/\sigma_k^2) - \log \sigma_k^2\), where \(f_{\chi^2_1}\) is the density function for the \(\chi^2_1\) distribution. The final CV based prediction log likelihood criterion is then \(-2\sum_{k=1}^{K} \sum_{i=1}^{d_k} \ell_{ki}(z_{k(-i)}^{(2)})\). Again, the model with the smallest criterion is preferred.

Note that the above two procedures select one model to be used for all \(K\) datasets. This may not be ideal if we are interested in analyses within datasets but not across datasets. In this case, CV based procedures can be easily adapted to dataset-specific PRESS \(\sum_{i=1}^{d_k} e_{ki}'^2\) and/or dataset-specific \((-2\times)\) prediction log likelihood \(-2\sum_{i=1}^{d_k} \ell_{ki}(z_{k(-i)}^{(2)})\) as model selection criteria. Note that even if these procedures are dataset-specific, information from other datasets is borrowed through the prediction \(z_{k(-i)}^{(2)}\). Use of these dataset-specific criteria would allow different variance pooling strategies for different datasets. As we will demonstrate through simulation in the next section, in some cases it may be advantageous to pool error variance estimates across two datasets 1 and 2 for analyzing dataset 1 but not for analyzing dataset 2. Thus such a breakdown of model selection criteria can be quite useful. Note that it is rarely straightforward to break other information criteria (e.g., AIC or BIC) down in this manner, because of the difficulty in reasonably decomposing the penalty on log likelihood.

### 2.2.3 AICc under heteroscedasticity

Hurvich and Tsai [Hurvich and Tsai, 1989] developed AICc for selecting an appropriate mean model in a linear model context under the normality assumption. As an estimate of expected estimated Kullback-Leibler divergence, AICc is exactly unbiased and has the same variance as the asymptotically unbiased AIC. Hence Burnham and Anderson [Burnham and Anderson, 2002] recommended routine use of AICc over AIC.

However, such analytic small sample bias correction has to be dealt with case by case,
i.e., the correction will be different for different models. For example, the correction under a heterogeneous variance assumption is different than under a homogeneous variance assumption. Unfortunately, to our knowledge, implementations in common statistical software do not acknowledge such differences.

Under the fully heteroscedastic model across $K$ datasets, the correct AICc formula is

$$-2 \sum_{k=1}^{K} (\log \text{REML}_k) + 2 \sum_{k=1}^{K} \frac{d_k}{d_k - 2},$$

where REML$_k$ is the maximized residual likelihood for dataset $k$. To see why this formula makes sense, first consider the $k$th dataset alone, where the homoscedastic AICc is known to be $-2 \log \text{REML}_k + 2d_k/(d_k - 2)$. That is, the bias in using maximized residual log likelihood to estimate the expected mean prediction log likelihood for the $k$th dataset is exactly $d_k/(d_k - 2)$. Since the $K$ datasets are independent, their log likelihoods are additive and the final bias is simply the sum of the bias of each individual maximized residual log likelihood, which justifies the heteroscedastic AICc formula.

Because $2d_k/(d_k - 2)$ is always greater than 2 when $d_k > 2$, the AICc penalty is larger than the AIC penalty $2K$. When $d_k \to \infty \forall k$, the two criteria are equivalent. However, the SAS/STAT PROC MIXED procedure with the GROUP option for specifying heterogeneous variance across datasets uses $2K \left( \sum_{k=1}^{K} d_k \right) \left[ \left( \sum_{k=1}^{K} d_k \right) - K - 1 \right]^{-1}$ as the penalty term, which converges to AIC if $d_k \to \infty$ for any $k$. Hence we would expect that this incorrect AICc will not perform well when some $d_k$’s are small but others are large.

### 2.2.4 Combining model selection criteria from multiple genes

In a usual gene-by-gene analysis of microarray data, one can compute information criteria for each gene separately. For better interpretability, it is often desirable to fit the same model to each gene, and hence an overall criterion for choosing one model based on data from all genes is needed.

We suggest using the sum of gene specific information criteria as an overall measure for model selection. This is a sensible strategy in general and the obvious strategy for the special case of independence across genes. For AIC, the log likelihood is additive under independence
and the penalty term (the number of parameters) is also additive. So the sum of gene specific AIC’s equals the AIC directly calculated by assuming an independence model on all G genes. The same argument also applies to prediction log likelihood CV and the correct AICc developed in the last subsection, but not for the AICc reported by SAS/STAT, nor for BIC, CAIC, or HQIC since their penalty terms cannot be added directly.

The Akaike’s weights [Burnham and Anderson, 2002] can be used to justify the use of sums as an overall model selection criterion for BIC, CAIC, and HQIC, in addition to AIC and AICc. Considering q candidate models, the Akaike’s weight for model m and gene g is defined as

$$w_{mg} = \frac{\exp\{-0.5\Delta IC_{mg}\}}{\sum_{m'=1}^{q}\exp\{-0.5\Delta IC_{m'g}\}},$$

where $\Delta IC_{mg} = IC_{mg} - \min_{1 \leq m' \leq q} IC_{m'g}$, and IC$_{mg}$ is the information criterion for model m and gene g, m = 1, 2, ..., q, and g = 1, 2, ..., G. This weight can be easily interpreted as the posterior probability of model m for gene g, and the choice of different information criteria is equivalent to choosing a different prior probability for each model (see [Burnham and Anderson, 2002] for details).

Hence, by assuming independence across genes, the posterior probability of selecting model m for all G genes is the product $w_m = \prod_{g=1}^{G} w_{mg}$, which is a monotonically decreasing function of the sum of information criteria $\sum_{g=1}^{G} IC_{mg}$. In other words, the model selected with the smallest sum of information criteria across genes always matches the model with the largest posterior probability among the q − 1 alternative overall models.

To combine gene-specific PRESS into an overall model selection measure, the sum is not as reasonable because genes with large variances tend to have large PRESS and the sum may be dominated by such genes. Hence, we use gene voting to select the overall model, so that the model selected for all genes will be the one selected by the plurality of gene-specific PRESS statistics.

Although the above arguments rely on independence across genes, these combination procedures are still natural intuitive measures under dependence. We do not expect that dependence will seriously bias the selection of models. Taking the most extreme case as an example, suppose we only have two genes which are nearly completely dependent. Then
the gene-wise model selection criteria will almost always select the same model, and the combination of criteria using either the sum or the votes will also select the same model. Hence the combination does not favor either simpler or more complex models, even in this highly dependent case. Therefore, these combination procedures remain sensible and useful in practice. Furthermore, although our approach was developed under the assumption of independence across genes, note that we evaluate its performance using a simulation procedure that includes correlation across genes as described in the next subsection.

2.2.5 Simulation based on real microarray data

Simulation studies to evaluate the performance of model selection methods often generate data from parametric models. Although such results are helpful to aid understanding the pros and cons of each method, they provide little information about actual performance on real datasets, especially when we have few clues about the general dependence structure among a large number of genes. To overcome such difficulties, we replace traditional model-based simulation with subsampling from an actual microarray dataset that involves at least one large treatment group. Treating the large treatment group as a population, we can simulate various multiple-dataset scenarios by drawing subsamples from the population and perturbing the data as described below. Details about the populations actually used in our study are provided at the end of this subsection.

Let $G$ denote the number of genes and $n$ the number of biological samples in the population. Let $E$ denote a $G \times n$ residual matrix obtained by subtracting the gene-specific mean log-scale expression value from the log-scale expression values of each gene. Because the population size $n$ is large, we ignored the small dependence across the columns of $E$. Also, we computed the sample standard deviation for each gene $g$ as $\sigma_{0g}$, and treated these as known values due to the large number of degrees of freedom.

To generate two datasets, each involving two treatments, we first randomly sampled $2n_1 + 2n_2$ columns from $E$ without replacement to form a $G \times (2n_1 + 2n_2)$ matrix, with the first $2n_1$ columns $Y_1$ denoted as dataset 1 and the other $2n_2$ columns $Y_2$ denoted as dataset 2. Within
each dataset, a balanced 2-treatment comparison design was used. Sample sizes $n_1$ and $n_2$ were set to 3, 5, or 7, which are similar to those used in many microarray studies.

Table 2.1 Simulation setting for the ratio of variances.

<table>
<thead>
<tr>
<th>$\mu_r$</th>
<th>$\sigma_r^2$</th>
<th>$E(r_g)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1 (null)</td>
</tr>
<tr>
<td>-0.057800</td>
<td>0.342$^2$</td>
<td>1</td>
</tr>
<tr>
<td>1.040812</td>
<td>0.342$^2$</td>
<td>3</td>
</tr>
<tr>
<td>1.551638</td>
<td>0.342$^2$</td>
<td>5</td>
</tr>
<tr>
<td>2.244785</td>
<td>0.342$^2$</td>
<td>10</td>
</tr>
</tbody>
</table>

Next, we independently generated $G$ random variables $r_g$, $g = 1, 2, \ldots, G$, from a log Normal($\mu_r, \sigma_r^2$) distribution, where the settings of $\mu_r$ and $\sigma_r^2$ are listed in Table 2.1. Then the $g$th row of $Y_2$ was multiplied by $\sqrt{r_g}$ for all $g = 1, 2, \ldots, G$. Hence $r_g$ is the ratio of the error variance in the 2nd dataset to the error variance in the 1st dataset for the $g$th gene. Note that, when $\mu_r = \sigma_r^2 = 0$, all $r_g = 1$, i.e., the error variances are equal across the two datasets. When $\sigma_r^2 > 0$, $\mu_r$ was chosen such that $E(r_g)$ was 1, 3, 5, or 10 respectively. $\sigma_r^2 = 0.342^2$ was used here such that the error variance correlation between the two datasets on the log scale is about 0.8, which is very similar to the observed log variance correlations between different treatments in the real microarray datasets we examined. In other words, our simulation setup acknowledges that genes with small error variances in one dataset also tend to have small variances in the other dataset, which is also a biologically rational assumption.

In order to assess the ability of different model selection methods to detect differentially expressed genes and to control false discovery rates, we perturbed the mean of a subset of genes in each dataset to mimic responses to the treatment. Since we believe that in real biological systems the standardized effective sizes are probably not related to the error variances, we added to each row $g$ of the first treatment group in the $k$th dataset $\sigma_{0g}r_g^{(k)}u_{gk}$, where $u_{gk}$ was independently sampled from a mixture of zero with probability 0.8 and a standard normal distribution with probability 0.2 for all $g = 1, 2, \ldots, G$ and $k = 1, 2$, and $r_g^{(k)} = 1$ when $k = 1$ and $r_g^{(k)} = \sqrt{r_g}$ when $k = 2$. Thus on average, 20% of genes in each dataset were simulated to be differentially expressed.
For each of the 15 simulation settings (3 sample sizes × 5 variance ratios), 100 independent dataset-pairs were simulated. Gene-wise linear models were fit to each simulated dataset-pair in the statistical computing environment R [R Development Core Team, 2009] using the model selection methods proposed in subsections 2.2.1 through 2.2.3. The accuracy in ranking differentially expressed genes was assessed through the area under the receiver operating characteristic (AUROC) curves using the package ROCR [Sing et al., 2005]. False discovery rate control procedures of Benjamini and Hochberg (BH) [Benjamini and Hochberg, 1995] and Storey and Tibshirani (ST) [Storey and Tibshirani, 2003] were used.

For each method, both results from selecting a variance model for each individual gene and results from selecting an overall variance model for all genes were obtained. For AIC, AICc assuming heterogeneity of variances, BIC, CAIC, HQIC, and the cross-validated prediction log likelihood, the overall model was determined through sums of these criteria. For PRESS, the overall model was determined through gene-wise voting. For our modified MRPP procedure, three p-value cutoffs 1/35 ≈ 0.03, 2/35 ≈ 0.06, or 4/35 ≈ 0.11 were examined, and the overall model was determined by the multivariate test applied to all genes instead of from summarizing gene-wise MRPP tests.

In addition to the variance model selection approaches, the very popular limma method of Smyth [2004] was also included as a representative of the variance shrinkage approaches. Limma assumes an inverse gamma prior on the gene-wise variances and performs well in our experience. Furthermore, an easy to use R package is available for the necessary computation. Although there could be differences in terms of performance between limma and other shrinkage methods, we believe the general conclusions would be very similar when compared with our model selection methods. In our simulation study, all default parameter settings in the limma package (version 2.18.3) were used.

To further compare our proposed modified MRPP procedure to other commonly used tests of heterogeneity of variances for univariate data, we conducted a separate two-sample comparison study, again using subsamples from real microarray data, except that the subsampling of each gene was performed separately to break the correlation across the genes. In this way, the
resulting gene-wise \( p \)-values across genes can be pooled to provide clearer information about the properties of these tests under the null and under the alternative hypotheses. The alternative tests being compared are Levene’s test, the Brown-Forsythe test, the \( F \)-test, and a new test, denoted hereafter as reduced Levene’s test, which applies ANOVA on the absolute values of the decorrelated and reduced datasets, as in our modified MRPP. 

To measure departure from uniformity under the null for each of the tests, the resulting \( G \) \( p \)-values were first binned into \( B \) bins, with bin boundaries determined as the non-redundant set of observed MRPP \( p \)-values, together with two natural boundaries 0 and 1. This binning is intended to remove the difference caused by the discrete nature of MRPP \( p \)-values, so that results from all tests are comparable. Next, Kullback-Leibler divergence from uniformity as a function of the likelihood ratio was computed for \( p \)-values from each method separately, as \( \sum_{b=1}^{B} w_b \left[ \log w_b - \log \left( C_b/G \right) \right] \), where \( w_b \) is the width of bin \( b \) and \( C_b \) is the number of \( p \)-values falling in bin \( b \). Larger Kullback-Leibler divergence indicates larger departure from uniformity. 

Under the alternative hypotheses, samples from one treatment are multiplied by \( \sqrt{r_g} \) as before. Because the actual sizes of these approximate tests are not exactly the same, using a fixed cutoff for nominal \( p \)-values does not provide fair comparisons. So we computed the probability (i.e., the proportion among the \( G \) genes) of the observed test statistic under the alternative hypothesis being at least as extreme as the observed test statistic under the null hypothesis for each simulation. The larger the probability is, the more powerful the test to detect heterogeneities. This measure is in the same spirit as AUROC or the signed rank test. All these simulations were repeated 50 times, and different sample sizes for each treatment were used (Tables 2.9 to 2.11). 

Our entire simulation process was repeated using three different populations, each derived from a real microarray dataset. The datasets are all publicly available in the GEO database hosted by the National Center for Biotechnology Information (http://www.ncbi.nlm.nih.gov/geo/) with accession numbers GSE755 [Tian et al., 2003], GSE4115 [Spira et al., 2007] and GSE5406 [Hannenhalli et al., 2006], respectively. The normalized “series” matrix files were downloaded from the GEO website, and the base-2 logarithm was taken to produce our raw
data. For each dataset, only the treatment group with the largest sample size was used as the population from which to simulate multiple datasets, as described previously in this subsection.

2.3 Simulation results and discussion

Our simulation results from the three real populations differed only slightly. Thus, we only report results from the population derived from dataset GSE5406 here. To save space, results with different sample sizes are not all reported, but we will mention the trend of change as sample sizes increase.

2.3.1 Proportion of correctly selected models

The proportion of correctly selected models using each of the model selection procedures is in Tables 2.2 and 2.3 for sample sizes 3 and 7, respectively. For all following tables, standard errors are shown in parentheses.

We can see from Table 2.2 that the PRESS criterion generally prefers larger models in small samples. That is, if the correct model is homogeneous, PRESS has lower probability of selecting such a model; but if the correct model is heterogeneous, PRESS performs best to choose the larger model. This is most obvious for gene-by-gene model selection, but when we consider selecting an overall model for all genes by voting, PRESS still picks out the correct model > 90% of the time even if the true model is homogeneous. Table 2.3 verifies this and further shows that, when sample size increases, the performance of PRESS also improves accordingly.

On the contrary, the AICc, CAIC and the CV log likelihood criteria seem to prefer smaller models (Table 2.2). They mostly pick the right model when the true model has homogeneous variances across datasets, but they tend not to choose the correct model when the true variance model is heterogeneous until \( E(r_g) \) is fairly large. However, when sample size increases (Table 2.3), their performance also improves, in particular AICc. This is reasonable because AICc converges to AIC asymptotically.

The performance of AIC, BIC and HQIC generally lies between the two extremes (Tables 2.2 and 2.3). However, when sample size increases, AIC still shows a relatively higher probability
Table 2.2 Proportion (×100) of correctly selected models from dataset GSE5406 when sample size $n_1 = n_2 = 3$

<table>
<thead>
<tr>
<th>Method</th>
<th>$r_g = 1$</th>
<th>$\mathbb{E}(r_g) = 1$</th>
<th>$\mathbb{E}(r_g) = 3$</th>
<th>$\mathbb{E}(r_g) = 5$</th>
<th>$\mathbb{E}(r_g) = 10$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>overall</td>
<td>per gene</td>
<td>overall</td>
<td>per gene</td>
<td>overall</td>
</tr>
<tr>
<td>AIC</td>
<td>98.0(1.4)</td>
<td>75.7(0.4)</td>
<td>4.0(2.0)</td>
<td>26.4(0.4)</td>
<td>70.0(4.6)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>41.1(1.0)</td>
<td></td>
<td>97.0(1.7)</td>
</tr>
<tr>
<td>AICc</td>
<td>100.0(0.0)</td>
<td>94.7(0.2)</td>
<td>0.0(0.0)</td>
<td>13.4(0.6)</td>
<td>2.0(1.4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>22.9(0.9)</td>
<td></td>
<td>39.0(4.9)</td>
</tr>
<tr>
<td>BIC</td>
<td>98.0(1.4)</td>
<td>76.6(0.4)</td>
<td>1.0(1.0)</td>
<td>25.4(0.4)</td>
<td>61.0(4.9)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>40.1(1.0)</td>
<td></td>
<td>97.0(1.7)</td>
</tr>
<tr>
<td>CAIC</td>
<td>100.0(0.0)</td>
<td>85.4(0.3)</td>
<td>0.0(0.0)</td>
<td>16.1(0.3)</td>
<td>13.0(3.4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>28.9(0.9)</td>
<td></td>
<td>53.0(5.0)</td>
</tr>
<tr>
<td>HQIC</td>
<td>62.0(4.8)</td>
<td>68.0(0.4)</td>
<td>61.0(4.9)</td>
<td>34.1(0.4)</td>
<td>100.0(0.0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>49.1(1.0)</td>
<td></td>
<td>100.0(0.0)</td>
</tr>
<tr>
<td>CV-lik.full</td>
<td>100.0(0.0)</td>
<td>74.1(0.3)</td>
<td>0.0(0.0)</td>
<td>27.5(0.3)</td>
<td>0.0(0.0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>38.8(0.8)</td>
<td></td>
<td>0.0(0.0)</td>
</tr>
<tr>
<td>PRESS.full</td>
<td>92.0(2.7)</td>
<td>55.3(0.4)</td>
<td>16.0(3.7)</td>
<td>46.6(0.4)</td>
<td>85.0(3.6)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>58.7(0.8)</td>
<td></td>
<td>99.0(1.0)</td>
</tr>
<tr>
<td>MRPP_{0.03}</td>
<td>91.0(2.9)</td>
<td>95.0(0.1)</td>
<td>8.0(0.0)</td>
<td>4.9(0.1)</td>
<td>70.0(4.6)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>8.8(0.3)</td>
<td></td>
<td>92.0(2.7)</td>
</tr>
<tr>
<td>MRPP_{0.06}</td>
<td>83.0(3.8)</td>
<td>91.0(0.2)</td>
<td>14.0(3.5)</td>
<td>9.0(0.2)</td>
<td>81.0(3.9)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>14.8(0.5)</td>
<td></td>
<td>96.0(2.0)</td>
</tr>
<tr>
<td>MRPP_{0.11}</td>
<td>70.0(4.6)</td>
<td>84.0(0.3)</td>
<td>25.0(4.3)</td>
<td>16.4(0.3)</td>
<td>88.0(3.2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>24.9(0.7)</td>
<td></td>
<td>99.0(1.0)</td>
</tr>
</tbody>
</table>

Table 2.3 Proportion (×100) of correctly selected models from dataset GSE5406 when sample size $n_1 = n_2 = 7$

<table>
<thead>
<tr>
<th>Method</th>
<th>$r_g = 1$</th>
<th>$\mathbb{E}(r_g) = 1$</th>
<th>$\mathbb{E}(r_g) = 3$</th>
<th>$\mathbb{E}(r_g) = 5$</th>
<th>$\mathbb{E}(r_g) = 10$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>overall</td>
<td>per gene</td>
<td>overall</td>
<td>per gene</td>
<td>overall</td>
</tr>
<tr>
<td>AIC</td>
<td>74.0(1.4)</td>
<td>72.0(0.5)</td>
<td>50.0(5.0)</td>
<td>31.9(0.4)</td>
<td>100.0(0.0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>62.6(1.1)</td>
<td></td>
<td>100.0(0.0)</td>
</tr>
<tr>
<td>AICc</td>
<td>99.0(1.0)</td>
<td>78.0(0.1)</td>
<td>5.0(2.2)</td>
<td>25.4(0.4)</td>
<td>98.0(1.4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>56.4(1.1)</td>
<td></td>
<td>100.0(0.0)</td>
</tr>
<tr>
<td>BIC</td>
<td>100.0(0.0)</td>
<td>82.0(0.4)</td>
<td>1.0(1.0)</td>
<td>20.9(0.4)</td>
<td>91.0(2.9)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>51.2(1.1)</td>
<td></td>
<td>100.0(0.0)</td>
</tr>
<tr>
<td>CAIC</td>
<td>100.0(0.0)</td>
<td>88.0(0.3)</td>
<td>0.0(0.0)</td>
<td>15.0(0.3)</td>
<td>68.0(4.7)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>43.1(1.2)</td>
<td></td>
<td>99.0(1.0)</td>
</tr>
<tr>
<td>HQIC</td>
<td>96.0(2.0)</td>
<td>75.0(0.4)</td>
<td>19.0(3.9)</td>
<td>28.4(0.4)</td>
<td>100.0(0.0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>59.4(1.1)</td>
<td></td>
<td>100.0(0.0)</td>
</tr>
<tr>
<td>CV-lik.full</td>
<td>100.0(0.0)</td>
<td>75.0(0.4)</td>
<td>0.0(0.0)</td>
<td>28.2(0.4)</td>
<td>81.0(3.9)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>56.2(1.0)</td>
<td></td>
<td>100.0(0.0)</td>
</tr>
<tr>
<td>PRESS.full</td>
<td>96.0(2.0)</td>
<td>59.0(0.5)</td>
<td>11.0(3.1)</td>
<td>45.1(0.4)</td>
<td>100.0(0.0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>72.5(0.9)</td>
<td></td>
<td>100.0(0.0)</td>
</tr>
<tr>
<td>MRPP_{0.03}</td>
<td>93.0(2.5)</td>
<td>95.0(0.2)</td>
<td>14.0(3.5)</td>
<td>6.8(0.2)</td>
<td>97.0(1.7)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>25.2(0.8)</td>
<td></td>
<td>100.0(0.0)</td>
</tr>
<tr>
<td>MRPP_{0.06}</td>
<td>87.0(3.4)</td>
<td>91.0(0.2)</td>
<td>21.0(4.1)</td>
<td>11.3(0.3)</td>
<td>99.0(1.0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>34.8(1.0)</td>
<td></td>
<td>100.0(0.0)</td>
</tr>
<tr>
<td>MRPP_{0.11}</td>
<td>77.0(4.2)</td>
<td>84.0(0.3)</td>
<td>23.0(4.2)</td>
<td>19.1(0.4)</td>
<td>100.0(0.0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>47.1(1.0)</td>
<td></td>
<td>100.0(0.0)</td>
</tr>
</tbody>
</table>

The above tables show the proportion (×100) of correctly selected models from dataset GSE5406 when sample size $n_1 = n_2 = 3$ and $n_1 = n_2 = 7$. Each table entry represents the proportion of correctly selected models for different methods and $r_g$ values.
of overfitting, whereas the overfitting for HQIC in small samples diminishes as sample size increases. This is because the penalty for AIC is independent of sample size, whereas BIC and HQIC use sample size as auxiliary information to achieve model dimension consistency.

For the model selection criteria other than PRESS and modified MRPP, the use of sum of individual gene-wise criteria seems to have good performance (Tables 2.2 and 2.3). For all methods, except CV log likelihood under small sample sizes, the probability of selecting the correct overall model is generally larger than the corresponding probability of selecting the correct individual gene model. This demonstrates the advantage of sharing information across genes, especially when the majority of genes provide concordant information about which model is preferred. The CV log likelihood prefers smaller models too often and using the sum does not help much. However, when sample size increases, it performs similarly to other criteria.

For the modified MRPP test, when the null hypothesis is true, i.e., when the correct model is homogeneous, the probability of selecting the right model is generally slightly smaller than $1 - \alpha$ (Tables 2.2 and 2.3), which indicates that the test is only approximate, tending to produce slightly liberal $p$-values. This problem is more severe for the multivariate version of the test to select an overall model, but less so for univariate tests to select individual gene models. Theoretically, when the responses are indeed normally distributed, the decorrelation in our modified MRPP offers complete independence. The better control for type I error shown in univariate tests than multivariate tests suggests that the marginal distribution of each gene’s expression is probably not far from normality, but the joint distribution of all genes deviates further from multivariate normality due to complicated unknown dependencies among genes. However, when sample size increases, the control of type I error for the multivariate test also gets tighter gradually.

Another phenomenon to note is that the univariate modified MRPP tests do not yield a high proportion of rejected hypotheses when the true model is heterogeneous (Tables 2.2 and 2.3), which suggests the use of other model selection criteria for individual gene models. However, when we consider the multivariate test to choose an overall model, the power of the modified MRPP increases considerably. This is because the test statistic we use is very sensitive when
the direction of variability change is generally the same for the majority of genes. In this case, the high-dimensionality amplifies such a concordant directional change, and the test power is actually improved by the high dimensionality.

Although there seem to be some shortcomings, i.e., being approximate and not powerful in the univariate case, our modified MRPP test actually suffers less severely from such problems than other tests, as shown in Subsection 2.3.4.

Lastly, from the results shown in Tables 2.2 and 2.3, we see that when the variances across the two datasets are different but $E(r_g) = 1$, none of the model selection procedures work well. However, this is not necessarily a disadvantage, because even if the correct model is heterogeneous, we need enough data to support the use of such a larger model. If the correct but larger model differs little from the smaller model, the cost in losing precision of estimates may not justify the use of the correct model. This will be further demonstrated in the next subsection.

2.3.2 Ranking genes in terms of differential expression

Although the proportion of correctly selected models is a good intuitive measure of performance of model selection procedures and is used in many simulation studies, it cannot reveal how well each procedure can pick the differentially expressed genes out of the other tens of thousands of equivalently expressed genes. Hence, we next conduct gene-wise $F$-tests under the model selected by each procedure to rank the genes according to their $p$-values and compare the area under the corresponding ROC curves to see how methods differ in terms of detecting differential expression. Results from study GSE5406 when models are selected for each individual gene are shown in Tables 2.4 and 2.5, with Table 2.4 showing results with sample size $= 7$, and Table 2.5 showing results with sample size $= 3$. These results are summarized by contrasts of interest. Contrast $k$ represents the comparison of treatment means within the $k$th dataset, $k = 1$ or 2. By simulation, the first dataset generally has smaller variance than the second dataset. Further, to ease the direct comparison of the AUROC’s across different methods, the standard errors reported in parentheses are computed after removing the common dataset effects by assuming a
two-way (dataset & methods) linear model on the AUROC’s. Note that although the differences in AUROC values shown in these tables are small, they are still important because 1) compared with standard errors, the difference is usually significant, and 2) the total number of genes is quite large so that even a small difference in AUROC may reflect substantial changes in the rankings of a large number of genes.

Tables 2.4 and 2.5 show that using the correct model does not necessarily give the best ranking of genes, even if sample sizes are not small. When $E(r_g)$ is large, the correct model, i.e., the “separate” row in the tables, generally performs well and the AUROC’s are big. But when the true model is heterogeneous but $E(r_g)$ is close to 1, the correct model is actually the worst in terms of AUROC. In both this case and when the true model is homogeneous, the best procedure is to always perform a pooled analysis of the two datasets, whether it is the correct model or not. A partial explanation is that when $E(r_g)$ is close to 1, we need much larger sample sizes to get stable estimates from using the larger, but correct, model.

However, this is not the whole story. A special feature shown in Tables 2.4 and 2.5 is that, for contrast 2, no matter what the correct model is, performing a pooled analysis is always better than performing a separate analysis, even if $E(r_g)$ is very large. The same phenomenon is also observed in the other two datasets GSE4115 and GSE755. Further, when we greatly increase the sample size, the same phenomenon still occurs (results not shown). Hence, this is not a phenomenon that can be explained by the lack of stable estimates under small sample sizes.

Since this phenomenon only occurs for the contrast in the second dataset, which has larger variances than the first, one might conjecture that the pooled variance estimate in this case is a shrinkage estimate that actually has smaller squared error risk and/or Stein-type risk. However, this does not fully explain the phenomenon either. A related observation by [Tong and Wang, 2007] demonstrated that optimal shrinkage estimation of variances does not always perform best. Our results confirm this observation using real datasets; using pooled estimates of variances is better in terms of AUROC for contrast 2 even if the variance estimates are far from the true parameters in the heterogeneous model. Hence our tentative conclusion is that
Table 2.4  Area ($\times 1000$) under the ROC curve for selecting individual gene model from dataset GSE5406 and $n_1 = n_2 = 7$

<table>
<thead>
<tr>
<th>Method</th>
<th>Contrast 1</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Contrast 2</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$r_g=1$</td>
<td>$E(r_g)=1$</td>
<td>$E(r_g)=3$</td>
<td>$E(r_g)=5$</td>
<td>$E(r_g)=10$</td>
<td>$r_g=1$</td>
<td>$E(r_g)=1$</td>
<td>$E(r_g)=3$</td>
<td>$E(r_g)=5$</td>
</tr>
<tr>
<td>AIC</td>
<td>715.9(0.0)</td>
<td>717.6(0.0)</td>
<td>716.5(0.1)</td>
<td>716.5(0.0)</td>
<td>716.9(0.0)</td>
<td>717.5(0.0)</td>
<td>712.3(0.1)</td>
<td>710.8(0.0)</td>
<td>710.7(0.0)</td>
</tr>
<tr>
<td>AICc</td>
<td>716.1(0.0)</td>
<td>717.6(0.0)</td>
<td>716.0(0.1)</td>
<td>715.9(0.1)</td>
<td>716.7(0.0)</td>
<td>717.7(0.0)</td>
<td>712.4(0.0)</td>
<td>710.6(0.0)</td>
<td>710.3(0.0)</td>
</tr>
<tr>
<td>BIC</td>
<td>716.4(0.0)</td>
<td>717.7(0.0)</td>
<td>715.5(0.1)</td>
<td>715.4(0.1)</td>
<td>716.4(0.0)</td>
<td>717.9(0.0)</td>
<td>712.5(0.0)</td>
<td>710.4(0.0)</td>
<td>709.9(0.0)</td>
</tr>
<tr>
<td>CAIC</td>
<td>716.7(0.0)</td>
<td>717.8(0.0)</td>
<td>714.8(0.1)</td>
<td>714.2(0.1)</td>
<td>715.5(0.1)</td>
<td>718.1(0.0)</td>
<td>712.5(0.0)</td>
<td>710.3(0.1)</td>
<td>709.3(0.1)</td>
</tr>
<tr>
<td>HQIC</td>
<td>716.0(0.0)</td>
<td>717.6(0.0)</td>
<td>716.3(0.1)</td>
<td>716.2(0.1)</td>
<td>716.8(0.0)</td>
<td>717.6(0.0)</td>
<td>712.4(0.0)</td>
<td>710.7(0.0)</td>
<td>710.5(0.0)</td>
</tr>
<tr>
<td>CV-lik.full</td>
<td>716.0(0.0)</td>
<td>717.6(0.0)</td>
<td>715.3(0.1)</td>
<td>714.5(0.1)</td>
<td>714.5(0.1)</td>
<td>717.6(0.0)</td>
<td>712.2(0.0)</td>
<td>710.7(0.0)</td>
<td>710.2(0.0)</td>
</tr>
<tr>
<td>CV-lik.specific</td>
<td>715.9(0.0)</td>
<td>717.5(0.0)</td>
<td>714.6(0.1)</td>
<td>713.1(0.1)</td>
<td>712.7(0.1)</td>
<td>717.5(0.0)</td>
<td>712.2(0.0)</td>
<td>711.2(0.0)</td>
<td>711.2(0.0)</td>
</tr>
<tr>
<td>PRESS.full</td>
<td>715.5(0.0)</td>
<td>717.4(0.1)</td>
<td>716.9(0.1)</td>
<td>716.8(0.0)</td>
<td>717.1(0.0)</td>
<td>717.2(0.1)</td>
<td>712.2(0.1)</td>
<td>711.3(0.0)</td>
<td>711.4(0.0)</td>
</tr>
<tr>
<td>PRESS.specific</td>
<td>716.0(0.1)</td>
<td>717.7(0.1)</td>
<td>717.5(0.1)</td>
<td>717.3(0.0)</td>
<td>717.2(0.0)</td>
<td>717.7(0.1)</td>
<td>712.6(0.1)</td>
<td>710.9(0.1)</td>
<td>710.5(0.1)</td>
</tr>
<tr>
<td>MRPP$\theta_{0.03}$</td>
<td>717.3(0.1)</td>
<td>718.2(0.1)</td>
<td>712.6(0.2)</td>
<td>708.6(0.2)</td>
<td>707.3(0.2)</td>
<td>718.4(0.1)</td>
<td>712.6(0.1)</td>
<td>711.1(0.1)</td>
<td>709.0(0.1)</td>
</tr>
<tr>
<td>MRPP$\theta_{0.06}$</td>
<td>717.0(0.1)</td>
<td>718.0(0.1)</td>
<td>713.2(0.1)</td>
<td>710.9(0.1)</td>
<td>711.3(0.1)</td>
<td>718.1(0.1)</td>
<td>712.6(0.1)</td>
<td>710.7(0.1)</td>
<td>709.2(0.1)</td>
</tr>
<tr>
<td>MRPP$\theta_{0.11}$</td>
<td>716.5(0.1)</td>
<td>717.7(0.0)</td>
<td>714.4(0.1)</td>
<td>713.5(0.1)</td>
<td>714.6(0.1)</td>
<td>717.9(0.1)</td>
<td>712.5(0.1)</td>
<td>710.7(0.0)</td>
<td>709.8(0.0)</td>
</tr>
<tr>
<td>separate</td>
<td>715.2(0.1)</td>
<td>717.3(0.1)</td>
<td>717.3(0.0)</td>
<td>717.3(0.0)</td>
<td>716.9(0.1)</td>
<td>712.2(0.1)</td>
<td>712.2(0.0)</td>
<td>712.2(0.0)</td>
<td>712.2(0.0)</td>
</tr>
<tr>
<td>pool</td>
<td>717.9(0.1)</td>
<td>718.4(0.1)</td>
<td>715.4(0.2)</td>
<td>713.7(0.3)</td>
<td>711.8(0.3)</td>
<td>718.9(0.1)</td>
<td>712.8(0.2)</td>
<td>714.1(0.1)</td>
<td>714.0(0.1)</td>
</tr>
<tr>
<td>limma</td>
<td>715.7(0.1)</td>
<td>717.8(0.1)</td>
<td>717.8(0.1)</td>
<td>717.8(0.1)</td>
<td>718.5(0.1)</td>
<td>713.3(0.1)</td>
<td>713.3(0.1)</td>
<td>713.3(0.1)</td>
<td>713.3(0.1)</td>
</tr>
</tbody>
</table>
Table 2.5  Area (×1000) under the ROC curve for selecting individual gene model from dataset GSE5406 and \( n_1 = n_2 = 3 \)

<table>
<thead>
<tr>
<th>Method</th>
<th>( r_g = 1 )</th>
<th>( \mathbb{E}(r_g) = 1 )</th>
<th>( \mathbb{E}(r_g) = 3 )</th>
<th>( \mathbb{E}(r_g) = 5 )</th>
<th>( \mathbb{E}(r_g) = 10 )</th>
<th>( r_g = 1 )</th>
<th>( \mathbb{E}(r_g) = 1 )</th>
<th>( \mathbb{E}(r_g) = 3 )</th>
<th>( \mathbb{E}(r_g) = 5 )</th>
<th>( \mathbb{E}(r_g) = 10 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIC</td>
<td>641.0(0.1)</td>
<td>631.2(0.1)</td>
<td>636.3(0.1)</td>
<td>635.2(0.2)</td>
<td>634.9(0.2)</td>
<td>636.1(0.1)</td>
<td>637.3(0.1)</td>
<td>633.9(0.1)</td>
<td>631.8(0.1)</td>
<td>630.7(0.1)</td>
</tr>
<tr>
<td>AICc</td>
<td>643.0(0.1)</td>
<td>632.9(0.1)</td>
<td>634.3(0.2)</td>
<td>630.1(0.2)</td>
<td>625.4(0.2)</td>
<td>638.3(0.1)</td>
<td>638.7(0.1)</td>
<td>637.2(0.1)</td>
<td>634.2(0.1)</td>
<td>629.5(0.2)</td>
</tr>
<tr>
<td>BIC</td>
<td>641.0(0.1)</td>
<td>631.2(0.1)</td>
<td>636.2(0.1)</td>
<td>635.1(0.2)</td>
<td>634.7(0.2)</td>
<td>636.2(0.1)</td>
<td>637.4(0.1)</td>
<td>633.9(0.1)</td>
<td>631.7(0.1)</td>
<td>630.6(0.1)</td>
</tr>
<tr>
<td>CAIC</td>
<td>641.6(0.1)</td>
<td>631.8(0.1)</td>
<td>635.2(0.1)</td>
<td>633.0(0.1)</td>
<td>631.9(0.2)</td>
<td>636.9(0.1)</td>
<td>637.9(0.1)</td>
<td>634.9(0.1)</td>
<td>632.0(0.1)</td>
<td>629.5(0.1)</td>
</tr>
<tr>
<td>HQIC</td>
<td>640.6(0.1)</td>
<td>630.8(0.1)</td>
<td>637.0(0.1)</td>
<td>636.3(0.2)</td>
<td>635.9(0.2)</td>
<td>635.6(0.1)</td>
<td>637.1(0.1)</td>
<td>633.6(0.1)</td>
<td>632.0(0.1)</td>
<td>631.5(0.1)</td>
</tr>
<tr>
<td>CV-lik-full</td>
<td>641.2(0.1)</td>
<td>631.5(0.1)</td>
<td>635.2(0.1)</td>
<td>632.6(0.1)</td>
<td>629.1(0.2)</td>
<td>636.4(0.1)</td>
<td>637.4(0.1)</td>
<td>634.6(0.1)</td>
<td>632.5(0.1)</td>
<td>630.7(0.1)</td>
</tr>
<tr>
<td>CV-lik-specific</td>
<td>640.9(0.1)</td>
<td>631.1(0.1)</td>
<td>634.7(0.1)</td>
<td>631.5(0.2)</td>
<td>626.8(0.2)</td>
<td>636.0(0.1)</td>
<td>637.2(0.1)</td>
<td>634.3(0.1)</td>
<td>632.9(0.1)</td>
<td>632.2(0.1)</td>
</tr>
<tr>
<td>PRESSfull</td>
<td>640.2(0.1)</td>
<td>630.5(0.1)</td>
<td>636.8(0.1)</td>
<td>636.0(0.2)</td>
<td>635.3(0.2)</td>
<td>635.3(0.1)</td>
<td>636.8(0.1)</td>
<td>633.7(0.1)</td>
<td>632.6(0.1)</td>
<td>632.2(0.1)</td>
</tr>
<tr>
<td>PRESSspecific</td>
<td>640.8(0.1)</td>
<td>631.0(0.1)</td>
<td>638.1(0.2)</td>
<td>637.6(0.2)</td>
<td>637.2(0.2)</td>
<td>635.9(0.1)</td>
<td>637.4(0.1)</td>
<td>634.4(0.1)</td>
<td>632.8(0.1)</td>
<td>631.2(0.1)</td>
</tr>
<tr>
<td>MRPP0.03</td>
<td>643.6(0.1)</td>
<td>633.4(0.1)</td>
<td>635.7(0.2)</td>
<td>631.1(0.3)</td>
<td>629.9(0.5)</td>
<td>638.8(0.1)</td>
<td>639.3(0.1)</td>
<td>638.5(0.1)</td>
<td>636.6(0.1)</td>
<td>633.2(0.1)</td>
</tr>
<tr>
<td>MRPP0.06</td>
<td>643.0(0.1)</td>
<td>633.0(0.1)</td>
<td>635.0(0.2)</td>
<td>630.2(0.3)</td>
<td>622.6(0.5)</td>
<td>638.2(0.1)</td>
<td>638.9(0.1)</td>
<td>637.5(0.1)</td>
<td>635.2(0.1)</td>
<td>631.8(0.1)</td>
</tr>
<tr>
<td>MRPP0.11</td>
<td>642.2(0.1)</td>
<td>632.3(0.1)</td>
<td>634.5(0.2)</td>
<td>630.5(0.3)</td>
<td>624.8(0.3)</td>
<td>637.5(0.1)</td>
<td>638.3(0.1)</td>
<td>636.0(0.1)</td>
<td>633.5(0.1)</td>
<td>630.5(0.1)</td>
</tr>
<tr>
<td>separate</td>
<td>639.1(0.2)</td>
<td>629.6(0.2)</td>
<td>637.0(0.2)</td>
<td>637.0(0.2)</td>
<td>637.0(0.2)</td>
<td>634.3(0.2)</td>
<td>635.9(0.1)</td>
<td>634.9(0.1)</td>
<td>634.9(0.1)</td>
<td>634.9(0.1)</td>
</tr>
<tr>
<td>pool</td>
<td>644.4(0.2)</td>
<td>634.0(0.2)</td>
<td>638.1(0.3)</td>
<td>636.1(0.3)</td>
<td>633.5(0.4)</td>
<td>639.7(0.2)</td>
<td>640.1(0.2)</td>
<td>640.5(0.1)</td>
<td>639.8(0.1)</td>
<td>638.6(0.1)</td>
</tr>
<tr>
<td>limma</td>
<td>643.1(0.3)</td>
<td>633.2(0.3)</td>
<td>641.1(0.4)</td>
<td>641.1(0.4)</td>
<td>641.0(0.2)</td>
<td>642.1(0.3)</td>
<td>641.1(0.2)</td>
<td>641.1(0.2)</td>
<td>641.1(0.2)</td>
<td>641.1(0.3)</td>
</tr>
</tbody>
</table>
AUROC is simply a different criterion than the usual loss functions for the variances. The AUROC tries to estimate the probability of giving higher ranking to a random differentially expressed gene than to a random equivalently expressed gene. It may or may not correspond to using good variance estimates, and it does depend upon which contrast is of interest. Further studies on methods optimizing AUROC should be valuable.

One can also see from Tables 2.4 and 2.5 that, for the cross-validation procedures, criteria based on full data and criteria only based on specific parts of the data do not necessarily select the same model. For contrast 1, contrast-specific CV log likelihood deteriorates the AUROC compared with the full CV log likelihood, whereas contrast-specific PRESS improves AUROC compared with the full PRESS. However, for contrast 2, this is further complicated by $E(r_g)$; when $E(r_g)$ is large, the contrast-specific CV log likelihood also improves AUROC, but on the contrary, contrast-specific PRESS produces worse results this time; and the conclusion reverses when $E(r_g) = 1$ or is close to 1. Hence, these cross-validation procedures that are based on only part of the data are preferred to the ordinary ones only under some but not all situations, and care has to be taken in practice to choose a good procedure.

Conclusions from selecting an overall model for all genes (data not shown) generally agree with those from gene-wise model selection (Table 2.4). Moreover, when $E(r_g)$ is larger, most of the model selection procedures will choose the correct model, and hence their differences in AUROC’s are largely indiscernible from each other.

When comparing model selection procedures with the variance shrinkage procedure limma, sample size plays an important role. When sample size is small (Table 2.5), variance shrinkage often outperforms model selection procedures by a large margin, except in a few cases where it is slightly worse than the best performing procedure. On the other hand, when sample sizes increase (Table 2.4), the advantage of variance shrinkage gradually fades away and shrinkage performs similarly to model selection procedures. This phenomenon is intuitive, because when sample sizes are small, the total information contained in the additional data is still relatively scarce compared with the information contained in the large number of genes from a single dataset. In this case, pooling information across genes is more beneficial. Other other hand,
when sample sizes increase (or when more datasets are incorporated into the joint analysis),
more and more information can be obtained from additional datasets and the estimates of
gene-wise variance become more and more precise. Of course, the number of genes within the
dataset from a single study does not increase for a fixed microarray platform, so there is no
more information to be borrowed from other genes within a single dataset. Hence shrinkage is
not necessarily preferred in such cases. Although we cannot arbitrarily increase sample size in
our simulations due the constraints of total population sample size, it is reasonable to believe
that when we greatly increase the sample size by including more datasets for a joint analysis,
shrinkage methods would eventually have little impact on the results.

So, our general recommendation for pooling variance estimates across datasets to rank genes
from most significant to least significant is 1) to use pooled analysis when variances are judged
using our methods to be approximately equal across datasets, 2) to use a pooled analysis even
when variances differ across datasets if the contrasts of interest are within the datasets with
larger variance, and 3) to use a separate analysis when variances differ across datasets and the
contrasts of interest are within the datasets with smaller variances. In our experience, shrinking
variance estimates by borrowing information across genes is seldom harmful and usually helpful.
Thus, whether gene-specific variance estimates are obtained by pooling across datasets or not,
we recommend shrinking gene-specific variance estimates by borrowing information across genes
using a procedure like \textit{limma}.

\textbf{2.3.3 Control of FDR}

For the massive amount of hypothesis testing in microarray data, controlling family-wise
type I errors is too conservative. One often chooses to control FDR at some pre-specified level.
It is of interest to see whether FDR is still under control after selecting the variance part of the
model. Selected results of the mean false discovery proportions (FDP\%) and the number of
rejected hypotheses (#Rej) are shown in Tables 2.6 to 2.8.

Table 2.6 shows results when the true model is homogeneous and sample size is small for
the contrast in dataset 1, using the BH procedure and with model selection performed on
Table 2.6  False discovery proportions and number of rejections for contrast  
1 from individual gene model selection on dataset GSE5406 using  
the BH method when $n_1 = n_2 = 3$ and $r = 1$ for all genes  

<table>
<thead>
<tr>
<th>Method</th>
<th>FDR = 5%</th>
<th>FDR = 10%</th>
<th>FDR = 15%</th>
<th>FDR = 20%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FDP%</td>
<td>#Rej</td>
<td>FDP%</td>
<td>#Rej</td>
</tr>
<tr>
<td>AIC</td>
<td>5.9(1.1)</td>
<td>10.8(2.4)</td>
<td>11.6(1.4)</td>
<td>56.2(14.1)</td>
</tr>
<tr>
<td>AICc</td>
<td>6.7(1.2)</td>
<td>13.3(2.4)</td>
<td>11.5(1.4)</td>
<td>53.6(10.4)</td>
</tr>
<tr>
<td>BIC</td>
<td>5.9(1.1)</td>
<td>11.2(2.4)</td>
<td>11.6(1.4)</td>
<td>56.8(14.2)</td>
</tr>
<tr>
<td>CAIC</td>
<td>6.6(1.2)</td>
<td>13.1(2.7)</td>
<td>11.7(1.4)</td>
<td>60.1(13.6)</td>
</tr>
<tr>
<td>HQIC</td>
<td>5.7(1.2)</td>
<td>9.3(1.9)</td>
<td>11.1(1.4)</td>
<td>50.2(13.6)</td>
</tr>
<tr>
<td>CV-lik_full</td>
<td>5.5(1.0)</td>
<td>9.6(1.7)</td>
<td>11.3(1.6)</td>
<td>45.3(10.5)</td>
</tr>
<tr>
<td>CV-lik_specific</td>
<td>5.4(1.4)</td>
<td>6.9(1.2)</td>
<td>10.6(1.6)</td>
<td>32.3(7.8)</td>
</tr>
<tr>
<td>PRESS_full</td>
<td>5.6(1.4)</td>
<td>7.0(1.3)</td>
<td>11.0(1.7)</td>
<td>35.9(9.8)</td>
</tr>
<tr>
<td>PRESS_specific</td>
<td>4.7(1.0)</td>
<td>5.9(0.9)</td>
<td>12.7(1.7)</td>
<td>38.2(10.5)</td>
</tr>
<tr>
<td>MRPP_0.03</td>
<td>5.7(1.0)</td>
<td>9.9(1.6)</td>
<td>9.3(1.3)</td>
<td>42.1(8.1)</td>
</tr>
<tr>
<td>MRPP_0.06</td>
<td>6.0(1.1)</td>
<td>10.2(1.7)</td>
<td>10.2(1.3)</td>
<td>44.4(9.0)</td>
</tr>
<tr>
<td>MRPP_0.11</td>
<td>6.0(1.1)</td>
<td>10.3(1.8)</td>
<td>11.4(1.4)</td>
<td>47.9(10.3)</td>
</tr>
<tr>
<td>separate</td>
<td>3.8(1.6)</td>
<td>0.6(0.1)</td>
<td>4.8(1.5)</td>
<td>3.4(1.8)</td>
</tr>
<tr>
<td>pool</td>
<td>3.4(0.7)</td>
<td>8.4(1.4)</td>
<td>6.0(1.0)</td>
<td>31.0(6.0)</td>
</tr>
<tr>
<td>limma</td>
<td>4.0(1.5)</td>
<td>14.6(5.4)</td>
<td>5.4(1.4)</td>
<td>54.2(17.1)</td>
</tr>
<tr>
<td>Method</td>
<td>FDR = 5%</td>
<td>FDR = 10%</td>
<td>FDR = 15%</td>
<td>FDR = 20%</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------</td>
</tr>
<tr>
<td></td>
<td>FDP%</td>
<td>#Rej</td>
<td>FDP%</td>
<td>#Rej</td>
</tr>
<tr>
<td>AIC</td>
<td>4.8(0.5)</td>
<td>479.8(17.2)</td>
<td>9.1(0.9)</td>
<td>737.5(27.0)</td>
</tr>
<tr>
<td>AICc</td>
<td>4.8(0.5)</td>
<td>485.3(16.5)</td>
<td>9.0(0.8)</td>
<td>740.5(26.1)</td>
</tr>
<tr>
<td>BIC</td>
<td>4.8(0.5)</td>
<td>487.5(16.1)</td>
<td>8.9(0.8)</td>
<td>740.0(25.4)</td>
</tr>
<tr>
<td>CAIC</td>
<td>4.6(0.5)</td>
<td>485.8(15.6)</td>
<td>8.7(0.8)</td>
<td>735.6(24.6)</td>
</tr>
<tr>
<td>HQIC</td>
<td>4.8(0.6)</td>
<td>483.4(16.9)</td>
<td>9.1(0.9)</td>
<td>740.8(26.6)</td>
</tr>
<tr>
<td>CV-lik_full</td>
<td>4.7(0.6)</td>
<td>476.4(16.6)</td>
<td>8.9(0.9)</td>
<td>728.1(26.3)</td>
</tr>
<tr>
<td>CV-lik_specific</td>
<td>4.3(0.5)</td>
<td>439.4(16.1)</td>
<td>8.3(0.8)</td>
<td>677.9(25.0)</td>
</tr>
<tr>
<td>PRESS_full</td>
<td>4.6(0.5)</td>
<td>453.3(17.6)</td>
<td>8.9(0.8)</td>
<td>707.2(27.6)</td>
</tr>
<tr>
<td>PRESS_specific</td>
<td>5.0(0.6)</td>
<td>497.2(17.9)</td>
<td>9.6(0.9)</td>
<td>778.1(29.4)</td>
</tr>
<tr>
<td>MRPP 0.03</td>
<td>4.2(0.5)</td>
<td>465.2(18.0)</td>
<td>8.0(0.8)</td>
<td>707.8(24.1)</td>
</tr>
<tr>
<td>MRPP 0.06</td>
<td>4.4(0.5)</td>
<td>470.7(15.3)</td>
<td>8.4(0.8)</td>
<td>717.2(24.8)</td>
</tr>
<tr>
<td>MRPP 0.11</td>
<td>4.6(0.5)</td>
<td>474.3(15.8)</td>
<td>8.7(0.8)</td>
<td>724.3(25.6)</td>
</tr>
<tr>
<td>separate</td>
<td>3.5(0.5)</td>
<td>296.3(16.4)</td>
<td>6.6(0.7)</td>
<td>522.9(25.6)</td>
</tr>
<tr>
<td>pool</td>
<td>3.5(0.5)</td>
<td>442.5(14.8)</td>
<td>7.1(0.8)</td>
<td>676.8(23.5)</td>
</tr>
<tr>
<td>limma</td>
<td>3.2(0.5)</td>
<td>371.4(19.4)</td>
<td>6.4(0.8)</td>
<td>601.1(29.8)</td>
</tr>
</tbody>
</table>

Table 2.7  False discovery proportions and number of rejections for contrast 1 from individual gene model selection on dataset GSE5406 using the BH method when $n_1 = n_2 = 7$ and $r = 1$ for all genes.
Table 2.8  False discovery proportions and number of rejections for contrast 2 from individual gene model selection on dataset GSE5406 using the BH method when $n_1 = n_2 = 3$ and $\mathbb{E}(r_g) = 10$ for all genes

<table>
<thead>
<tr>
<th>Method</th>
<th>FDR = 5% FDP%</th>
<th>#Rej</th>
<th>FDR = 10% FDP%</th>
<th>#Rej</th>
<th>FDR = 15% FDP%</th>
<th>#Rej</th>
<th>FDR = 20% FDP%</th>
<th>#Rej</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIC</td>
<td>31.9(1.3)</td>
<td>164.6(13.7)</td>
<td>39.2(1.2)</td>
<td>332.1(26.1)</td>
<td>43.5(1.2)</td>
<td>498.7(38.0)</td>
<td>46.8(1.1)</td>
<td>674.7(51.1)</td>
</tr>
<tr>
<td>AICc</td>
<td>32.8(1.3)</td>
<td>386.4(29.2)</td>
<td>41.6(1.2)</td>
<td>795.8(54.7)</td>
<td>47.1(1.2)</td>
<td>1204.4(77.7)</td>
<td>51.0(1.1)</td>
<td>1632.3(100.6)</td>
</tr>
<tr>
<td>BIC</td>
<td>31.9(1.3)</td>
<td>171.2(14.1)</td>
<td>39.4(1.2)</td>
<td>343.8(26.8)</td>
<td>43.7(1.2)</td>
<td>515.6(39.0)</td>
<td>47.0(1.1)</td>
<td>698.0(52.5)</td>
</tr>
<tr>
<td>CAIC</td>
<td>32.3(1.3)</td>
<td>244.5(19.3)</td>
<td>40.1(1.2)</td>
<td>494.6(36.1)</td>
<td>45.2(1.1)</td>
<td>742.8(52.4)</td>
<td>48.6(1.1)</td>
<td>996.3(69.0)</td>
</tr>
<tr>
<td>HQIC</td>
<td>31.1(1.4)</td>
<td>122.6(10.6)</td>
<td>38.7(1.2)</td>
<td>250.4(20.6)</td>
<td>42.6(1.1)</td>
<td>383.1(30.7)</td>
<td>45.5(1.1)</td>
<td>523.8(42.2)</td>
</tr>
<tr>
<td>CV-lik._full</td>
<td>31.2(1.3)</td>
<td>182.9(15.2)</td>
<td>38.5(1.2)</td>
<td>388.8(30.0)</td>
<td>43.3(1.2)</td>
<td>603.7(45.0)</td>
<td>46.5(1.1)</td>
<td>831.8(60.9)</td>
</tr>
<tr>
<td>CV-lik._specific</td>
<td>29.4(1.4)</td>
<td>90.5(8.4)</td>
<td>36.7(1.2)</td>
<td>193.8(17.1)</td>
<td>40.5(1.1)</td>
<td>299.9(25.9)</td>
<td>43.5(1.1)</td>
<td>423.1(36.3)</td>
</tr>
<tr>
<td>PRESSfull</td>
<td>28.9(1.5)</td>
<td>83.7(7.9)</td>
<td>36.6(1.2)</td>
<td>181.6(16.1)</td>
<td>40.3(1.2)</td>
<td>286.9(25.1)</td>
<td>43.2(1.1)</td>
<td>406.7(35.6)</td>
</tr>
<tr>
<td>PRESS_specific</td>
<td>29.5(1.3)</td>
<td>172.3(14.7)</td>
<td>37.3(1.2)</td>
<td>394.9(31.5)</td>
<td>42.1(1.2)</td>
<td>639.1(48.8)</td>
<td>45.9(1.1)</td>
<td>910.8(68.3)</td>
</tr>
<tr>
<td>MRPP0.03</td>
<td>32.5(1.3)</td>
<td>441.5(33.3)</td>
<td>41.3(1.2)</td>
<td>959.6(65.9)</td>
<td>47.6(1.2)</td>
<td>1487.6(94.5)</td>
<td>51.4(1.1)</td>
<td>2048.0(121.9)</td>
</tr>
<tr>
<td>MRPP0.06</td>
<td>32.5(1.3)</td>
<td>376.0(28.8)</td>
<td>40.9(1.2)</td>
<td>809.9(55.8)</td>
<td>46.2(1.2)</td>
<td>1250.3(81.0)</td>
<td>50.3(1.1)</td>
<td>1723.9(106.2)</td>
</tr>
<tr>
<td>MRPP0.11</td>
<td>31.7(1.3)</td>
<td>282.6(22.1)</td>
<td>39.7(1.2)</td>
<td>598.2(43.3)</td>
<td>44.9(1.2)</td>
<td>927.5(63.7)</td>
<td>48.7(1.1)</td>
<td>1279.2(85.1)</td>
</tr>
<tr>
<td>separate</td>
<td>5.0(1.9)</td>
<td>0.5(0.1)</td>
<td>8.7(2.5)</td>
<td>0.6(0.1)</td>
<td>10.0(2.4)</td>
<td>1.6(0.4)</td>
<td>12.2(2.2)</td>
<td>5.7(1.5)</td>
</tr>
<tr>
<td>pool</td>
<td>32.0(1.3)</td>
<td>557.9(42.1)</td>
<td>42.0(1.3)</td>
<td>1239.8(84.5)</td>
<td>48.2(1.2)</td>
<td>1958.4(121.0)</td>
<td>52.9(1.2)</td>
<td>2717.2(153.9)</td>
</tr>
<tr>
<td>limma</td>
<td>1.9(1.2)</td>
<td>2.6(0.7)</td>
<td>5.0(1.5)</td>
<td>19.6(5.3)</td>
<td>6.9(1.4)</td>
<td>50.6(12.7)</td>
<td>8.7(1.5)</td>
<td>103.2(23.8)</td>
</tr>
</tbody>
</table>
a gene-by-gene basis. We see that always using the correct model, \textit{i.e.}, the pooled analysis, generally controls the FDR below the desired level. Although always using the separate analysis also successfully controls the FDR, the number of rejections (\textit{i.e.}, power) is much lower compared with using the correct model.

Also note in Table 2.6 that, although both the separate analysis and the pooled analysis control FDR, this is no longer the case when some genes use a separate analysis and others use the pooled analysis. For most model selection procedures, gene-by-gene model selection tends to increase the false discovery proportions. Because the BH procedure does not estimate the proportion of null hypotheses, it is generally considered as a conservative method compared to methods that are more adaptive, \textit{e.g.}, the ST procedure. Not surprisingly, if we use the ST procedure after gene-by-gene model selection, the actual FDPs increase further above the desired level (results not shown).

However, as sample size increases (Table 2.7), the liberalism of \(p\)-values is largely alleviated and the control of FDR is still successful for BH, even if after gene-wise model selection. The ST method is still slightly liberal when sample size increases, but the severity is lower and probably ignorable in practice (results not shown). Moreover, variance shrinkage through \textit{limma} generally outperforms model selection when sample size is small (Table 2.6), but is not as powerful as the best model selection procedures when sample size increases (Table 2.7). This is consistent with the observation on ranking genes in the previous subsection.

When \(\mathbb{E}(r_g)\) increases to 10, the FDRs are all controlled at the desired levels, even after gene-by-gene model selection (results not shown). However, the pooled analysis is extremely under powered for contrasts in dataset 1. This is because dataset 1 is simulated with smaller variances than dataset 2 and the pooled analysis for contrasts in dataset 1 always vastly overestimates the variances and reduces power. At higher FDR levels, the CV log likelihood and the MRPP procedure for gene-by-gene analysis result in lower power than the other model selection methods, probably due to their preference for the smaller model under this simulation situation. The variance shrinkage approach is not seriously affected by \(\mathbb{E}(r_g)\) but is affected mainly by sample size.
However, when we consider the contrast within the second dataset, where the true variances are much larger than the first, the only method that controls the FDR properly uses a separate analysis for all genes (Table 2.8). Pooled analysis and all model selection procedures produce mean FDP far above the desired level. Note, however, that in this situation the pooled analysis is able to produce the best ranking of genes, in terms of AUROC.

When variance model selection is conducted on the whole genome scale, the final FDP is largely dependent on how often the procedure chooses the separate model. For example, in the case shown in Table 2.8 but with model selection methods applied to all genes together, only AICc and CV log likelihood choose the pooled model frequently, and hence their AUROC’s are better but FDR is not controlled. Other methods mostly choose a separate model, and their FDR is below the desired level, although their lists of differentially expressed genes are worse than using the pooled model.

Hence, one faces the dilemma of whether we should care more about gene ranking or care more about control of FDR. Similar to the recommendation in [Demirkale et al., 2009], we suggest the following strategy: when combined sample size is small or when model selection procedures suggest that the variances are too different to be combined across studies, use a separate analysis to determine the number of genes that can be declared differentially expressed and a pooled analysis to determine which of the genes are declared as differentially expressed. This hybrid approach will both control FDR well below the desired level and provide a good list of candidate genes for further study.

2.3.4 Size and power of modified MRPP compared with alternative univariate tests

Among other model selection procedures considered in this study, our modified MRPP is the only one that is based on hypothesis testing, and behaves largely different than other information criteria based methods. Hence it is more reasonable to compare our modified MRPP with other commonly used hypothesis tests for unequal variances in the univariate setting, where all such tests are applicable.
Table 2.11 Probability that the test statistic under the alternative is at least \( \times 7.10 \).  
\[
\begin{array}{ccccccc}
\text{Sample sizes} & \text{Kullback-Leibler divergence } \times 1000 \\
 & \text{Lev} & \text{RLev} & \text{BF} & \text{F} & \text{MRPP} \\
3,3 & 88.0(0.4) & 3.0(0.1) & 104.4(0.5) & 4.4(0.1) & 2.0(0.1) \\
3,4 & 67.2(0.3) & 4.7(0.1) & 18.2(0.2) & 7.7(0.1) & 2.9(0.1) \\
3,7 & 40.9(0.3) & 6.0(0.1) & 106.8(0.8) & 10.6(0.1) & 2.9(0.1) \\
3,10 & 32.4(0.2) & 8.2(0.1) & 34.4(0.3) & 11.7(0.1) & 3.0(0.1) \\
4,4 & 48.2(0.3) & 4.9(0.1) & 2.9(0.1) & 11.0(0.1) & 3.7(0.1) \\
4,7 & 31.8(0.2) & 7.6(0.1) & 11.8(0.1) & 18.8(0.2) & 5.8(0.1) \\
4,10 & 28.9(0.2) & 11.4(0.1) & 9.7(0.1) & 24.2(0.2) & 8.3(0.1) \\
7,7 & 31.4(0.2) & 18.1(0.1) & 46.3(0.5) & 41.3(0.3) & 16.6(0.1) \\
7,10 & 27.1(0.2) & 17.9(0.2) & 22.2(0.2) & 46.5(0.2) & 17.3(0.1) \\
\end{array}
\]

Table 2.9 Estimated Kullback-Leibler divergence of nominal to theoretical \( p \)-value distributions

Table 2.10 Probability that the test statistic under the alternative is at least as extreme as the test statistic under the null when \( \mathbb{E}(r_g) = 1 \)

\[
\begin{array}{ccccccc}
\text{Sample sizes} & \text{Probability } \times 1000 \\
 & \text{Lev} & \text{RLev} & \text{BF} & \text{F} & \text{MRPP} \\
3,3 & 530.2(0.4) & 530.0(0.4) & 530.4(0.5) & 530.4(0.5) & 530.0(0.4) \\
3,4 & 527.7(0.4) & 531.1(0.4) & 520.6(0.4) & 533.3(0.4) & 531.0(0.4) \\
3,7 & 523.6(0.5) & 532.5(0.5) & 515.7(0.5) & 534.8(0.5) & 532.2(0.5) \\
3,10 & 521.3(0.5) & 532.5(0.5) & 510.8(0.5) & 535.7(0.4) & 532.2(0.5) \\
4,4 & 538.0(0.5) & 537.6(0.5) & 537.5(0.4) & 538.4(0.5) & 537.6(0.5) \\
4,7 & 535.4(0.5) & 540.9(0.5) & 535.8(0.5) & 541.9(0.5) & 540.6(0.5) \\
4,10 & 533.7(0.5) & 540.7(0.4) & 531.3(0.4) & 542.1(0.4) & 540.4(0.4) \\
7,7 & 552.8(0.5) & 552.1(0.6) & 553.1(0.5) & 552.8(0.5) & 552.2(0.6) \\
7,10 & 553.5(0.4) & 555.3(0.5) & 550.9(0.4) & 556.1(0.4) & 555.1(0.5) \\
\end{array}
\]

Table 2.11 Probability that the test statistic under the alternative is at least as extreme as the test statistic under the null when \( \mathbb{E}(r_g) = 10 \)

\[
\begin{array}{ccccccc}
\text{Sample sizes} & \text{Probability } \times 1000 \\
 & \text{Lev} & \text{RLev} & \text{BF} & \text{F} & \text{MRPP} \\
3,3 & 730.7(0.4) & 724.6(0.5) & 732.0(0.4) & 733.0(0.5) & 730.1(0.5) \\
3,4 & 775.1(0.3) & 743.1(0.4) & 826.4(0.3) & 754.4(0.4) & 768.0(0.4) \\
3,7 & 827.8(0.4) & 769.8(0.4) & 876.1(0.3) & 782.6(0.4) & 816.1(0.4) \\
3,10 & 850.3(0.3) & 782.8(0.4) & 904.4(0.2) & 796.9(0.4) & 836.5(0.3) \\
4,4 & 773.8(0.5) & 768.2(0.5) & 769.6(0.5) & 778.8(0.4) & 774.9(0.5) \\
4,7 & 832.4(0.4) & 798.0(0.4) & 829.0(0.4) & 815.1(0.3) & 831.8(0.4) \\
4,10 & 854.4(0.3) & 809.1(0.3) & 864.6(0.3) & 829.6(0.3) & 854.0(0.3) \\
7,7 & 852.0(0.4) & 847.1(0.4) & 855.7(0.4) & 855.0(0.4) & 853.3(0.4) \\
7,10 & 877.9(0.3) & 862.6(0.4) & 896.7(0.3) & 875.5(0.3) & 883.5(0.4) \\
\end{array}
\]
Table 2.9 shows the estimated Kullback-Leibler divergence of the nominal $p$-value distribution under the null hypothesis of each test compared to the theoretical uniform distribution, from a two-sample comparison design, with varying sample sizes. One can see that both methods based on our decorrelated and reduced dataset, i.e., the reduced Levene’s test ("RLev") and MRPP, have small Kullback-Leibler distances under all sample sizes. The original Levene’s test ("Lev") generally has a poor null distribution. The $F$-test approximates the null well when sample sizes are small, but not when sample sizes increase. The Brown-Forsythe test ("BF") has a close to uniform null distribution when both sample sizes are even, but it becomes worse if one of the sample sizes is odd, and is usually the worst compared to other methods when both sample sizes are odd. This is because the definition of median depends on the parity of sample size. In terms of null distribution, our proposed RLev and MRPP outperform other methods under our realistic simulation settings.

Table 2.10 shows the probability that the test statistic under the alternative is at least as extreme as the test statistic under the null when $E(r_g) = 1$. This corresponds to small departure of the alternative from the null. We can see that whenever the two sample sizes are the same, all five tests have similar power. Otherwise, the $F$-test, the modified MRPP, and the reduced Levene’s test have better power than the original Levene’s test and the Brown-Forsythe test.

Table 2.11 shows the same probability when $E(r_g) = 10$, which corresponds to a large difference between the null and the alternative. In this case, the reduced Levene’s test does not perform well, but the Brown-Forsythe test is often the best, usually closely followed by Levene’s test and the modified MRPP. The performance of the $F$-test seems to depend on the balancedness of the design—when sample sizes are equal, its power is high; but when sample sizes are different, its power deteriorates greatly.

Considering both small and large departures from the null hypotheses, the modified MRPP is the only one that always has relatively good power. Since it also controls type I error better than others (Table 2.9) and directly applies to any high dimensional situation, it is our recommended test for heterogeneity in practice to replace the aforementioned alternatives.
2.4 Summary

In this study, we proposed several new model selection procedures for selecting the variance part of linear models. Our modified MRPP test has tighter control of type I errors and also has good power compared to other methods. Our cross-validation procedures are able to differentiate linear models that only differ in the variance assumptions. We also give the correct AICc formula that removes the bias in AIC when multiple variances need to be estimated independently.

Through real data based simulation, we found that using the correct models does not necessarily provide the best separation between differentially and equivalently expressed genes, although using the correct models can control FDR at desired levels. A hybrid procedure to decouple FDR control and differential expression detection is recommended, as in [Demirkale et al., 2009].

Variance model selection for mixed linear models is more complicated, primarily because the estimated variance components may lie on the boundary of the parameter space with positive probability [Greven and Kneib, 2009]. We may envisage using some of our methods, e.g., AICc, in special types of mixed models, but it would be valuable in future studies to extend our methods to general mixed models.

Our simulations suggest that neither shrinkage estimation within a dataset nor model selection across several datasets is always preferred to the other. Fortunately, it is straightforward to combine both approaches by first pooling across datasets (if our proposed methods suggest that pooling will be beneficial) and then shrinking the resulting estimates by borrowing information across genes using a procedure like limma.

In summary, for microarray data analysis, our general recommendation for ranking genes is to use a pooled analysis only when variances are judged to be equal or when variances differ but the contrasts of interest are within the datasets with larger variance. For control of FDR, pooled analysis should only be applied when variances are judged to be equal and the combined sample size is moderately large, irrespective of which analysis has been used to rank genes. For both ranking genes and controlling FDR, shrinkage estimation of variances across genes is
recommended, irrespective of whether additional datasets will be used to estimate variances. If homogeneity of variances will be tested separately for each gene, then the modified MRPP procedure is preferred to other univariate tests.

Bibliography


CHAPTER 3. IMPROVED ESTIMATION OF THE NONCENTRALITY PARAMETER DISTRIBUTION FROM A LARGE NUMBER OF $t$-STATISTICS, WITH APPLICATIONS TO FALSE DISCOVERY RATE ESTIMATION IN MICROARRAY DATA ANALYSIS

A paper to be submitted to *Biometrics*

Long Qu, Dan Nettleton, Jack C.M. Dekkers

**Abstract**

Given a large number of $t$-statistics, we consider the problem of approximating the distribution of noncentrality parameters (NCPs) by a continuous density. This problem is closely related to the control of false discovery rates (FDR) in massive hypothesis testing applications, e.g., microarray gene expression analysis. Our methodology is similar to, but improves upon, the existing approach by Ruppert et al. [2007]. We provide parametric, nonparametric, and semiparametric estimators for the distribution of NCPs, as well as estimates of the FDR and local FDR. In the parametric situation, we assume that the noncentrality parameters follow a distribution that leads to an analytically available marginal distribution for the test statistics. In the nonparametric situation, we use convex combinations of basis density functions to estimate the density of the NCPs. A sequential quadratic programming procedure is developed to maximize the penalized likelihood. Models are selected with the approximate network information criterion. A semiparametric estimator is also developed to combine both parametric and nonparametric fits. Simulations show that, under a variety of situations, our density estimates are closer to the underlying truth and our FDR estimates are improved compared with alternative methods. A real myeloma dataset is used in simulations to confirm the good
performance characteristics even when the assumptions of our approach are only approximately satisfied.

3.1 Introduction

In microarray gene expression profiling experiments, a large number of hypothesis tests are often performed. For example, in a two-treatment microarray experiment, t-tests will be simultaneously conducted for each of the tens of thousands of genes. To adjust for multiplicity, control of the FDR or variants thereof [Benjamini and Hochberg, 1995; Storey and Tibshirani, 2003] is the current standard practice. Most methods for estimating and controlling the FDR require an estimate of the proportion, \( \pi_0 \), of truly null hypotheses among a large number of tests. Due to the problem of unidentifiability of \( \pi_0 \) [Genovese and Wasserman, 2002], the vast majority of methods choose to estimate an identifiable upper bound of \( \pi_0 \), either parametrically [Allison et al., 2002] or nonparametrically [Benjamini and Hochberg, 2000; Broberg, 2004, 2005; Cheng, 2006; Dalmasso et al., 2007; Efron, 2005; He and Ng, 1999; Jin and Cai, 2007; Langaas et al., 2005; Mosig et al., 2001; Nettleton et al., 2006; Pounds and Cheng, 2004; Scheid and Spang, 2004; Schweder and Spjøtvoll, 1982; Storey and Tibshirani, 2003; Tsai et al., 2003; Turkheimer et al., 2001; Wu et al., 2006]. Two exceptions are Lai [2007] and Ruppert et al. [2007], both of which try to target \( \pi_0 \) itself instead of its upper bound. Lai [2007] gets around the unidentifiability problem by assuming that for each hypothesis, two independent tests are available, so that a moment based estimator for \( \pi_0 \) can be found. In practice, data splitting is used to provide the two independent tests. The performance of this estimator is satisfactory if the NCPs corresponding to the alternative hypotheses are far away from zero. However, if most of the NCPs from the alternative hypotheses concentrate closely around zero, the estimator is positively biased, with a performance very similar to those that estimate the upper bound of \( \pi_0 \).

The seminal work of Ruppert et al. [2007] assumed that, for a large number of t-tests, the empirical distribution of the absolute values of the NCPs from the alternative hypotheses can be approximated by a smooth continuous distribution with density \( g \). They modeled \( g \)
with B-splines that were normalized to be valid densities. After binning the $p$-values, they developed penalized least squares estimators for $\pi_0$ and $g$ through quadratic programming, with a penalty on the smoothness of $g$. Tuning parameters were then chosen by approximate generalized cross-validation, with the approximativeness coming from ignoring the convex linear constraints in estimation. This estimator has shown the ability to reduce the bias in estimating $\pi_0$ in simulation studies. Moreover, the method also provides important information about how large the NCPs are, which can be useful for planning future experiments. Their analysis of a barley infection dataset showed that $g$ indeed had most of its probability close to zero, which implies that relatively large sample sizes would be required to discover a large proportion of the differentially expressed genes while controlling FDR at reasonable levels.

Although the approach of Ruppert et al. [2007] (denoted as the RNH method hereafter) has advantages over competing methods, we have identified several aspects of the approach that can be improved. First, the method uses $p$-values from two-sided $t$-tests, which carry no information about the direction of changes for differentially expressed genes. As Sun and Cai [2007] point out, ignoring information about the direction of changes will be inefficient unless the NCP distribution is perfectly symmetric about zero.

A second issue related to efficiency is that the RNH method starts from binning the $p$-values so that quadratic programming can be used to solve the estimation problem. Binning inevitably causes some loss of information. Thus, the resulting estimators are not as efficient as those based on the original data.

A third issue of the RNH approach arises due to the difficulty in evaluating the mixture distribution of triangular linear B-spline density with the noncentral $t$-density. To evaluate this infinite mixture, the RNH method relies on numerical integrations. As stated in Ruppert et al. [2007], the actual computation only requires less than 10 seconds if the integral is directly available to be loaded into memory, but preparing the integral alone can take more than several minutes. Because high-throughput technologies keep developing, the number of tests to be conducted also increases and numerical integration could eventually become rather awkward in terms of speed.
A fourth issue that applies to the unweighted version of the RNH estimator of $g$ is a lack of stability when the average power of the tests is high. The resulting estimate of $g$ can have unusual bumps near the right boundary of the assumed support that are hard to interpret biologically. These bumps are typically caused by a few large bin counts that contribute more to the residual sums of squares than the vast majority of small bin counts and force the resulting estimate of $g$ to be unduly rough to capture the noise in the large bin counts. This problem can be partly alleviated by weighting, with weights determined by estimates from an initial unweighted fit, but this involves more computation and does not always eliminate the problem.

Lastly, when choosing the tuning parameter, the RNH method uses the degrees of freedom from an unconstrained estimate that does not guarantee $g$ to be a valid density, but the final estimate of $g$ solves a constrained optimization problem. Intuition suggests that the constrained estimates should cost fewer degrees of freedom than the unconstrained estimates, as long as at least some constraints are active at the final solution.

In this study, we develop new methods that avoid or alleviate the aforementioned shortcomings of the RNH method. To make our presentation easier to follow, we start the next section with a simple parametric model, which assumes that the NCPs are normally distributed. In section 3.3, we extend the parametric model to the non-parametric situation, which is closer to Ruppert et al. [2007]. A combination of parametric and nonparametric fits is considered in section 3.4, leading to a semiparametric method. Section 3.5 explains how our methods can be used to estimate false discovery rates. Simulation results are provided in section 3.6 to demonstrate the performance of our methods. The paper concludes with a brief summary in Section 3.7.

### 3.2 A parametric model for the $t$-statistics

We first consider parametric estimation of $\pi_0$ and the density of NCPs. Let $t_i$ be the $t$-statistic for the $i$th null hypothesis $H_i$, where $i = 1, 2, \ldots, G$ and $G$ is a large integer. If $H_i$ is true, then $t_i$ follows a central $t$-distribution, with appropriate degrees of freedom $\nu_i$. If $H_i$ is false, then $t_i$ follows a noncentral $t$-distribution with density function $h_t(\cdot; \nu_i, \delta_i)$, where $\delta_i$
is the non-centrality parameter and \( \nu_i \) is the degrees of freedom. The statistics \( t_1, t_2, \ldots, t_G \) are assumed to be independent and the proportion of the true null hypotheses is denoted as \( \pi_0 \in [0, 1] \).

We complete our parametric model by further assuming that the \( \delta_i \) are independently and identically distributed as \( \text{Normal}(\mu, \gamma^2) \), for all \( i \) such that \( H_i \) is false. This implies that both positive and negative values of \( \delta_i \) are allowed, and that the distribution need not be symmetric about zero. This avoids the first shortcoming of the RNH method. Given a \( \delta_i \) value, \( t_i \) is identically distributed as the random variable

\[
\frac{Z + \delta_i}{\sqrt{S/\nu_i}},
\]

where \( Z \sim \text{Normal}(0, 1) \) is independent of \( S \sim \chi^2_{\nu_i} \). Since the numerator \( Z + \delta_i \) is marginally distributed as \( \text{Normal}(\mu, 1+\gamma^2) \) and is independent of the denominator, the marginal distribution of \( t_i \) is the same as the random variable

\[
\sqrt{1 + \gamma^2} \frac{Z + \mu}{\sqrt{S/\nu_i}},
\]

i.e., \( t_i \) is marginally distributed as a scaled noncentral \( t \) random variable, with \( \nu_i \) degrees of freedom, noncentrality parameter \( \mu/\sqrt{1 + \gamma^2} \), and scale factor \( s = \sqrt{1 + \gamma^2} \).

In summary, the distribution of each \( t_i \) is a mixture of a central \( t \)-distribution and a scaled noncentral \( t \)-distribution, with density

\[
f_{\nu_i}(t_i; \mu, \gamma^2) = \pi_0 h_t(t_i; \nu_i, 0) + (1 - \pi_0) h_t \left( \frac{t_i}{s}; \nu_i, \frac{\mu}{s} \right) \frac{1}{s}.
\] (3.1)

For experiments where the amount of up-regulation can be reasonably assumed to be the same as the amount of down-regulation, the distribution of the \( \delta_i \) can be assumed to be centered on zero, i.e., \( \mu = 0 \). In this case, the marginal distribution of \( t_i \) becomes a mixture of a central \( t \)-distribution and a scaled central \( t \)-distribution, \( h_t(t_i/s; \nu_i, 0)/s \).

Because there are only 3 free parameters (\( \pi_0, \mu, \) and \( \gamma^2 > 0 \)) in model (3.1), maximum likelihood estimates can be easily obtained numerically, e.g., through a grid search or Newton-type optimization. As all of the data points are efficiently used in the maximum likelihood estimation, the binning data shortcoming in Ruppert et al. [2007] is avoided.
3.3 A nonparametric model for the $t$-statistics

Comparing model (3.1) with that of Ruppert et al. [2007], the latter is much more flexible in shape because the density of $|\delta_i|$ is modeled by a convex combination of normalized linear B-splines. Motivated by this idea, the rigidity of model (3.1) can be similarly removed by allowing the density of the $\delta_i$ to be a convex combination of a large number of basis distributions.

Given a user chosen moderately large integer, $K$, and an approximate interval, $[L,U]$, where the density of $\delta_i$ is concentrated, we assume that the $\delta_i$ values are independent draws from a mixture of $K$ normal distributions with the means located on equally spaced knots over $[L,U]$ and standard deviations given by the distances between adjacent knots. That is, the density of each $\delta_i$ is given by

$$p(\cdot) = \sum_{k=1}^{K} \beta_k \phi(\cdot; \tilde{\mu}_k, \tilde{\gamma}^2)$$  \hspace{1cm} (3.2)

where

$$\tilde{\gamma} = (U - L)/(K - 1),$$

$$\tilde{\mu}_k = L + (k - 1)\tilde{\gamma},$$

$\phi(\cdot; \tilde{\mu}_k, \tilde{\gamma}^2)$ is the density of the Normal($\tilde{\mu}_k, \tilde{\gamma}^2$) distribution, and $\beta_1, \beta_2, \ldots, \beta_K$ are unknown mixing proportions subject to $\sum_{j=1}^{K} \beta_j = 1$ and $\beta_k \geq 0$ for all $k = 1, 2, \ldots, K$. The actual values of $K$, $L$, and $U$ are not important, as long as $K$ is not too small and $[L,U]$ is not too narrow. In practice, we simply choose $K = 100$, with $L$ and $U$ being the lower and upper 0.01 quantile of all $t$-statistics, respectively.

Strictly speaking, this model is still parametric. However, because the Gaussian density is approximately zero outside 3 standard deviations around the mean, each basis function primarily covers a span of only 7 knots. Thus, it is similar to a fifth-degree B-spline spanning 7 knots. Given this “nearly local” coverage feature, we can see that as long as $K$ is sufficiently large, our model should behave nearly nonparametrically, just as the B-spline model of Ruppert et al. [2007] does.

Besides allowing both positive and negative $\delta_i$, our model using Gaussian radial basis functions has another benefit over the B-spline model in that, by arguments similar to those
in section 3.2, the marginal distribution of $t$-statistics is still analytically available. More specifically, the marginal density of $t_i$, for all $i = 1, 2, \ldots, G$, is

$$
\tilde{f}_{\nu_i} (t_i; \theta) = \pi_0 h_t(t_i; \nu_i, 0) + (1 - \pi_0) \sum_{k=1}^{K} \beta_k h_t(t_i/\tilde{s}; \nu_i, \tilde{\mu_k}/\tilde{s})/\tilde{s} = h_t(t_i; \nu_i, 0) + b_i'\theta,
$$

(3.3)

where $\tilde{s} = \sqrt{1 + \tilde{\gamma}^2}$ and $\theta = [\theta_1, \theta_2, \ldots, \theta_K]'$ is a reparameterized $K$-vector of unknown parameters such that $\pi_0 = 1 - \theta_1^T 1$ and $\beta_k = (1 - \pi_0)^{-1} \theta_k$ for $k = 1, 2, \ldots, K$, and $b_i$ is a $K$-vector with the $k$th element $h_t(t_i/\tilde{s}; \nu_i, \tilde{\mu_k}/\tilde{s})/\tilde{s} - h_t(t_i; \nu_i, 0)$. Correspondingly, the constraints become $\sum_{j=1}^{K} \theta_j \leq 1$ and $\theta_k \geq 0$ for all $k = 1, 2, \ldots, K$ under this parameterization.

### 3.3.1 Penalized likelihood

When $K$ is large, the model is nearly over specified, and the parameter estimates are unstable. Thus, we introduce a smoothness penalty on the log likelihood function to stabilize the estimates. A popular and intuitive penalty [Wahba, 1990] is based on integrated squared $r$th order derivatives of $p(\cdot)$,

$$
D(\theta) = \frac{\lambda}{2} \int_{-\infty}^{\infty} \left( \frac{d^r}{dx^r} p(x) \right)^2 dx = \frac{\lambda}{2 (\theta^T 1)^2} \theta^T \Omega \theta,
$$

(3.4)

where $\lambda > 0$ is a tuning parameter that controls the degree of smoothness of $p(\cdot)$ and $\Omega$ is a $K \times K$ matrix with $(j,k)$th element

$$
\Omega_{jk} = \int_{-\infty}^{\infty} \frac{d^r}{dx^r} \phi (x; \tilde{\mu_j}, \tilde{\gamma}^2) \frac{d^r}{dx^r} \phi (x; \tilde{\mu_k}, \tilde{\gamma}^2) \, dx.
$$

In nonparametric regression, a value of 2 is often chosen for $r$ so that linear regression functions are not penalized. For density estimation, the RNH method penalizes the finite difference of coefficients, an approximation to the first derivative ($r = 1$), emphasizing that the uniform density (constant function) does not receive any penalty. However, as previously mentioned, the distribution of noncentrality parameters is often more concentrated around zero. Based on this reasoning, we suggest setting $r = 3$, because this is the lowest derivative that allows a dome-shaped function (a quadratic function here) to be unpenalized. The mode
could be very close to zero so that probability is concentrated around zero. As will be seen in simulations based on real data, this choice works well. Furthermore, when \( r = 3 \), elements of \( \Omega \) are also analytically available, with

\[
\Omega_{jk} = \frac{\exp\left\{ -\frac{(\tilde{\mu}_j - \tilde{\mu}_k)^2}{2(2\tilde{\gamma}^2)^2} \right\}}{2\pi(2\tilde{\gamma}^2)^{13}} \left[ 15(2\tilde{\gamma}^2) - 45(2\tilde{\gamma}^2)^2(\tilde{\mu}_j - \tilde{\mu}_k)^2 + 15(2\tilde{\gamma}^2)(\tilde{\mu}_j - \tilde{\mu}_k)^4 - (\tilde{\mu}_j - \tilde{\mu}_k)^6 \right].
\]  

(3.5)

To estimate \( \theta \), let \( \ell_i(\theta) = -\log[f_{\nu_i}(t_i; \theta)] + D(\theta)/G \). The negative penalized log likelihood is then \( \ell(\theta) = \sum_{i=1}^{G} \ell_i(\theta) \). Our regularized estimate of \( \theta \) is given by

\[
\hat{\theta} = \arg \min_{\{\theta : C(\theta) \geq 0\}} \ell(\theta),
\]  

(3.6)

where the \( K + 1 \) constraints are given by

\[
C(\theta) = [I_K, -1]'\theta - [0', -1]' \geq 0,
\]

where \( I_K \) is the \( K \times K \) identity matrix and “\( \geq \)” denotes component-wise comparison.

### 3.3.2 Sequential quadratic programming

Optimization problem (3.6) is a non-linear programming problem subject to linear inequality constraints. A commonly used practical algorithm is sequential quadratic programming (SQP). SQP originated with applying Newton’s method to the Lagrangian function, which turns out to be equivalent to solving a quadratic programming problem in each iteration of Newton’s update. See Nocedal and Wright [2006, Chapter 18] for details. Since all of the constraints are linear in problem (3.6), the Hessian matrix of the Lagrangian function is the same as that of \( \ell(\cdot) \), and the SQP algorithm is particularly easy as the Lagrange multipliers do not need to be re-computed in each iteration. For a given \( \lambda \), the pseudo-code of our simple SQP algorithm is listed in Algorithm 3.1.

In Algorithm 3.1, the gradient and Hessian of \( \ell \) and the gradient of \( C \) are given by

\[
\nabla \ell(\theta) = \sum_{i=1}^{G} \nabla \ell_i(\theta), \quad \nabla \ell_i(\theta) = -\frac{1}{f_{\nu_i}(t_i; \theta)} b_i + \frac{\lambda}{G(1'\theta)^2} \Omega \theta - \frac{\lambda \theta' \Omega \theta}{G(1'\theta)^3} 1,
\]  

(3.7)
Algorithm 3.1 Pseudo-code for the SQP algorithm

1: Initialize $\theta$ to satisfy $C(\theta) \geq 0$;
2: repeat
3: compute $\nabla \ell(\theta)$, $\nabla^2 \ell(\theta)$ and $\nabla C(\theta)$ according to (3.7~3.9);
4: adjust $\nabla^2 \ell(\theta)$ to its nearest positive definite matrix;
5: $\hat{\vartheta} \leftarrow \text{arg min}_{\vartheta} \nabla \ell(\theta) \vartheta + 0.5 \vartheta^\prime \nabla^2 \ell(\theta) \vartheta \text{ subject to } \nabla C(\theta) \vartheta + C(\theta) \geq 0$;
6: $\theta \leftarrow \theta + \alpha \hat{\vartheta}$ where $0 \leq \alpha \leq 1$ is chosen such that $C(\theta) \geq 0$ and $\ell(\theta)$ does not increase;
7: until convergence

$$\nabla^2 \ell(\theta) = \sum_{i=1}^{G} \frac{1}{f_{\nu_i}(t_i; \theta)} b_i b_i^\prime + \frac{\lambda}{(1^\prime \theta)^2} \Omega - \frac{2\lambda}{(1^\prime \theta)^3} (\Omega \theta 1^\prime + 1^\prime \theta' \Omega) + \frac{3\lambda \theta' \Omega \theta}{(1^\prime \theta)^4} 11^\prime,$$  \quad (3.8)

and

$$\nabla C(\theta) = [I_K, -1]. \quad (3.9)$$

The step size factor $\alpha$ can be chosen by simple step-halving, \textit{i.e.}, starting from $\alpha = 1$, reducing $\alpha$ by half until the constraints are met and the objective is not increased.

The quadratic programming problem in each iteration can be easily solved using common statistical software, \textit{e.g.}, the \texttt{solve.QP} function in the \texttt{R} package \texttt{quadprog}, the \texttt{lsei} function in the \texttt{R} package \texttt{limSolve}, the \texttt{quadprog} function in optimization toolbox of \texttt{MATLAB}, or the \texttt{OPTQP} procedure in \texttt{SAS/OR}. Our code is implemented in the freely available package \texttt{pi0} in \texttt{R}.

### 3.3.3 Tuning parameter selection

Likelihood-based cross-validation can be applied to choose a good tuning parameter $\lambda$. But this will be time consuming as the model must be fitted to many subsets of the original data. Alternatively, the network information criterion (NIC) can be used to guide the choice of $\lambda$. NIC is a generalized version of Akaike’s information criterion (AIC) \cite[section 7.1~7.3 of Burnham and Anderson, 2002; Takeuchi, 1976] that can be applied to regularized estimation problems \cite[section 2.2 and 4.3 of Ripley, 1996; Moody, 1991, 1992]. The critical quantity in NIC is the effective model dimension $d$, defined as $\text{tr} [J^{-1} K]$ for optimization problem (3.6) \textit{without} constraints, where $J = E [\nabla^2 \ell(\theta_0)]$ and $K = \sum_{i=1}^{G} \text{Var} [\nabla \ell_i(\theta_0)]$, and the unknown $\theta_0$
minimizes $E[\ell(\theta)]$. Hence, an estimate of $d$ can be obtained by plugging in the estimates of $J$ and $K$.

For our constrained optimization problem, once a parameter estimate hits the boundary, it is no longer a free parameter and the gradient in this dimension is not necessarily zero. Hence, it is more reasonable to remove all of the parameters that are estimated to be on the boundary before computing an estimate of $d$.

Let $\hat{\theta}$ be the estimate of $\theta$, with the $k$th element $\hat{\theta}_k$, and let $\hat{\pi}_0 = 1 - \sum_{k=1}^{K} \hat{\theta}_k$. In most cases, $\sum_{k=1}^{K} \hat{\theta}_k < 1$, i.e., $\hat{\pi}_0$ is not on the lower boundary. Our parameterization in (3.3) and (3.5) is convenient in this case. Let $I = \{k : \hat{\theta}_k \neq 0, k = 1, 2, \ldots, K\}$ be the index set for all free parameters. Then, the estimates of $J$ and $K$ are given by $\hat{J} = \nabla^2_{I} \ell(\hat{\theta})$ and $\hat{K} = \sum_{i=1}^{G} \nabla_{I, i} \ell(\hat{\theta})[\nabla_{I, i} \ell(\hat{\theta})]'$, respectively. The subscript notation $I$ denotes first extracting only those rows (and also columns, if applied to a matrix) corresponding to indices in $I$ and then re-ordering the extracted rows (and columns) such that its index is increasing. Thus, $\nabla^2_{I} \ell(\hat{\theta})$ is an $|I| \times |I|$ Hessian matrix for the free parameters, and $\nabla_{I} \ell(\hat{\theta})$ is an $|I|$-vector gradient for free parameters. In this case, the estimate of $d$ is simply $\hat{d} = \text{tr}\left[\hat{J}^{-1}\hat{K}\right]$.

In the rare case where $\sum_{k=1}^{K} \hat{\theta}_k = 1$, i.e., $\hat{\pi}_0 = 0$ hits the lower boundary, (3.3) is overparametrized, since one of the $\hat{\theta}_k$ can be completely determined given the rest. Without loss of generality, we choose to remove the first $\theta_k$ such that $\hat{\theta}_k \neq 0$, i.e., letting $\theta_\mathcal{I} = [1, 0]' + \mathcal{J}' \theta^*$ where $\mathcal{J} = [-1, I_{|\mathcal{I}|}, -1]$ is the Jacobian of the transformation, and $\theta^*$ is considered as the new parameter of dimension $|\mathcal{I}| - 1$. We then re-compute the gradient and Hessian with respect to $\theta^*$. It turns out that, whenever $\sum_{k=1}^{K} \hat{\theta}_k = 1$, $d$ can be estimated by $\hat{d} = \text{tr}\left[\left\{\mathcal{J} \mathcal{J}'\right\}^{-1} \mathcal{J} \mathcal{K} \mathcal{J}'\right]$.

For a fixed tuning parameter $\lambda$, the final model selection criterion is defined as $\text{NIC} = 2\ell(\hat{\theta}) + 2\hat{d}$. We search over a grid of $\lambda$ on the log scale, and the one that minimizes $\text{NIC}$ is chosen to give the final estimate of $\theta$.

### 3.4 A semiparametric approach

Parametric methods often have better efficiency when the model is a good approximation to the unknown data generating mechanism, but can produce seriously biased estimators when the
approximation is poor. On the other hand, nonparametric alternatives sacrifice some efficiency to keep the bias at a lower level no matter what the underlying truth is. Semiparametric approaches are a natural and promising compromise between parametric and nonparametric approaches. There are many different ways of formulating semiparametric models. Actually, Ruppert et al. [2007] considered their method to be semiparametric because the distribution of noncentrality parameters is modeled nonparametrically and the conditional distribution of $t$-statistics given the noncentrality parameters is modeled parametrically. In this study, our terminology concentrates on the distribution of noncentrality parameters. Thus, the RNH method will be considered nonparametric instead of semiparametric.

Olkin and Spiegelman [1987] proposed a simple but effective semiparametric density estimation method, which is a two-component mixture of a parametric density estimate and a nonparametric density estimate, with the mixing proportion estimated by the maximum likelihood method. Here, we follow their approach to combine a parametric fit with a nonparametric fit for the latent noncentrality parameters.

In the semiparametric model, we assume that each $t$-statistic $t_i$ is marginally distributed according to either a parametric estimate or a nonparametric estimate, with density function

$$
\hat{f}_{\nu_i}(t_i) = \rho \hat{f}_{\nu_i}(t_i) + (1 - \rho) \hat{\tilde{f}}_{\nu_i}(t_i),
$$

(3.10)

where $\hat{f}_{\nu_i}$ is an estimate of (3.1), $\hat{\tilde{f}}_{\nu_i}$ is an estimate of (3.3), and $\rho \in [0, 1]$ is an extra mixing parameter. Note that $\hat{f}_{\nu_i}$ and $\hat{\tilde{f}}_{\nu_i}$ are treated as known models with all parameters being substituted with reasonable estimates. One might consider this as a two-stage approach, where both parametric and nonparametric estimates are obtained in initial separate analyses, and the combination is performed conditionally on these estimates by maximizing $\prod_{i=1}^G \hat{f}_{\nu_i}(t_i)$ with respect to $\rho$ only. The optimization problem is one-dimensional and can be easily solved, e.g., by a golden-section search algorithm.

The final estimate of the NCP density is again a two-component mixture,

$$
\hat{\rho}\phi(\cdot; \hat{\mu}, \hat{\gamma}^2) + (1 - \hat{\rho}) \sum_{k=1}^K \hat{\beta}_k \phi(\cdot; \hat{\mu}_k, \hat{\gamma}^2),
$$

where $\hat{\rho}$ is estimated from the semiparametric model (3.10), $\hat{\mu}$ and $\hat{\gamma}^2$ are estimated from the
parametric model (3.1), and the $\hat{\beta}_k$ are estimated from the nonparametric model (3.3). When the parametric fit agrees well with the data, $\rho$ would be estimated to be close to 1, and the final density estimate gains the efficiency of the parametric fit. When the parametric model is too rigid to fit the data well, $\rho$ would be estimated to be close to 0, and the final density estimate would be a flexible nonparametric fit.

3.5 False discovery rate estimation

All of the above models have the form of a mixture of a null component and a non-null component. Given the estimates of the necessary parameters, false discovery rates and variants can be easily computed. For a particular gene $i$, the posterior probability of equivalent expression (PPEE), i.e., the local false discovery rate (LFDR), can be estimated by

$$\hat{\text{LFDR}}_i = \frac{\hat{\pi}_0 h(t_i; \nu_i, 0)}{\hat{f}_{\nu_i}(t_i)},$$

where $\hat{\pi}_0$ is the estimated proportion of true null hypotheses, and $\hat{f}_{\nu_i}$ is the estimated marginal density of $t_i$ from the appropriate model. For example, given an estimate $\hat{\theta}$ of $\theta$ from the nonparametric model, $\hat{\pi}_0 = \tilde{\pi}_0$ and $\hat{f}_{\nu_i}(\cdot) = \tilde{f}_{\nu_i}(\cdot; \hat{\theta})$. The LFDR is the quantity of interest if we wish to determine whether the null or alternative hypothesis is true for this gene.

On the other hand, for a list of genes, $L \subseteq \{1, 2, \ldots, G\}$, whose null hypotheses are rejected, the false discovery rate (FDR) for the whole list can be estimated by the average of the LFDRs for the genes on the list,

$$\hat{\text{FDR}}_L(1) = \frac{1}{|L|} \sum_{i \in L} \hat{\text{LFDR}}_i.$$

In many cases, list $L$ is constructed by including all of the genes with a $p$-value no larger than some pre-chosen cutoff $\alpha$, then a popular alternative estimate of FDR is

$$\hat{\text{FDR}}_L(2) = \frac{\hat{\pi}_0 G \alpha}{|L|},$$

which is very similar to the method of Storey and Tibshirani [2003]. When the distribution of NCPs is symmetric about zero, the two FDR estimators usually result in similar estimates. However, as Sun and Cai [2007] argued, when the distribution of NCPs is asymmetric about
zero, using a \( p \)-value cutoff is suboptimal. Constructing the list \( L \) based on LFDR is preferred in that case.

3.6 Simulation study

To evaluate our methods, compared with that of Ruppert et al. [2007], we performed two simulation studies, one under the ideal situation that the \( t \)-statistics are \( t \)-distributed and independent of each other, and the other using error distributions subsampled from a real microarray dataset such that the dependencies among the \( t \)-statistics are preserved and the distribution of the \( t \)-statistics is only approximate.

In the first simulation study, we set the proportion, \( \pi_0 \), of true null hypotheses to 0.39, 0.69, or 0.99, and the degrees of freedom for all of the \( t \)-statistics to either \( \nu_i = 4 \) or 10. We consider the following five distributions for NCPs:

1. \( N(0, 1.5^2) \), the case when the parametric model assuming a zero mean is correct;
2. \( N(1.5, 1.5^2) \), the case when the parametric model assuming a nonzero mean is correct;
3. \( N(0, 3^2) \), the case when the parametric model assuming a zero mean is correct, but \( |\delta_i| \) tends to be relatively large;
4. Weibull(1.6, 2.6) \(-2.33 \), the case where NCPs are distributed asymmetrically around zero, but still have a mean of zero;
5. Cauchy(0, 1), the case where the NCPs are very heavily tailed.

For each combination of \( \pi_0, \nu_i \), and NCP distribution, we simulated \( G = 10,000 \) \( t \)-statistics from a mixture of central \( t \)-distribution and noncentral \( t \)-distribution, with mixing proportion \( \pi_0 \).

For the second simulation study, the settings of \( \pi_0, \nu_i \), and the distribution of NCPs were the same as the first one. However, the \( t \)-statistics were not directly simulated from the mixture of \( t \)-distributions. Instead, we simulated data based upon the myeloma study by Tian et al. [2003] (Gene Expression Omnibus accession# GSE755). Specifically, the “series matrix” file
with $G = 12\,625$ genes (rows) was downloaded from http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE755. The 137 tumor samples (columns) with focal lesions detectable by magnetic resonance imaging technology were treated as a large homogeneous biological population of errors. We separately computed a population standard deviation $\sigma_i$ for each gene $i = 1, 2, \ldots, 12\,625$. Then, for each simulation, we sampled a total of $\nu_i + 2$ columns without replacement from this large population. For the $i$th row of the first $n = \nu_i/2 + 1$ sampled columns, we added $\delta_i \sigma_i \sqrt{2/n}$ to each value, where $\delta_i$ is the target NCP, which has probability $\pi_0$ of being zero and probability $1 - \pi_0$ of being sampled from one of the five NCP distributions described above. This created a difference in means between the first $n$ columns and the other columns with the specified NCP distribution. We then performed a two-sample $t$-test for each row, assuming that the first $n$ columns came from treatment 1 and the rest from treatment 2. In this way, the joint error distribution was simulated to be very similar to that in the real dataset, i.e., not necessarily marginally normal and not necessarily independent across genes.

For both simulation studies, a total of 200 datasets were simulated. For each dataset, genes with $p \leq 0.05$ were considered as differentially expressed (DE). We then computed the proportion of falsely rejected hypotheses in this DE list, and compared it with its FDR estimates. Due to its close relationship with the method of Storey and Tibshirani [2003], probably the most commonly used one in practice, $\hat{\text{FDR}}^{(2)}_L$ was used as the FDR estimator in our simulations. Note that, when the underlying NCP distribution is not symmetric about zero, this choice of DE list may not be optimal. However, this choice can always produce a single list that is independent of $\pi_0$ or the density of $t$-statistics. Therefore, it is convenient to compare the performance based a single list for each simulated dataset. For each simulation setting, we considered the following four questions when comparing the different methods:

1. How well is $\pi_0$ estimated?

2. How well is FDR estimated? This was measured by the difference between the estimated FDRs and the true proportion of false discoveries (FDP) in the DE list. For a good FDR estimator, the expectation of this difference should be close to zero.
3. How well is the distribution of NCPs estimated? This was measured by the Hellinger divergence between the estimated NCP distribution and the true NCP distribution, where the Hellinger divergence between densities $y_1$ and $y_2$ is defined as

$$
\left[ \frac{1}{2} \int \left( \sqrt{y_1(x)} - \sqrt{y_2(x)} \right)^2 \, dx \right]^{1/2},
$$

which always lies between 0 and 1, with smaller values indicating better agreement between the two densities.

4. How well are genes ranked? This was measured by the partial area under the receiver characteristic curves (pAUROC) with false positive rate $\leq 0.1$ [Baker and Pinsky, 2001], with larger values indicating better top-ranked gene lists. For the methods we proposed, genes are ranked according to their LFDRs; for the RNH method and the method of Storey and Tibshirani [2003], genes are ranked based on their $p$-values.

The methods we compared in these simulations were the parametric model assuming a zero mean (par0), the parametric model assuming a nonzero mean (par), the nonparametric model (npar), the semiparametric model (spar), the unweighted single-iteration version of the RNH method (RNH1), the weighted double-iteration version of the RNH method (RNH2) that uses the estimates from the first iteration to re-weight binned counts in the second iteration, and the $q$-value smoother method (qvalue) of Storey and Tibshirani [2003]. Because the $q$-value smoother method only estimates $\pi_0$ and FDR, it was removed when comparing the estimates of NCP distributions. For the two methods from Ruppert et al. [2007], their default settings were used, and the estimated absolute NCP densities were reflected about zero and renormalized to determine the estimate of the NCP density.

Due to the large amount of simulation results, we only include and discuss in the main text the following representative situations.

Figures 3.1 and 3.2 show the results when $\pi_0 = 0.39, \nu_i = 4$, and $\delta_i \sim N(1.5, 1.5^2)$ under ideal and data-based error distributions, respectively. This is a situation where we can clearly see the advantage of estimating the distribution of NCPs over absolute NCPs. In terms of estimating $\pi_0$ and FDR, our par, npar and spar methods only showed some negligible biases
Figure 3.1  Simulation results when $\pi_0 = 0.39$, $\nu_i = 4$, and $\delta_i \sim N(1.5, 1.5^2)$ under ideal error distributions.
Figure 3.2  Simulation results when $\pi_0 = 0.39$, $\nu_i = 4$, and $\delta_i \sim N(1.5, 1.5^2)$ under data-based error distributions.
and the variabilities were also smaller than the variabilities of the two RNH methods. The
$q$-value smoother, on the other hand, had a very large positive bias, because this approach can
only estimate an upper bound for $\pi_0$. In terms of estimating the distribution of NCPs, the
reflection of the estimated densities from RNH1 and RNH2 were clearly beaten by methods
that do not assume symmetry around zero. Even if we consider estimating the density of
absolute NCPs, folding our estimates at zero still outperformed the estimates from RNH1 and
RNH2 (not shown). In terms of ranking of genes, methods not assuming symmetry of the NCP
distribution about zero outperformed those based on $p$-values or assuming the symmetry, which
confirmed the argument of Sun and Cai [2007]. Compared with the ideal simulation setting,
we see that all of the methods were affected by inflated variability under the data-based error
distributions. However, our par, npar, and spar methods still outperformed the others in this
situation.

Figure 3.3 shows the results when $\pi_0 = 0.39$, $\nu_i = 4$, and $\delta_i \sim \text{Cauchy}(0, 1)$, and Figure 3.4
shows similar results except that $\delta_i \sim \text{N}(0, 3^2)$, both under ideal situations. These two situations
are where RNH1 and RNH2 often failed to produce reasonable results under their default
settings. We can see that in terms of estimating $\pi_0$ and FDR, our npar and spar methods
usually had only small positive biases, whereas the RNH1 and RNH2 estimates were far off
the target. In terms of recovering the distribution of NCPs, npar and spar also had smaller
Hellinger divergences. Re-weighting the binned counts in RNH2 seemed helpful in terms of
recovering the distribution of NCPs, but was still worse than our methods. The main reason
for the less satisfactory performance of the RNH methods is that the default settings are not
robust to the heavy tails of the NCP distribution, and the penalized least squares fit is largely
influenced by the few bins with very large counts. Our npar and spar methods remove the
requirement of binning $p$-values and were less sensitive to such problems. Parametric models
assuming the normality of NCPs were best only when normality held. Moreover, because the
true underlying NCP distributions are symmetric about zero, there are no differences in terms
of ranking of genes. Overall, npar and spar were the winners in these two simulation settings.

Figures 3.5 and 3.6 show the results when $\pi_0 = 0.39$ and $\delta_i \sim \text{Weibull}(1.6, 2.6) - 2.33$
Figure 3.3 Simulation results when $\pi_0 = 0.39$, $\nu_i = 4$, and $\delta_i \sim \text{Cauchy}(0, 1)$ under ideal error distributions.
Figure 3.4  Simulation results when $\pi_0 = 0.39$, $\nu = 4$, and $\delta_i \sim N(0, 3^2)$ under ideal error distributions.
$\pi_0 = 0.39$

$\nu = 4$, Weib$(1.6, 2.6) - 2.33$

Figure 3.5 Simulation results when $\pi_0 = 0.39$, $\nu_i = 4$, and $\delta_i \sim \text{Weibull}(1.6, 2.6) - 2.33$ under ideal error distributions.
Figure 3.6 Simulation results when \( \pi_0 = 0.39 \), \( \nu_i = 10 \), and \( \delta_i \sim \text{Weibull}(1.6, 2.6) - 2.33 \) under ideal error distributions.
Figure 3.7  Simulation results when $\pi_0 = 0.99$, $\nu_i = 4$, and $\delta_i \sim N(0, 1.5^2)$ under ideal error distributions.
Figure 3.8  Simulation results when $\pi_0 = 0.99$, $\nu_i = 4$, and $\delta_i \sim N(0, 1.5^2)$ under data-based error distributions.
under the ideal simulation setting, with \( \nu_i = 4 \) and \( \nu_i = 10 \), respectively. The shifted Weibull distribution is asymmetric about zero but still has a mean of zero. These are the more complex situations where good methods for estimating \( \pi_0 \) and FDR are not necessarily good methods for estimating the distribution of NCPs or ranking genes. The two parametric methods produced the best estimates of \( \pi_0 \) and FDR, even when parametric assumptions were not satisfied. However, in terms of estimating the distribution of NCPs, nonparametric and semiparametric methods were the best performers with the smallest Hellinger divergences. Their ranking of genes was also best among the methods. Their estimates of \( \pi_0 \) and FDR, however, were more negatively biased. The conclusions were similar when we increased the degrees of freedom from 4 to 10 (Figure 3.6). Hence, the parametric models seem robust to mild asymmetry for the purpose of controlling FDR, but for examining the distribution of NCPs and ranking genes, more flexible nonparametric and semiparametric methods should be employed.

Figures 3.7 and 3.8 show the results when \( \pi_0 = 0.99, \nu_i = 4, \) and \( \delta_i \sim N(0, 1.5^2) \), under ideal and data-based error distributions, respectively. Because \( \pi_0 \) was very close to 1, few genes had nonzero NCPs. Hence, the datasets contained little information regarding the distribution of nonzero NCPs. In such difficult situations, neither our methods nor those based on Ruppert et al. [2007] performed reasonably. Even though par0 is the correct model here, it did not produce very useful results. Moreover, when the error distributions were dependent and not necessarily normal, the results were even worse. However, in practice, such problematic situations can often be detected before analysis by checking a histogram of the \( p \)-values. If the \( p \)-value histogram looks nearly flat, i.e., nearly Uniform(0, 1), then we may immediately realize that good estimates for the distribution of NCPs may not be achievable. In such cases, we can safely assume \( \pi_0 = 1 \) and apply the original FDR control method by Benjamini and Hochberg [1995].

### 3.7 Summary

In this study, we developed parametric, nonparametric, and semiparametric methodologies for estimating the distribution of noncentrality parameters from a large number of parametric
test statistics. The results showed that the estimation of noncentrality parameter distributions can be improved by applying nonparametric or semiparametric methods, compared with that of Ruppert et al. [2007], under a variety of situations. When parametric assumptions approximately hold, our parametric methods also provide good estimates of FDR and the proportion of true null hypotheses. Situations where such methods do not perform well were also identified and can be easily checked before analysis. The dependency across genes and/or non-normality generally inflates the estimation variability. It will be interesting to actively utilize the dependence information across genes in future studies.

Bibliography


false discovery rate. IEEE/ACM Transactions on Computational Biology and Bioinformatics 1, 98–108.


CHAPTER 4.  A HIERARCHICAL SEMIPARAMETRIC MODEL FOR INCORPORATING INTER-GENE INFORMATION FOR ANALYSIS OF GENOMIC DATA

A paper to be submitted to *Biometrics*

Long Qu, Dan Nettleton, Jack C.M. Dekkers

Abstract

For analysis of genomic data, e.g., microarray data from gene expression profiling experiments, the two-component mixture model has been widely used in practice to detect differentially expressed genes. However, it naively imposes strong exchangeability assumptions across genes and does not make active use of *a priori* information about inter-gene relationships that is currently available, e.g., gene annotations through the Gene Ontology (GO) project. We propose a general strategy that first generates a set of covariates that summarizes the inter-gene information and then extends the two-component mixture model into a hierarchical semiparametric model utilizing the generated covariates through latent nonparametric regression. Simulations and analysis of real microarray data show that our method can outperform the naïve two-component mixture model.

4.1 Introduction to the naïve two-component mixture model

High-throughput genomic technologies, e.g., microarray-based technologies, are now popular for exploring biological information at the genomic scale. For example, in a two-treatment comparison microarray experiment, transcription levels of tens of thousands of genes can
be measured simultaneously, and genes can be ranked according to strength of evidence for
eexpression level difference between the two treatment groups. A subset of top-ranked genes are
often candidates for further costly in-depth study. Hence, a good ranking of genes is essential
to gain biological insights and to reduce the chance of negative findings in future biological
research.

A popular approach for such data is to fit a two-component mixture model, where each
gene is assumed to be either differentially expressed (DE) or equivalently expressed (EE),
according to independent hypothetical Bernoulli trials. For simplicity, one often first performs
a hypothesis test of differential expression for each gene separately and then models the whole
set of resulting test statistics. Let $T_i$ be the test statistic for gene $i$, where $i = 1, 2, \ldots, G$
and $G$ is a large integer. $T_i$ might be chosen as a Student’s $t$-statistic or a $p$-value from any valid
test. The naïve two-component mixture model states that $T_i$ are independently and identically
distributed (i.i.d.) as

$$
\begin{cases}
F_0, & \text{with probability } \pi_0; \\
F_1, & \text{with probability } 1 - \pi_0;
\end{cases}
$$

where $\pi_0 \in [0, 1]$ is the mixing proportion, $F_0$ is the distribution of $T_i$ given that gene $i$ is EE
and $F_1$ is the distribution of $T_i$ when gene $i$ is DE. When $T_i$ is the $p$-value for gene $i$, then $F_0$
is often known (either continuous or discrete uniform distribution) and $F_1$ is a distribution
stochastically smaller than $F_0$. For example, Allison et al. [2002] and Pounds and Morris
[2003] both used a Beta distribution to model $F_1$, whereas Langaas et al. [2005] and Ruppert
et al. [2007] estimated $F_1$ nonparametrically. When $T_i$ is Student’s $t$-statistic, $F_0$ would be a
central $t$-distribution with appropriate degrees of freedom, and $F_1$ would be a discrete mixture
of a large number of noncentral $t$-distributions. Note that the null distribution $F_0$ is usually
completely specified, whereas the alternative distribution $F_1$ almost always has free parameters
and possibly an infinite number of free parameters if modeled nonparametrically. Efron [2004],
Jin and Cai [2007] and Wei and Pan [2008] also allowed $F_0$ to be estimated from data, but we
will not pursue this approach in our current study.

For the rest of our paper, we will choose Student’s $t$-statistic as our test statistic. However,
our method of incorporating inter-gene information is not limited to this particular choice, and extensions can be easily made to other test statistics, e.g., the $F$-statistic from a general linear test. For $p$-values computed from $t$-statistics, Ruppert et al. [2007] presented a method for approximating the distribution of noncentrality parameters by a continuous distribution. After transforming from $p$-values to $t$-statistics, their model is equivalent to assuming that each $T_i$ has density

$$f(t) = \pi_0 g(t; \nu, 0) + (1 - \pi_0) \int_{-\infty}^{\infty} g(t; \nu, \delta) h(\delta; \gamma) d\delta,$$

where $g(t; \nu, \delta)$ is the noncentral $t$-distribution with $\nu$ degrees of freedom and noncentrality parameter $\delta$, and $h(\delta; \gamma)$ is any valid density defined over $\mathbb{R}$ with a vector of unknown parameters $\gamma$. Ruppert et al. [2007] used nonparametric splines to model a folded version of $h$ and provided an estimation method using quadratic programming and numerical integration. Through their analyses of real microarray data, they found that noncentrality parameter distributions often concentrate their probability near zero and assign diminishing probability to regions farther from zero. This supports the idea that most DE genes have only small effects whereas a small number of DE genes show large effects in typical microarray experiments.

Based on these observations, we choose to simplify the model by replacing $h$ with a similarly shaped zero mean normal density with variance $\gamma^2 > 0$. The mean zero assumption has several advantages. First, it is consistent with the biological observation that the overall gene expression level across all genes rarely changes, i.e., the amount of up-regulation usually cancels with down-regulation in most biological systems. Second, by assuming zero mean, the single scalar parameter $\gamma$ can be used as an overall measure of how large the effects are. For a fixed sample size, large $\gamma$ indicates large effects for DE genes. Third, under the zero mean assumption, the integral over noncentrality parameters $\delta$ in model (4.1) can be analytically evaluated as a scaled central $t$-distribution, and model (4.1) simplifies to

$$f(t) = \pi_0 g(t; \nu, 0) + (1 - \pi_0) \frac{1}{s} g \left( \frac{t}{s}; \nu, 0 \right),$$

where $s = \sqrt{1 + \gamma^2}$. Since $s > 1$, this alternative distribution always has heavier tails than the central $t$-distribution. With analytical integration available, the very high computational burden of the numerical integration process in Ruppert et al. [2007] is avoided, and $f(t)$ can
be computed almost instantaneously using common statistical software like R [R Development Core Team, 2009].

Although two-component mixture models similar to (4.1) have been successfully used in many applications, they all suffer from the na"ive ness of an unrealistic independence assumption. Such a model states that, for a gene of interest, if a Bernoulli trial with success probability 1 − π₀ results in 1, the gene is DE regardless of the results for any other genes. However, it is well known that genes interact with each other to carry out biological functions. If we know that all other genes in a particular biochemical pathway are DE, then, without observing the data, we will believe the next gene in this pathway is very likely also DE. Hence, the success probability in the hypothetical Bernoulli trial for this gene should be increased towards 1. This dependency among genes creates a major hurdle for the analysis of high-dimensional data because the sample sizes used in microarray experiments are often very low, whereas the dimensionality is much larger than the sample sizes. Hence, the data itself does not contain enough information about the dependency structure across genes.

In this paper, we address the problem by first generating a set of pseudo-covariates from a priori information about the relationships across genes and then extending the naïve model (4.2) to a regression model over the generated pseudo-covariates.

4.2 Summarizing a priori inter-gene information

A major difficulty in using such information to assist statistical analysis is that the information is almost always represented in very complicated forms, not suitable for direct use by most existing statistical methods. Consider the widely used GO project for assisting analysis of microarray data. GO associates a set of controlled vocabulary of terms with each gene. The vocabulary itself is organized as a directed acyclic graph (DAG). Common practice for analyzing microarray data is to first produce a list of DE genes and then check whether any of the GO terms are over-represented in the DE list [e.g., Dennis et al., 2003]. Such analysis ignores the DAG structure of GO and rests on unrealistic independence assumptions. Although there have been attempts to mitigate such shortcomings [Liang and Nettleton, 2010] under some situations, a general framework for the problem is lacking.

In this study, we provide a general strategy to incorporate complicated a priori inter-genic information into statistical modeling. The overall flow chart is schematically illustrated in Figure 4.1. First, we transform the arbitrarily complicated prior information into a dissimilarity matrix across genes. This is the only interface step that depends on the source and form of the prior biological information. Once we have a dissimilarity matrix, multidimensional scaling (MDS) will be used to further transform the matrix into a few coordinates that can serve as “covariates” in a traditional sense. Our assumptions is that genes regulated similarly will have similar MDS coordinates. We do not attempt to have a clear biological interpretation of these coordinates, so long as they capture the dissimilarities as best as they can. With such generated pseudo-covariates available, the last step is to use them exactly as in traditional statistical modeling techniques, e.g., using some form of regression. Hence, the gap between the complicated representation of prior information and the applicability of traditional modeling tools can be filled. In this section and the next, we will discuss this framework in more detail.

4.2.1 Dissimilarity matrices across genes

Dissimilarities are widely used in multivariate statistics and are conceptually well accepted by biologists. For example, biologists routinely use dissimilarity-based clustering analysis for finding similarly expressed genes from microarray data [Eisen et al., 1998] or for reproducing
Input prior annotations for genes

Calculate semantic dissimilarity matrix of genes

Use multidimensional scaling to generate pseudocovariate(s)

Fit semiparametric hierarchical model to genewise $t$-statistics using constructed pseudocovariate(s)

Output local false discovery rates

Input microarray experimental data

Calculate genewise $t$-statistics for interesting hypotheses

Figure 4.1 Flow chart of analysis
phylogenetic history from gene sequence data [Sneath and Sokal, 1962]. Dissimilarity-based multidimensional scaling has been widely used in ecology for decades [Kenkel and Orlóci, 1986]. Hence, inter-gene relationships summarized using dissimilarities will be relatively easy for biologists to interpret.

In the case of GO, Fröhlich et al. [2006, 2007], Speer et al. [2005], Mistry and Pavlidis [2008], Schlicker et al. [2006], Alexa et al. [2006], del Pozo et al. [2008], and Lerman and Shakhnovich [2007] all provided various definitions of semantic distances between pairs of ontology terms, either from information theoretic considerations or using the diffusion kernel technique. Further, dissimilarities between pairs of genes can then be calculated by summarizing the dissimilarities of the GO terms associated with the genes. The R package GOsim provides several popular methods [Fröhlich et al., 2007]. Hence, the so-computed dissimilarities have taken the special DAG structure of GO into account. A simpler method, implemented in the very popular online analysis tool DAVID [Dennis et al., 2003, http://david.abcc.ncifcrf.gov/], is to compute the Cohen’s χ-coefficient [Cohen, 1960] between pairs of genes to assess similarity on the basis of shared annotation terms, ignoring the DAG structure of GO. In short, one can choose any of such methods that are available to get a $G \times G$ dissimilarity matrix across the $G$ genes.

There are many other sensible ways of obtaining a dissimilarity matrix besides those based on GO. For example, one may take similar datasets deposited at many microarray databases, e.g., the Gene Expression Omnibus (GEO) maintained by National Center for Biotechnology Information [Barrett et al., 2005], and use these existing datasets to compute data-based dissimilarity measures, e.g., $1 - \text{correlation matrices}$. Further, for species with (nearly) completed genome sequence information, a dissimilarity matrix across genes can be obtained from sequence alignment scores on the flanking regulatory regions of the genes of interest.

As many new methods are still being invented, these existing methods have already demonstrated their usefulness in many applications. Even if all other methods fail, one can still compute a (possibly shrunken) sample correlation matrix across genes using the data to be analyzed [Schäfer and Strimmer, 2005]. We do not attempt to review all possible ways to get
gene dissimilarity matrices here. Subject matter considerations are often needed to choose the most appropriate source of prior information and/or dissimilarity measure for a particular dataset. Data-based measures similar to Figure 4.6 might be plotted to assess the usefulness of the dissimilarities. Our main point is that one can very conveniently summarize the a priori inter-gene relationship information into a dissimilarity matrix.

4.2.2 Feature generation through multidimensional scaling

Once we have a dissimilarity matrix, MDS can be applied to generate a set of features for each gene. MDS is a set of well studied exploratory dimension reduction techniques, originally designed for psychometrical experiments, for finding lower dimensional representations of objects, such that the higher dimensional distances between the objects are best preserved in lower dimensions. The input of the methods is a dissimilarity matrix, and the output is the coordinates for each object in a pre-specified small number of dimensions. The advantage of MDS is that it only operates on dissimilarity matrices, not the original complicated annotations of genes. This opens the door for us to apply MDS to the distance matrices calculated using other complex knowledge, e.g., the semantic distances obtained from GO. In this study, we only consider the metric MDS, where the dissimilarities themselves are the target to preserve in lower dimensions, in contrast to non-metric MDS where only the ordering of the dissimilarities are to be approximately preserved.

Given a $G \times G$ dissimilarity matrix, $\Delta$, with $(i,j)$th element $\delta_{ij}$, MDS tries to find a $G \times J$ real matrix $X \equiv [x_1, x_2, \ldots, x_G]'$ that minimizes a chosen loss function $L(X, \Delta)$ for a given dimension $J$. Many choices of loss function are popular. For example, the so-called classical scaling, which is closely related to principal component analysis, chooses the loss as

$$\left\| -XX' - \frac{1}{2}(I - P_1)\Delta^{(2)}(I - P_1) \right\|^2$$

where $\Delta^{(2)}$ is the element-wise squared $\Delta$ matrix, and $P_1$ is the orthogonal projection matrix for a vector of ones. This choice bears some optimality property and is computationally less intensive, but the squared dissimilarities emphasize large differences. On the other hand, loss
functions proportional to
\[ \sum \sum_{i<j} \delta_{ij}^{-q} (\delta_{ij} - \|x_i - x_j\|)^2 \]
for some predetermined \( q \) will down weight larger dissimilarities when \( q > 0 \). In particular, when \( q = 0 \), the loss is often called raw stress; when \( q = 1 \) the procedure is equivalent to the widely used Sammon’s mapping [Sammon, 1969]; and when \( q = 2 \), it becomes the elastic scaling of McGee [1966]. For biological data, usually small dissimilarities represent some form of cooperation between genes, whereas large dissimilarities are not as meaningful. Hence, one should prefer the loss functions that best preserve small dissimilarities in lower dimensions. See Borg and Groenen [2005] and Cox and Cox [2001] for a fuller discussion of other choices.

For explorative analysis, the dimensionality \( J \) is usually chosen as 2 or 3 for ease of visualization. More sophisticated methods are available to choose a proper \( J \), especially for classical scaling [section 2.2 of Cox and Cox, 2001]. However, such methods are usually designed for other purposes, not necessarily providing good choices for our use of the pseudo-covariates. Another possibility is to treat this as a general tuning parameter and use cross-validation or some other information criteria to achieve a good choice (see section 4.4.1). In practice, it is advisable to consider several choices of \( J \) and compare and contrast results.

### 4.3 Hierarchical semiparametric model

Using MDS, we achieve at a set of pseudo-covariates to be used in conjunction with model (4.2). Because our underlying rationale is that

related genes \( \rightarrow \) small distances \( \rightarrow \) similar pseudo-covariates \( \rightarrow \) similar probability of DE,

it is natural to extend model (4.2) to allow \( \pi_0 \) to be gene specific and to depend on the pseudo-covariates. Because we do not have any knowledge about how the pseudo-covariates affect DE probability, we should consider flexible nonparametric models that allow the pseudo-covariates to enter the model through general smooth functions. The following latent additive logistic regression model is our preferred choice.
Let $x_i$ denote the $J$-vector of pseudo-covariates for gene $i$ and $x_{ij}$ be the $j$th element of $x_i$. We assume the gene-specific Student’s $t$-statistics, $t_i$, for $i = 1, 2, \ldots, G$, are conditionally independent, given $x_1, x_2, \ldots, x_G$, and that the conditional density of $t_i$ is

$$
f(t_i) = \pi_{0i} g_{0i} + (1 - \pi_{0i}) g_{1i}, \quad \text{with}$$

$$
\logit(1 - \pi_{0i}) = \mu + \sum_{j=1}^J y_j(x_{ij}),
$$

(4.3)

where

$$
g_{0i} = g(t_i; \nu, 0),$$

$$
g_{1i} = g(t_i/s; \nu, 0)/s,$$

$y_j(\cdot)$ is a general smooth function and $\mu$ is an intercept parameter. Furthermore, we model $y_j$ with quadratic B-splines as

$$
y_j(\cdot) = m_j(\cdot)' \beta_j,
$$

(4.4)

where $m_j(\cdot)$ is a vector valued quadratic B-spline basis function, with a moderately large number $K$ internal knots placed at equally spaced sample quantiles of the $j$th pseudo-covariate across all $G$ genes, and $\beta_j$ is a $(K + 3)$-vector of unknown parameters to be estimated from data. Although $\beta_j$ is not identifiable due to redundancy with $\mu$, we only care about the identifiable quantities $\logit(1 - \pi_{0i})$ for $i = 1, 2, \ldots, G$. Without loss of generality, we restrict $\sum_{i=1}^G y_j(x_{ij}) = 0$, for all $j = 1, 2, \ldots, J$. Collectively, we express the additive logistic model using matrix notation as

$$
\logit (1 - \pi_0) = M \beta \iff \pi_0 = 1 - \logit^{-1}(M \beta),
$$

(4.5)

where $\pi_0$ is a $G$-vector $\{\pi_{0i}\}_{i=1}^G$, $M$ is a $G \times [1 + J(K + 3)]$ matrix with the $i$th row being

$$
M_{(i)} = [1, m_1(x_{i1})', m_2(x_{i2})', \ldots, m_J(x_{iJ})']',
$$

and $\beta = [\mu, \beta_1', \beta_2', \ldots, \beta_J']'$. 

Once we have reasonable estimate of the parameters $s$ and $\beta$ (see next section), inference for gene $i$ is based on plug-in estimate of the posterior probability of DE, $1 - \ell_i$, where $\ell_i$ is the local false discovery rate (lfdr),

$$
\ell_i(s, \beta) = \pi_{0i} g_{0i}/f(t_i).
$$

(4.6)
For a list of genes, the false discovery rate can be estimated by the average of \( \ell_i \) in the list.

Equivalently, our model can be considered as a hierarchical semiparametric model. Let the latent random variable \( Z_i \) denote the DE status, \( i.e., \) if \( Z_i = 1 \), gene \( i \) is DE; and if \( Z_i = 0 \), gene \( i \) is EE. Conditional on \( Z_i \), our parametric model is a scaled \( t \)-distribution, with the scale factor being 1 if \( Z_i = 0 \) and greater than 1 if \( Z_i = 1 \). Marginally, the \( Z_i \) are modeled by nonparametric additive logistic regression. If two genes \( i \) and \( j \) are closely related, their dissimilarity \( \delta_{ij} \) will be small, and their pseudo-covariates \( x_i \) and \( x_j \) will be close. Hence, their prior DE probabilities \( 1 - \pi_{0i} \) and \( 1 - \pi_{0j} \) will tend to be close, depending on \( \beta \). In this way, information about inter-gene relationships has been incorporated into the inference for each gene.

### 4.4 Parameter estimation

Because we choose \( K \) to be moderately large, our model is highly parametrized. Therefore, we use penalized likelihood estimation to get stable estimates of \( \beta \). The penalty for lack of smoothness that we subtract from our log likelihood function, \( L(s, \beta) = \sum_{i=1}^{G} \log [f(t_i)] \), is

\[
0.5\lambda \sum_{j=1}^{J} \beta_j' \Omega_j \beta_j,
\]

where \( \lambda \) is an extra smoothing parameter and \( \Omega_j \) is a \((K + 3) \times (K + 3) \) matrix with \( (u, w) \)th element

\[
\int_{-\infty}^{\infty} m'_{ju}(z) m'_{jw}(z) \, dz,
\]

where \( m'_{ju}(\cdot) \) and \( m'_{jw}(\cdot) \) are the first derivative of the \( u \)th and \( w \)th component function of \( m_j(\cdot) \), respectively. We can collectively express the penalty as \( 0.5\lambda \beta' \Omega \beta \), where \( \Omega = \bigoplus_{j=0}^{J} \Omega_j = \text{diag}\{\Omega_0, \Omega_1, \ldots, \Omega_J\} \) and \( \Omega_0 \) is a \( 1 \times 1 \) zero matrix.

The idea of choosing first derivative penalty in (4.7) is intuitive. If the tuning parameter \( \lambda = 0 \), the resulting fit would be an interpolator through the knots. If \( \lambda \to \infty \), the penalty will tolerate no nonzero first derivatives and our model should behave just as model (4.2), where no \textit{a priori} information is used and the prior DE probability for all genes equals to a constant \( \logit^{-1}(\mu) \). This differs from the common practice of smoothing where the limiting case is a
linear fit, instead of a constant fit. By analogy to the derivation of smoothing splines [Green, 1994], where knots are placed on every distinct data point, this choice of penalty suggests that the optimal degree of B-spline expansion should be 2 (quadratic), instead of the commonly used cubic spline. Because the number of genes, \( G \), is usually very large, there is no need to use all data points as knots. As long as \( K \ll G \) is moderately large, the fit is generally very similar to a smoothing spline fit [O’Sullivan, 1986; Hastie et al., 2009].

Denote the negative penalized log likelihood (NPLL) based on data point \( i \) as

\[
NPLL_i = -\log [f(t_i)] + \frac{1}{2G} \lambda \beta' \Omega \beta,
\]

where \( f(t_i) \) is given in (4.3). Then the objective function to be minimized is \( \text{NPLL} = \sum_{i=1}^{G} NPLL_i \).

For a given \( \lambda \), both generalized Expectation-Maximization (EM) algorithm (Appendix 4.A) and Newtonian algorithms (Appendix 4.B) can be used to minimize NPLL. The advantage of the generalized EM algorithm includes intuitiveness, ease of implementation, relative robustness to poor starting values, and its generation of a sequence of estimates that lead to a non-decreasing penalized log likelihood. However, the convergence rate is usually slower than gradient based optimization methods, especially when the amount the missing data is high and when the estimates are getting close to the solution. Hence, in practice, we adopt a hybrid approach by first running a small number of iterations of the generalized EM algorithm and then switching to a general Newtonian optimization routine, such as the popular quasi-Newton BFGS algorithm [Nocedal and Wright, 1999] or the Newton-Raphson algorithm.

### 4.4.1 Tuning parameter selection

The smoothing parameter \( \lambda \) is a tuning parameter in our nonparametric model. Also, the number of pseudo-covariates \( J \) can also be treated as a tuning parameter. Although one can further allow each pseudo-covariate to have a different smoothing parameter, we will not adopt this extension here. Commonly used methods for selecting tuning parameters include cross-validation, generalized cross-validation (GCV), and information criteria (e.g., Akaike’s information criterion or AIC).
To use \( B \)-fold cross-validation, we first randomly split the data into \( B \) approximately equally sized subsets and use only \( B - 1 \) of the subsets to fit the model but compute a negative prediction log likelihood as a loss function on the data points in the excluded subset. The procedure is repeated another \( B - 1 \) times, one for each of the subsets of data. The final cross-validation score is the average of negative prediction log likelihoods across all data points. Because microarray datasets are large, this procedure is time consuming, especially when \( B \) is large. From our experience, \( B = 5 \) or 10 works well.

Alternatively, information criteria can be used to choose tuning parameters. This requires an estimate of the dimension of the model, or the effective number of parameters (enp). Among many other definitions, the network information criterion (NIC) is an appropriate one for penalized likelihood [section 2.2 and 4.3 of Ripley, 1996; Moody, 1991, 1992]. NIC is essentially Takeuchi’s information criterion applied to regularized fitting [section 7.1–7.3 of Burnham and Anderson, 2002; Takeuchi, 1976].

Let \( g(r, \beta) \) and \( H(r, \beta) \) denote the gradient and the Hessian of NPLL, respectively (see Appendix 4.B for explicit formulas), where \( r = \log(s - 1) \). Further let

\[
d = \left\{ \frac{\partial}{\partial r} \text{NPLL}_i \bigg|_{r=\hat{r} \atop \beta=\hat{\beta}} \right\}_i^G, \quad D = \text{diag} \left\{ \frac{g_{0i} - g_{1i}}{f(t_i)} \pi_{0i}(1 - \pi_{0i}) \bigg|_{r=\hat{r} \atop \beta=\hat{\beta}} \right\}_i^G,
\]

where \( \hat{r} \) and \( \hat{\beta} \) are the maximum penalized likelihood estimates.

Then, in the ideal case, the enp is defined as \( \text{tr} \left[ J^{-1} K \right] \) where \( J = \mathbb{E}[H(r_0, \beta_0)] \) and \( K = \text{Var}[g(r_0, \beta_0)] \), and the unknown \( r_0 \) and \( \beta_0 \) minimize \( \mathbb{E}[\text{NPLL}] \). Here we take the estimate of enp as

\[
\hat{\text{enp}} = \text{tr} \left[ J^{-1} \hat{K} \right] \quad \text{where} \quad \hat{J} = H(\hat{r}, \hat{\beta}) \quad \text{and} \quad \hat{K} = Q'Q, \quad \text{where}
\]

\[
Q = \begin{bmatrix} d, DM & + \frac{\lambda}{G} 1 \beta' \Omega \end{bmatrix}.
\]

Since the column sums of \( Q \) are \( g(\hat{r}, \hat{\beta}) = 0 \), \( \hat{K} \) can be simplified as

\[
\hat{K} = \begin{bmatrix} d'd & d'DM \\ M'Dd & M'D^2M - \frac{\lambda^2}{G} \Omega \hat{\beta}' \Omega \end{bmatrix}.
\]

The final estimate of the NIC criterion is given by \( \text{NIC}_0 = 2\text{NPLL} + 2\hat{\text{enp}} \), where \( \text{NPLL} \) is the negative penalized log likelihood at the estimated parameters. In practice, we can remove the
penalty term to give \( \text{NIC} = 2\text{NIC}_0 - \lambda \hat{\beta} \Omega \hat{\beta} \). If \( \lambda \to \infty, \hat{\text{enp}} \to 2 \), i.e., only the intercept \( \mu \) and the scale parameter \( s \) are always effective.

In practice, we search a grid of \( \lambda \) on the logarithmic scale to use NIC. Starting from a nearly infinite \( \lambda \), say \( 10^8 \), we fit the model and estimate \( \text{enp} \). This start point is easiest in terms of optimization, as there are only \( \sim 2 \) parameters. Then we divide \( \lambda \) by a factor of, say 1.5, and repeat the procedure using the estimates from previous \( \lambda \) as starting values. We choose the \( \lambda \) that minimizes NIC (Figure 4.2).

From our experience, NIC agrees with cross-validation scores very well as long as \( \lambda \) is not too small. When \( \lambda \) is very small, there could be many stationary points as possible solutions to the minimization problem, and it is difficult to choose a good one. Hence, both the likelihood and \( \text{enp} \) may be unreliable. For example, if the number of knots is very large and the \( \lambda \) is very small, estimated \( \text{enp} \) values will not necessarily decrease as \( \lambda \) increases. This contradicts the fact that a larger \( \lambda \) value corresponds to a smoother fit. The situation is extreme and rarely happens in our experience. Most of the time, a clear minimum NIC shows up before \( \lambda \) is very close to zero. However, if the problem does arise, we suggest to reducing the number of knots and/or switching to \( B \)-fold cross-validation.

### 4.4.2 Interval estimation

Because our estimation procedure computes matrices \( J \) and \( K \), asymptotic confidence intervals are immediately available. Loosely speaking, we can think of the maximum penalized likelihood estimates \( (\hat{r}, \hat{\beta})' \) as being approximately multivariate normally distributed with mean \( (r_0, \beta_0)' \) and covariance matrix \( J^{-1}KJ^{-1} \). The sandwich-type estimator for the covariance matrix is naturally \( \hat{\Sigma} = \hat{J}^{-1}\hat{K}\hat{J}^{-1} \). Confidence intervals of functions of the parameters can be obtained through the Delta method. Specifically, we are interested in interval estimates for the function \( Y_0 = M\beta_0 \) and the local false discovery rates \( \ell_i \).

\( Y_0 \) is a linear function of \( \beta_0 \), hence the covariance matrix of \( M\hat{\beta} \) is approximately \( \hat{\Sigma}^{(Y)} = [0, M]\hat{\Sigma}[0, M]' \). A Wald-type point-wise confidence interval for the \( i \)th element of \( Y_0 \) is given
by

\[
\left[ M'_{ij} \hat{\beta} - T_{(1 - \alpha/2)} \sqrt{\hat{\Sigma}^{(Y)}_{ii}}, M'_{ij} \hat{\beta} + T_{(1 - \alpha/2)} \sqrt{\hat{\Sigma}^{(Y)}_{ii}} \right],
\]

where \( T_{(1 - \alpha/2)} \) is the \((1 - \alpha/2)\)th quantile for the \(t\)-distribution with \((G - \hat{\text{enp}})\) degrees of freedom. Since \( G \) is large, this is very close to the corresponding quantile for a standard normal distribution.

Besides using NIC to select a model, it is also possible to construct a test against the null model to decide if we should adopt the more complicated model. Let \( \beta_{0(-1)} \) and \( \hat{\beta}_{(-1)} \) denote the corresponding vectors excluding the first element, then the null hypothesis to be tested is \( H_0: \beta_{0(-1)} = 0 \), i.e., whether all other parameters except the scale factor and the intercept are zero. Let \( \hat{\Sigma}_{(-1,-2)} \) denote the submatrix of \( \hat{\Sigma} \) after removing the first two rows and the first two columns, then the Wald-type \( F \)-statistic is

\[
F = \frac{1}{J(K + 2)} \hat{\beta}_{(-1)}' \hat{\Sigma}^{-1}_{(-1,-2)} \hat{\beta}_{(-1)},
\]

which can be compared with the \((1 - \alpha)\)th quantile of the \(F\)-distribution with \( J(K + 2) \) and \((G - \hat{\text{enp}})\) degrees of freedom, and \( H_0 \) will be rejected if the \( F \)-statistic is larger than the quantile.

For the local false discovery rates, the natural bound \([0,1]\) should be taken into account when constructing confidence intervals. We again choose the logit transformation to remove the boundary constraints by considering,

\[
\logit(1 - \ell_i) = M'_{(i)} \beta_0 + \log[g(t_i/s_0; \nu, 0)/s_0] + \log[g(t_i; \nu, 0)],
\]

where \( s_0 = 1 + \exp(r_0) \). Since

\[
\frac{\partial}{\partial \beta} \logit(1 - \ell_i) = M_{(i)} \quad \text{and} \quad \frac{\partial}{\partial r} \logit(1 - \ell_i) = (t_i^2 - s_0^2)/(t_i^2 + \nu s_0^2)^{-1} \nu (s_0 - 1)/s_0,
\]

the confidence limits for \( \ell_i \) is given by

\[
1 - \logit^{-1} \left\{ \logit(1 - \hat{\ell}_i) \pm T_{(1 - \alpha/2)} \sqrt{\hat{\Sigma}^{(\logit[1-\ell])}_{ii}} \right\}
\]

where \( \hat{\ell}_i \) is the maximum penalized likelihood estimate of \( \ell_i \),

\[
\hat{\Sigma}^{(\logit[1-\ell])} = [d, M] \hat{\Sigma} [d, M]',
\]
and

\[ d_\ell = \left\{ \frac{\partial}{\partial r} \logit(1 - \ell_i) \right\}_{s=\hat{s}} \left. \bigg|_{s=\hat{s}} \right\}^G_{i=1}. \]

The so constructed confidence intervals will never exceed the \([0, 1]\) bounds.

### 4.5 Simulation study

To evaluate the effectiveness of our model, we set up a simulation study under simple settings. In the first simulation, we set the number of genes \( G = 20000 \), the scale factor \( s = 1.64 \) (i.e., the variance of the noncentrality parameter \( \gamma^2 = 1.69 \)), and the degrees of freedom \( \nu = 8 \). We used a single underlying smooth function \( \logit(1 - \pi_{0i}) = y(x_i) = \sin \frac{\pi x_i}{2000} - 0.5 \) to generate \( \pi_{0i} \) values, with \( x_i = i \) for all \( i = 1, 2, \ldots, G \). We considered two pseudo-covariates: the true covariate used to generate the \( \pi_{0i} \) values and an irrelevant covariate obtained by randomly shuffling the integers 1, 2, \ldots, \( G \). Selection of tuning parameter \( \lambda \) was based on NIC, over a fine grid on the logarithmic scale from \( 10^{-4} \) to \( 10^8 \). The number of internal knots was fixed at 100.
Figure 4.2 shows typical patterns of NIC vs. $\log_{10}\lambda$. When the true covariate was included in the model, the NIC had a clear minimum between $10^3$ and $10^4$. When only the irrelevant covariate was included in the model, the minimum NIC occurred at $\lambda = 10^8$, which suggests a null model where the $\pi_{0i}$ values are constant as a function of the irrelevant covariate. The minimum NIC for the correct model, i.e., the model with the true covariate only, was lower than the other two models. Hence, NIC worked properly in choosing the correct covariate in this simulation.

Figure 4.3 shows the fitted function along with point-wise asymptotic 95% confidence bands and the true function to be estimated. The fit based on the NIC-selected tuning parameter generally captured the trend in the true function. However, there was some over fitting, evidenced by irregularities in the fitted curve. The confidence band covered the true function for almost every $x$, but the band length was much higher at the boundaries. Also the “cutting top, filling bottom” effect suggests that the bias is larger in regions with more curvature. Nevertheless, the fitted function was still in good accordance with the truth.
Figure 4.4 shows a plot of estimated $\ell_i$ values vs. their corresponding $t$-statistics. As expected, larger values of $|t_i|$ are associated with smaller values of $\ell_i$; however, the relationship is clearly not monotone. Using prior information to account for relatedness among genes led to substantial adjustment to the order of the $1 - \ell_i$ values relative to the order of the $|t_i|$ values. This is evidenced by the scatter of points around the null model fit (that uses no information about inter-gene relationships) in Figure 4.4.

To demonstrate the advantage of our model in ranking DE genes, the same simulation was repeated 100 times. Also, we included two other simulation settings, one with constant $y(\cdot)$ and the other with a two dimensional $y(\cdot, \cdot)$ (Table 4.1). For each simulation, the genes were ranked by the estimated $\ell_i$ values. Table 4.1 presents the resulting average area under the receiver operating characteristic (AUROC) curves and the associated effective number of parameters, with standard error in the parentheses. The larger the AUROC, the better the ranking. We can see that, in all simulation settings, the null models that ignored the information in the covariates always had low AUROC, whereas the ideal models that used the true (but unknown in practice) $y$ had the highest AUROC. The models that estimated $y$ using
The correct covariate(s) were only slightly worse than ideal model. If irrelevant covariate(s) were added to the model, the AUROCs slightly deteriorated and more than necessary enp were used. If the model only included irrelevant covariate(s) or if the true \( y \) was constant, the AUROCs from fitted models were very similar to using the null model, with slight over fitting. When NIC was used to choose the covariates in the model, the resulting AUROCs fell between those obtained by using the true covariate(s) and using both the true covariate(s) and irrelevant covariate(s). Note that the AUROCs for the NIC approach were much closer to those obtained using the true \( y \) than to those obtained using the null model.

In summary, if the pseudo-covariates indeed contain information, our method improves the AUROC by a significant amount compared with a usual null model. On the other hand, if the pseudo-covariates are completely useless, then our method still performs similarly to the null model, representing good robustness to using poor pseudo-covariates. Therefore, there is a potential for substantially improved ranking of DE genes with little risk when using our method rather than the traditional null model.

<table>
<thead>
<tr>
<th>Simulation model</th>
<th>Fitting model</th>
<th>AUROC</th>
<th>enp</th>
</tr>
</thead>
<tbody>
<tr>
<td>( y(x) = 0 )</td>
<td>null</td>
<td>0.644 (0.004)</td>
<td>2.000 (-)</td>
</tr>
<tr>
<td></td>
<td>known ( y(x) )</td>
<td>0.644 (0.004)</td>
<td>2.000 (-)</td>
</tr>
<tr>
<td></td>
<td>irrelevant ( x )</td>
<td>0.642 (0.005)</td>
<td>4.057 (4.573)</td>
</tr>
<tr>
<td>( y(x) = -0.5 + \sin \frac{x \pi}{2000} )</td>
<td>null</td>
<td>0.644 (0.004)</td>
<td>2.000 (-)</td>
</tr>
<tr>
<td></td>
<td>known ( y(x) )</td>
<td>0.740 (0.004)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>true ( x )</td>
<td>0.731 (0.004)</td>
<td>30.422 (3.195)</td>
</tr>
<tr>
<td></td>
<td>irrelevant ( x )</td>
<td>0.643 (0.004)</td>
<td>3.817 (4.093)</td>
</tr>
<tr>
<td></td>
<td>true &amp; irrelevant ( x )</td>
<td>0.724 (0.005)</td>
<td>51.468 (4.994)</td>
</tr>
<tr>
<td></td>
<td>NIC chosen</td>
<td>0.730 (0.004)</td>
<td>31.160 (5.090)</td>
</tr>
<tr>
<td>( y(x_1, x_2) = -1 + \sin \frac{x_1 \pi}{2000} + \cos \frac{x_2 \pi}{2000} )</td>
<td>null</td>
<td>0.643 (0.004)</td>
<td>2.000 (-)</td>
</tr>
<tr>
<td></td>
<td>known ( y(x) )</td>
<td>0.783 (0.004)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>true ( x )'s</td>
<td>0.766 (0.006)</td>
<td>57.103 (4.133)</td>
</tr>
<tr>
<td></td>
<td>one true ( x ) only</td>
<td>0.719 (0.004)</td>
<td>28.103 (2.919)</td>
</tr>
<tr>
<td></td>
<td>one true and one irrelevant ( x )</td>
<td>0.713 (0.004)</td>
<td>46.421 (5.458)</td>
</tr>
<tr>
<td></td>
<td>NIC chosen</td>
<td>0.766 (0.006)</td>
<td>57.103 (4.133)</td>
</tr>
</tbody>
</table>
4.6 Analysis of pig fasting data

Our method can also show benefits when analyzing real microarray data. The following data were obtained from a study exploring the effects of fasting on the liver in pigs [Lkhagvadorj et al., 2009]. This study followed an extended randomized complete block design with 4 blocks and a $2 \times 2$ factorial treatment structure (fasting status: yes/no; and melanocortin-4 receptor genotype: D298/N298). The liver samples from a total of 24 pigs were collected and hybridized to the Affymetrix Porcine Genome Array with one array for each pig. After summarizing expression measures using Affymetrix’s MAS5 algorithm and median centering normalization, a gene-wise linear model was fit to each of the 24,123 probe sets on the log scale. The $t$-statistics were obtained for testing the main effect of fasting, with 17 degrees of freedom. The genotype effects and the interaction effects were minimal in this data and were not further analyzed.

The *a priori* inter-gene information was obtained from GO annotation. In brief, porcine sequences were compared with human sequences, and the GO annotations from the most similar human genes were used to annotate $G = 11,512$ porcine probe sets used in this study. For more detail on the annotation process, see Couture et al. [2009]. The gene similarity matrix for these 11,512 probe sets was calculated by the GOSim packages version 1.1.5.4 in R, using the Biological Process (BP) ontology and all default settings. A small fraction (0.3%) of the entries in the similarity matrix exceeded the upper bound 1, and were truncated at 1. Canonical scaling was then performed on the dissimilarity matrix (*i.e.*, $1 -$ similarity matrix), and the first 10 coordinates were obtained for our semiparametric modeling. To evaluate the robustness of our method with respect to misspecified prior information, we also generated a dissimilarity matrix using dissimilarities that were randomly shuffled across genes. Our semiparametric model was fit using 100 internal knots on each dimension.

Plots of the first ten eigenvalues of the dissimilarity matrix and the NIC values are shown in Figure 4.5. There seemed to be a small gap between the first two eigenvalues and the rest. The NICs kept decreasing as more pseudo-covariates were added. But the decrease slowed down after including the first two. Although one may wish to try including more pseudo-covariate until a minimum NIC is found, it would be time consuming in terms of computing the coordinates.
Figure 4.5  Eigenvalues and NIC for pig fasting data

Figure 4.6  Correlatedness of gene lists
through canonical scaling and/or fitting the model with a very large number of parameters. Hence, a choice of 2 pseudo-covariates was selected as a reasonable compromise between a complex model and the null model that ignores inter-gene information.

For real microarray data, we do not know exactly which genes are DE. Hence, it is not possible to calculate AUROC to demonstrate the superiority of our method. Instead, we computed pairwise sample correlations between genes using the residuals from the fit of the gene-wise linear model. Then, for a gene list consisting of $G_0$ genes with smallest estimated $\ell_i$ values, we calculated the average absolute pairwise correlations of residuals for pairs of genes on the list, denoted as $\bar{\Upsilon}(G_0)$. When $G_0 = 1$, we define $\bar{\Upsilon}(1) = 1$. When $G_0 = G = 11512$, $\bar{\Upsilon}(G_0)$ is the average of absolute pairwise correlations among all $G$ genes analyzed. For a particular $G_0$, higher $\bar{\Upsilon}(G_0)$ suggests the genes in the list tend to work together and share similar biological expression regulations. Thus, higher values of $\bar{\Upsilon}(G_0)$ suggest a more biologically meaningful gene list. To get a single measure for the overall ranking of genes, we further computed the average $\bar{\bar{\Upsilon}} = \frac{1}{G} \sum_{u=1}^{G} \bar{\Upsilon}(u)$. Again, higher values indicate good a ranking.

Note that this assessment of our method based on inter-gene residual correlations is not using any information twice. The information we used to fit the model includes GO annotation and marginal $t$-statistics. However, the assessment is based on the correlation among residuals that plays no part in either GO annotations or the $|t_i|$ values.

Figure 4.6 shows a plot of $\bar{\bar{\Upsilon}}$ against the number of pseudo-covariates. The null model, i.e., no pseudo-covariates, produced a rather low value. Increasing the number of pseudo-covariates to 2 gave a higher $\bar{\bar{\Upsilon}}$ and including more pseudo-covariates did not improve $\bar{\bar{\Upsilon}}$ further. Hence, using $J = 2$ was a good choice and our method outperformed the null model in producing more correlated (hence more biologically meaningful) DE gene lists on this real dataset. Using randomly shuffled dissimilarities produced $\bar{\bar{\Upsilon}}$ values similar to the one not using prior information. Hence, our method appeared to be robust to misspecified prior information.
4.7 Discussion

We have proposed to use multidimensional scaling to generate a set of pseudo-covariates that can capture inter-gene relationship information, and have extended the naïve two-component mixture model to a semiparametric hierarchical logistic regression model that can take advantage of the information contained in the pseudo-covariates. Penalized likelihood estimation methods have been devised and implemented in an R package hisemi.

Various extensions of our approach are possible. For example, the parametric assumption for the distribution of noncentrality parameters could be relaxed by using methods similar to Ruppert et al. [2007] to obtain a nonparametric estimate of the distribution of noncentrality parameters. In future work, we also plan to investigate a kernel-based local likelihood method in place of our additive spline model.

In terms of using a priori inter-gene relationship information, several other approaches exist in the literature. Wei and Li [2007] and Wei and Pan [2008] used an a priori graph in detecting differential expression. Huang and Pan [2006] and Pan [2006] incorporated prior information into clustering analysis. Pan [2009] and Li et al. [2010] proposed methods for using inter-gene relationship information in linkage analysis. Pan et al. [2009] and Li and Li [2008] considered the feature selection problem in the presence of inter-feature relationship information. Most of these methods assume the availability of a graph that summarizes the prior knowledge, whereas our method is slightly more general in that the prior knowledge may or may not be a graph. Rather, any representation that can be summarized into a dissimilarity matrix can be used in our approach.

Although the methods of Wei and Li [2007] and Wei and Pan [2008] can be used to address the same problem that we consider, the computational expense of their procedures makes a simulation-based comparison of our approach with theirs prohibitively time consuming. Furthermore, aside from computational expense, the question of how to most appropriately simulate data to allow a fair comparison of methods is nontrivial. Thus, we focus on an examination of conceptual similarities and differences between our method and these most closely related procedures.
Wei and Li [2007] modeled $Z_i$ values from a discrete Markov random field model, assuming logit$(1 - \pi_i)$ to be linear in the proportion of genes that are truly DE in its neighborhood, defined on a graph that contains inter-gene relationship information. Their initial algorithm could only report the most likely state of each $Z_i$, but Wei and Pan [2010] developed a fully Bayesian Markov chain Monte Carlo (MCMC) algorithm for this model to estimate other interesting quantities. This model directly builds overall dependency across $Z_i$ values, through specifying only conditional probabilities. A disadvantage is that the joint likelihood has no closed form representation, and approximations like the pseudolikelihood of Besag [1986] must be used. Our proposed method builds dependency through conditioning on pseudo-covariates in a regression-like model, whose joint conditional likelihood is easily available for maximization. Another difference is that Wei and Li [2007] also modeled the distribution of raw data, whereas we only modeled a summary statistic to reduce computing demand and to simplify interpretation. Using summary statistics also facilitates the use of the methods in other applications, e.g., array-based comparative genomic hybridization (CGH) data, whose raw data model may not follow that for expression array data. A potential advantage of modeling raw data is to allow shrinkage estimation of variances, which is the approach considered by Li et al. [2008], where a discrete Markov random field model is combined with an inverse scaled $\chi^2$ prior on the variances. However, our analysis strategy also allows such extensions.

The approach taken by Wei and Pan [2008] had another layer of hierarchy of latent variables, which are assumed to be realizations from a Gaussian Markov random field model. They then modeled logit$(1 - \pi_i)$ to be linear in a contrast of the latent variables. Wei and Pan [2010] further introduced different ad hoc constraints in the MCMC estimation algorithm to improve its performance. The latent variables in the model are not unlike our pseudo-covariates, both of which summarize the inter-gene relationship information. The difference is that our model conditions on these pseudo-covariates, whereas Wei and Pan [2010] averaged over the latent variables through the Gaussian Markov random field, having a more Bayesian flavor. Our conditional approach has a clear computational advantage. It allows re-use of the generated pseudo-covariates, such that a database of pseudo-covariates can be potentially constructed.
so that users can extract precomputed covariates and apply them to their new data. For the averaging approach, this is not possible; users have to integrate over the very high dimensional space through MCMC each time they obtain a new dataset. Moreover, the specification of prior probabilities in the Bayesian approach can be a challenging and rather subjective process.

Furthermore, neither the approach of Wei and Li [2007] nor Wei and Pan [2008] includes the naïve two-component mixture model as a special case. In contrast, our model succinctly uses a single smoothing parameter to represent both the naïve model and the more complicated model on a continuous scale, so that the users can easily compare results by changing the value of the smoothing parameter.

The advantages of the two Markov random field models observed in Wei and Pan [2010] are also preserved by our method. Namely, our simulations showed that the resulting ranking of differentially expressed genes is much better than the usual null model if the covariates contain useful information, and that the model is robust to incorrect information because the performance deteriorated very little when the prior inter-gene information was completely irrelevant. Analysis of real pig fasting data also demonstrated that the genes in the list produced from our method were more correlated with each other than those identified by the null model fit. It would be interesting to investigate our methods on other types of data, e.g., the array-CGH data, array-based chromatin immunoprecipitation (ChIP-chip) data, or gene association data, to which variants of Markov random field models have been successfully applied. Our main message is that combining multidimensional scaling with nonparametric smoothing is a very attractive strategy for analyzing microarray data, when prior information about inter-gene relationships is available.

Appendix

4.A A generalized Expectation-Maximization algorithm

Since model (4.3) can be augmented with latent random variables \( Z_1, Z_2, \ldots, Z_G \), the Expectation-Maximization (EM) algorithm [Dempster et al., 1977; Green, 1990] is a natural choice to maximize the penalized log likelihood. The pseudo-code is listed in Algorithm 4.1,
and detailed below.

Algorithm 4.1 Pseudo-code for the EM algorithm

1: Initialize $\beta$ and $s$
2: repeat
3:  compute $\ell_i$ for all $i = 1, 2, \ldots, G$ \{E-step\}
4:  $s \leftarrow \arg \max_{1 < s < \|t\|_{\infty}} \sum_{i=1}^{G} (1 - \ell_i) \log [g(t_i/s; \nu, 0)/s]$ \{M-step for $s$\}
5: repeat \{M-step for $\beta$\}
6:  $\beta \leftarrow \beta - \alpha \left[ \tilde{H}(\beta) + \tau I \right]^{-1} \tilde{g}(\beta)$
7: until $\|\tilde{g}(\beta)\|_{\infty} < \epsilon$
8: until convergence

Suppose, at iteration $r$ of the EM algorithm, the estimated parameters are $(s^{(r)}, \beta^{(r)})$. Then given this estimate, the posterior distribution of $z_i$ given $t_i$ is Bernoulli$(1 - \ell_i^{(r)})$, where $\ell_i^{(r)} = \ell_i(s^{(r)}, \beta^{(r)})$ is computed from (4.6). In the E-step, one computes the expectation, with respect to the conditional distribution of $\{Z_i\}_{i=1}^{G}$ given $\{t_i\}_{i=1}^{G}$, of the complete log likelihood based on $\{t_i, z_i\}_{i=1}^{G}$,

$$E_{z_i|t_i} \left\{ \sum_{i=1}^{G} \log \left[ \pi_0(1 - z_i)g(t_i; \nu, 0) + (1 - \pi_0)z_ig(t_i/s; \nu, 0)/s \right] \right\}$$

$$\propto \sum_{i=1}^{G} \left\{ \left( 1 - \ell_i^{(r)} \right) M_i' \beta - \log \left[ 1 + \exp(M_i' \beta) \right] \right\}$$

$$+ \sum_{i=1}^{G} \left( 1 - \ell_i^{(r)} \right) \log \left[ g \left( \frac{t_i}{s}; \nu, 0 \right) \frac{1}{s} \right],$$

(4.8)

To get maximum likelihood estimates, one needs to maximize eq.(4.8) in the M-step. Here, we maximize $[\text{eq.} (4.8) - 0.5\lambda \beta' \Omega \beta]$ for penalized likelihood estimation. This change does not violate the validity of the EM algorithm [Green, 1990].

Note that second summation in (4.8) is free from $\beta$; thus, it can be maximized separately by a golden section search [Kiefer, 1953] to get an updated scale factor,

$$s^{(r+1)} = \arg \max_{1 < s < \|t\|_{\infty}} \sum_{i=1}^{G} \left( 1 - \ell_i^{(r)} \right) \log [g(t_i/s; \nu, 0)/s]$$

where $\|t\|_{\infty} = \max_i |t_i|$ is an upper bound on sensible values of $s$. This maximization can be easily done with the optimize function in R.
For maximization with respect to $\beta$, we choose to use the Newton-Raphson procedure. Specifically, we define the objective function to be minimized as

$$\sum_{i=1}^{G} \left\{ -\left(1 - \ell_i^{(r)}\right) M_{(i)}' \beta + \log \left[1 + \exp(M_{(i)}' \beta)\right] \right\} + 0.5 \lambda \beta' \Omega \beta.$$  

(4.9)

The gradient is

$$\hat{g}(\beta) = M'w_{\hat{g}}(\beta) + \lambda \Omega' \beta,$$

where $w_{\hat{g}}(\beta) = \left\{ \ell_i^{(r)} - \pi_{0i}(\beta) \right\}_{i=1}^{G}$, and $\pi_{0i}(\beta)$ is given by (4.5). The Hessian matrix is

$$\hat{H}(\beta) = M'W_{\hat{H}}(\beta)M + \lambda \Omega,$$

where $W_{\hat{H}}(\beta) = \text{diag}\{\pi_{0i}(\beta) [1 - \pi_{0i}(\beta)] \}_{i=1}^{G}$. The Newton-Raphson update of $\beta$ is given by

$$\beta^{(\text{new})} \leftarrow \beta^{(\text{old})} - \alpha \left[ \hat{H}(\beta^{(\text{old})}) + \tau I \right]^{-1} \hat{g}(\beta^{(\text{old})}),$$

where $\beta^{(\text{old})}$ and $\beta^{(\text{new})}$ denote the parameter estimates at the current and the next Newton-Raphson iteration, respectively, $\alpha > 0$ is a step length, and $\tau \geq 0$ is a small ridge factor that guarantees the Hessian to be positive definite and computationally well-conditioned. Usually $\alpha = 1$ and $\tau = 0$ work well. However, if problem occurs, i.e., non-positive definite Hessian and/or non-decreasing objective function (4.9), it may be necessary to reduce $\alpha$ and/or increase $\tau$. Again, such optimization routines are included in most statistical software, e.g., the \texttt{nlminb} function in \texttt{R}.

For a full Newton-Raphson optimization in the M-step, one repeatedly update parameters until $\beta^{(\text{old})}$ and $\beta^{(\text{new})}$ are close enough. For example, one may choose $\|\hat{g}(\beta^{(\text{new})})\|_{\infty} < \epsilon$ as a termination criterion for some small $\epsilon > 0$, and set $\beta^{(r+1)} = \beta^{(\text{new})}$. However, from our experience, a one-step update is usually sufficient by letting $\epsilon = \infty$, since the EM algorithm only requires that the penalized log likelihood increases at each iteration. This results in the so-called generalized EM algorithm.
Appendix

4.B The gradient and Hessian for use in Newtonian algorithms

To use quasi-Newton or Newton-Raphson algorithms, we provide the gradients and Hessian of the penalized log likelihood with respect to parameters below.

Denote the negative penalized log likelihood (NPLL) based on data point $i$ as

$$NPLL_i = -\log [f(t_i)] + \frac{1}{2G} \lambda \beta' \Omega \beta,$$

where $f(t_i)$ is given in (4.3). Then the objective function to be minimized is $NPLL = \sum_{i=1}^{G} NPLL_i$. The vector of derivatives with respect to $\beta$ can be written as

$$\frac{\partial}{\partial \beta} NPLL = M' \omega_\beta + \lambda \Omega \beta,$$

where the $G$-vector

$$\omega_\beta = \left\{ \frac{g_{0i} - g_{1i}}{f(t_i)} \pi_0 (1 - \pi_0) \right\}_{i=1}^{G}.$$

For $s > 1$, we reparameterize as $s = 1 + \exp(r)$, i.e., $r = \log(s-1) \in \mathbb{R}$, to remove the boundary constraint. Hence $ds = \exp(r) \, dr = (s-1) \, dr$. So, with respect to $r$,

$$\frac{\partial}{\partial r} NPLL_i = \left( \frac{1}{s-1} \right) (1 - \ell_i) \frac{t_i^2 - s^2}{\ell_i^2 / \nu + s^2}$$

and

$$\frac{\partial}{\partial r} NPLL = \sum_{i=1}^{G} \frac{\partial}{\partial r} NPLL_i.$$

The matrix of second derivatives with respect to $\beta$ can be written as,

$$\frac{\partial^2}{\partial \beta \partial \beta'} NPLL = M' W_{\beta \beta} M + \lambda \Omega,$$

where the $G \times G$ diagonal matrix $W_{\beta \beta}$ is the product of

$$\text{diag} \{ \omega_\beta \} \quad \text{and} \quad \text{diag} \{ \pi_0 \ell_i (1 - \pi_0) (1 - \ell_i) \}_{i=1}^{G}.$$

The vector of cross derivatives is

$$\frac{\partial^2}{\partial \beta \partial r} NPLL = M' \omega_{\beta r},$$

where the $G$-vector

$$\omega_{\beta r} = \left\{ \ell_i \frac{\partial}{\partial r} NPLL_i \right\}_{i=1}^{G}.$$
Finally the second derivative with respect to $r$ is given by

$$\frac{\partial^2}{\partial r^2} NPLL = (s - 1) \sum_{i=1}^{G} \frac{\partial}{\partial r} NPLL_i \left[ \frac{\ell_i \left( t_i^2 - s^2 \right)}{s \left( t_i^2 / \nu + s^2 \right)} + \frac{1}{s(s - 1)} - \frac{2s}{t_i^2 - s^2} - \frac{2s}{t_i^2 / \nu + s^2} \right].$$

In summary, the gradient vector and the Hessian matrix are

$$\mathbf{g}(r, \beta) = \begin{bmatrix} \frac{\partial}{\partial r} NPLL \\ \mathbf{M}' \mathbf{w}_\beta + \lambda \mathbf{\Omega} \beta \end{bmatrix} \quad \text{and} \quad \mathbf{H}(r, \beta) = \begin{bmatrix} \frac{\partial^2}{\partial r^2} NPLL & \mathbf{w}'_{\beta r} \mathbf{M} \\ \mathbf{M}' \mathbf{w}_{\beta r} & \mathbf{M}' \mathbf{W}_{\beta \beta} \mathbf{M} + \lambda \mathbf{\Omega} \end{bmatrix},$$

respectively. The Newton-Raphson algorithm proceeds similarly to the one in the M-step in Algorithm 4.1, except that the gradient and Hessian are different. We will not detail it again.

Quasi-Newton procedures (e.g., BFGS) do not need the Hessian matrix. Our code is provided in an R package hisemi. Although some of the matrices may be large, they are highly sparse. Using sparse matrix routines, our implementation is reasonably fast for practical use.

**Bibliography**


Matthews, L., Gopinath, G., Gillespie, M., Caudy, M., Croft, D., de Bono, B., Garapati, P., Hemish, J., Hermjakob, H., Jassal, B., Kanapin, A., Lewis, S., Mahajan, S., May, B.,


R Development Core Team (2009). *R: A Language and Environment for Statistical Computing.*
R Foundation for Statistical Computing, Vienna, Austria.


Vastrik, I., D'Eustachio, P., Schmidt, E., Gopinath, G., Croft, D., de Bono, B., Gillespie, M.,


CHAPTER 5. GENERAL CONCLUSIONS

In this thesis research, we studied several topics related with statistical inference for gene expression profiling data, and developed and evaluated new methods for detecting differential expression.

In Chapter 2, we proposed several new model selection procedures for selecting the variance part of linear models. Our modified multi-response permutation (MRPP) test has tighter control of type I errors and also has good power compared to other methods. Our cross-validation procedures are able to differentiate linear models that only differ in the variance assumptions. We also give the right AICc (corrected Akaike’s Information Criterion) formula that removes the bias in AIC when multiple variances need to be estimated independently. Simulations based on real microarray datasets indicated that using the correct variance models does not necessarily provide the best separation between differentially and equivalently expressed genes, although using the correct models can control false discovery rates (FDR) at desired levels.

Model selection is an alternative approach to shrinkage estimation. Chapter 2 considered the problem of model selection across datasets, whereas most existing research considered shrinkage estimation across genes within datasets. It would be interesting to investigate other combinations in future studies, e.g., variance model selection across genes, variance model selection both across genes and across datasets, shrinkage estimation across genes followed by variance model selection across datasets, etc.

Chapter 3 and Chapter 4 are related to analyses that involve a large number of t-tests. In Chapter 3, we developed parametric, nonparametric and semiparametric methodologies for estimating the distribution of noncentrality parameters. We demonstrated that our proposed methods improve existing methods. However, the assumption of independence across genes is not
eliminated from our methods. On the other hand, Chapter 4 addressed the problem of modeling dependency across genes using \textit{a priori} information summarized by multidimensional scaling (MDS). However, the model for the noncentrality parameters was assumed to be parametric. Hence, the methods in Chapter 3 outperform those in Chapter 4 in terms of the flexibility of the noncentrality parameter distribution, and the methods in Chapter 4 outperform those in Chapter 3 in terms of modeling dependency across genes. For future studies, it is natural to combine these two classes of methods by allowing the noncentrality parameter distribution to be more flexible and allowing \textit{a priori} information to be incorporated in statistical inference.

Chapter 2 and Chapter 4 shared some similarity in terms of combining multiple sources of information into a single framework, whereas still maintaining statistical rigor. In the surge of systems biology studies in the “small $N$, large $p$” situation, the ability to make use of heterogeneous sources of information is essential and requires further innovations.

Another area that deserves further research is the adaptation of the developed methods to gene expression profiling experiments using high-throughput sequencing technologies. Chapters 2 through 4 are dependent on linear models and/or availability of $t$-tests, which may not directly apply to the counting responses from sequencing experiments. However, the underlying principles can still be transferred to sequencing experiments. That is, to borrow information from other genes, other datasets, and/or other sources of relationship across genes.


distributions: noncentral chisquare, noncentral t and the distribution of the square of the


Bolstad, B., Collin, F., Brettschneider, J., Simpson, K., Cope, L., Irizarry, R., and Speed,
W. Huber, R. Irizarry, and S. Dudoit, editors, *Bioinformatics and Computational Biology
Solutions Using R and Bioconductor*, Statistics for Biology and Health, pages 33–47. New
York: Springer.

normalization methods for high density oligonucleotide array data based on variance and
bias. *Bioinformatics* 19, 185–93.

edition.


Breiman, L. (2001). Statistical modeling: The two cultures (with comments and a rejoinder by


McCarthy, M. I., Abecasis, G. R., Cardon, L. R., Goldstein, D. B., Little, J., Ioannidis, J. P. A.,
and Hirschhorn, J. N. (2008). Genome-wide association studies for complex traits: consensus,


Medvedovic, M., Yeung, K., and Bumgarner, R. (2004). Bayesian mixture model based clustering


Springer-Verlag, 2nd edition.


Moody, J. E. (1991). Note on generalization, regularization and architecture selection in
nonlinear learning systems. In *Neural Networks for Signal Processing [1991]., Proceedings of

Moody, J. E. (1992). The effective number of parameters: An analysis of generalization and
regularization in nonlinear learning systems. In *Advances in Neural Information Processing


