Statistical Methods for High Throughput Genomics

by

Chi Ho Lo

B.Sc., The University of Hong Kong, 2003
M.Phil., The University of Hong Kong, 2005

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

The Faculty of Graduate Studies

(Statistics)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

September 2009

Abstract

The advancement of biotechnologies has led to indispensable high-throughput techniques for biological and medical research. Microarray is applied to monitor the expression levels of thousands of genes simultaneously, while flow cytometry (FCM) offers rapid quantification of multi-parametric properties for millions of cells. In this thesis, we develop approaches based on mixture modeling to deal with the statistical issues arising from both high-throughput biological data sources.

Inference about differential expression is a typical objective in analysis of gene expression data. The use of Bayesian hierarchical gamma-gamma and lognormal-normal models is popular for this type of problem. Some unrealistic assumptions, however, have been made in these frameworks. In view of this, we propose flexible forms of mixture models based on an empirical Bayes approach to extend both frameworks so as to release the unrealistic assumptions, and develop EM-type algorithms for parameter estimation. The extended frameworks have been shown to significantly reduce the false positive rate whilst maintaining a high sensitivity, and are more robust to model misspecification.

FCM analysis currently relies on the sequential application of a series of manually defined 1D or 2D data filters to identify cell populations of interest. This process is time-consuming and ignores the high-dimensionality of FCM data. We reframe this as a clustering problem, and propose a robust model-based clustering approach based on t mixture models with the Box-Cox transformation for identifying cell populations. We describe an EM algorithm to simultaneously handle parameter estimation along with transformation selection and outlier identification, issues of mutual influence. Empirical studies have shown that this approach is well adapted to FCM

data, in which a high abundance of outliers and asymmetric cell populations are frequently observed. Finally, in recognition of concern for an efficient automated FCM analysis platform, we have developed an **R** package called `flowClust` to automate the gating analysis with the proposed methodology. Focus during package development has been put on the computational efficiency and convenience of use at users' end. The package offers a wealth of tools to summarize and visualize features of the clustering results, and is well integrated with other FCM packages.

Table of Contents

List of Tables

List of Figures

Acknowledgements

I dedicate my greatest gratitude to my supervisors, Raphael Gottardo and Ryan R. Brinkman, for their inspirational guidance and continuous support throughout this journey. I would also like to thank my committee members, Jenny Bryan and Kevin Murphy, for their invaluable advice on my research and career.

In addition to the aforementioned exceptional scholars, I feel honored to study at the UBC Statistics Department filled with excellent mentors including Paul Gustafson, Harry Joe, John Petkau and Matías Salibián-Barrera (in alphabetical order). I have learnt so much from their insightful advice and tremendous knowledge of statistics.

My list of acknowledgements also goes to

- Ryan R. Brinkman (again!), Ali Bashashati and Josef Spidlen, among others, for their assistance and helpful discussion during my internship at the British Columbia Cancer Research Centre;

- Florian Hahne, Martin Morgan, Patrick Aboyoun and Marc Carlson from Fred Hutchinson Cancer Research Centre for their professional advice on the technical issues in software development;

- Bakul Dalal, Maura Gasparetto, Mario Roederer, Clayton Smith and the British Columbia Cancer Research Centre for kindly offering clinical data and providing assistance to interpret the data;

- WestGrid for providing tremendous computational resources which are indispensable to my research, and its technical support team, Roman Baranowski in particular, for their prompt and efficient response to my endless questions about the priority system for resource allocation;

- Tony W. K. Fung for enriching my research experience with his excellent supervision during my Master's at the University of Hong Kong;
- the office ladies, namely, Peggy Ng, Elaine Salameh and Viena Tran, from whom I truly feel that they always provide efficient assistance with sincere care rather than merely carrying out their duties.

The research constituting this thesis has received funding support from Genome Canada, MITACS, NIH, NSERC, PIMS, and University Graduate Fellowships.

Statement of Co-Authorship

This thesis is completed under the supervision of Dr. Raphael Gottardo and Dr. Ryan R. Brinkman.

Chapter 2 of this thesis is co-authored with Dr. Raphael Gottardo. I developed the methodology, performed the analysis, and prepared the manuscript.

Chapter 3 is co-authored with Dr. Raphael Gottardo. I identified the research problem, conceived of the study, developed the methodology, performed the analysis, and prepared the manuscript.

Chapter 4 is co-authored with Dr. Ryan R. Brinkman and Dr. Raphael Gottardo. I developed the methodology, performed the analysis, and prepared the manuscript.

Chapter 5 is co-authored with Dr. Florian Hahne, Dr. Ryan R. Brinkman and Dr. Raphael Gottardo. I conceived of the study, developed the methodology and software, performed the analysis, and prepared the manuscript.

Chapter 1

Introduction

Recent technological advances in molecular biology have enabled the rapid quantification of characteristics for an enormous number of genes or cells under the same experimental condition. Microarray has been being a popular technique for monitoring the expression levels of thousands of genes for more than a decade, while flow cytometry (FCM) offers quantification of multi-parametric properties for up to millions of cells. To date, extensive applications of these two high-throughput technologies can be found in health research, medical diagnosis and treatment, drug discovery and vaccine development (Schena *et al.*, 1995; DeRisi *et al.*, 1996; Behr *et al.*, 1999; Debouck and Goodfellow, 1999; Hengel and Nicholson, 2001; Braylan, 2004; Illoh, 2004; Mandy, 2004; Orfao *et al.*, 2004; Bolton and Roederer, 2009).

The interest in studying changes in gene expression levels over experimental conditions have led to the development of a wealth of methodology for identifying differentially expressed genes. Meanwhile, the tremendous attention built towards FCM in recent years has urged the need for both methodological and software development for an automated analysis platform of gating, the process of identifying cell populations. In this thesis, we show that the aforementioned issues can be recast into problems of clustering, the process of looking for homogeneous groups of observations in statistics. We introduce flexible forms of finite mixture models (Titterington *et al.*, 1985; Banfield and Raftery, 1993; McLachlan and Peel, 2000; Fraley and Raftery, 2002), commonly applied as a statistical tool for clustering, which serve as the modeling basis for approaches developed to deal with the issues arising from both high-throughput biological data sources.

In this chapter, we review the technology of microarrays and several popular methods for differential gene expression. We then give a brief account

of the FCM technology as well as a few attempts to automate the gating analysis to date. Next, in Chapter 2, we introduce mixture models based on a flexible empirical Bayes approach to detect differentially expressed genes in microarray data. This approach releases the unreasonable assumptions and enhances the flexibility of models introduced in Newton *et al.* (2001) and Kendziorski *et al.* (2003). In Chapter 3, we develop a unified framework to *simultaneously* handle data transformation, outlier identification and clustering, issues which are of mutual influence. This methodology stems from a mixture model using the multivariate t distributions with the Box-Cox transformation, which can be viewed as a new class of distributions extending the t distribution. We proceed to present in Chapter 4 the result obtained on applying the proposed methodology to FCM data, from which cell populations asymmetric in shape and an abundance of outliers are often observed. In Chapter 5 we introduce an open-source software package called `flowClust` to implement the methodology introduced in Chapters 3 and 4. This publicly available package addresses a bottleneck to FCM that there is a dearth of software tools to manage, analyze and present data on a sound theoretical ground. Finally, we conclude in Chapter 6 with a discussion of the overall contribution of this research work, and directions for future extensions.

1.1 Differential Gene Expression Analysis of Microarray Data

1.1.1 The Technology of Microarrays

The structure, function, development and reproduction of an organism depends on the type and amount of proteins present in each cell and tissue. A protein is a sequence of up to 20 types of amino acids, which is specified by the nucleotide sequence of the encoding gene(s). The synthesis of proteins consists of two major stages, transcription and translation, and is described by the central dogma of molecular biology (Crick, 1970); see Figure 1.1. The genetic information encoded by the deoxyribonucleic acid (DNA) is first

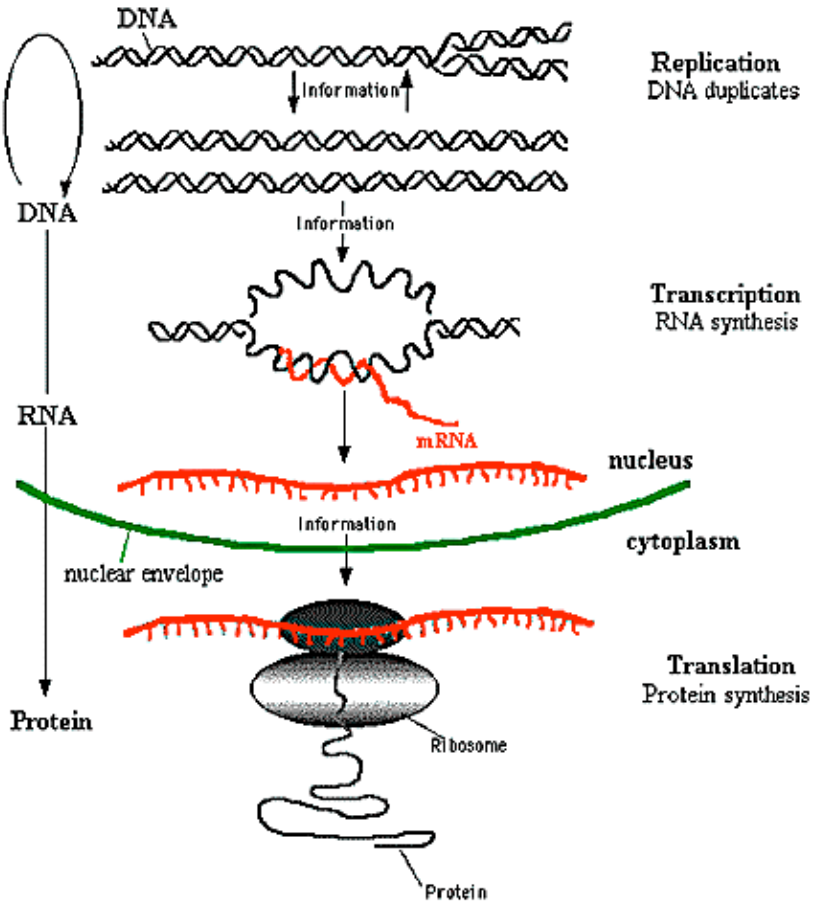


Figure 1.1: The central dogma of molecular biology. The synthesis of proteins constitutes two major stages, transcription and translation. Part of the DNA is first transcribed into the single-stranded mRNA taking a complementary sequence. The mRNA then migrates from the nucleus to the cytoplasm, and is translated into proteins. (Picture source: access-excellence.org)

transcribed into the messenger ribonucleic acid (mRNA), a single-stranded sequence complementary to the base sequence in the DNA. The mRNA then migrates from the nucleus to the cytoplasm, and is translated into proteins.

When a gene is transcribed and then translated, we say that it is expressed. Cells under different conditions tend to express different sets of genes, and thereby synthesize different proteins. To understand a biological process, it is important to know what proteins are being processed. Nonetheless, due to the complex structures, the analysis of proteins is difficult. Based on the fact that the mRNA gets translated into proteins, the analysis of gene expression helps provide information about the biological process of interest. This is where microarrays, a technology which facilitates the simultaneous measurement of expression levels of thousands of genes, come forth.

The microarray technology relies on two key chemical processes, reverse transcription and hybridization. The process of reverse transcription creates single-stranded complementary DNA (cDNA) copy of mRNA transcripts experimentally isolated from a cell. Hybridization is the process of combining two single strands of DNA or RNA into a single molecule. Two strands which are perfectly complementary to each other tend to bind together, resulting in specific hybridization. The term "specific" is used as opposed to the case in which binding randomly occurs between two strands that do not form a complementary pair.

Microarrays can be classified into two categories: the cDNA microarrays and the oligonucleotide arrays. Below we give a brief account of each of these two categories.

cDNA Microarrays

A cDNA microarray consists of thousands of microscopic spots attached to a solid surface, with each spot containing a massive number of identical DNA sequences to serve as probes. The choice of probes may be customized in-house to satisfy specific experimental needs. In a typical dual-color cDNA microarray experiment, two mRNA samples extracted from different exper-

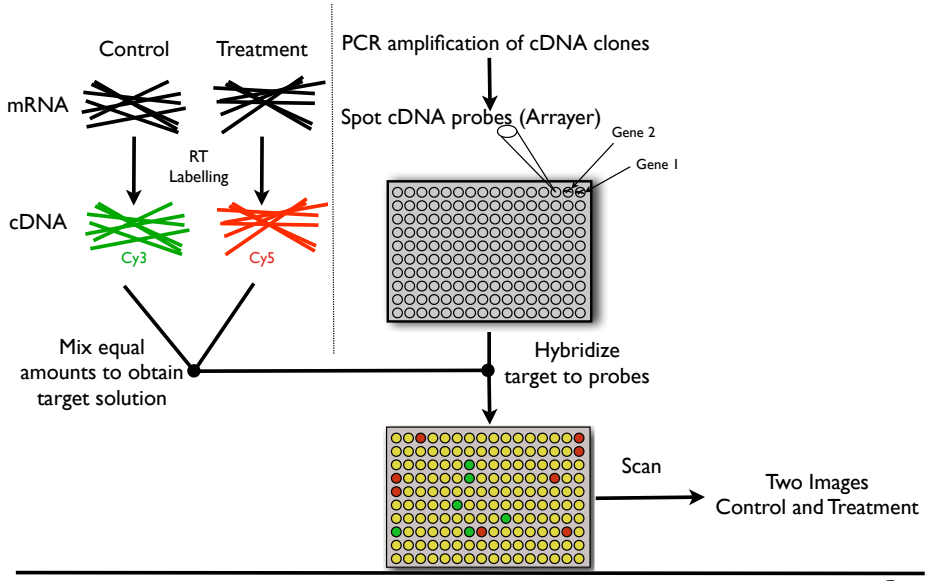


Figure 1.2: The cDNA microarray experiment. A cDNA microarray is a glass microscope slide spotted with individual DNA sequences as probes. The cDNA solutions, prepared from mRNA by reverse transcription, are labeled with green and red dyes respectively to identify the source (control and treatment). The mixed cDNA target solution is then hybridized with the probes on the microarray. The array is scanned twice to obtain images for the red and green intensities.

imental conditions are reverse-transcribed into cDNA, labeled with different fluorescent dyes (red and green), mixed and targeted against the probes on the microarray. Owing to the property of preferential binding of a labeled cDNA molecule (target) to a probe containing the complementary sequence, specific hybridization occurs, under some stringent environment. The array is then scanned and the red and green intensities for each spot are measured. Figure 1.2 illustrates such an experiment.

Oligonucleotide Arrays

In a high-density oligonucleotide array, the probes are composed of short DNA sequences known as oligonucleotides. Affymetrix GeneChip arrays,

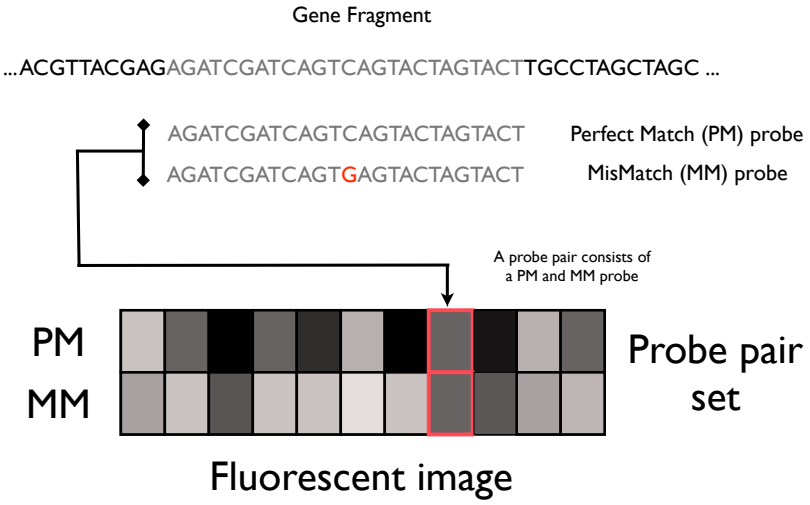


Figure 1.3: The representation of a gene with a probe set on Affymetrix GeneChip arrays. A probe set consists of 11–20 distinct probe pairs. Each perfect match (PM) probe contains an excerpted sequence (25bp long) of the gene. A mismatch (MM) probe is created from a PM probe by changing the middle base to its complement.

which consist of up to 33,000 genes with probes containing oligonucleotides 25bp long, are the most popular in this technology. Each gene is represented by a set of 11–20 distinct probe pairs. The perfect match (PM) probe of each probe pair contains a section of the mRNA molecule of interest; the mismatch (MM) probe is created by changing the middle (13th) base of the PM probe. The use of probe sets to represent genes reduces the chance of non-specific hybridization by including only probes unique to the genome, while the presence of MM probes helps quantify the non-specific hybridization that still occurs. A graphical depiction of the relationship between gene sequence and probe set is given in Figure 1.3. Various methods are available for computing the expression summary values from the probe intensities, for example, gcRMA (Wu *et al.*, 2004), RMA (Irizarry *et al.*, 2003), MAS 5 (Affymetrix Manual, 2001), and dChip (Li and Wong, 2001).

Compared to cDNA microarrays, high-density oligonucleotide arrays have

a lower chance of non-specific hybridization and a high detection specificity, and allow for more genes to be probed in one experiment. However, an Affymetrix GeneChip array does not support a dual-channel system and can only be exposed to one sample in each experiment. Also, as an off-the-shelf product, the probes on the array are not to be customized.

1.1.2 Methods for Detecting Differentially Expressed Genes

The analysis of differential gene expression helps us understand how genes are differentially expressed under different conditions, for example, normal and cancer. In recent years, there has been a considerable amount of work on the detection of differentially expressed genes. In the following we give a review of some representative methods.

t Tests and Variants

Simplistic statistical treatments include the use of two-sample t tests on the log intensities, or one-sample t tests on the log intensity ratios for each gene (Callow *et al.*, 2000). A gene is declared to be differentially expressed if its p-value is less than a threshold, for example, 0.05. Because of the large number of hypothesis tests, adjustment methods such as Bonferroni or Holm-Bonferroni should be employed to control the familywise error rate, the probability of yielding one or more false positives. In addition, due to the small number of replicates in microarray experiments, the gene-specific variance can be poorly estimated. Baldi and Long (2001) suggested using a modified t test statistic where the denominator is regularized by using a weighted average of the gene-specific and global variance estimates:

$$\frac{\nu_0 s_0^2 + (R-1)s_g^2}{\nu_0 + R - 2}, \tag{1.1}$$

where R is the number of replicates, s_0^2 and s_g^2 are respectively the estimates for the global and gene-specific variances, and ν_0 is a tuning parameter that governs the contribution of the global variance. Here, "global" may refer to all the genes, or those in the neighborhood of gene g. This regularized

variance estimate is derived from the mean of the posterior distribution of the gene-specific variance in a Bayesian framework.

Significance Analysis of Microarrays (SAM)

SAM, proposed by Tusher *et al.* (2001), uses a regularized t statistic d_g where a constant c is added to the gene-specific standard error s_g:

$$d_g = \frac{\overline{M}_g}{c + s_g}, \tag{1.2}$$

where $\overline{M}_g$ denotes the average log intensity ratio for gene g. The value of c was suggested by Efron *et al.* (2001) to be the 90th percentile of all s_g. To estimate empirically the distribution of the statistic d_g under the null hypothesis (no differential expression), different permutations of the replicates are considered, and the statistic shown in Eq.(1.2) is recomputed for each permutation. The average of the statistics over all permutations, denoted as $\tilde{d}_g$, is then determined for each gene. By considering the displacement of d_g from $\tilde{d}_g$, and a threshold Δ, asymmetric cutoffs are obtained as the smallest d_g such that $d_g - \tilde{d}_g > \Delta$, and the largest d_g such that $d_g - \tilde{d}_g < -\Delta$. The threshold Δ is determined by controlling the false discovery rate (FDR), the proportion of falsely identified genes among the genes declared differentially expressed, at 10% or a reasonable level. In SAM, the FDR is estimated as the ratio of the average number of genes called significant from those permuted datasets to that number from the original dataset.

Lönnstedt and Speed's B statistic

Making use of an empirical Bayes normal mixture model, Lönnstedt and Speed (2002) proposed the log posterior odds statistic, more conveniently called the B statistic, to determine differentially expressed genes. More explicitly, the log intensity ratio M_{gr} for gene g on the r-th replicate is assumed to follow a normal distribution $\mathrm{N}(\mu_g, k\tau_g^{-1})$. Let I_g be the indicator variable such that $I_g = 1$ if gene g is differentially expressed and $I_g = 0$ otherwise. The parameters μ_g and τ_g have the following conjugate prior

distribution:

$$\begin{aligned}\tau_g &\sim \mathrm{Ga}(\nu/2, 1) \\ \mu_g|\tau_g &= \begin{cases} 0 & \text{if } I_g = 0 \\ \mathrm{N}(0, ck\tau_g^{-1}) & \text{if } I_g = 1 \end{cases}\end{aligned}$$

A mixture structure is implicitly assumed by the above specification on μ_g. Let $p = \Pr(I_g = 1)$ be the proportion of differentially expressed genes. The log posterior odds of differential expression is derived to be

$$\begin{aligned} B_g &= \log \frac{\Pr(I_g = 1|\mathbf{M})}{\Pr(I_g = 0|\mathbf{M})} \\ &= \log \frac{p}{1-p} \frac{f(\mathbf{M}_g|I_g = 1)}{f(\mathbf{M}_g|I_g = 0)}. \end{aligned} \tag{1.3}$$

A large value of B_g is in favor of the alternative hypothesis of differential expression. Note, however, that due to computational difficulty, the authors did not estimate the proportion p and fixed it *a priori*. In consequence, a cutoff on B_g for declaring differential expression could not be determined on an objective ground.

Linear Models for Microarray Data (LIMMA)

LIMMA (Smyth, 2004) reformulates the aforementioned hierarchical model of Lönnstedt and Speed (2002) in the context of general linear models to cater for the case of a different number of replicates in different conditions and the case of multiple conditions. In addition, LIMMA uses a moderated t statistic in place of the posterior odds statistic given by Eq.(1.3) for inferencing about differential expression:

$$\tilde{t}_g = \frac{\overline{M}_g}{\tilde{s}_g/\sqrt{R}}, \tag{1.4}$$

where the posterior variance estimate

$$\tilde{s}_g^2 = \frac{\nu s^2 + (R-1)s_g^2}{\nu + R - 1} \tag{1.5}$$

provides shrinkage of the sample variance s_g^2 towards a pooled estimate s^2, resulting in more stable inference when the number of replicates is small. The computation of the moderated t statistic does not depend on p in Eq.(1.3), the potentially contentious parameter that is left un-estimated in Lönnstedt and Speed (2002).

Efron's Local False Discovery Rate (fdr)

Efron (2004) proposed an empirical Bayes approach combined with a local version of the false discovery rate to test for differential expression. In this method, t test statistics are first obtained, one for each gene. The associated p-values are converted into z-scores defined as $z_g = \Phi^{-1}(p_g)$, where Φ indicates the standard normal distribution function. A two-component mixture,

$$f(z_g) = p_0 f_0(z_g) + p_1 f_1(z_g), \tag{1.6}$$

where f_0 and f_1 refer to the density of the z-scores under the null (no differential expression) and alternative (differential expression) hypotheses respectively, and p_0 and p_1 are the proportion of true null and alternative hypotheses, is used to model the z-scores. The mixture density f and the null density f_0 are then empirically estimated. For each gene, inference is based on the local false discovery rate, which is defined as

$$\text{fdr}(z_g) \equiv \frac{\hat{f}_0(z_g)}{\hat{f}(z_g)}. \tag{1.7}$$

The notation fdr is deliberately shown in lowercase to signify its difference from the usual definition of false discovery rate proposed by Benjamini and Hochberg (1995). A gene with an fdr lower than some threshold, say, 10%, will be called differentially expressed.

Empirical Bayes Gamma-Gamma and Lognormal-Normal Models (EBarrays)

Newton *et al.* (2001) developed a method for detecting changes in gene expression using a hierarchical gamma-gamma (GG) model. Kendziorski *et al.* (2003) extended this to multiple replicates with multiple conditions, and provided the option of using a hierarchical lognormal-normal (LNN) model. For the GG model, each observation is modeled by a gamma distribution with shape a and rate θ. Strength is borrowed across genes by assuming a gamma prior on θ. Denote by $\mathbf{x}_g$ and $\mathbf{y}_g$ the intensities for gene g in two conditions respectively. The following two-component mixture is used to model the data:

$$p(\mathbf{x}_g, \mathbf{y}_g) = p\ p_A(\mathbf{x}_g, \mathbf{y}_g) + (1-p)\ p_0(\mathbf{x}_g, \mathbf{y}_g), \tag{1.8}$$

where

$$p_A(\mathbf{x}_g, \mathbf{y}_g) = \int \Big(\prod_r p(x_{gr}|a, \theta_{gx})\Big)\pi(\theta_{gx})\ d\theta_{gx} \cdot \int \Big(\prod_r p(y_{gr}|a, \theta_{gy})\Big)\pi(\theta_{gy})\ d\theta_{gy} \tag{1.9}$$

is the marginal density for differentially expressed genes using condition-specific rate parameters, and

$$p_0(\mathbf{x}_g, \mathbf{y}_g) = \int \Big(\prod_r p(x_{gr}|a, \theta_g)\ \prod_r p(y_{gr}|a, \theta_g)\Big)\pi(\theta_g)\ d\theta_g \tag{1.10}$$

is for non-differentially expressed genes using a common rate parameter. The LNN model assumes a lognormal sampling distribution for each observation with mean μ and variance σ^2. A conjugate normal prior is imposed on μ. The corresponding mixture model takes a form similar to that shown in Eq.(1.8), where the marginal densities are derived by considering $\mu_{gx} \neq \mu_{gy}$ (differential expression) and $\mu_{gx} = \mu_{gy}$ (no differential expression) respectively. Note that, in both models, the assumption of a constant coefficient of variation for all genes has been implicitly made. For both models, the prior can be integrated out and the EM algorithm can be used to estimate the unknown parameters. Inference is based on the posterior probabilities

of differential expression.

Bayesian Robust Inference for Differential Gene Expression (BRIDGE)

Gottardo *et al.* (2006) developed a robust Bayesian hierarchical model for testing differential expression. The model may be viewed as an extension of the LNN specification of EBarrays. An enhanced flexibility is achieved by measures taken to release the implicitly made assumption of the constant coefficient of variation in EBarrays, and to account for outliers. BRIDGE includes an exchangeable prior for the variances, which allows different variances for the genes whilst still achieving shrinkage of extreme variances. In addition, observations are modeled using a t distribution, which accounts for outliers. A fully Bayesian approach is adopted, by assuming vague priors on the parameters and carrying out parameter estimation using Markov chain Monte Carlo (MCMC) algorithms (Gelfand and Smith, 1990).

Due to the relatively small number of replicates in microarray experiments, combining information across genes in statistical analysis is vital. A Bayesian framework fits such a scenario very well, and is adopted by most of the aforementioned methods. Also, mixture modeling is a popular strategy among these methods. Incidentally, the majority of the aforementioned methods are also applicable to oligonucleotide arrays upon slight or no modification, although they are presented primarily for cDNA microarrays. Also, most of the methods have included an extension to multiple conditions.

1.2 Gating Analysis of Flow Cytometry Data

1.2.1 The Technology of Flow Cytometry

Flow cytometry (FCM) is a high-throughput technology that offers automated quantification of a set of physical and chemical characteristics for up to millions of cells in a sample. The characteristics measured for each cell include size, granularity or internal complexity, and fluorescence inten-

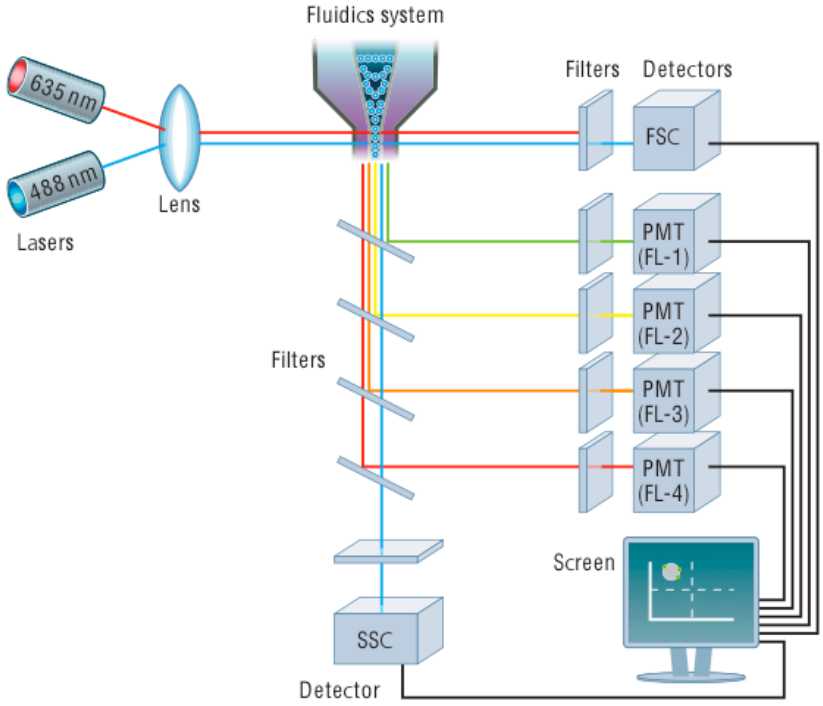


Figure 1.4: The schematic overview of a typical flow cytometer setup. Cells in the fluidics system are aligned in single file. When a cell intercepts the light source, light scattering and emission will occur. The scattered light is collected by the forward scatter (FSC) and sideward scatter (SSC) detectors. The emitted light is collected by the photomultiplier tubes (PMT), each of which targets at an individual narrow range of wavelengths. (Picture source: ab-direct.com)

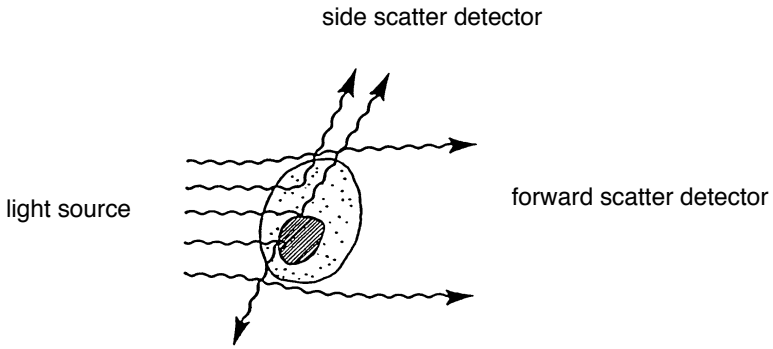


Figure 1.5: The occurrence of light scattering. The forward scatter detector measures the amount of light diffracted by a cell in the forward direction. The sideward scatter detector measures the amount of refracted or reflected light. (Picture source: BD Biosciences)

sity. FCM is widely used in health research and treatment for a variety of tasks, such as the monitoring of the course and treatment of HIV infection, the diagnosis and monitoring of leukemia and lymphoma patients, the cross-matching of organs for transplantation, and research on vaccine development (Hengel and Nicholson, 2001; Bagwell, 2004; Braylan, 2004; Illoh, 2004; Krutzik *et al.*, 2004; Mandy, 2004; Orfao *et al.*, 2004; Bolton and Roederer, 2009).

Figure 1.4 gives a schematic overview of a typical flow cytometer setup. Cells are introduced into the sample core of the flow cytometer, where hydrodynamic forces align the cells to move in single file and at speeds up to $70,000$ cells per second. When a cell intercepts a light source (e.g., laser), light scattering will occur. The forward scatter (FSC) detector in Figure 1.5 measures the amount of light diffracted in the forward direction, and is proportional to the cell-surface area or size. The sideward scatter (SSC) detector measures the amount of light refracted or reflected by any interface within the cell, and is proportional to cell granularity or internal complexity.

Before they are introduced into the flow cytometer, cells have been tagged with fluorescently conjugated antibodies bound to the antigens (Fig-

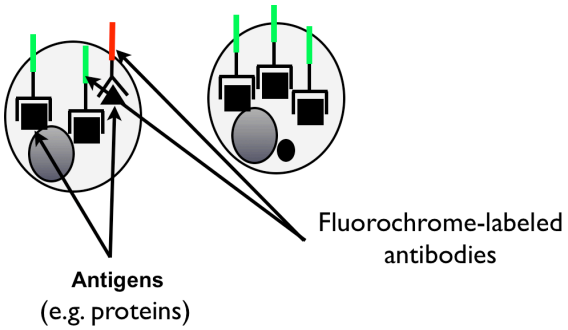


Figure 1.6: Specific binding of fluorochrome-labeled antibodies to antigens.

ure 1.6). The fluorochromes will be excited by the light source to incur light emission, when a cell intercepts the laser. The emitted light will be diverted to a series of fluorescent detectors. Each of the fluorescent detectors targets at an individual narrow range of wavelengths, close to the peak emission wavelength characteristic of an individual fluorochrome. The fluorescent signals detected are proportional to the amount of individual fluorochromes present. As each fluorochrome is conjugated to an antibody, the signal can be used to measure the amount of individual antigens.

Cells of different types have different correlated measurements of FSC and SSC. Antigens are also present in the cells at different amounts. The fluorescent signals, combined with the FSC and SSC measurements, can therefore be used to identify cell populations (homogeneous groups of cells that display a particular function) and their relative abundance in a sample.

1.2.2 Methods for Identifying Cell Populations

One major component of FCM analysis involves the process of identifying cell populations. This is referred to as gating in the FCM community. Conventionally, the identification of cell populations relies on applying sequentially a series of manually drawn filters, i.e., gates, to subset and select regions in 1D or 2D graphical representations of the data; see Figure 1.7 for an example. In such an analysis, the choice of which sequence of parameters

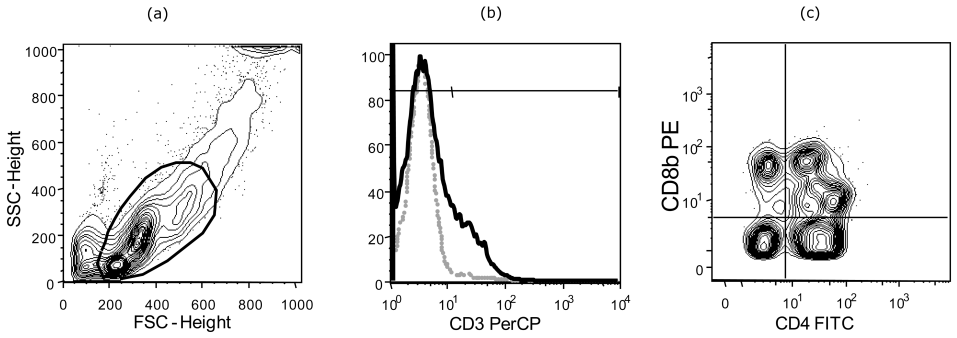


Figure 1.7: A typical manual gating analysis. (a) Data are projected onto the FSC and SSC dimensions to identify basic cell populations. An expert researcher draws a gate on the contour density plot to define the lymphocyte population. (b) The selected cells are then projected on the CD3 dimension, and $CD3^+$ cells are defined through an interval gate with the marked threshold as the lower bound. (c) A quadrant gate is applied on the projections along the CD4 and CD8β dimensions. Cells within the upper right gate is referred to as $CD3^+CD4^+CD8\beta^+$, the cell population of interest in this analysis.

to gate on and where to position the gates are highly subjective. It also ignores the high-multidimensionality of FCM data, which may convey information that cannot be displayed on 1D or 2D projections. In addition, there is a major concern for the manually time-consuming input to the manual gating analysis. It is not uncommon for an FCM study to include thousands of samples to analyze, thanks to the high-throughput technological advancement in generating FCM data. Attempts for partial automation have been made by using software to automatically apply a template gating sequence with the same set of gates on all samples. Nevertheless, the improvement in overall efficiency is limited as the variation between samples has not been taken into account, and therefore sample-by-sample manual adjustment of the position of the gates cannot be avoided. As noted in Lizard (2007), the lack of an automated analysis platform to parallel the high-throughput data-generation platform has become a major bottleneck for FCM.

In statistics, gating may be reframed as a clustering problem. There have

been few attempts to devise an automated platform with a sound statistical framework for gating analysis. Below is a brief account of the methods that have been applied to automate the gating analysis, with the latter two being recent new additions.

K-means Clustering

K-means clustering (MacQueen, 1967) is a relatively early attempt for FCM analysis. Its objective is to obtain a partition $\boldsymbol{\mathcal{P}}$ with K clusters $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_K$, corresponding to cell populations in FCM data, such that the within-cluster sum of squares is minimized:

$$\arg\min_{\boldsymbol{\mathcal{P}}} \sum_{k=1}^{K} \sum_{\mathbf{y}_i \in \mathcal{P}_k} (\mathbf{y}_i - \boldsymbol{\mu}_k)^T (\mathbf{y}_i - \boldsymbol{\mu}_k), \tag{1.11}$$

where $\mathbf{y}_i$ is an observation vector and $\boldsymbol{\mu}_k$ is the mean of $\mathcal{P}_k$. The algorithm starts with K randomly selected points as means. A K-cluster partition of the data is obtained by assigning each observation to the nearest mean, followed by a recomputation of the cluster means. Such a procedure repeats until there is no change in the assignment of observations. This method was found to be equivalent to the classification EM algorithm (Celeux and Govaert, 1992, 1995) for a Gaussian mixture model with equal mixing proportions, and a scalar multiple of the identity matrix as a common covariance matrix.

Bayesian Mixture Modeling using Gaussian Distributions

Chan *et al.* (2008) proposed a Bayesian approach of representing cell populations with Gaussian components in a mixture model. Explicitly, observation $\mathbf{y}_i$ in an FCM dataset is modeled as

$$f(\mathbf{y}_i | w_1, \ldots, w_K, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K, \boldsymbol{\Gamma}_1, \ldots, \boldsymbol{\Gamma}_K) = \sum_{k=1}^{K} w_k \phi(\mathbf{y}_i | \boldsymbol{\mu}_k, \boldsymbol{\Gamma}_k^{-1}), \tag{1.12}$$

where $\phi(\cdot|\boldsymbol{\mu}_k, \boldsymbol{\Gamma}_k^{-1})$ is the multivariate Gaussian density with mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Gamma}_k^{-1}$, and w_k is the mixing proportion. The parameters in Eq.(1.12) are assumed to follow a conjugate prior distribution:

$$\begin{aligned}(w_1, \ldots, w_K) &\sim \mathrm{D}(\alpha_1, \ldots, \alpha_K) \\ \boldsymbol{\mu}_k &\sim \mathrm{N}(\mathbf{m}_k, \lambda_k \boldsymbol{\Gamma}_k^{-1}) \\ \boldsymbol{\Gamma}_k &\sim \mathrm{W}(r_k, \mathbf{V}_k)\end{aligned}$$

where $\mathrm{D}(\cdot)$ and $\mathrm{W}(\cdot)$ denote the Dirichlet and Wishart distributions respectively. Parameter estimation is performed using Gibbs sampling described in Lavine and West (1992). The number of mixture components is selected via the Bayesian Information Criterion (Schwarz, 1978). Attempts to resolve the inadequacy of Gaussian distributions to asymmetric components are made by merging components which share a common local mode in the fitted mixture distribution. Nevertheless, Gaussian mixture models are known to be vulnerable to the presence of outliers, which are frequently observed in FCM data. Moreover, considering the large size of FCM datasets, the use of MCMC techniques for parameter estimation requires an enormous amount of computational time. In FCM analysis usually involving a large number of samples, time is of the essence.

Mixture Modeling using Skew t Distributions

Another model-based clustering approach of automating the gating analysis was proposed by Pyne *et al.* (2009), who addressed the issues of asymmetric cell populations and outliers with the skew t distributions (Sahu *et al.*, 2003). Each component in the mixture is modeled by a p-dimensional skew t distribution with the following density:

$$f(\mathbf{y}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\delta}, \nu) = 2\ \varphi(\mathbf{y}|\boldsymbol{\mu}, \boldsymbol{\Omega}, \nu)\ T\left(\frac{\beta}{\sigma}\sqrt{\frac{\nu+p}{\nu+\eta}}\ \middle|\ \nu\right), \tag{1.13}$$

where $\varphi(\cdot|\boldsymbol{\mu}, \boldsymbol{\Omega}, \nu)$ is the multivariate t density with mean $\boldsymbol{\mu}$ ($\nu > 1$), variance $\nu(\nu-2)^{-1}\boldsymbol{\Omega}$ ($\nu > 2$) and degrees of freedom ν, $T(\cdot|\nu)$ is the univariate

standard t distribution function with ν degrees of freedom, $\boldsymbol{\Omega} = \boldsymbol{\mu}+\boldsymbol{\delta}\boldsymbol{\delta}^T$, $\beta = \boldsymbol{\delta}^T\boldsymbol{\Omega}^{-1}(\mathbf{y}-\boldsymbol{\mu})$, $\sigma^2 = 1-\boldsymbol{\delta}^T\boldsymbol{\Omega}^{-1}\boldsymbol{\delta}$, and $\eta = (\mathbf{y}-\boldsymbol{\mu})^T\boldsymbol{\Omega}^{-1}(\mathbf{y}-\boldsymbol{\mu})$. A stochastic representation of the distribution is given by

$$\mathbf{Y} = \boldsymbol{\mu} + \mathbf{U} + \boldsymbol{\delta}|Z|, \tag{1.14}$$

where

$$\begin{aligned}
\mathbf{U} &\sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma}/\tau) \\
Z &\sim \mathrm{N}(0, 1/\tau) \\
\tau &\sim \mathrm{Ga}(\nu/2, \nu/2)
\end{aligned}$$

and $\mathbf{U}, Z$ and τ are all independently distributed. Deduced from Eq.(1.14), extensions of the Monte Carlo EM algorithm (Wei and Tanner, 1990) have been developed for parameter estimation complicated by analytically intractable quantities in the E step (Lin, 2009).

Bibliography

Affymetrix Manual (2001). *Affymetrix Microarray Suite User Guide Version 5.0.* Santa Clara, CA.

Bagwell, C. B. (2004). DNA histogram analysis for node-negative breast cancer. *Cytometry Part A*, 58A(1):76–78.

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

Banfield, J. D. and Raftery, A. E. (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 49:803–821.

Behr, M. A., Wilson, M. A., Gill, W. P., Salamon, H., Schoolnik, G. K., Rane, S., and Small, P. M. (1999). Comparative genomics of BCG vaccines by whole-genome DNA microarray. *Science*, 284(5419):1520–1523.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B*, 57(1):289–300.

Bolton, D. L. and Roederer, M. (2009). Flow cytometry and the future of vaccine development. *Expert Review of Vaccines*, 8(6):779–789.

Braylan, R. C. (2004). Impact of flow cytometry on the diagnosis and characterization of lymphomas, chronic lymphoproliferative disorders and plasma cell neoplasias. *Cytometry Part A*, 58A(1):57–61.

Callow, M. J., Dudoit, S., Gong, E. L., Speed, T. P., and Rubin, E. M. (2000). Microarray expression profiling identifies genes with altered expression in HDL-deficient mice. *Genome Research*, 10:2022–2029.

Celeux, G. and Govaert, G. (1992). A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics and Data Analysis*, 14(3):315–332.

Celeux, G. and Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern Recognition*, 28(5):781–793.

Chan, C., Feng, F., Ottinger, J., Foster, D., West, M., and Kepler, T. B. (2008). Statistical mixture modeling for cell subtype identification in flow cytometry. *Cytometry Part A*, 73A:693–701.

Crick, F. (1970). Central dogma of molecular biology. *Nature*, 227:561–563.

Debouck, C. and Goodfellow, P. N. (1999). DNA microarrays in drug discovery and development. *Nature Genetics*, 21(1 Suppl):48–50.

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

Efron, B. (2004). Large-scale simultaneous hypothesis testing: the choice of a null hypothesis. *Journal of the American Statistical Association*, 99:96–104.

Efron, B., Tibshirani, R., Storey, J. D., and Tusher, V. (2001). Empirical Bayes analysis of a microarray experiment. *Journal of the American Statistical Association*, 96(456):1151–1160.

Fraley, C. and Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631.

Gelfand, A. E. and Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85:398–409.

Gottardo, R., Raftery, A. E., Yeung, K. Y., and Bumgarner, R. E. (2006). Bayesian robust inference for differential gene expression in microarrays with multiple samples. *Biometrics*, 62(1):10–18.

Hengel, R. L. and Nicholson, J. K. (2001). An update on the use of flow cytometry in HIV infection and AIDS. *Clinics in Laboratory Medicine*, 21(4):841–856.

Illoh, O. C. (2004). Current applications of flow cytometry in the diagnosis of primary immunodeficiency diseases. *Archives of Pathology and Laboratory Medicine*, 128(1):23–31.

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

Kendziorski, C., Newton, M., Lan, H., and Gould, M. N. (2003). On parametric empirical Bayes methods for comparing multiple groups using replicated gene expression profiles. *Statistics in Medicine*, 22:3899–3914.

Krutzik, P. O., Irish, J. M., Nolan, G. P., and Perez, O. D. (2004). Analysis of protein phosphorylation and cellular signaling events by flow cytometry: techniques and clinical applications. *Clinical Immunology*, 110(3):206–221.

Lavine, M. and West, M. (1992). A Bayesian method for classification and discrimination. *Canadian Journal of Statistics*, 20:451–461.

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

Lin, T. I. (2009). Robust mixture modeling using multivariate skew t distributions. *Statistics and Computing*, (In press).

Lizard, G. (2007). Flow cytometry analyses and bioinformatics: interest in new softwares to optimize novel technologies and to favor the emergence of innovative concepts in cell research. *Cytometry Part A*, 71A:646–647.

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

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In LeCam, L. and Neyman, J., editors, *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, Berkeley. University of California Press.

Mandy, F. F. (2004). Twenty-five years of clinical flow cytometry: AIDS accelerated global instrument distribution. *Cytometry Part A*, 58A(1):55–56.

McLachlan, G. and Peel, D. (2000). *Finite Mixture Models*. Wiley-Interscience, New York.

Newton, M. C., Kendziorski, C. M., Richmond, C. S., Blattner, F. R., and Tsui, K. W. (2001). On differential variability of expression ratios: improving statistical inference about gene expression changes from microarray data. *Journal of Computational Biology*, 8:37–52.

Orfao, A., Ortuno, F., de Santiago, M., Lopez, A., and San Miguel, J. (2004). Immunophenotyping of acute leukemias and myelodysplastic syndromes. *Cytometry Part A*, 58A(1):62–71.

Pyne, S., Hu, X., Wang, K., Rossin, E., Lin, T.-I., Maier, L. M., Baecher-Allan, C., McLachlan, G. J., Tamayo, P., Hafler, D. A., De Jager, P. L., and Mesirov, J. P. (2009). Automated high-dimensional flow cytometric data analysis. *Proceedings of the National Academy of Sciences of the United States of America*, 106(21):8519–8524.

Sahu, S. K., Dey, D. K., and Branco, M. D. (2003). A new class of multivariate skew distributions with applications to Bayesian regression. *Canadian Journal of Statistics*, 31(2):129–150.

Schena, M., Shalon, D., Davis, R. W., and Brown, P. O. (1995). Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270:467–470.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6:461–464.

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

Titterington, D. M., Smith, A. F. M., and Makov, U. E. (1985). *Statistical Analysis of Finite Mixture Distributions*. Wiley, Chichester, UK.

Tusher, V., Tibshirani, R., and Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences of the United States of America*, 98(9):5116–5121.

Wei, G. C. G. and Tanner, M. A. (1990). A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85:699–704.

Wu, Z., Irizarry, R. A., Gentleman, R., Martinez-Murillo, F., and Spencer, F. (2004). A model-based background adjustment for oligonucleotide expression arrays. *Journal of the American Statistical Association*, 99:909–917.

Chapter 2

Flexible Empirical Bayes Models for Differential Gene Expression*

2.1 Introduction

As a natural development following the success of genome sequencing, DNA microarray technology has emerged for the sake of exploring the functioning of genomes (Schena *et al.*, 1995). By exploiting the ability of a single-strand nucleic acid molecule to hybridize to a complementary sequence, researchers can simultaneously measure the expression levels of thousands of genes within a cell. A common task with microarray is to determine which genes are differentially expressed under two different conditions.

In recent years, there has been a considerable amount of work on the detection of differentially expressed genes. An early statistical treatment can be found in Chen *et al.* (1997). A common approach is to test a hypothesis for each gene using variants of t or F-statistics and then try to correct for multiple testing (Efron *et al.*, 2001; Tusher *et al.*, 2001; Dudoit *et al.*, 2002). Due to the small number of replicates, variation in gene expression can be poorly estimated. Baldi and Long (2001) and Tusher *et al.* (2001) suggested using a modified t statistic where the denominator has been regularized by adding a small constant to the gene specific variance estimate. Similar to an empirical Bayes approach this results in shrinkage of the em-

* A version of this chapter has been published. Lo, K. and Gottardo, R. (2007). Flexible empirical Bayes models for differential gene expression. *Bioinformatics*, 23(3):328–335.

pirical variance estimates towards a common estimate. Lönnstedt and Speed (2002) proposed an empirical Bayes normal mixture model for gene expression data, which was later extended to the two condition case by Gottardo *et al.* (2003) and to more general linear models by Smyth (2004) and Cui *et al.* (2005), though Smyth (2004) and Cui *et al.* (2005) did not use mixture models but simply empirical Bayes normal models for variance regularization. In each case, the authors derived explicit gene specific statistics and did not consider the problem of estimating p the proportion of differentially expressed genes. Newton *et al.* (2001) developed a method for detecting changes in gene expression in a single two-channel cDNA slide using a hierarchical gamma-gamma (GG) model. Kendziorski *et al.* (2003) extended this to replicate chips with multiple conditions, and provided the option of using a hierarchical lognormal-normal (LNN) model. Both models are implemented in an **R** package called `EBarrays` (Empirical Bayes microarrays) and from now on we use the name EBarrays to refer to the methodology. Both EBarrays model specifications rely on the assumption of a constant coefficient of variation across genes. In this chapter, we extend both models by releasing this assumption and introduce EM type algorithms for parameter estimation, thus extending the work of Lönnstedt and Speed (2002) and Gottardo *et al.* (2003) as well.

The structure of this chapter is as follows. The extended forms of the two EBarrays hierarchical models and the estimation procedures are presented in Section 2.2. In Section 2.3, the performance of the extended models is examined on three experimental datasets and compared to five other baseline and commonly used methods. Section 2.4 presents a simulation study to further compare our empirical Bayes approach to the other methods. Finally, in Section 2.5 we discuss our results and possible extensions.

2.2 A Bayesian Framework for Identifying Differential Expression

2.2.1 A Hierarchical Model for Measured Intensities

In a typical microarray experiment, two conditions are compared for gene expression. Let us denote by X_{gr} and Y_{gr} the intensities of gene g from the rth replicate in the two conditions respectively. Measurements between the two conditions are assumed to be independent. The proposed model is an extension of the EBarrays framework (Newton *et al.*, 2001; Kendziorski *et al.*, 2003). Extensions to the original two types of model formulation are considered in turn below.

The Extended Gamma-Gamma Model

Here, a Gamma distribution is used to model the measured intensities of a given gene. Explicitly, the probability density of X_{gr} (resp. Y_{gr}) with shape and rate parameters a_g and θ_{gx} (resp. θ_{gy}) is given by

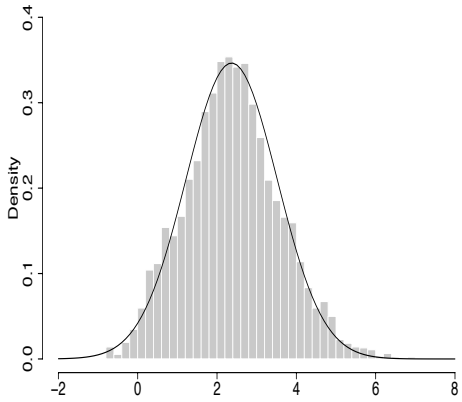
$$p(x|a_g, \theta_{gx}) = \frac{1}{\Gamma(a_g)} \theta_{gx}^{a_g} x^{a_g - 1} \exp(-x\theta_{gx}) \quad \text{for } x > 0. \tag{2.1}$$

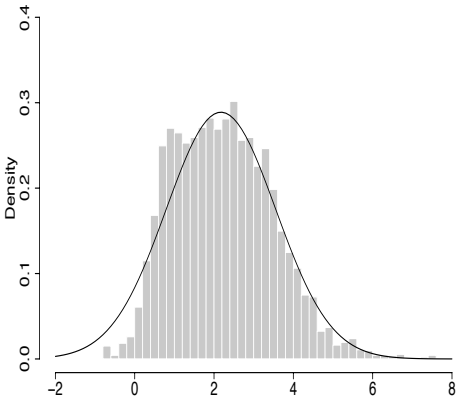
To borrow strength across genes we assume an exchangeable Gamma(a_0, ν) prior for the rate parameters, and a Lognormal(η, ξ) prior for the shape parameters. The Gamma prior is used for simplicity as it is conjugate to the sampling distribution (Newton *et al.*, 2001) while the Lognormal prior is suggested by a histogram plot of the empirical shape parameters estimated by the method of moments (See Figure 2.1). The hyperparameters a_0, ν, η and ξ are assumed unknown and will be estimated as part of our approach.

The proposed model extends the EBarrays GG model by placing a prior on the shape parameter. In the original GG model, the shape parameter a was assumed to be constant and common to all genes whereas now it is gene specific. However, strength is borrowed across genes through the prior distribution. By "borrowing strength", we mean that information from all genes is used when estimating a_g, which comes from the hyperparmeters

(a) HIV-1A

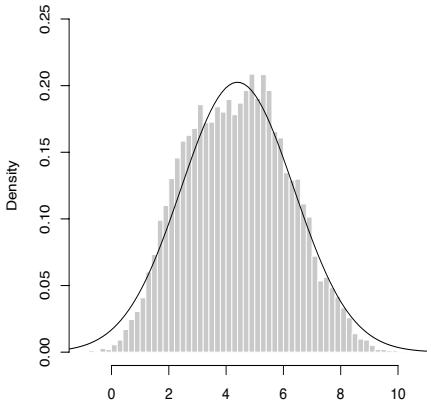
(b) HIV-1B





(c) HGU95A, gcRMA

(d) HGU133A, gcRMA



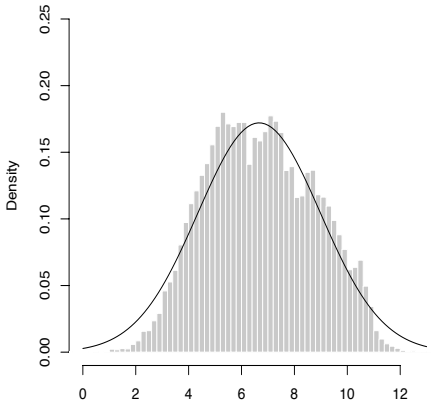


Figure 2.1: Histograms of robust empirical estimates of a_g's with fitted Lognormal density curves shown on a log scale under the extended Gamma-Gamma modeling framework. The hyperparameters ξ and η are estimated using a robust version of the method of moments. The graphs for the HGU95A and HGU133A spike-in data are for one of the comparisons made between two array groups.

through the prior.

The Extended Lognormal-Normal Model

The second formulation is an extension of the EBarrays LNN framework. The intensities are assumed to be lognormally distributed, i.e., the log-transformed intensities are from a normal distribution, and we write $\log X_{gr} \sim \mathrm{N}(\mu_{gx}, \tau_{gx}^{-1})$ and $\log Y_{gr} \sim \mathrm{N}(\mu_{gy}, \tau_{gy}^{-1})$ respectively. A conjugate prior is imposed on the mean μ_{gx} (resp. μ_{gy}) and precision τ_{gx} (resp. τ_{gy}). Explicitly, we set $\mu_{gx}|\tau_{gx} \sim \mathrm{N}(m, k\tau_{gx}^{-1})$ and $\tau_{gx} \sim \mathrm{Gamma}(\alpha, \beta)$ respectively. In the original LNN model, the precision τ was assumed to be constant and common to all genes. Our proposed formulation extends the EBarrays model by releasing the assumption of a constant coefficient of variation $\sqrt{\exp(\tau^{-1}) - 1}$, which is equivalent to the assumption of a constant variance τ^{-1} on the log scale. Note that our proposed formulation is also the framework of Gottardo *et al.* (2003). However, we use an EM based algorithm to estimate the unknown parameters, including the proportion of differentially expressed genes.

On assuming a prior on both μ_{gx} (resp. μ_{gy}) and τ_{gx} (resp. τ_{gy}) common to all genes, strength is borrowed across genes through both means and variances of the distributions when making inferences. Again, we mean that information from all genes is used when estimating both μ_{gx} (resp. μ_{gy}) and τ_{gx} (resp. τ_{gy}). In particular, this is essential for variances — due to the small number of replicates variance estimates can be very noisy. Similar ideas have been used in Smyth (2004) and Cui *et al.* (2005), where the authors concentrated on variance regularization.

2.2.2 A Mixture Model for Differential Expression

We use a mixture model to identify differentially expressed genes. We assume that *a priori* $\theta_{gx} = \theta_{gy}$ (resp. $\mu_{gx} = \mu_{gy}$) with probability $1 - p$ and $\theta_{gx} \neq \theta_{gy}$ (resp. $\mu_{gx} \neq \mu_{gy}$) with probability p. For the latter case, the model specification is just as stated in Section 2.2.1, while the former case is modeled through setting the gene-specific parameters common to both

conditions.

Let us denote by z_g the indicator variable equal to one if there is real change in expression for gene g and zero otherwise. Then one can define the posterior probability of change, $\Pr(z_g = 1|\mathbf{x}_g, \mathbf{y}_g, p, \boldsymbol{\psi})$, where $\mathbf{x}_g = (x_{g1}, x_{g2}, \ldots, x_{gR_x})'$ and $\mathbf{y}_g = (y_{g1}, y_{g2}, \ldots, y_{gR_y})'$ and $\boldsymbol{\psi}$ denotes the vector of unknown hyperparameters. Applying the Bayes rule, we obtain

$$\begin{aligned}\hat{z}_g &= \Pr(z_g = 1|\mathbf{x}_g, \mathbf{y}_g, p, \boldsymbol{\psi}) \\ &= \frac{p\ p_A(\mathbf{x}_g, \mathbf{y}_g|\boldsymbol{\psi})}{p\ p_A(\mathbf{x}_g, \mathbf{y}_g|\boldsymbol{\psi}) + (1-p)p_0(\mathbf{x}_g, \mathbf{y}_g|\boldsymbol{\psi})},\end{aligned} \tag{2.2}$$

where $p_A(\mathbf{x}_g, \mathbf{y}_g|\boldsymbol{\psi})$ and $p_0(\mathbf{x}_g, \mathbf{y}_g|\boldsymbol{\psi})$ denote the joint marginal density of the measured intensities of gene g under both the alternative (differential expression) and null (no differential expression) models respectively given $\boldsymbol{\psi}$. The marginal density for the extended LNN model can be computed explicitly and is given in Appendix A.1. For the extended GG model only θ_g can be integrated out, and the corresponding "conditional" marginal density is given in Appendix A.1. In the next section we describe an approximate estimation procedure to deal with this difficulty.

2.2.3 Parameter Estimation using the EM-algorithm

Here we start with the extended LNN model as the estimation procedure is straightforward. The vector of unknown parameters $\boldsymbol{\Phi} = (\boldsymbol{\psi}', p)'$, where $\boldsymbol{\psi} = (m, k, \alpha, \beta)'$, can be estimated by maximizing the integrated likelihood using the EM-algorithm (Dempster *et al.*, 1977). The estimation of p is important since it calibrates the posterior probability of change for multiple testing, as seen in Eq.(2.2). Such estimation is also part of some multiple testing procedure such as Storey's q-value (Storey, 2003). Estimation of the parameter p can be difficult (Smyth, 2004; Bhowmick *et al.*, 2006), and as suggested by Newton *et al.* (2001) we place a Beta$(2, 2)$ prior over p, which avoids numerical issues when p gets close to 0 or 1. Given the large number of genes, the prior on p has essentially no effect on the final estimation, and thus on the number of genes called differentially expressed.

Treating the z_g's as missing data, the complete-data log-likelihood is given by

$$l_c(\mathbf{\Phi}|\mathbf{x},\mathbf{y},\mathbf{z}) = \sum_g \Big[z_g \log\ p_A(\mathbf{x}_g,\mathbf{y}_g|\boldsymbol{\psi}) + (1-z_g)\log\ p_0(\mathbf{x}_g,\mathbf{y}_g|\boldsymbol{\psi}) + (1+z_g)\log\ p + (2-z_g)\log(1-p)\Big]. \tag{2.3}$$

During the E-step, the expectation is obtained by replacing z_g by $\hat{z}_g$ as given by Eq.(2.2) while the M-step consists of maximizing the conditional expectation with respect to the parameter vector $\mathbf{\Phi} = (\boldsymbol{\psi}', p)'$. At convergence, the estimated parameters can be substituted into Eq.(2.2) to compute the posterior probability of change for each gene.

Because the prior of the extended GG model is not conjugate to the sampling distribution, only the marginal density conditional on a_g is analytically available for each gene. We refer to it as the conditional marginal density. To incorporate information about the prior for the a_g's, we propose to estimate the hyperparameters η and ξ beforehand through an empirical Bayes approach using the method of moments (see Appendix A.2 for details), and add $\log[\pi(a_g|\eta,\xi)]$ to the log conditional density as a penalty term. Again, treating the z_g's as missing data, the corresponding modified complete-data log-likelihood can be written as

$$\tilde{l}_c(\mathbf{\Phi}|\mathbf{x},\mathbf{y},\mathbf{z}) = \sum_g \Big\{ z_g \log\ p_A(\mathbf{x}_g,\mathbf{y}_g|\boldsymbol{\psi},a_g) + (1-z_g)\log\ p_0(\mathbf{x}_g,\mathbf{y}_g|\boldsymbol{\psi},a_g) + (1+z_g)\log(p) + (2-z_g)\log(1-p) + \log\pi(a_g|\eta,\xi)\Big\}, \tag{2.4}$$

where $\boldsymbol{\psi} = (a_0,\nu)'$. The vector of parameters to be estimated becomes $\mathbf{\Phi} = (a_1, a_2, \ldots, a_G, \boldsymbol{\psi}', p)'$.

Similar to the extended LNN model, we can use the EM algorithm to maximize the modified marginal likelihood. During the E-step, to obtain the conditional expectation of the modified complete-data log-likelihood, z_g in Eq.(2.4) is replaced by $\hat{z}_g$ as in Eq.(2.2). The M-step consists of maximizing

$\tilde{l}_c$ given the current z_g's. Such maximization can be difficult given the high dimensionality of $\mathbf{\Phi}$ and here we suggest to exploit the conditional structure of the model during the maximization step, namely that given $\boldsymbol{\psi}$ and p, the genes are conditionally independent and each a_g can be maximized over separately. Let us split the unknown parameters into two groups, namely, $\mathbf{\Phi}_1 = (a_1, a_2, \ldots, a_G)'$ (gene-specific shape parameters) and $\mathbf{\Phi}_2 = (\boldsymbol{\psi}', p)'$ (global parameters). Then the M-step would consist of iteratively maximizing over $\mathbf{\Phi}_1$ given $\mathbf{\Phi}_2$ and $\mathbf{\Phi}_2$ given $\mathbf{\Phi}_1$. Here, we decided to maximize over $\mathbf{\Phi}_1$ only once during the first iteration to reduce the computational burden, and then take EM-iterations with respect to $\mathbf{\Phi}_2$ only until convergence. It turns out that the estimates obtained were very similar to the ones obtained when maximizing over both $\mathbf{\Phi}_1$ and $\mathbf{\Phi}_2$, while significantly reducing the computing time.

Details about the estimation of (η, ξ) and initialization of the EM algorithm can be found in Appendix A.3.

2.3 Application to Experimental Data

2.3.1 Data Description

To illustrate our methodology we use three publicly available microarray datasets: one cDNA experiment and two Affymetrix spike-in experiments. All three have the advantage that in each case the true state (differentially expressed or not) of all or some of the genes is known.

The HIV-1 Data

The expression levels of 4608 cellular RNA transcripts were measured one hour after infection with human immunodeficiency virus type 1 (HIV-1) using four replicates on four different slides. 13 HIV-1 genes have been included in the set of RNA transcripts to serve as positive controls, i.e., genes known in advance to be differentially expressed. Meanwhile, 29 non-human genes have also been included and act as negative controls, i.e., genes known to be not differentially expressed. Another dataset was obtained by repeating

the four aforementioned experiments but with an RNA preparation different from that for the first dataset. For easy reference, we label the two datasets as HIV-1A and HIV-1B respectively. See van't Wout *et al.* (2003) for more details of the HIV-1 data. The data were lowess normalized using a global lowess normalization step (Yang *et al.*, 2002).

The HGU95A Spike-In Data

This dataset was obtained from a spike-in study by Affymetrix used to develop and validate the MAS 5.0 (Affymetrix Manual, 2001) platform. The concentrations of 14 spiked-in human gene groups in 14 groups of HGU95A GeneChip© arrays were arranged in a Latin square design. The concentrations of the 14 groups in the first array group are $0, 0.25, 0.5, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512$ and 1024 pM respectively. Each subsequent array group rotates the spike-in concentrations by one group such that each human gene was spiked-in at a particular concentration level on exactly one array group, and each concentration level came with exactly one spiked-in gene group in each array group. There are three technical replicates in each array group. The third array group has been removed from the analysis as one of its replicates was missing. We use a set of 16 spiked-in genes in our list in recognition of the extras reported by Hsieh *et al.* (2003) and Cope *et al.* (2004). Analysis is performed on each set of probe summary indices computed using gcRMA (Wu *et al.*, 2004), RMA (Irizarry *et al.*, 2003a), MAS 5 and dChip (Li and Wong, 2001) respectively.

The HGU133A Spike-In Data

This dataset was obtained from another spike-in study done with HGU133A arrays. A total of 42 spiked-in genes were organized in 14 groups, and the concentrations used were $0, 0.125, 0.25,\ 0.5, 1, 2, 4, 8, 16, 32, 64, 128, 256$ and 512 pM. The arrangement of the spike-in concentrations was similar to the Latin square design stated above. Again, there are three technical replicates in each array group. For more information see Irizarry *et al.* (2003b). In addition to the original 42, we claim that another 20 genes

should also be included in the spiked-in gene list as they consistently show significant differential expression across the array groups in the exploratory data analysis. Similar observations have been made by Sheffler *et al.* (2005). Moreover, the probe sets of three genes contain probe sequences exactly matching those for the spiked-ins. These probes should be hybridized by the spike-ins as well. As a result, our expanded spiked-in gene list contains 65 entries in total.

2.3.2 Results

We compare our proposed methods – extended GG (eGG) and extended LNN (eLNN) models – to five other methods, namely, EBarrays GG and LNN models, the popular Significance Analysis of Microarrays (SAM) (Tusher *et al.*, 2001), Linear Models for Microarray data (LIMMA) (Smyth, 2004), and a fully Bayesian approach named BRIDGE (Gottardo *et al.*, 2006a). The results have been organized in Tables 2.1–2.3.

In the analysis of the HIV-1 data, we obtain the number of genes called differentially expressed (DE) for each method. Among those genes called

Table 2.1: Analysis of differential expression with the HIV-1 data.

(a) HIV-1A

Method	DE	TP*	FP*
GG	24	13	0
LNN	18	13	1
eGG	13	13	0
eLNN	14	13	0
LIMMA	13	13	0
SAM	13	13	0
BRIDGE	14	13	0

(b) HIV-1B

Method	DE	TP*	FP*
GG	18	11	1
LNN	18	11	1
eGG	12	11	0
eLNN	12	11	0
LIMMA	11	11	0
SAM	13	11	0
BRIDGE	11	11	0

The FDR is controlled at 0.1.

* The numbers of TP and FP are based on the controls, namely, the 13 (resp. 12 in the second experiment) HIV-1 and the 29 non-human genes of which the states are known in advance. They do not represent the true numbers of TP and FP in the entire data.

DE, we look at the number of true positives (TP), i.e., genes known to be DE in advance, and the number of false positives (FP), i.e., genes known to be not DE. Gottardo *et al.* (2006b) showed that one of the HIV genes, which was expected to be highly differentially expressed had a very small estimated log ratio and did not properly hybridize in the second experiment (HIV-1B). We removed the corresponding gene from the list of known differentially expressed genes. Thus there are 13 genes known to be DE in the first experiment and 12 in the second. To compare the performance between the seven methods, we intend to control the false discovery rate (FDR) at a fixed level of 0.1. The FDR cutoffs can be selected using a direct posterior probability calculation as described in Newton *et al.* (2004). For the HIV-1A dataset, when the FDR is controlled at 0.1, all methods can identify the 13 positive controls. Meanwhile, EBarrays LNN has made one FP. Similar result is observed when the HIV-1B dataset is considered. All methods detect 11 out of the 12 positive controls but both versions of EBarrays (GG and LNN) have made one FP. Concluded from the HIV-1 datasets, along with LIMMA, SAM and BRIDGE our proposed eGG and eLNN methods appear to perform the best as they recognize the most positive controls and do not get any FP.

For the HGU95A spike-in data, after removing the array group with one missing replicate, we have a set of 13 array groups. To evaluate the different methods we compare the first array group to the other array groups, leading to 12 comparisons. Since dChip may return negative probe summary indices, which cannot be processed by the aforementioned methods, those genes with negative summary indices were filtered out. This excluded 5.5 spike-ins on average. This time, since we know the actual status of each gene, we can check the true FDR of each method against the desired FDR. In addition, we look at the number of false negatives (FN) as a power assessment.

Unlike the results on the HIV-1 data, SAM does not show a competitive performance. A large number of FN (>11) have been observed with SAM for both gcRMA and RMA summary indices, considering that there are only 16 entries in our spiked-in gene list. eLNN and LIMMA have the actual FDR closest to the desired FDR in general, though they have a relatively large

Table 2.2: Analysis of differential expression with the HGU95A spike-in data.

(a) gcRMA

Method	FN	FDR
GG	2.42	0.22
LNN	1.83	0.22
eGG	1.58	0.28
eLNN	5.83	0.09
LIMMA	4.33	0
SAM	11.25	0.05
BRIDGE	3.6	0.06

(b) RMA

Method	FN	FDR
GG	2.42	0.28
LNN	2.42	0.25
eGG	2.25	0.2
eLNN	3.25	0.15
LIMMA	3.08	0.08
SAM	12.58	0.23
BRIDGE	2.33	0.17

(c) MAS 5

Method	FN	FDR
GG	6.5	0.7
LNN	5.42	0.84
eGG	4.33	0.53
eLNN	7.08	0.26
LIMMA	5.58	0.27
SAM	5.83	0.27
BRIDGE	12.08	0

(d) dChip

Method	FN	FDR
GG	3.25	0.7
LNN	3.58	0.74
eGG	2.83	0.43
eLNN	6.08	0.34
LIMMA	4.83	0.3
SAM	3	0.45
BRIDGE	4.00	0.34

The FDR is controlled at 0.1. The values of FN and FDR shown are the averages across the 12 comparisons.

number of FN cases regarding MAS 5 and dChip summary indices. The actual FDRs for EBarrays GG and LNN methods are too high compared to the other methods, and our proposed extended versions have lowered the rates by a wide margin while keeping relatively small FN rates.

The HGU133A spike-in data have a set of 14 array groups, and therefore 13 comparisons have been made. A total of 14 out of 65 spiked-in genes on average have been filtered from the analysis with dChip due to negative summary indices. The relative performance of the six methods is similar to that for the HGU95A data. It is worth mentioning that eGG is the only method that can sustain the FN cases to a low number for all four types

Table 2.3: Analysis of differential expression with the HGU133A spike-in data.

(a) gcRMA

Method	FN	FDR
GG	5.85	0.2
LNN	5.92	0.2
eGG	6.46	0.23
eLNN	13.08	0.07
LIMMA	10.38	0.08
SAM	22.23	0.12
BRIDGE	6.01	0.11

(b) RMA

Method	FN	FDR
GG	4.38	0.14
LNN	4.46	0.13
eGG	5.23	0.06
eLNN	6.69	0.09
LIMMA	6.15	0.03
SAM	17.15	0.1
BRIDGE	4.53	0.08

(c) MAS 5

Method	FN	FDR
GG	15.77	0.89
LNN	15.85	0.87
eGG	9.23	0.59
eLNN	15.77	0.23
LIMMA	13.85	0.31
SAM	13.77	0.28
BRIDGE	18.46	0.25

(d) dChip

Method	FN	FDR
GG	9.31	0.48
LNN	9.69	0.58
eGG	6.69	0.44
eLNN	11.31	0.3
LIMMA	9.38	0.26
SAM	5.08	0.28
BRIDGE	6.92	0.51

The FDR is controlled at 0.1. The values of FN and FDR shown are the averages across the 13 comparisons.

of probe summary indices, though its FDR is higher than the desired one. SAM has considerably more FN cases than the other methods for gcRMA and RMA, while its FDR is close to the desired one. Similarly, eLNN and LIMMA exhibit good FDR performance but with better FN rates. Again, the FDRs for EBarrays GG and LNN methods are at quite a high level, while their extended versions (eGG and eLNN) have significantly reduced the rates while keeping relatively small FN rates.

2.4 Simulation Studies

2.4.1 Data Generation

We now use a series of simulations to study the performance of our empirical Bayes framework under different model specifications compared to the original EBarrays framework and the methods presented in Section 2.3.2. In order to do so, we generated data from the following models: EBarrays GG ($a = 5, a_0 = 0.8, \nu = 15$), EBarrays LNN ($m = 5, \sigma^2 = 2, \tau^{-1} = 0.25$, σ^2 being the variance parameter of the prior of μ_{gx} or μ_{gy}), extended GG ($\eta = 2, \xi = 1, a_0 = 1, \nu = 20$) and extended LNN ($m = 5, k = 12, \alpha = 2, \beta = 0.5$). The values of the parameters are set in the proximity of the estimates from the HIV-1 data. We fixed the number of genes to 500, the number of replicates to three in each group and generated 100 datasets under each of the above models for two different values of $p = \{0.1, 0.2\}$.

2.4.2 Results

The seven methods mentioned in Section 2.3.2 are applied to each simulated dataset to make inference about differential expression. Results are summarized graphically in two ways: a plot of the actual FDR against the desired FDR, and a plot of the number of FP against the number of FN. The curves show the average results across the 100 simulated datasets. For each dataset, results are collected by setting the cutoffs for the posterior probabilities or p-values at different points in turn in detecting differential expression.

As expected, the EBarrays GG and LNN models perform quite poorly compared to the eGG and eLNN models when the variance is not constant and clearly under estimate the FDR (Figures 2.2 and 2.3). On the other hand, the eGG and eLNN models are comparable to EBarrays when the variance is constant, showing that strength borrowing across genes is working well (Figures 2.4 and 2.5). Finally, both GG and eGG (resp. LNN and eLNN) appear to perform relatively well under LNN and eLNN (resp. GG and eGG) model specifications respectively. This confirms previous simula-

tion studies (Kendziorski *et al.*, 2003).

Overall, SAM is not performing very well and tend to under estimate the FDR by a large amount. Meanwhile, LIMMA and BRIDGE consistently show good performance for data generated from the four models, suggesting that they are good candidates for identifying differential expression under a wide variety of settings.

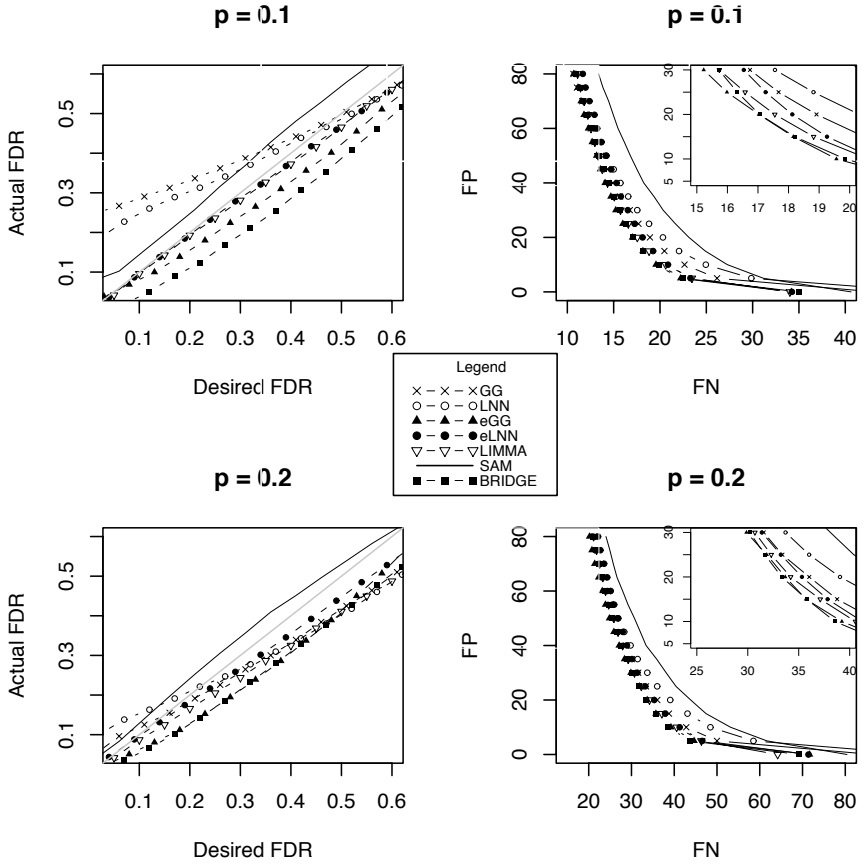


Figure 2.2: Simulation results generated from the extended GG model.

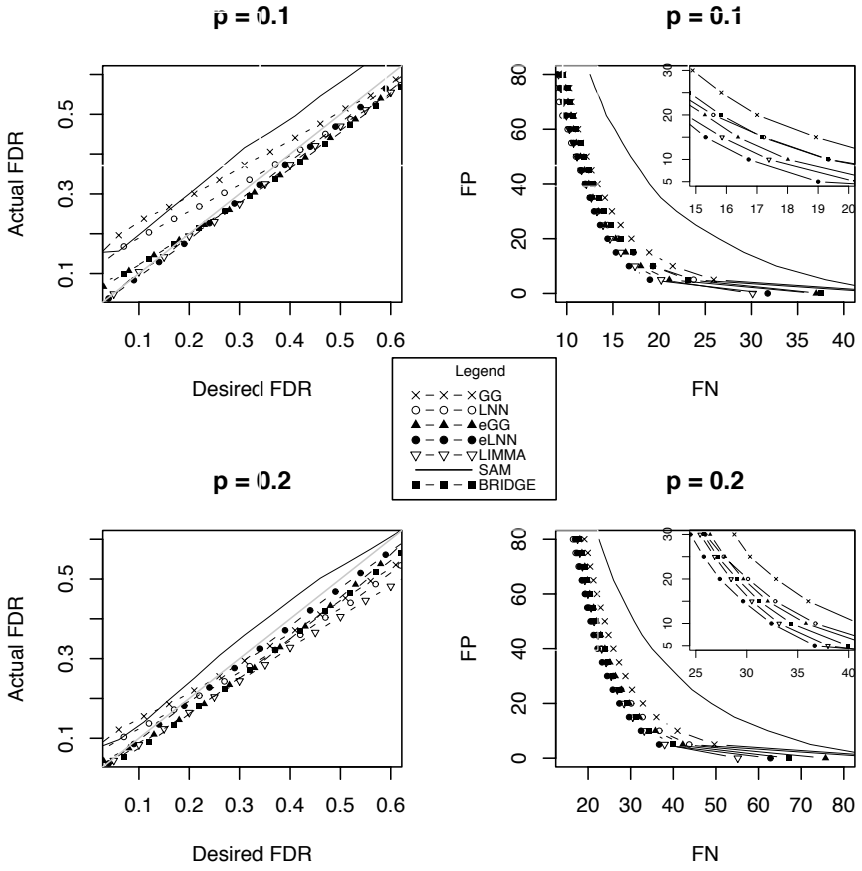


Figure 2.3: Simulation results generated from the extended LNN model.

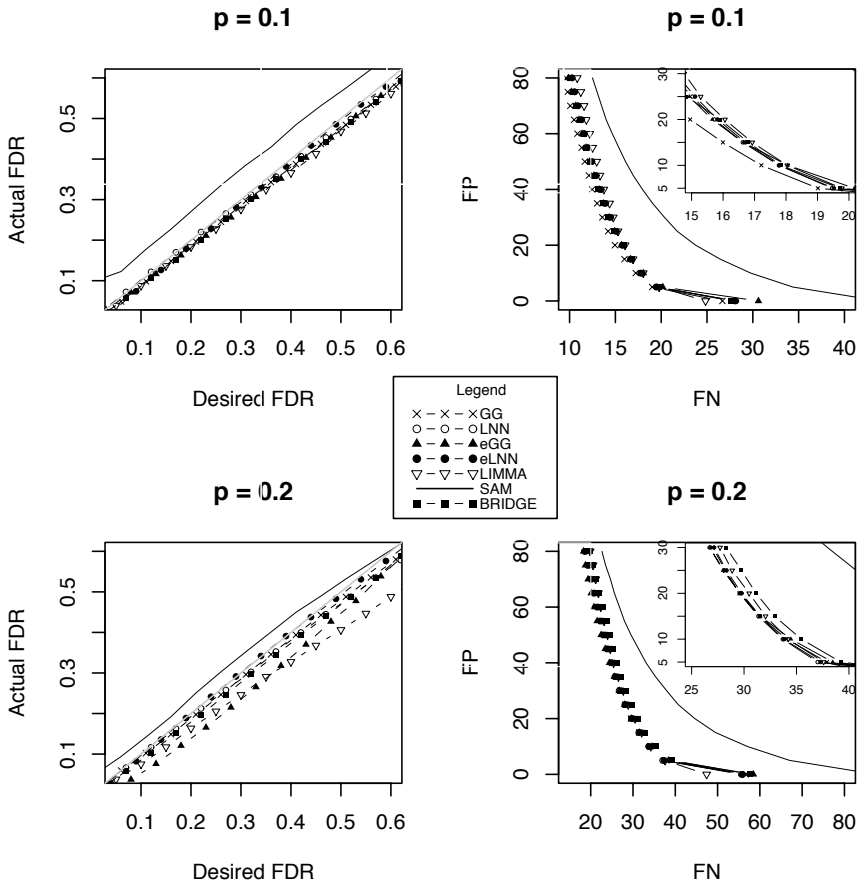


Figure 2.4: Simulation results generated from the EBarrays GG model.

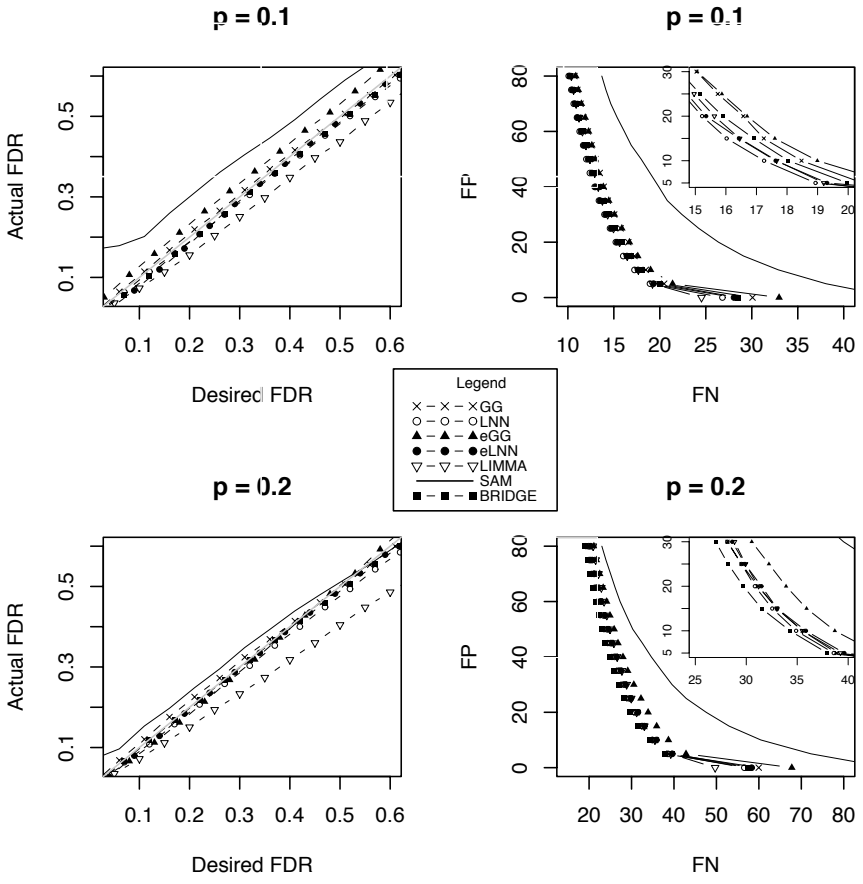


Figure 2.5: Simulation results generated from the EBarrays LNN model.

2.5 Discussion

We have extended the EBarrays empirical Bayes framework for differential gene expression by releasing the constant coefficient of variation assumption, and introducing two algorithms that can be used for parameter estimation. Using both experimental and simulated data we have shown that the extended framework clearly improves the original framework. In addition, it appears that the eLNN model performs better than the eGG one as shown with the spike-in data, and that it is comparable to BRIDGE, a more computational fully Bayesian approach. This is not the case for the original EBarrays framework, where the GG model generally performs better. This confirms previous findings of Gottardo *et al.* (2006a) and suggests that EBarrays GG is more robust to the model misspecification of a constant coefficient of variation compared to the LNN formulation. However, when the EBarrays model formulations are extended and the constant coefficient of variation assumption is released, the LNN model seems more appropriate.

In spite of the complications accompanying the model enhancements relative to the original EBarrays framework, the proposed methodology remains to be highly competitive in terms of processing time. In the analysis with the HGU133A data of >20000 genes, it takes about five minutes to complete the eGG or eLNN analysis of one comparison between the two array groups each with three replicates on the **R** platform.

In this chapter, we have compared our approach with five alternatives, but there are many other methods for detecting differentially expressed genes with gene expression data. We chose these five because they are either obvious baseline methods or widely used; they are also representative of other methods. More comparisons between statistical tests can be found in Cui and Churchill (2003). Among explicit adjustments for multiple testing, we considered the FDR control method as it is interpretable under each method.

For simplicity and ease of comparison, we assumed that we were in a situation with only two conditions of interest. However, the methodology could easily be extended to the multiple condition case (Kendziorski *et al.*, 2003) or more complex ANOVA-type designs (Cui and Churchill, 2003; Smyth, 2004).

Bibliography

Affymetrix Manual (2001). *Affymetrix Microarray Suite User Guide Version 5.0.* Santa Clara, CA.

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

Bhowmick, D., Davison, A. C., Goldstein, D. R., and Ruffieux, Y. (2006). A Laplace mixture model for identification of differential expression in microarray experiments. *Biostatistics*, 7(4):630–641.

Chen, Y., Dougherty, E. R., and Bittner, M. L. (1997). Ratio-based decisions and the quantitative analysis of cDNA microarray images. *Journal of Biomedical Optics*, 2(4):364–374.

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

Cui, X. and Churchill, G. A. (2003). Statistical tests for differential expression in cDNA microarray experiments. *Genome Biology*, 4(4):210.

Cui, X., Hwang, J. T., Qiu, J., Blades, N. J., and Churchill, G. A. (2005). Improved statistical tests for differential gene expression by shrinking variance components estimates. *Biostatistics*, 6(1):59–75.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39:1–38.

Dudoit, S., Yang, Y. H., Callow, M. J., and Speed, T. P. (2002). Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. *Statistica Sinica*, 12(1):111–139.

Efron, B., Tibshirani, R., Storey, J. D., and Tusher, V. (2001). Empirical Bayes analysis of a microarray experiment. *Journal of the American Statistical Association*, 96(456):1151–1160.

Gottardo, R., Pannucci, J. A., Kuske, C. R., and Brettin, T. (2003). Statistical analysis of microarray data: a Bayesian approach. *Biostatistics*, 4:597–620.

Gottardo, R., Raftery, A. E., Yeung, K. Y., and Bumgarner, R. E. (2006a). Bayesian robust inference for differential gene expression in microarrays with multiple samples. *Biometrics*, 62(1):10–18.

Gottardo, R., Raftery, A. E., Yeung, K. Y., and Bumgarner, R. E. (2006b). Quality control and robust estimation for cDNA microarrays with replicates. *Journal of the American Statistical Association*, 101(473):30–40.

Hsieh, W. P., Chu, T. M., and Wolfinger, R. D. (2003). Who are those strangers in the Latin square? In Johnson, K. F. and Lin, S. M., editors, *Methods of Microarray Data Analysis III: Papers from CAMDA'02*, pages 199–208. Kluwer, Boston.

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

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

Kendziorski, C., Newton, M., Lan, H., and Gould, M. N. (2003). On parametric empirical Bayes methods for comparing multiple groups using replicated gene expression profiles. *Statistics in Medicine*, 22:3899–3914.

Li, C. and Wong, W. H. (2001). Model-based analysis of oligonucleotide arrays: expression index computation and outlier detection. *Proceedings*

of the National Academy of Sciences of the United States of America, 98:31–36.

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

Newton, M., Noueiry, A., Sarkar, D., and Ahlquist, P. (2004). Detecting differential gene expression with a semiparametric hierarchical mixture method. *Biostatistics*, 5(2):155–176.

Newton, M. C., Kendziorski, C. M., Richmond, C. S., Blattner, F. R., and Tsui, K. W. (2001). On differential variability of expression ratios: improving statistical inference about gene expression changes from microarray data. *Journal of Computational Biology*, 8:37–52.

Schena, M., Shalon, D., Davis, R. W., and Brown, P. O. (1995). Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270:467–470.

Sheffler, W., Upfal, E., Sedivy, J., and Noble, W. S. (2005). A learned comparative expression measure for Affymetrix GeneChip DNA microarrays. *Proceedings of IEEE Computational Systems Bioinformatics Conference*, pages 144–154.

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

Storey, J. D. (2003). The positive false discovery rate: a Bayesian interpretation and the q-value. *Annals of Statistics*, 31:2013–2035.

Tusher, V., Tibshirani, R., and Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences of the United States of America*, 98(9):5116–5121.

van't Wout, A. B., Lehrman, G. K., Mikheeva, S. A., O'Keeffe, G. C., Katze, M. G., Bumgarner, R. E., Geiss, G. K., and Mullins, J. I. (2003). Cellular gene expression upon human immunodeficiency virus type 1 infection of $CD4^+$-T-cell lines. *Journal of Virology*, 77:1392–1402.

Wu, Z., Irizarry, R. A., Gentleman, R., Martinez-Murillo, F., and Spencer, F. (2004). A model-based background adjustment for oligonucleotide expression arrays. *Journal of the American Statistical Association*, 99:909–917.

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

Chapter 3

Flexible Mixture Modeling via the Multivariate t Distribution with the Box-Cox Transformation*

3.1 Introduction

In statistics, model-based clustering (Titterington *et al.*, 1985; Banfield and Raftery, 1993; McLachlan and Peel, 2000; Fraley and Raftery, 2002) is a popular unsupervised approach to look for homogeneous groups of observations. The most commonly used model-based clustering approach is based on finite normal mixture models, which has been shown to give good results in various applied fields, for example, gene expression (Yeung *et al.*, 2001; McLachlan *et al.*, 2002; Pan *et al.*, 2002), image analysis (Wehrens *et al.*, 2004; Fraley *et al.*, 2005; Li *et al.*, 2005), medical diagnosis (Schroeter *et al.*, 1998; Forbes *et al.*, 2006) and astronomy (Kriessler and Beers, 1997; Mukherjee *et al.*, 1998).

However, normal mixture models rely heavily on the assumption that each component follows a normal distribution symmetric in shape, which is often unrealistic. A common remedy for the asymmetry issue is to look for transformations of the data that make the normality assumption more realistic. Box and Cox (1964) discussed the power transformation in the

* A version of this chapter has been submitted for publication. Lo, K. and Gottardo, R. (2009). Flexible Mixture Modeling via the Multivariate t Distribution with the Box-Cox Transformation.

context of linear regression, which has also been applied to normal mixture models (Schork and Schork, 1988; Gutierrez *et al.*, 1995).

Another line of attempts to resolve the asymmetry observed in data is to enhance the flexibility of the normal distribution by introducing skewness. Azzalini (1985) developed a class of univariate skew normal distributions with the introduction of a shape parameter to account for the skewness, which had been put to use in a mixture modeling context by Lin *et al.* (2007b). A multivariate version of the skew normal distributions was first proposed by Azzalini and Dalla Valle (1996), with various generalizations or modifications ensuing. One such modification was found in Sahu *et al.* (2003), who developed a new class of multivariate skew elliptically symmetric distributions with applications to Bayesian regression models, and included the multivariate skew normal distribution as a special case. As opposed to Azzalini and Dalla Valle's (1996) formulation of the skew normal distribution, the correlation structure in that of Sahu *et al.* (2003) is not affected by the introduction of skewness in the sense that independence between elements of a random vector is preserved irrespective of changes in the skewness parameters. The latter formulation was adopted by Lin (2009a), who introduced the multivariate skew normal mixture model and described an ECM algorithm (Meng and Rubin, 1993) for maximum likelihood estimation. However, the implementation of this methodology is extremely computationally intensive. A simplified version of Sahu *et al.*'s (2003) formulation has recently been suggested by Pyne *et al.* (2009), who parameterized skewness in the form of a vector in place of a matrix. As a result of this simplification, the computational complexity of parameter estimation has been reduced considerably.

In addition to non-normality, there is also the problem of outlier identification in mixture modeling. Outliers can have a significant effect on the resulting clustering. For example, they will usually lead to overestimating the number of components in order to provide a good representation of the data (Fraley and Raftery, 2002). If a more robust model is used, fewer clusters may suffice. Outliers can be handled in the model-based clustering framework, by either replacing the normal distribution with a more robust

one (e.g., t; see Peel and McLachlan, 2000; McLachlan and Peel, 2000) or adding an extra component to accommodate the outliers (e.g., uniform; see Schroeter *et al.*, 1998).

Transformation selection and outlier identification are two issues which can have heavy mutual influence (Carroll, 1982; Atkinson, 1988). While a stepwise approach in which transformation is preselected ahead of outlier detection (or vice versa) may be considered, it is unlikely to tackle the problem well in general, as the preselected transformation may be influenced by the presence of outliers. One possible means of handling the two issues simultaneously is through the application of skew t distributions (Azzalini and Capitanio, 2003; Sahu *et al.*, 2003) in mixture modeling. Such an attempt was given by Lin *et al.* (2007a), who proposed a skew t mixture model based on the formulation of Azzalini and Capitanio (2003), but it is confined to the univariate case. Not until recently has a multivariate version of the skew t mixture model come to light. Lin (2009b) and Pyne *et al.* (2009) adopted a similar approach to the case of skew normal in defining the multivariate skew t distribution, thereby simplifying Sahu *et al.*'s (2003) formulation with a vector in place of a skewness matrix.

In view of the aforementioned issues, we propose a unified framework based on mixture models using a new class of skewed distributions, namely, the multivariate t distributions with the Box-Cox transformation, to handle transformation selection and outlier identification simultaneously. The t distribution provides a robust mechanism against outliers with its heavier tails relative to the normal distribution (Lange *et al.*, 1989). The Box-Cox transformation is a type of power transformation, which can bring skewed data back to symmetry, a property of both the normal and t distributions. Along with the introduction of the mixture model using this new class of distributions, we also describe a convenient means of parameter estimation via the EM algorithm (Dempster *et al.*, 1977). Whilst the proposed framework holds a big appeal of being computationally much simpler than mixture modeling using skew t distributions, it performs well in various scenarios compared to a wealth of competing approaches, as shown in subsequent sections of this chapter. A simplified form of our proposed framework has been ap-

plied to flow cytometry, which shows a favorable performance in identifying cell populations (Chapter 4). This chapter presents a comprehensive framework that substantially enriches that previous simplified version, including the selection of component-specific transformations, and the estimation of data outlyingness. In addition, it focuses at the computational development of the proposed methodology, and includes a large-scale comparison with competing approaches such as those using the skew normal or t mixture distributions.

The structure of this chapter is as follows. In Section 3.2 we first introduce the new class of skewed distributions, the multivariate t distributions with the Box-Cox transformation. Then we introduce the mixture model using the proposed distributions, and present details including outlier identification, density estimation and the selection of the number of components. In addition, we describe an EM algorithm (Dempster *et al.*, 1977) to simultaneously handle parameter estimation and transformation selection for our proposed mixture model. In Section 3.3, the performance of the proposed framework is examined on real datasets and compared to a wealth of commonly used approaches. Section 3.4 presents extensive simulation studies to further evaluate our proposed framework relative to the other approaches. Finally, in Section 3.5 we summarize and discuss our findings.

3.2 Methodology

3.2.1 Preliminaries

The Multivariate t Distribution

The multivariate t distribution has found its use as a robust modeling tool in various fields of applied statistics like linear and non-linear regression, time series, and pedigree analysis; see Lange *et al.* (1989) and Kotz and Nadarajah (2004) for examples. The t distribution is applied in place of the normal distribution when the latter fails to offer long enough tails for the error distribution. Formally, a random vector $\mathbf{y}$ of length p is said to follow a p-dimensional multivariate t distribution with mean $\boldsymbol{\mu}$ ($\nu > 1$), covariance

matrix $\nu(\nu-2)^{-1}\mathbf{\Sigma}$ ($\nu > 2$) and ν degrees of freedom if its density function is given by

$$\varphi_p(\mathbf{y}|\boldsymbol{\mu}, \mathbf{\Sigma}, \nu) = \frac{\Gamma(\frac{\nu+p}{2})|\mathbf{\Sigma}|^{-1/2}}{(\pi\nu)^{p/2}\Gamma(\frac{\nu}{2})\{1+(\mathbf{y}-\boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(\mathbf{y}-\boldsymbol{\mu})/\nu\}^{\frac{\nu+p}{2}}}. \qquad (3.1)$$

The degrees of freedom ν may be viewed as a robustness tuning parameter, as it controls the fatness of the tails of the distribution. When $\nu \to \infty$, the t distribution approaches a p-dimensional multivariate normal distribution with mean $\boldsymbol{\mu}$, covariance matrix $\mathbf{\Sigma}$, and density function

$$\phi_p(\mathbf{y}|\boldsymbol{\mu}, \mathbf{\Sigma}) = \frac{1}{(2\pi)^{p/2}|\mathbf{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(\mathbf{y}-\boldsymbol{\mu})\right\}. \qquad (3.2)$$

An account of the development of the maximum likelihood estimation of the multivariate t distribution can be found in Liu and Rubin (1995), Liu (1997) and Peel and McLachlan (2000). The estimation involves the use of the EM algorithm or its variants including the ECM (Meng and Rubin, 1993) and ECME (Liu and Rubin, 1994) algorithms. The crux of these algorithms constitutes the fact that we can parameterize a t distribution using a normal-gamma compound distribution. The degrees of freedom ν may be jointly estimated along with other unknown parameters, or fixed *a priori* when the sample size is small. In the latter case, the setting with $\nu = 4$ has been found to provide good protection against outliers and work well in many applications (see, for example, Lange *et al.*, 1989; Stephens, 2000).

Box-Cox Transformation

The power transformation proposed by Box and Cox (1964) was originally introduced to make data with asymmetric distributions fulfill the normality assumption in a regression model. The Box-Cox transformation of an

observation y is defined as follows:

$$y^{(\lambda)} = \begin{cases} \frac{y^{\lambda}-1}{\lambda} & \lambda \neq 0 \\ \log y & \lambda = 0 \end{cases} \tag{3.3}$$

where λ is referred to as the transformation parameter. Note that this function is defined for positive values of y only. In view of the need to handle negative-valued data in some applications, we adopt a modified version (Bickel and Doksum, 1981) of the Box-Cox transformation which is also defined for negative values:

$$y^{(\lambda)} = \frac{\text{sgn}(y)|y|^{\lambda} - 1}{\lambda}, \qquad \lambda > 0. \tag{3.4}$$

There exist several modified versions of the Box-Cox transformation to handle negative-valued data, for example, the log-shift transformation, which was also proposed in Box and Cox's (1964) paper for the original Box-Cox transformation. The advantage of our choice given by Eq.(3.4) is that, while continuity is maintained across the whole range of the data, it retains the simplicity of the form of the transformation without introducing additional parameters; when all data are positive, it reduces to the original version.

3.2.2 The Multivariate t Distribution with the Box-Cox Transformation

In this subsection, we propose a new class of distributions, namely, the multivariate t distributions with the Box-Cox transformation (tBC), to handle transformation and to accommodate outliers simultaneously. Explicitly, a random vector $\mathbf{y}$ of length p following such a distribution has a density function specified by

$$\varphi_p(\mathbf{y}^{(\lambda)}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu) \cdot |J_p(\mathbf{y}; \lambda)|, \tag{3.5}$$

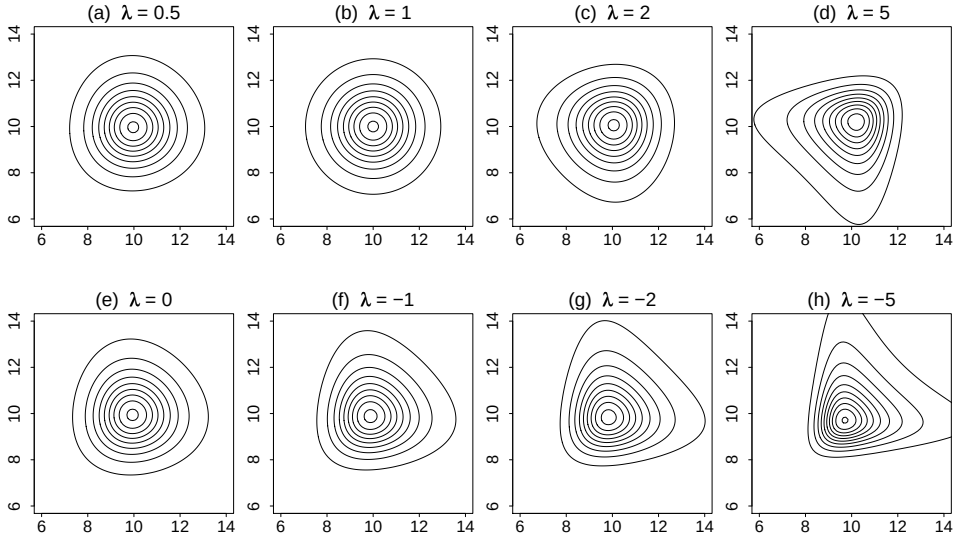


Figure 3.1: Contour plots revealing the shape of bivariate t distributions with the Box-Cox transformation for different values of the transformation parameter. Each distribution has a mean of 10 and unit variance along each dimension. The degrees of freedom parameter is fixed at eight. The values of the transformation parameter λ range from -5 (extremely right-skewed) to 5 (extremely left-skewed).

where $|J_p(\mathbf{y};\lambda)| = |y_1^{\lambda-1} y_2^{\lambda-1} \cdots y_p^{\lambda-1}|$ is the Jacobian induced by the Box-Cox transformation. Equivalently speaking, the random vector $\mathbf{y}$ follows a multivariate t distribution after being Box-Cox transformed. It is difficult to derive the exact mean and variance of the distribution in closed form. However, using first-order Taylor series expansion, approximations for the mean and covariance matrix can be derived. The mean can be approximated by a vector of length p with the j-th element being $\text{sgn}(\lambda\mu_j + 1)\ |\lambda\mu_j + 1|^{1/\lambda}$, and the variance by $\nu/(\nu-2)\ D_p(\boldsymbol{\mu};\lambda)\ \boldsymbol{\Sigma}\ D_p(\boldsymbol{\mu};\lambda)$, where $D_p(\boldsymbol{\mu};\lambda)$ is a diagonal matrix of order p with the j-th diagonal element being $|\lambda\mu_j + 1|^{1/\lambda - 1}$. The various shapes that can be represented by the tBC are shown in Figure 3.1.

Analogous to the case of the t distribution without transformation, the tBC approaches a multivariate normal distribution with the Box-Cox trans-

formation (NBC) when $\nu \to \infty$. In addition, this class of distributions also includes the untransformed version of the multivariate t or normal distribution. The untransformed t or normal distribution is recovered by setting λ in Eq.(3.5) to one, although there is a translation of one unit to the left in each direction on the original scale (due to the term $-1/\lambda$ in Eq.(3.4)).

The flexible class of tBC offers robustness against both outliers and asymmetry observed in data. Comparatively, the t distribution alone is deemed robust in the sense that it offers a mechanism to accommodate outliers. As noted by Lange *et al.* (1989), however, the t distribution is not robust against asymmetric error distributions. When asymmetry is observed, data transformation is desired for the sake of restoring symmetry, and subsequently drawing proper inferences. The introduction of the tBC is therefore in line with Lange *et al.*'s notion.

3.2.3 The Mixture Model of t Distributions with the Box-Cox Transformation

The Model

Making use of the tBC introduced in the last subsection, we now define a G-component mixture model in which each component is described by a tBC. Given data $\mathbf{y}$, with independent p-dimensional observation vectors $\mathbf{y}_i, i = 1, \ldots, n$, the likelihood for the tBC mixture model is given as follows:

$$L(\boldsymbol{\Psi}|\mathbf{y}) = \prod_{i=1}^{n} \sum_{g=1}^{G} w_g \varphi_p(\mathbf{y}_i^{(\lambda_g)}|\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g, \nu_g) \cdot |J_p(\mathbf{y}_i; \lambda_g)|, \qquad \sum_{g=1}^{G} w_g = 1. \quad (3.6)$$

The mixing proportion w_g is the probability that an observation belongs to the g-th component. Estimates of the unknown parameters $\boldsymbol{\Psi} = (\boldsymbol{\Psi}_1, \ldots, \boldsymbol{\Psi}_G)$ where $\boldsymbol{\Psi}_g = (w_g, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g, \nu_g, \lambda_g)$ can be obtained conveniently using an EM algorithm described in the next subsection. Analogous to the case of tBC, the mixture distribution approaches that for an NBC mixture model with $\varphi_p(\cdot|\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g, \nu_g)$ being replaced by $\phi_p(\cdot|\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$ when $\nu_g \to \infty$ for all g. Also, the class of tBC mixture models includes the conventional, untransformed t

or normal mixture model, obtained by fixing $\lambda_g = 1$ for all g. Note that a restricted form of Eq.(3.6) has been previously applied to identify cell populations in flow cytometry data, on setting a global transformation parameter $\lambda = \lambda_g$ and fixing $\nu_g = 4$ for all g (Chapter 4).

Maximum Likelihood Estimation

In this subsection we illustrate how transformation selection can be handled along with parameter estimation simultaneously via an EM algorithm. As in the algorithm for a t mixture model described in Peel and McLachlan (2000), we first define two types of missing data to augment the set of complete data. One is the unobserved component membership $\mathbf{z}_i = (z_{i1}, \ldots, z_{iG})$ with

$$z_{ig} = \begin{cases} 1 & \text{if } \mathbf{y}_i \text{ belongs to the } g\text{-th component} \\ 0 & \text{otherwise} \end{cases}$$

associated with each observation $\mathbf{y}_i$. Each vector $\mathbf{Z}_i$ follows independently a multinomial distribution with one trial and event properties $\mathbf{w} = (w_1, \ldots, w_G)$, denoted as $\mathbf{Z}_i \sim \mathcal{M}_G(1, \mathbf{w})$. Another type of missing data is the weight u_i, coming from the normal-gamma compound parameterization for the t distribution, such that

$$\mathbf{Y}_i | u_i, z_{ig} = 1 \sim \mathrm{N}(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g / u_i) \tag{3.7}$$

independently for $i = 1, \ldots, n$, and $U_i \sim \mathrm{Ga}(\nu_g/2, \nu_g/2)$. The advantage of writing the model in this way is that, conditional upon the U_i's, the sampling errors are again normal but with different precisions, and estimation becomes

a weighted least squares problem. The complete-data log-likelihood becomes

$$
\begin{aligned}
l_c(\boldsymbol{\Psi}|\mathbf{y},\mathbf{z},\mathbf{u}) = \sum_{i=1}^{n}\sum_{g=1}^{G} z_{ig}\Big\{ & \log\Big[w_g\phi_p(\mathbf{y}_i^{(\lambda_g)}|\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g/u_i)\cdot|J_p(\mathbf{y}_i;\lambda_g)|\Big] \\
& + \log \mathrm{Ga}(u_i|\nu_g/2,\nu_g/2)\Big\} \\
= \sum_{i=1}^{n}\sum_{g=1}^{G} z_{ig}\Big\{ & \log w_g - \frac{p}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}_g| \\
& - \frac{u_i}{2}(\mathbf{y}_i^{(\lambda_g)}-\boldsymbol{\mu}_g)^T\boldsymbol{\Sigma}_g^{-1}(\mathbf{y}_i^{(\lambda_g)}-\boldsymbol{\mu}_g) + (\lambda_g-1)\sum_{j=1}^{p}\log|y_{ij}| \\
& + \frac{\nu_g}{2}\log\frac{\nu_g}{2} - \log\Gamma\Big(\frac{\nu_g}{2}\Big) + \frac{\nu_g}{2}(\log u_i - u_i) + \Big(\frac{p}{2}-1\Big)\log u_i\Big\},
\end{aligned}
\tag{3.8}
$$

where $\mathrm{Ga}(\cdot|\cdot)$ is the density function of u_i. The E-step of the EM algorithm involves the computation of the conditional expectation of the complete-data log-likelihood $E_{\boldsymbol{\Psi}}(l_c|\mathbf{y})$. To facilitate this, we need to compute $\tilde{z}_{ig} \equiv E_{\boldsymbol{\Psi}}(Z_{ig}|\mathbf{y}_i)$, $\tilde{u}_{ig} \equiv E_{\boldsymbol{\Psi}}(U_i|\mathbf{y}_i, z_{ig}=1)$ and $\tilde{s}_{ig} \equiv E_{\boldsymbol{\Psi}}(\log U_i|\mathbf{y}_i, z_{ig}=1)$:

$$
\tilde{z}_{ig} \leftarrow \frac{w_g\varphi_p(\mathbf{y}_i^{(\lambda_g)}|\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g,\nu_g)\cdot|J_p(\mathbf{y}_i;\lambda_g)|}{\sum_{k=1}^{G} w_k\varphi_p(\mathbf{y}_i^{(\lambda_k)}|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k,\nu_k)\cdot|J_p(\mathbf{y}_i;\lambda_k)|}, \tag{3.9}
$$

$$
\tilde{u}_{ig} \leftarrow \frac{\nu_g + p}{\nu_g + (\mathbf{y}_i^{(\lambda_g)}-\boldsymbol{\mu}_g)^T{\boldsymbol{\Sigma}_g}^{-1}(\mathbf{y}_i^{(\lambda_g)}-\boldsymbol{\mu}_g)} \tag{3.10}
$$

and

$$
\tilde{s}_{ig} \leftarrow \log\tilde{u}_{ig} + \psi\left(\frac{\nu_g+p}{2}\right) - \log\left(\frac{\nu_g+p}{2}\right), \tag{3.11}
$$

where $\psi(\cdot)$ is the digamma function. Note that, if we assume a global transformation parameter λ, then Eq.(3.9) used to compute $\tilde{z}_{ig}$ is slightly simplified as

$$
\tilde{z}_{ig} \leftarrow \frac{w_g\varphi_p(\mathbf{y}_i^{(\lambda)}|\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g,\nu_g)}{\sum_{k=1}^{G} w_k\varphi_p(\mathbf{y}_i^{(\lambda)}|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k,\nu_k)}. \tag{3.12}
$$

As can be seen in the following, $\tilde{s}_{ig}$ only appears in Eq.(3.18) or Eq.(3.19) for the update of the degrees of freedom ν_g. If we fix ν_g to some predetermined value, then $\tilde{s}_{ig}$ is not needed and Eq.(3.11) can be omitted. Upon plugging $\tilde{z}_{ig}$, $\tilde{u}_{ig}$ and $\tilde{s}_{ig}$ into Eq.(3.8) for z_{ig}, u_i and $\log u_i$ respectively, we obtain the conditional expectation of the complete-data log-likelihood.

In the M-step, we update the parameter estimates with values which maximize the conditional expectation of the complete-data log-likelihood. The mixing proportions are updated with the following formula:

$$\hat{w}_g \leftarrow \frac{n_g}{n}, \tag{3.13}$$

where $n_g \equiv \sum_i \tilde{z}_{ig}$. The estimation of $\boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}_g$ needs to be considered along with the transformation parameter λ_g of the Box-Cox transformation. Closed-form solutions for $\boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}_g$ are available conditional on λ_g as follows,

$$\hat{\boldsymbol{\mu}}_g = \frac{\sum_{i=1}^n \tilde{z}_{ig}\tilde{u}_{ig}\mathbf{y}_i^{(\lambda_g)}}{\sum_{i=1}^n \tilde{z}_{ig}\tilde{u}_{ig}} = h_1(\lambda_g); \tag{3.14}$$

$$\hat{\boldsymbol{\Sigma}}_g = \frac{\sum_{i=1}^n \tilde{z}_{ig}\tilde{u}_{ig}(\mathbf{y}_i^{(\lambda_g)} - \hat{\boldsymbol{\mu}}_g)(\mathbf{y}_i^{(\lambda_g)} - \hat{\boldsymbol{\mu}}_g)^T}{n_g} = h_2(\lambda_g). \tag{3.15}$$

No closed-form solution is available for λ_g, but on substituting $\hat{\boldsymbol{\mu}}_g = h_1(\lambda_g)$ and $\hat{\boldsymbol{\Sigma}}_g = h_2(\lambda_g)$ into the conditional expectation of the complete-data log-likelihood for $\boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}_g$ respectively, the problem reduces to a one-dimensional search of λ_g. Explicitly, the optimization is recast as a one-dimensional root-finding problem of the equation $\partial E_{\boldsymbol{\Psi}}(l_c|\mathbf{y})/\partial\lambda_g = 0$, in which

$$\begin{aligned}
\frac{\partial E_{\boldsymbol{\Psi}}(l_c|\mathbf{y})}{\partial\lambda_g} &= \frac{\partial}{\partial\lambda_g}\sum_{i=1}^n \tilde{z}_{ig}\Bigg\{ -\frac{\tilde{u}_{ig}}{2}\left[(\mathbf{y}_i^{(\lambda_g)} - \boldsymbol{\mu}_g)^T\boldsymbol{\Sigma}_g^{-1}(\mathbf{y}_i^{(\lambda_g)} - \boldsymbol{\mu}_g)\right] \\
&\qquad + (\lambda_g - 1)\sum_{j=1}^p \log|y_{ij}|\Bigg\} \\
&= \sum_{i=1}^n \left[-\tilde{z}_{ig}\tilde{u}_{ig}(\mathbf{y}_i^{(\lambda_g)} - \boldsymbol{\mu}_g)^T\boldsymbol{\Sigma}_g^{-1}\right]\frac{\partial\mathbf{y}_i^{(\lambda_g)}}{\partial\lambda_g} + \sum_{i=1}^n \tilde{z}_{ig}\sum_{j=1}^p \log|y_{ij}|
\end{aligned} \tag{3.16}$$

where $\partial \mathbf{y}_i^{(\lambda_g)}/\partial\lambda_g$ is a vector of length p whose j-th element is $\lambda_g^{-2}\big[\mathrm{sgn}(y_{ij})|y_{ij}|^{\lambda_g}(\lambda_g \log|y_{ij}| - 1) + 1\big]$, and $\boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}_g$ are replaced with $\hat{\boldsymbol{\mu}}_g = h_1(\lambda_g)$ and $\hat{\boldsymbol{\Sigma}}_g = h_2(\lambda_g)$ respectively. The equation may be solved numerically using, for example, Brent's (1973) algorithm. If we assume a global transformation parameter λ instead, the left hand side of the equation to consider is slightly modified from Eq.(3.16) as

$$\frac{\partial E_{\boldsymbol{\Psi}}(l_c|\mathbf{y})}{\partial\lambda} = \sum_{i=1}^{n}\left\{\sum_{g=1}^{G}\left[-\tilde{z}_{ig}\tilde{u}_{ig}(\mathbf{y}_i^{(\lambda)} - \boldsymbol{\mu}_g)^T\boldsymbol{\Sigma}_g^{-1}\right]\right\}\frac{\partial \mathbf{y}_i^{(\lambda)}}{\partial\lambda} + \sum_{i=1}^{n}\sum_{j=1}^{p}\log|y_{ij}|. \tag{3.17}$$

Once a numerical estimate of λ_g has been obtained, we substitute it back into Eqs.(3.14)–(3.15) to update $\boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}_g$ respectively.

To complete the M-step, we need to update the estimate of the degrees of freedom ν_g, unless it is fixed *a priori*. From Eq.(3.8), we see that there are no overlaps between terms involving $(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g, \lambda_g)$ and those involving ν_g. Hence, the incorporation of the Box-Cox transformation does not complicate the estimation of ν_g. Again, since there is no closed-form solution available for ν_g, we turn it into a one-dimensional root-finding problem by considering the equation $\partial E_{\boldsymbol{\Psi}}(l_c|\mathbf{y})/\partial\nu_g = 0$, in which

$$\begin{aligned}\frac{\partial E_{\boldsymbol{\Psi}}(l_c|\mathbf{y})}{\partial\nu_g} &= \frac{\partial}{\partial\nu_g}\sum_{i=1}^{n}\tilde{z}_{ig}\left\{\frac{\nu_g}{2}\log\frac{\nu_g}{2} - \log\Gamma\Big(\frac{\nu_g}{2}\Big) + \frac{\nu_g}{2}\left(\tilde{s}_{ig} - \tilde{u}_{ig}\right)\right\} \\ &\propto n_g\left\{\log\frac{\nu_g}{2} + 1 - \psi\left(\frac{\nu_g}{2}\right)\right\} + \sum_{i=1}^{n}\tilde{z}_{ig}(\tilde{s}_{ig} - \tilde{u}_{ig}).\end{aligned} \tag{3.18}$$

If we assume a global degrees of freedom $\nu = \nu_g$ for all g, the derivative $\partial E_{\boldsymbol{\Psi}}(l_c|\mathbf{y})/\partial\nu$ is given by

$$\frac{\partial E_{\boldsymbol{\Psi}}(l_c|\mathbf{y})}{\partial\nu} \propto n\left\{\log\frac{\nu}{2} + 1 - \psi\left(\frac{\nu}{2}\right)\right\} + \sum_{i=1}^{n}\sum_{g=1}^{G}\tilde{z}_{ig}(\tilde{s}_{ig} - \tilde{u}_{ig}). \tag{3.19}$$

Alternatively, to improve the convergence, we may exploit the advantage of the ECME algorithm (Liu and Rubin, 1994) and switch to update ν by

optimizing the constrained actual log-likelihood function:

$$\hat{\nu} \leftarrow \arg\max_{\nu} \left\{ \sum_{i=1}^{n} \log \left(\sum_{g=1}^{G} w_g \varphi_p(\mathbf{y}_i^{(\lambda_g)} | \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g, \nu) \cdot |J_p(\mathbf{y}_i; \lambda_g)| \right) \right\}. \quad (3.20)$$

Apart from an intuitive sense that a faster convergence is expected on disregarding the information of the parameter estimates obtained from the previous iteration (which is carried over by the conditional expectation of the complete-data log-likelihood otherwise) as well as considering the actual likelihood instead of its approximation, it also saves a little computational burden by circumventing the computation of $\tilde{s}_{ig}$.

The EM algorithm alternates between the E and M-steps until convergence. The quantity $\tilde{z}_{ig}$ may be interpreted as the posterior probability that observation $\mathbf{y}_i$ belongs to the g-th component. The maximum *a posteriori* configuration results from assigning each observation to the component associated with the largest $\tilde{z}_{ig}$ value. The uncertainty corresponding to each assignment may be conveniently quantified as $1 - \max_g \tilde{z}_{ig}$ (Bensmail *et al.*, 1997).

Outlier Identification

Just like the case of $\tilde{z}_{ig}$, the introduction of $\tilde{u}_{ig}$ does not only facilitate the implementation of the EM algorithm, but also aids in the interpretation of the final estimated model. As seen from Eqs.(3.14)–(3.15), $\tilde{u}_{ig}$ serves as the weight in the weighted least squares estimation of $\boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}_g$. It holds a negative relationship with the Mahalanobis distance $(\mathbf{y}_i^{(\lambda_g)} - \boldsymbol{\mu}_g)^T \boldsymbol{\Sigma}_g{}^{-1} (\mathbf{y}_i^{(\lambda_g)} - \boldsymbol{\mu}_g)$ between $\mathbf{y}_i$ and $\boldsymbol{\mu}_g$ on the transformed scale, as given by Eq.(3.10). Hence, a small value of $\tilde{u}_{ig}$ would suggest that the corresponding observation is an outlier, and diminish its influence on the estimation of the parameters. In contrast, in the absence of such a mechanism, a normal mixture model is not robust against outliers, as the constraint $\sum_g \tilde{z}_{ig} = 1$ for all i restricts all observations to make equal contributions overall towards parameter estimation.

Exploiting such a mechanism, we may conveniently set up a rule of calling

an observation with the associated $\tilde{u}_{ig}$ value smaller than a threshold, say, 0.5, an outlier. Such a threshold may be selected on a theoretical basis by considering the one-to-one correspondence between $\tilde{u}_{ig}$ and the Mahalanobis distance which follows some standard, known distribution. On noting that

$$(\mathbf{y}_i^{(\lambda_g)} - \boldsymbol{\mu}_g)^T {\boldsymbol{\Sigma}_g}^{-1} (\mathbf{y}_i^{(\lambda_g)} - \boldsymbol{\mu}_g)/p \sim F(p, \nu_g), \tag{3.21}$$

where $\mathbf{y}_i^{(\lambda_g)}$ follows a p-dimensional t distribution with parameters $(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g, \nu_g)$ and $F(\cdot)$ denotes an F distribution, a threshold c for $\tilde{u}_{ig}$ may be determined by considering the desired threshold quantile level α of the distribution stated in Eq.(3.21):

$$c = \frac{\nu_g + p}{\nu_g + pF_{1-\alpha}(p, \nu_g)}, \tag{3.22}$$

where $F_{1-\alpha}(\cdot)$ denotes the α quantile of the F distribution such that $\Pr(F \geq F_{1-\alpha}) = 1 - \alpha$. For instance, if $\nu_g = 4, p = 5$, and the desired threshold quantile level is $\alpha = 0.9$, then the corresponding threshold for $\tilde{u}_{ig}$ is $c = 0.37$ given the 0.9 quantile $F_{0.1}(5, 4) = 4.051$. Any observation with the associated $\tilde{u}_{ig} < 0.37$ will be deemed an outlier.

From Eq.(3.10), we can also see how the degrees of freedom ν_g participates in robustifying the parameter estimation process. A smaller value of ν_g tends to downweight outliers to a greater extent, while a large enough value tends to regress all weights to one, approaching the case of the NBC model. In addition, the upper bound of $\tilde{u}_{ig}$ offers a guide for setting the degrees of freedom $\nu = \nu_g$ for all g, if it is preferred to be fixed in advance. The weight $\tilde{u}_{ig}$ takes a positive value on $(0, 1 + p/\nu_g)$, and for a moderate-valued ν_g, its mean is around one. To avoid a point in the vicinity of the central location of a mixture component from imposing excessive influence on the estimation of parameters, we may set ν accordingly such that the ratio p/ν is maintained at an appropriate level, for example, one to 1.5.

Density Estimation

One advantage of mixture modeling based on the normal distribution is that the marginal distribution for any subset of the dimensions is also normally

distributed with the mean and covariance matrix extracted from the conformable dimensions (Johnson and Wichern, 2002). This favorable property is also observed in the multivariate t distiribution (Liu and Rubin, 1995; Kotz and Nadarajah, 2004), making the estimation of the marginal density for any dimensions available at a very low computational cost. Consider the partition $\mathbf{Y} = (\mathbf{Y}_1, \mathbf{Y}_2)$ as an example. If $\mathbf{Y}$ comes from a multivariate t distribution with ν degrees of freedom and with mean and covariance matrix conformably partitioned as

$$\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \boldsymbol{\mu}_2) \quad \text{and} \quad \frac{\nu}{\nu - 2}\,\boldsymbol{\Sigma} = \frac{\nu}{\nu - 2}\begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix}$$

respectively, then its subset $\mathbf{Y}_1$ will follow a t distribution with mean $\boldsymbol{\mu}_1$, covariance matrix $\nu/(\nu-2)\boldsymbol{\Sigma}_{11}$ and the same ν degrees of freedom. This nice property is easily extended to a t mixture model with more than one component, and, in addition, preserved in our proposed tBC mixture model. One can easily derive the marginal density by extracting the conformable partitions from the means, covariance matrices and the Jacobian. The 90th percentile region of the mixture components shown in Figure 3.2 is produced by these means.

Selecting the Number of Components

When the number of mixture components is unknown, we apply the Bayesian Information Criterion (BIC) (Schwarz, 1978) to guide the selection. The BIC provides a convenient approximation to the integrated likelihood of a model and, in the context of mixture models, is defined as

$$\text{BIC}_G = 2 \log \tilde{L}_G - K_G \log n, \tag{3.23}$$

where $\tilde{L}_G$ is the likelihood value of Eq.(3.6) evaluated at the maximum likelihood estimates of $\boldsymbol{\Psi}$, and K_G is the number of independent parameters for a G-component mixture model. The BIC would then be computed for a range of possible values for G and the one with the largest BIC (or relatively

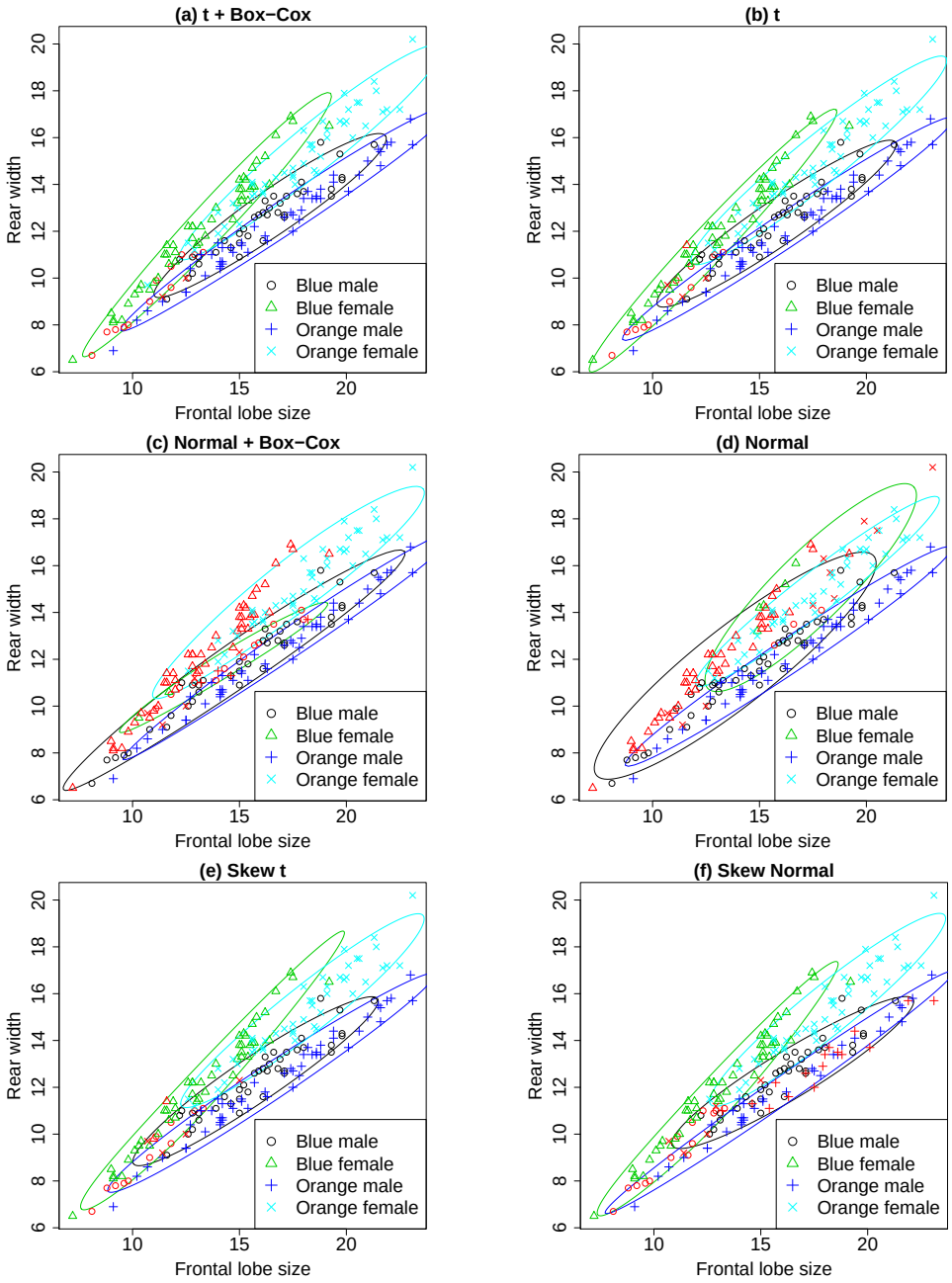


Figure 3.2: Scatterplots revealing the assignment of observations for different models applied to the crabs dataset, projected onto the dimensions of the frontal lobe size and the width of the rear region of the carapace. The solid lines represent the 90th percentile region of the components in the mixture models. The line colors match with the group they are labeled, determined in a way such that the lowest misclassification rate is derived. Misclassified observations are drawn in red, overriding the original colors used to reveal their true group memberships.

close to it) would be selected. Often, the BIC is applied in line with the principle of parsimony, by which we favor a simpler model if it does not incur a downgrade of the modeling performance. Suppose there are two tBC mixture models with G_1 and G_2 components respectively such that $G_1 < G_2$. Under the notion of this principle, we would prefer the simpler model, i.e., the one with G_1 components, unless a very strong evidence of improved performance signified by an increase of >10 (Kass and Raftery, 1995; Fraley and Raftery, 2002) is observed from BIC_{G_1} over BIC_{G_2}.

3.3 Application to Real Data

3.3.1 Data Description

To illustrate our methodology we use the following two real datasets.

The bankruptcy dataset

This dataset was obtained from a study which conducted financial ratio analysis to predict corporate bankruptcy (Altman, 1968). The sample consists of 66 manufacturing firms in the United States, of which 33 are bankrupt and the other 33 solvent. The data collected include the ratio of retained earnings (RE) to total assets, and the ratio of earnings before interest and taxes (EBIT) to total assets. They were derived from financial statements released two years prior to bankruptcy, and statements from the solvent firms during the same period.

The crabs dataset

Measurements in this dataset were collected from a study of rock crabs of genus *Leptograpsus* (Campbell and Mahon, 1974). The sample is composed of 50 crabs for each combination of species (blue and orange color forms) and sex (male and female), resulting in a total of 200 observations. There are five morphological measurements, namely, the frontal lobe size, the width of the rear region of the carapace, the length of the carapace along the midline,

the maximum width of the carapace, and the depth of the body, for each crab.

3.3.2 Results

We compare the performance of six mixture modeling approaches using different mixture distributions, namely, t with the Box-Cox transformation (tBC), t, normal with the Box-Cox transformation (NBC), normal, skew t, and skew normal. Since all observations in the two datasets come with known labels, we can assess and compare the models based on the following two criteria: misclassification rates and the number of components selected.

Classification

We fit the two datasets using the six aforementioned models in turn, on fixing the number of mixture components at the known values, i.e., two for the bankruptcy dataset and four for the crabs dataset. The same initialization strategy is applied to the EM algorithm for all the models. Each time, 10 random partitions are generated, each of which is followed by a few EM runs. The one delivering the highest likelihood value is taken as the initial configuration for the eventual EM algorithm. At convergence of the EM algorithm, misclassification rates, i.e., the proportions of observations assigned to the incorrect group, are computed. Each misclassification rate is determined as the minimum considering all permutations of the labels of the components.

Table 3.1 shows the misclassification rates for the different models. As can be seen, for the bankruptcy dataset, the tBC and NBC mixture models deliver misclassification rates (15.2% and 16.7% respectively) lower than the other methods by a large margin. By taking a graphical inspection of the results, we find that the poor classification performance of the other four methods is due to the inability to resolve the shape of the two groups of observations properly (Figures 3.3(b,d–f)). The challenge likely arises from the scattered group of bankrupt firms, with its most concentrated region located at the upper right corner and in close proximity to the dense group

Table 3.1: Misclassification rates for different models applied to the bankruptcy and crabs datasets.

Model	Bankruptcy	Crabs
tBC	**0.152 (10)**	**0.070 (14)**
t	0.273 (18)	0.075 (15)
NBC	0.167 (11)	0.345 (69)
Normal	0.318 (21)	0.290 (58)
Skew t	0.303 (20)	0.085 (17)
Skew Normal	0.394 (26)	0.175 (35)

The best results are shown in bold. The numbers of misclassified cases are given within parentheses.

of solvent firms. The sensitivity of normal mixture models to outliers is clearly demonstrated in this example: the obvious outlier at the bottom of the scatterplot leads to an excessively sparse component representing the bankrupt group. Consequently, most observations in the bankrupt group have been absorbed by the compact component representing the solvent group. The shapes of the components in the t, skew t and skew normal mixture models are not all the same, but it appears that for all of them the scattered group of bankrupt firms are split into two components with one absorbing a concentration extending to the left and the other to the bottom. In contrast, both the tBC and NBC mixture models provide a nice representation of both groups of observations (Figures 3.3(a,c)). The group of bankrupt firms is resolved quite successfully upon a proper transformation ($\hat{\lambda} \approx 0.5$ for both models) of the observations.

As another means of performance assessment, we look into the location of the misclassified observations in a plot of the ordered uncertainties (Figure 3.4). On observing that the misclassified observations have spread over the entire range of the uncertainties, it suggests that the t, skew t and skew normal mixture models simply provide an incorrect representation of the two groups (Figures 3.4(b,e,f)). The quality of the fit using the tBC and NBC mixture models respectively is confirmed by the corresponding uncertainty plots (Figures 3.4(a,c)). We can see that the observations associated with high uncertainties are also the ones most likely to be misclassified.

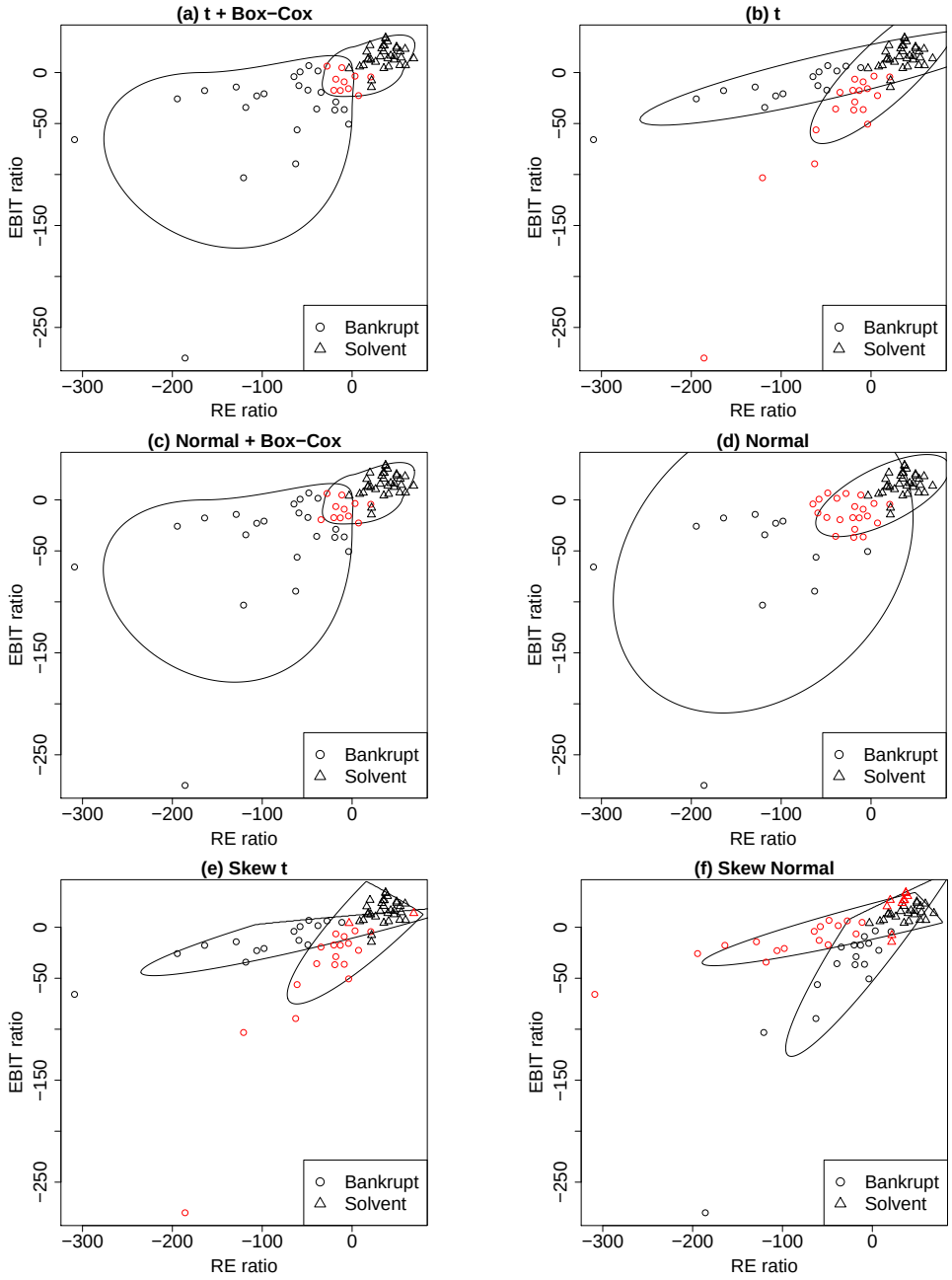


Figure 3.3: Scatterplots revealing the assignment of observations for different models applied to the bankruptcy dataset. The black solid lines represent the 90th percentile region of the components in the mixture models. Misclassified observations are drawn in red.

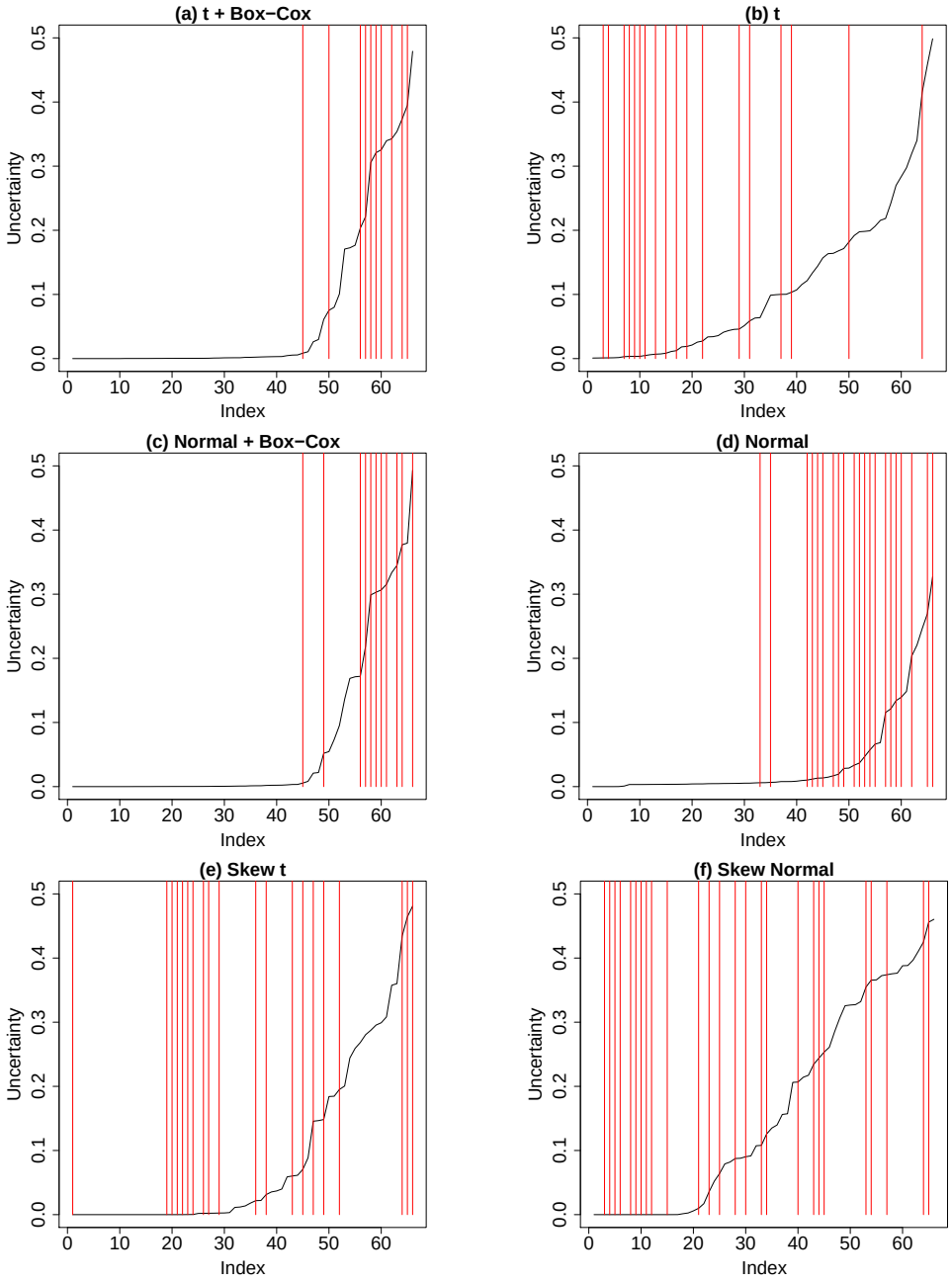


Figure 3.4: Plots revealing the location of misclassified observations relative to the ordered uncertainties of all observations for different models applied to the bankruptcy dataset. Locations of the misclassified observations are marked with red vertical lines.

The results on the crabs dataset once again show that the tBC mixture model delivers the best performance in terms of misclassification rate (7%). It is followed closely by the t (7.5%) and skew t (8.5%) mixture models. Figure 3.2 shows a scatterplot of the crabs dataset projected onto the first two dimensions, namely, the frontal lobe size and the width of the rear region of the carapace. However, unlike the case for the bankruptcy dataset with only two dimensions, a visually clear discrimination of the four groups in the crabs dataset cannot be achieved by projecting the observations onto any two out of the five dimensions. Therefore, we opt for displaying the crabs dataset on its second versus third principal components which provides a good visually discriminating effect. Figures 3.5(a,b) suggest that those few misclassified observations in the tBC and t mixture models are all likely in the overlapping region of neighboring groups, justifying that these models provide a good representation of all the four groups in the dataset. This is further confirmed by a check on the uncertainty plots, in which the misclassified observations are also among the ones with the highest uncertainties (Figures 3.6(a,b)). Meanwhile, from Figures 3.5(c,d,f) we find that, for the poorly performed NBC, normal and skew normal mixture models, misclassified cases are concentrated on one or two of the groups. Figures 3.2(c,d,f) reveal that these models incorrectly split the assignment of the observations from those groups into other components. As expected, these three poorly performing normal-based models have misclassified observations spreading over the entire range in the uncertainty plots (Figures 3.6(c,d,f)).

Selecting the Number of Components

To facilitate this part of analysis, when we apply the aforementioned models, we fit the data and compute the BIC once for each choice of the number of mixture components $G = 1, 2, \ldots, M$, where $M = 6$ for the bankruptcy dataset and $M = 8$ for the crabs dataset. These values are chosen for M because they are well above the true number of groups (two for the bankruptcy dataset and four for the crabs dataset) such that little change in the result is expected when we further increase M; numerical problems may arise when

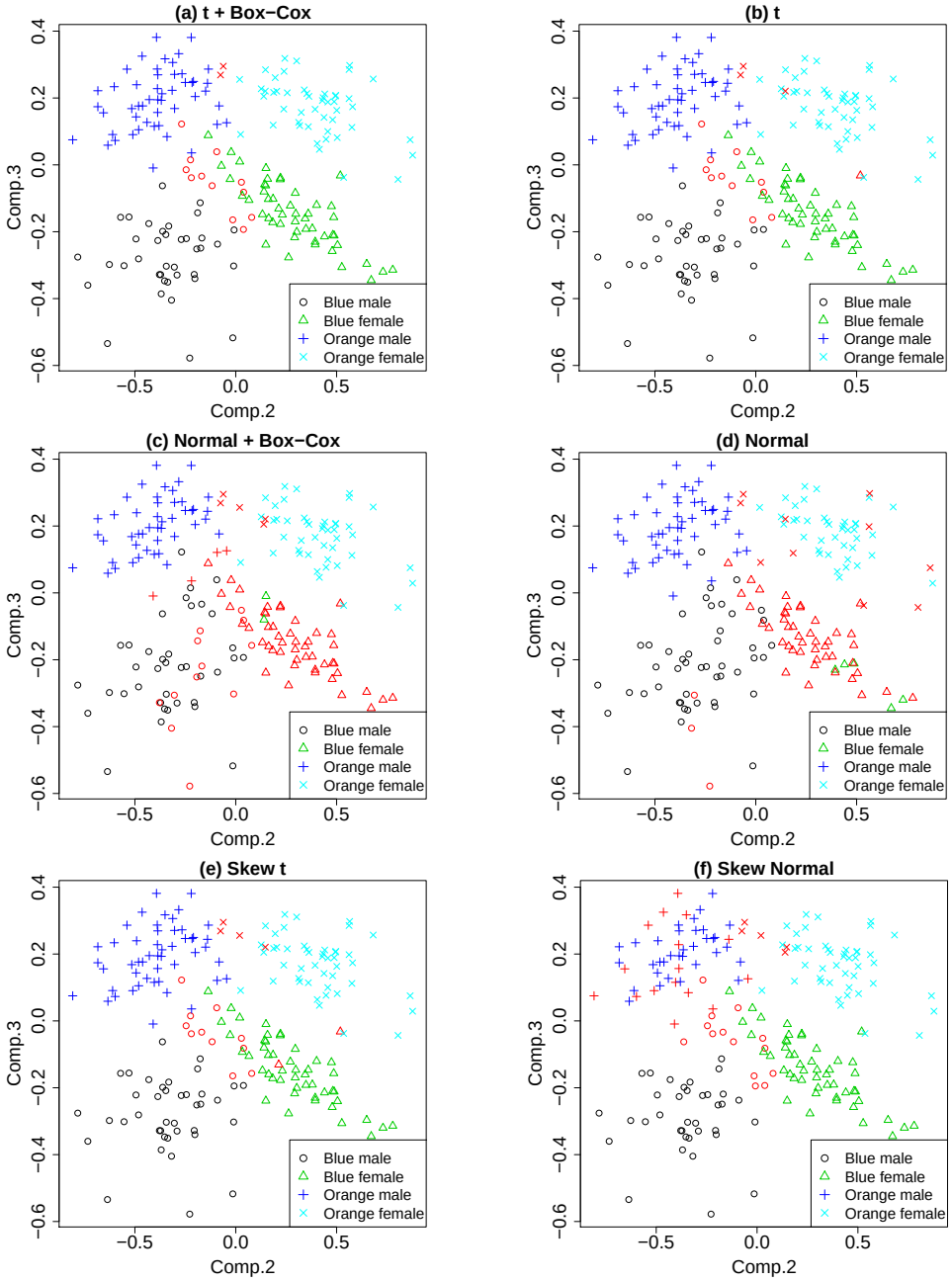


Figure 3.5: Plots revealing the assignment of observations for different models applied to the crabs dataset, displayed via the second and third principal components. Misclassified observations are drawn in red, overriding the original colors used to reveal their true group memberships.

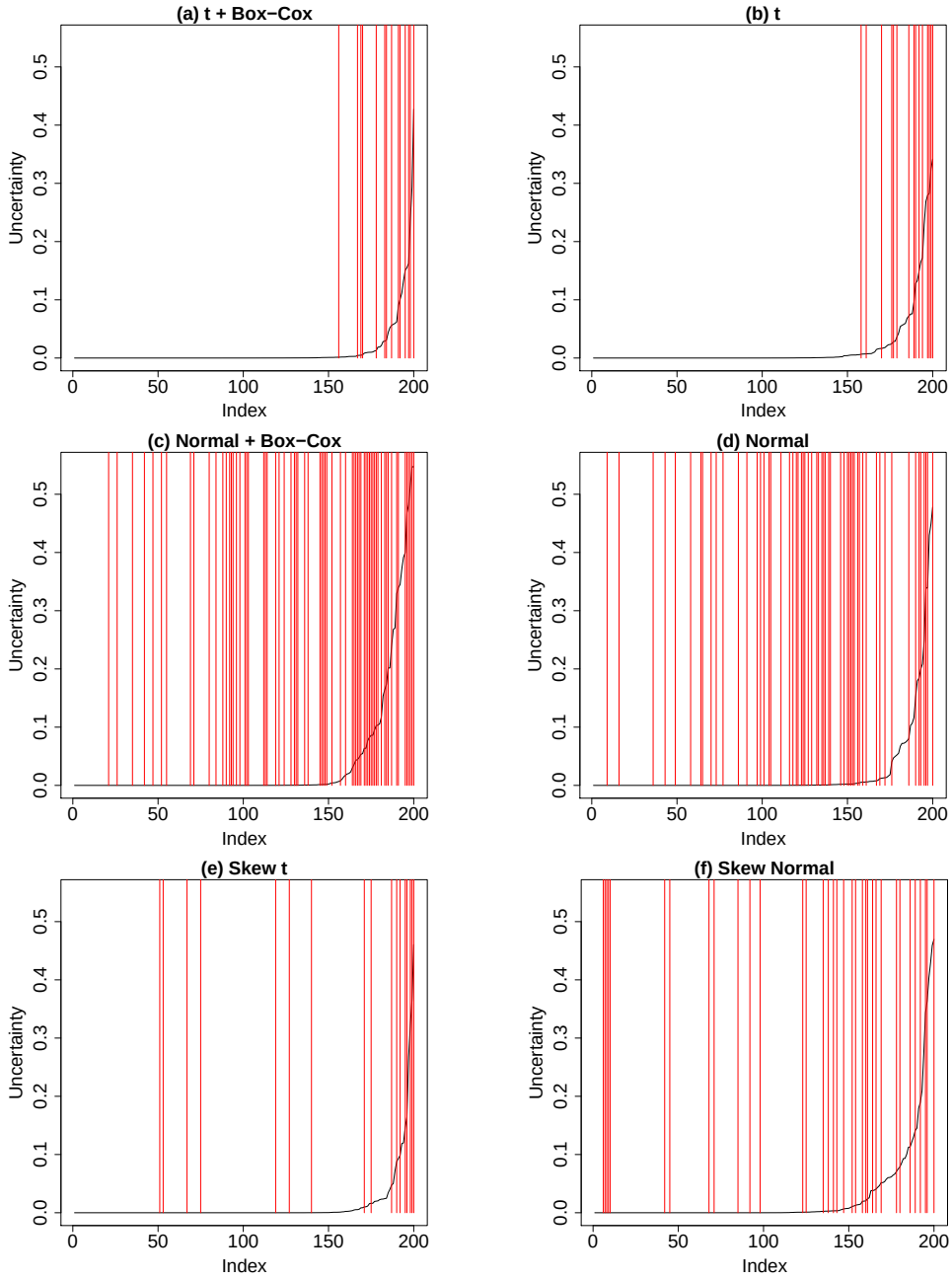


Figure 3.6: Plots revealing the location of misclassified observations relative to the ordered uncertainties of all observations for different models applied to the crabs dataset. Locations of the misclassified observations are marked with red vertical lines.

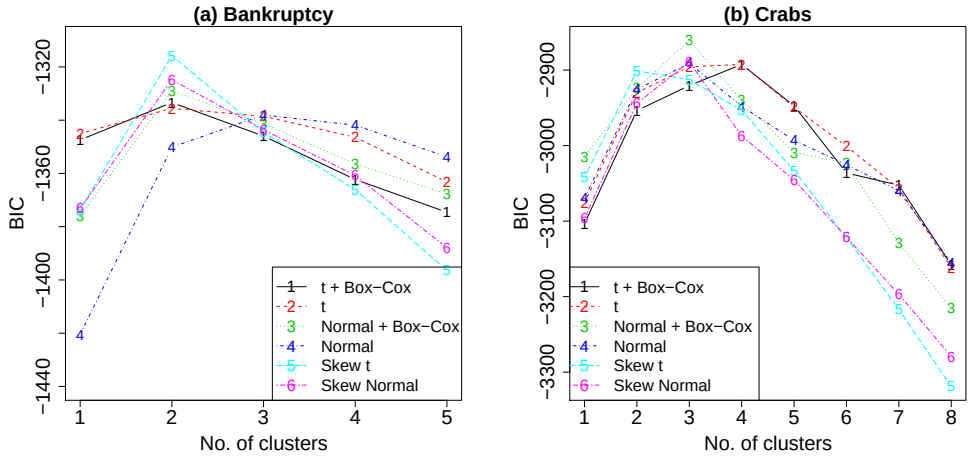


Figure 3.7: Plots of BIC against the number of components for the different models applied to the bankruptcy and crabs datasets.

M is too large, moreover. From the BIC curves shown in Figure 3.7, we observe that one single peak is observed for each modeling choice over the range of the number of components attempted. The number of components at which a peak is observed is deemed optimal by the BIC for the respective model. The BIC has selected the correct number of components (two) for all the mixture models except normal when applied to the bankruptcy dataset (Table 3.2). As to the crabs dataset in which the separation of the groups is less clear-cut, it poses a challenge of selecting the right number of components to most models. Only the tBC and t mixture models have resolved the correct number of components (four) guided by the BIC. This result further confirms with what we have observed in the last subsection that the four-component mixture model using the tBC or t mixture model provides the best representation of the data out of all candidates.

3.4 Simulation Studies

We have conducted a series of simulations to further evaluate the relative performance of our proposed framework to the other approaches presented

Table 3.2: The number of components selected by the BIC for different models applied to the bankruptcy and crabs datasets.

Model	Bankruptcy	Crabs
tBC	**2**	**4**
t	**2**	**4**
NBC	**2**	3
Normal	3	3
Skew t	**2**	2
Skew Normal	**2**	3

The best results are shown in bold.

in Section 3.3.2. The different approaches are evaluated for their sensitivity against model misspecification, using the following two criteria: the accuracy in the assignment of observations, and the accuracy in selecting the number of components.

3.4.1 Data Generation

To facilitate the comparison, we generate data from the following mixture models: tBC, skew t, t and normal. To assess the accuracy in the assignment of observations, two settings of parameter values have been adopted: one taken from the estimates obtained when applying each of the aforementioned models to the bankruptcy dataset, and the other one from the crabs dataset, with the number of components set as the respective known values. As a result, each dataset generated from the bankruptcy setting consists of two components and two dimensions, while that from the crabs setting has four components and five dimensions. For datasets generated under the bankruptcy setting we fix the number of observations at 200, while it is set as 500 for the crabs setting. 100 datasets are generated from each of the aforementioned models under each setting. To study the accuracy in selecting the number of components, we focus at the crabs setting. Pertaining to this criterion, the crabs setting offers a better platform to discriminate the relative performance of the different approaches for its larger number of groups and higher dimensions. 1000 observations are generated from the

Table 3.3: Average misclassification rates for different models applied to datasets generated under the bankruptcy or crabs setting.

		Model used to fit data					
		tBC	t	NBC	Normal	Skew t	Skew Norm.
Model used to generate data under the bankruptcy setting	tBC	**0.075**	0.124	**0.075**	0.126	0.142	0.110
	Skew t	0.100	0.094	0.225	0.189	**0.087**	0.191
	t	**0.109**	**0.109**	0.126	0.134	0.114	0.144
	Normal	0.032	0.032	0.033	**0.030**	0.032	0.031
Model used to generate data under the crabs setting	tBC	**0.011**	0.014	0.057	0.074	0.015	0.016
	Skew t	0.024	0.023	0.046	0.060	**0.020**	0.021
	t	0.024	**0.021**	0.048	0.070	0.023	0.023
	Normal	**0.027**	0.029	0.042	0.038	0.028	0.028

The best results are shown in bold.

crabs setting instead to avoid numerical problems that may arise from small components formed when the number of components is significantly larger than the true number.

3.4.2 Results

Classification

We apply the six approaches presented in Section 3.3.2 in turn to each generated dataset. Model fitting is done by presuming that the number of components is known, i.e., two for the bankruptcy setting and four for the crabs setting. Similar to the way we determined the misclassification rates in our real data analysis, we consider all permutations of the labels of the components and take the lowest one out of all misclassification rates computed. The performance of the different models is compared via the average misclassification rates.

As shown clearly in Table 3.3, our proposed tBC mixture model is the only model that remains the best or close to the best in all the comparisons made. It delivers the lowest misclassification rates under both settings (7.5% and 1.1% respectively) when data are generated from the tBC mixture model. The flexibility of the tBC mixture model is exhibited when we look into its performance in the scenario of model misspecification. It

remains close to the respective true model in those cases, and even delivers the lowest misclassification rate when the true model is t under the bankruptcy setting (10.9%), or normal under the crabs setting (2.7%). Contrariwise, when data are generated from the tBC mixture model, with a lack of mechanisms to handle asymmetric components, both the t and normal mixture models do not perform well. It is worth noting that even the skew t mixture model, which is intended for data departing from symmetry, also performs poorly; the associated misclassification rate is as high as 14.2% under the bankruptcy setting, while that for tBC is only 7.5%. When data are generated from the skew t mixture model, taking advantage of the correct specification the skew t mixture model performs well. The tBC mixture model also shows a competent performance, however. Meanwhile, the skew t mixture model performs satisfactorily when the true mixture model is t or normal. The normal mixture model cannot match the others at all when data are generated from models other than normal, showing its vulnerability to outliers and asymmetric components. In addition, it is interesting to notice that the normal mixture model gives a rather high misclassification rate (3.8%) relative to the levels attained by tBC, t and skew t (2.7%–2.9%) when it itself is the true model for data generation under the crabs setting. It seems that the t-based mixture models are more robust to initialization of the EM algorithm.

Selecting the Number of Components

In this part of study, each time when we apply a model to a dataset generated under the crabs setting, we set the number of components from one up to eight in turn. The number of components is then selected to be the one which delivers the highest BIC. Table 3.4 summarizes the result and gives the 90% coverage intervals of the number of components selected for each model out of the 100 repetitions.

The tBC mixture model selects the correct number of components (four) in the majority of repetitions, even in case of model misspecification. It is the only model that remains to contain only the true number of components in

Table 3.4: 90% coverage intervals of the number of components selected by the BIC for different models applied to datasets generated under the crabs setting.

		Model used to fit data					
		tBC	t	NBC	Normal	Skew t	Skew Norm.
Model used to generate data	tBC	**(4, 4)**	**(4, 4)**	**(4, 4)**	**(4, 4)**	(3, 5)	(3, 5)
	Skew t	**(4, 4)**	**(4, 4)**	(4, 5)	(4, 5)	**(4, 4)**	**(4, 4)**
	t	**(4, 4)**	**(4, 4)**	(4, 5)	(4, 5)	**(4, 4)**	**(4, 4)**
	Normal	**(4, 4)**	(4, 5)	**(4, 4)**	(4, 5)	**(4, 4)**	**(4, 4)**

The best results are shown in bold.

all the 90% coverage intervals. On the other hand, both the skew t and skew normal mixture models fail to distinguish the four groups properly in about 30% of the datasets generated from the tBC mixture model. Besides, both the NBC and normal mixture models, when applied to datasets generated from the t or skew t mixture model, tend to require an additional component to accommodate the data in an excess of outliers.

3.5 Discussion

In this chapter, we have introduced a new class of distributions, the t distributions with the Box-Cox transformation, for mixture modeling. The proposed methodology is in line with Lange *et al.*'s (1989) notion that transformation selection and outlier identification are two issues of mutual influence and therefore should be handled simultaneously. In our real data applications and simulation studies, we have shown the flexibility of this methodology in accommodating asymmetric components in the presence of outliers, and in coping with model misspecification. The vulnerability of the normal-based models to outliers is exposed in the analysis of the crabs dataset, in which the presence of outliers prevents a clear distinction of the four groups. A lack of mechanisms to downsize the influence of remote observations undermines the ability of these approaches to properly locate the cores of the four groups in the dataset. On the other hand, the analysis of the bankruptcy dataset provides a very good example of demonstrating

the importance of incorporating data transformation in clustering. In the absence of a means to accommodate components departing from symmetry, the t mixture model fails to provide a reasonable representation of the data, while the number of groups is known in advance. Our simulation studies have confirmed these findings.

As mentioned in the Introduction, although mixture modeling using our proposed tBC distributions and that using the skew t distributions follow two lines of development with more or less the same aim, our approach has an appeal of being computationally much simpler to implement. As noted in Lin (2009b), difficulties have been encountered in evaluating the conditional expectation of the complete-data log-likelihood in the E-step of the EM algorithm for the skew t mixture model. The objective function cannot be derived in closed form due to the presence of analytically intractable quantities. Numerical techniques for optimization as well as integration need to be employed extensively to update a vast amount of quantities in both the E and M-steps of the algorithm, undermining the computational stability therein. Besides, the parameterization that accounts for skewness in our proposed model originates from the family of power transformations, which is intuitively interpretable. It is less trivial to interpret the skewness vector parameterized in the skew t distribution, however. In addition, as presented in Section 3.2.3, the way to identify outliers using our approach is straightforward and on a theoretical ground. Exploiting the relationship between $\tilde{u}_{ig}$ and the quantile of an F distribution through Eq.(3.22), it is almost costless to proceed with outlier identification once the EM algorithm is completed. On the contrary, when the skew t mixture model is used, we cannot determine such a threshold by recasting it as a known quantity obtained from a standard distribution. In consequence, it demands extra computational effort to identify outliers, especially when the dimension of the data is high. Finally, perhaps most importantly, as demonstrated from our real data applications and simulation studies, the simplicity of the computational implementation of our proposed methodology is not achieved at the expense of the quality of performance. The results have shown that our proposed approach performs as well as that based on the skew t mixture

model, or even slightly better.

An open-source software package that facilitates flow cytometry analysis with the methodology proposed in this chapter has been developed and is available at Bioconductor (Gentleman *et al.*, 2004); see Chapter 5 for details. It is released as an **R** package called `flowClust` and addresses the vast demand for software development from the flow cytometry community. `flowClust` is dedicated to the automated identification of cell populations, and is well integrated into other flow cytometry packages. Meanwhile, we recognize the potential of the proposed methodology in other fields, and the importance of developing a general-purpose tool like MCLUST (Fraley and Raftery, 2002, 2006), the popular software that performs clustering analysis based on normal mixture models. We are going to work on such a general-purpose, standalone software that will serve as a contribution to the general public.

Bibliography

Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *Journal of Finance*, 23(4):589–609.

Atkinson, A. C. (1988). Transformations unmasked. *Technometrics*, 30:311–318.

Azzalini, A. (1985). A class of distributions which includes the normal ones. *Scandinavian Journal of Statistics*, 12:171–178.

Azzalini, A. and Capitanio, A. (2003). Distributions generated by perturbation of symmetry with emphasis on a multivariate skew t-distribution. *Journal of the Royal Statistical Society, Series B*, 65(2):367–389.

Azzalini, A. and Dalla Valle, A. (1996). The multivarite skew-normal distribution. *Biometrika*, 83(4):715–726.

Banfield, J. D. and Raftery, A. E. (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 49:803–821.

Bensmail, H., Celeux, G., Raftery, A. E., and Robert, C. P. (1997). Inference in model-based cluster analysis. *Statistics and Computing*, 7:1–10.

Bickel, P. J. and Doksum, K. A. (1981). An analysis of transformations revisited. *Journal of the American Statistical Association*, 76(374):296–311.

Box, G. E. P. and Cox, D. R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society, Series B*, 26:211–252.

Brent, R. (1973). *Algorithms for Minimization without Derivatives*. Prentice-Hall, Englewood Cliffs, NJ.

Campbell, N. A. and Mahon, R. J. (1974). A multivariate study of variation in two species of rock crab of the genus Leptograpsus. *Australian Journal of Zoology*, 22(3):417–425.

Carroll, R. J. (1982). Prediction and power transformations when the choice of power is restricted to a finite set. *Journal of the American Statistical Association*, 77(380):908–915.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39:1–38.

Forbes, F., Peyrard, N., Fraley, C., Georgian-Smith, D., Goldhaber, D. M., and Raftery, A. E. (2006). Model-based region-of-interest selection in dynamic breast MRI. *Journal of Computer Assisted Tomography*, 30:675–687.

Fraley, C., Raftery, A., and Wehrens, R. (2005). Incremental model-based clustering for large datasets with small clusters. *Journal of Computational and Graphical Statistics*, 14(3):529–546.

Fraley, C. and Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631.

Fraley, C. and Raftery, A. E. (2006). MCLUST version 3 for R: Normal mixture modeling and model-based clustering. Technical Report 504, University of Washington, Department of Statistics.

Gentleman, R. C., Carey, V. J., Bates, D. M., Bolstad, B., Dettling, M., Dudoit, S., Ellis, B., Gautier, L., Ge, Y., Gentry, J., Hornik, K., Hothorn, T., Huber, W., Iacus, S., Irizarry, R., Leisch, F., Li, C., Maechler, M., Rossini, A. J., Sawitzki, G., Smith, C., Smyth, G., Tierney, L., Yang, J. Y. H., and Zhang, J. (2004). Bioconductor: open software development for computational biology and bioinformatics. *Genome Biology*, 5(10):R80.

Gutierrez, R. G., Carroll, R. J., Wang, N., Lee, G.-H., and Taylor, B. H. (1995). Analysis of tomato root initiation using a normal mixture distribution. *Biometrics*, 51:1461–1468.

Johnson, R. A. and Wichern, D. W. (2002). *Applied Multivariate Statistical Analysis*. Prentice Hall, Upper Saddle River, NJ.

Kass, R. E. and Raftery, A. E. (1995). Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795.

Kotz, S. and Nadarajah, S. (2004). *Multivariate t Distributions and Their Applications*. Cambridge University Press, Cambridge.

Kriessler, J. R. and Beers, T. C. (1997). Substructure in galaxy clusters: a two-dimensional approach. *The Astronomical Journal*, 113:80–100.

Lange, K. L., Little, R. J. A., and Taylor, J. M. G. (1989). Robust statistical modeling using the t-distribution. *Journal of the American Statistical Association*, 84:881–896.

Li, Q., Fraley, C., Bumgarner, R. E., Yeung, K. Y., and Raftery, A. E. (2005). Donuts, scratches and blanks: Robust model-based segmentation of microarray images. *Bioinformatics*, 21(12):2875–2882.

Lin, T. I. (2009a). Maximum likelihood estimation for multivariate skew normal mixture models. *Journal of Multivariate Analysis*, 100(2):257–265.

Lin, T. I. (2009b). Robust mixture modeling using multivariate skew t distributions. *Statistics and Computing*, (In press).

Lin, T. I., Lee, J. C., and Hsieh, W. J. (2007a). Robust mixture modeling using the skew t distribution. *Statistics and Computing*, 17:81–92.

Lin, T. I., Lee, J. C., and Yen, S. Y. (2007b). Finite mixture modelling using the skew normal distribution. *Statistica Sinica*, 17:909–927.

Liu, C. (1997). ML estimation of the multivariate t distribution and the EM algorithm. *Journal of Multivariate Analysis*, 63:296–312.

Liu, C. and Rubin, D. (1994). The ECME algorithm: a simple extension of EM and ECM with faster monotone convergence. *Biometrika*, 81(4):633–648.

Liu, C. and Rubin, D. (1995). ML estimation of the t distribution using EM and its extensions, ECM and ECME. *Statistica Sinica*, 5:19–39.

McLachlan, G. and Peel, D. (2000). *Finite Mixture Models*. Wiley-Interscience, New York.

McLachlan, G. J., Bean, R. W., and Peel, D. (2002). A mixture model-based approach to the clustering of microarray expression data. *Bioinformatics*, 18(3):413–422.

Meng, X. L. and Rubin, D. B. (1993). Maximum likelihood estimation via the ECM algorithm: a general framework. *Biometrika*, 80:267–278.

Mukherjee, S., Feigelson, E. D., Babu, G. J., Murtagh, F., Fraley, C., and Raftery, A. E. (1998). Three types of gamma ray bursts. *The Astrophysical Journal*, 508:314–327.

Pan, W., Lin, J., and Le, C. T. (2002). Model-based cluster analysis of microarray gene-expression data. *Genome Biology*, 3(2):R9.

Peel, D. and McLachlan, G. J. (2000). Robust mixture modelling using the t distribution. *Statistics and Computing*, 10(4):339–348.

Pyne, S., Hu, X., Wang, K., Rossin, E., Lin, T.-I., Maier, L. M., Baecher-Allan, C., McLachlan, G. J., Tamayo, P., Hafler, D. A., De Jager, P. L., and Mesirov, J. P. (2009). Automated high-dimensional flow cytometric data analysis. *Proceedings of the National Academy of Sciences of the United States of America*, 106(21):8519–8524.

Sahu, S. K., Dey, D. K., and Branco, M. D. (2003). A new class of multivariate skew distributions with applications to Bayesian regression. *Canadian Journal of Statistics*, 31(2):129–150.

Schork, N. J. and Schork, M. A. (1988). Skewness and mixtures of normal distributions. *Communications in Statistics: Theory and Methods*, 17:3951–3969.

Schroeter, P., Vesin, J.-M., Langenberger, T., and Meuli, R. (1998). Robust parameter estimation of intensity distributions for brain magnetic resonance images. *IEEE Transactions on Medical Imaging*, 17(2):172–186.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6:461–464.

Stephens, M. (2000). Bayesian analysis of mixture models with an unknown number of components – an alternative to reversible jump methods. *Annals of Statistics*, 28:40–74.

Titterington, D. M., Smith, A. F. M., and Makov, U. E. (1985). *Statistical Analysis of Finite Mixture Distributions*. Wiley, Chichester, UK.

Wehrens, R., Buydens, L. M. C., Fraley, C., and Raftery, A. E. (2004). Model-based clustering for image segmentation and large datasets via sampling. *Journal of Classification*, 21:231–253.

Yeung, K. Y., Fraley, C., Murua, A., Raftery, A. E., and Ruzzo, W. L. (2001). Model-based clustering and data transformations for gene expression data. *Bioinformatics*, 17(10):977–987.

Chapter 4

Automated Gating of Flow Cytometry Data via Robust Model-Based Clustering*

4.1 Introduction

Flow cytometry (FCM) can be applied to analyze thousands of samples per day. However, as each dataset typically consists of multiparametric descriptions of millions of individual cells, data analysis can present a significant challenge. As a result, despite its widespread use, FCM has not reached its full potential because of the lack of an automated analysis platform to parallel the high-throughput data generation platform. As noted in Lizard (2007), in contrast to the tremendous interest in the FCM technology, there is a dearth of statistical and bioinformatics tools to manage, analyze, present, and disseminate FCM data. There is considerable demand for the development of appropriate software tools, as manual analysis of individual samples is error-prone, non-reproducible, non-standardized, not open to re-evaluation, and requires an inordinate amount of time, making it a limiting aspect of the technology (Roederer and Hardy, 2001; Roederer *et al.*, 2001a,b; de Rosa *et al.*, 2003; Bagwell, 2004; Braylan, 2004; Redelman, 2004; Tzircotis *et al.*, 2004; Spidlen *et al.*, 2006).

The process of identifying homogeneous groups of cells that display a particular function, known as gating, is one major component of FCM anal-

* A version of this chapter has been published. Lo, K., Brinkman, R. R. and Gottardo, R. (2008). Automated gating of flow cytometry data via robust model-based clustering. *Cytometry Part A*, 73A(4):321–332.

ysis. As mentioned in Chapter 1, currently, to a large extent, gating relies on using software to apply a series of manually drawn gates (i.e., data filters) that select regions in 2D graphical representations of FCM data. This process is based largely on intuition rather than standardized statistical inference (Parks, 1997; Suni *et al.*, 2003; Bagwell, 2004). It also ignores the high-dimensionality of FCM data, which may convey information that cannot be displayed in 1D or 2D projections. This is illustrated in Figure 4.1 with a synthetic dataset, consisting of two dimensions, generated from a t mixture model (McLachlan and Peel, 2000) with three components. While the three clusters can be identified using both dimensions, the structure is hardly recognized when the dataset is projected on either dimension. Such an example illustrates the potential loss of information if we disregard the multivariate nature of the data. The same problem occurs when projecting three (or more) dimensional data onto two dimensions.

Several attempts have been made to automate the gating process. Among those, the K-means algorithm (MacQueen, 1967) has found the most applications (Murphy, 1985; Demers *et al.*, 1992; Bakker Schut *et al.*, 1993; Wilkins *et al.*, 2001). Demers *et al.* (1992) have proposed an extension of K-means allowing for non-spherical clusters, but this algorithm has been shown to lead to performance inferior to fuzzy K-means clustering (Wilkins *et al.*, 2001). In fuzzy K-means (Rousseeuw *et al.*, 1996), each cell can belong to several clusters with different association degrees, rather than belonging completely to only one cluster. Even though fuzzy K-means takes into consideration some form of classification uncertainty, it is a heuristic-based algorithm and lacks a formal statistical foundation. Other popular choices include hierarchical clustering algorithms (e.g., linkage or Pearson coefficients method). However, these algorithms are not appropriate for FCM data, since the size of the pairwise distance matrix increases in the order of n^2 with the number of cells, unless they are applied to some preliminary partition of the data (Bakker Schut *et al.*, 1993), or they are used to cluster across samples, each of which is represented by a few statistics aggregating measurements of individual cells (Maynadié *et al.*, 2002; Lugli *et al.*, 2007). Classification and regression trees (Breiman *et al.*, 1984), artificial neural

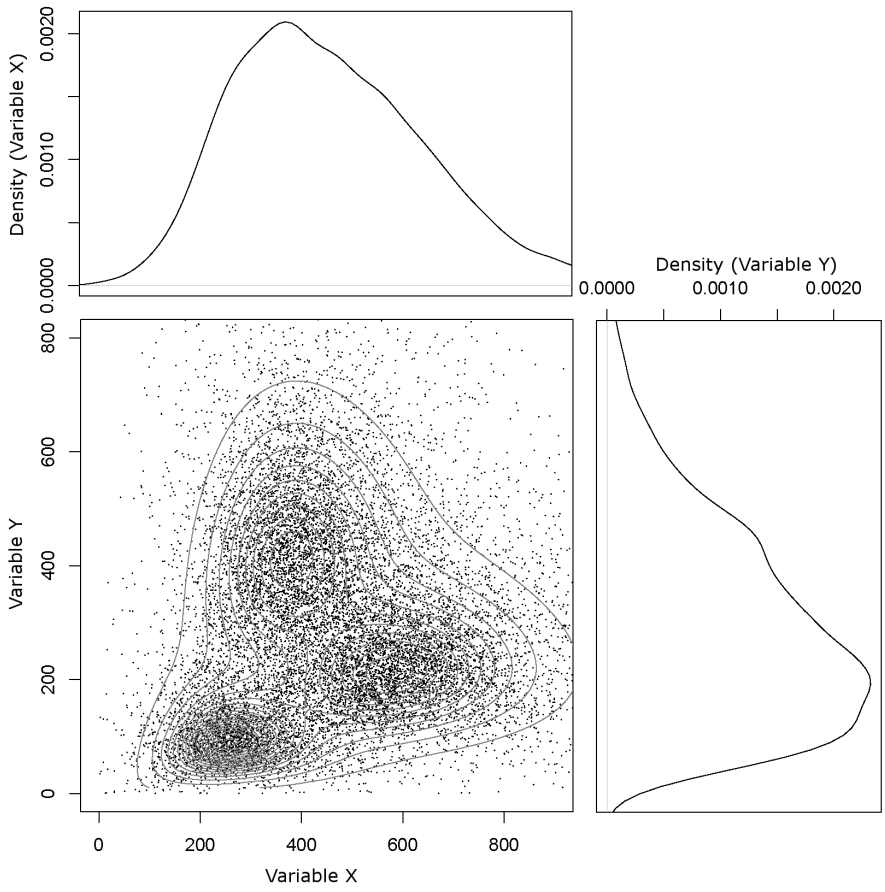


Figure 4.1: A synthetic 2D dataset with three mixture components. The three components can easily be identified when both dimensions are used (lower left), while the two density curves produced from projecting the data on either dimension fail to capture the structure.

networks (Boddy and Morris, 1999) and support vector machines (Burges, 1998; Schölkopf and Smola, 2002) have also been used in the context of FCM analyses (Beckman *et al.*, 1995; Kothari *et al.*, 1996; Boddy *et al.*, 2000; Morris *et al.*, 2001), but these supervised approaches require training data, which are not always available.

In statistics, the problem of finding homogeneous groups of observations is referred to as clustering. An increasingly popular choice is model-based clustering (Titterington *et al.*, 1985; McLachlan and Basford, 1988; Banfield and Raftery, 1993; McLachlan and Peel, 2000; Fraley and Raftery, 2002), which has been shown to give good results in many applied fields involving high dimensions (greater than ten); see, for example, Yeung *et al.* (2001), Fraley and Raftery (2002) and Pan *et al.* (2002). In this chapter, we propose to apply an unsupervised model-based clustering approach to identify cell populations in FCM analysis. In contrast to previous unsupervised methods (Murphy, 1985; Demers *et al.*, 1992; Bakker Schut *et al.*, 1993; Roederer and Hardy, 2001; Roederer *et al.*, 2001a,b; Wilkins *et al.*, 2001), our approach provides a formal unified statistical framework to answer central questions: How many populations are there? Should we transform the data? What model should we use? How should we deal with outliers (aberrant observations)? These questions are fundamental to FCM analysis where one does not usually know the number of populations, and where outliers are frequent. By performing clustering using all variables consisting of fluorescent markers, the full multidimensionality of the data is exploited, leading to more accurate and more reproducible identification of cell populations.

The most commonly used model-based clustering approach is based on finite Gaussian mixture models (Titterington *et al.*, 1985; McLachlan and Basford, 1988; McLachlan and Peel, 2000; Fraley and Raftery, 2002). However, Gaussian mixture models rely heavily on the assumption that each component follows a Gaussian distribution, which is often unrealistic. As a remedy, transformation of the data is often considered. On the other hand, there is the problem of outlier identification in mixture modeling. Transformation selection can be heavily influenced by the presence of outliers (Carroll, 1982; Atkinson, 1988), which are frequently observed in FCM data.

To handle the issues of transformation selection and outlier identification simultaneously, in Chapter 3 we have developed an automated clustering approach based on t mixture models with the Box-Cox transformation. The t distribution is similar in shape to the Gaussian distribution with heavier tails and thus provides a robust alternative (Lange *et al.*, 1989). The Box-Cox transformation is a type of power transformation, which can bring skewed data back to symmetry, a property of both the Gaussian and t distributions. In particular, the Box-Cox transformation is effective for data where the dispersion increases with the magnitude, a scenario not uncommon to FCM data.

4.2 Materials and Methods

4.2.1 Data Description

To demonstrate our proposed automated clustering we use two FCM datasets publicly available at `http://www.ficcs.org/software.html`.

The Rituximab Dataset

Flow cytometric high-content screening (Abraham *et al.*, 2004) was applied in a drug-screening project to identify agents that would enhance the anti-lymphoma activity of Rituximab, a therapeutic monoclonal antibody (Gasparetto *et al.*, 2004). 1600 different compounds were distributed into duplicate 96-well plates and then incubated overnight with the Daudi lymphoma cell line. Rituximab was then added to one of the duplicate plates and both plates were incubated for several more hours. In addition to cells treated with the compound alone, other controls included untreated cells and cells treated with Rituximab alone. During the entire culture period, cells were incubated with the thymidine analogue BrdU to label newly synthesized DNA. Following culture, cells were stained with anti-BrdU and the DNA binding dye 7-AAD. The proportion of cells in various phases of the cell cycle and undergoing apoptosis was measured with multiparameter FACS analysis.

The GvHD Dataset

Graft-versus-Host Disease (GvHD) occurs in allogeneic hematopoietic stem cell transplant recipients when donor-immune cells in the graft initiate an attack on the skin, gut, liver, and other tissues of the recipient. It is one of the most significant clinical problems in the field of allogeneic blood and marrow transplantation. FCM was used to collect data on patients subjected to bone marrow transplant with a goal of identifying biomarkers to predict the development of GvHD. The GvHD dataset is a collection of weekly peripheral blood samples obtained from 31 patients following allogeneic blood and marrow transplant (Brinkman *et al.*, 2007). Peripheral blood mononuclear cells were isolated using Ficoll-Hypaque and then cryopreserved for subsequent batch analysis. At the time of analysis, cells were thawed and aliquoted into 96-well plates at 1×10^4 to 1×10^5 cells per well. The 96-well plates were then stained with 10 different four-color antibody combinations. All staining and analysis procedures were miniaturized so that small number of cells could be stained in 96-well plates with optimally diluted fluorescently conjugated antibodies.

4.2.2 The Model

In statistics, model-based clustering (Titterington *et al.*, 1985; McLachlan and Basford, 1988; McLachlan and Peel, 2000; Fraley and Raftery, 2002) is a popular unsupervised approach to look for homogeneous groups of observations. The most commonly used model-based clustering approach is based on finite Gaussian mixture models, which have been shown to give good results in various applied fields (Banfield and Raftery, 1993; McLachlan and Peel, 2000; Fraley and Raftery, 2002, 2006). However, Gaussian mixture models might give poor representations of clusters in the presence of outliers, or when the clusters are far from elliptical in shape, phenomena commonly observed in FCM data. In view of this, we have proposed an approach based on t mixture models (McLachlan and Peel, 2000; Peel and McLachlan, 2000) coupled with a variant of the Box-Cox transformation (Bickel and Doksum, 1981), which is also defined for negative-valued

data, to handle the two aforementioned issues simultaneously. Please refer to Chapter 3 for a detailed account of an Expectation-Maximization (EM) algorithm (Dempster *et al.*, 1977) for the simultaneous estimation of all unknown parameters along with transformation selection. When the number of clusters is unknown, we use the Bayesian Information Criterion (BIC) (Schwarz, 1978), which gives good results in the context of mixture models (Fraley and Raftery, 1998, 2002).

While it is possible to estimate the degrees of freedom parameter ν of the t distribution for each component of the mixture model as part of the EM algorithm (Peel and McLachlan, 2000), fixing it to a reasonable predetermined value for all components reduces the computational burden while still providing robust results. A reasonable value for ν is four, which leads to a distribution similar to the Gaussian distribution, with slightly fatter tails accounting for outliers. Besides, the EM algorithm needs to be initialized. In this chapter, we apply a type of agglomerative hierarchical clustering based on Gaussian models (Banfield and Raftery, 1993; Fraley, 1998) for initialization. Model-based Gaussian hierarchical clustering is a stepwise process aimed to maximize the classification likelihood function (Banfield and Raftery, 1993; Celeux and Govaert, 1992). The process starts with treating each observation itself as one cluster, and then successively merges pairs of clusters leading to the highest increase in the likelihood until the desired number of clusters is reached. This initialization method is the same as the one used in the model-based clustering strategy proposed by Fraley and Raftery (2002, 2006), as implemented in the **R** package `mclust`. As mentioned in the Introduction, hierarchical clustering algorithms pose a problem with FCM data as they require the storage of a pairwise distance matrix which increases in the order of n^2 with the number of cells. In view of this, we apply hierarchical clustering to a subset of data, and perform one EM iteration to cluster the remaining data to complete the initial partition.

4.2.3 Density Estimation

To visualize FCM data, it may be convenient to project high-dimensional data on 1D or 2D density plots. One such application can be found in the analysis of the GvHD data, in which cells selected through the $CD3^+$ gate were projected on the CD4 and CD8β dimensions to produce contour plots (see Figures 4.2 and 4.3). Usually, nonparametric methods are applied to produce such plots. However, all nonparametric methods require a tuning parameter (e.g., bandwidth for kernel density estimation; see Silverman, 1986) to be specified to control the smoothness of these plots, and different softwares have different default settings. In the model-based clustering framework, such plots can be easily generated at a very low computational cost once estimates of the model parameters are available. The degree of smoothness is controlled by the number of components, which is chosen by the Bayesian Information Criterion (BIC) (Schwarz, 1978). Please see Section 3.2.3 for more details on implementation.

4.2.4 Sequential Approach to Clustering

In practice, gating is often done on a preselected subset of data chosen by projecting the data on the forward light scatter (FSC) and sideward light scatter (SSC) dimensions. These two variables, which measure the relative morphological properties (corresponding roughly to cell size and shape) of the cells, are often used to distinguish basic cell types (e.g., monocytes and lymphocytes) or to remove dead cells and cell debris. As a consequence, similar to Hahne *et al.* (2006), we have adopted a sequential approach to clustering. We first use the FSC and SSC variables to cluster the data and find basic cell populations, and then perform clustering on one or more populations of interest using all other variables consisting of fluorescent markers. However, our methodology could also be applied to any subset or the entire set of variables.

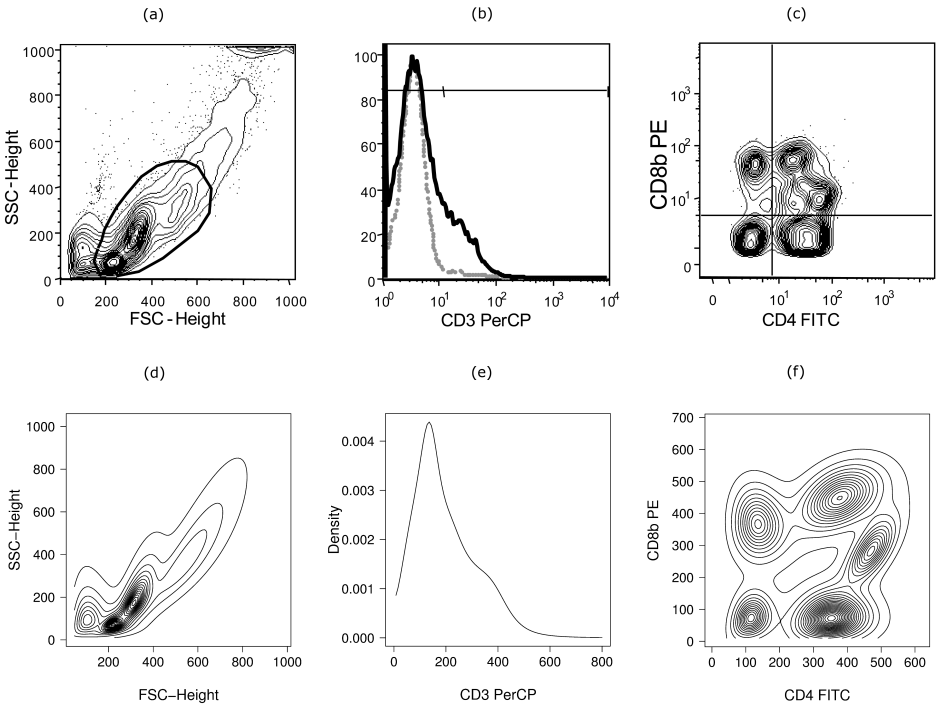


Figure 4.2: Strategy for clustering the GvHD positive sample to look for $CD3^+CD4^+CD8\beta^+$ cells. The manual gating strategy is shown in (a–c). (a) Using FlowJo, a gate was drawn by an expert researcher to define the lymphocyte population. (b) The selected cells were projected on the CD3 dimension, and $CD3^+$ cells were defined through setting an interval gate. (c) Cells within the upper right gate were referred to as $CD3^+CD4^+CD8\beta^+$. (d–f) A t mixture model with the Box-Cox transformation was used to mimic this manual selection process; here we display the corresponding density estimates. For FlowJo, the density estimates correspond to kernel estimates, while for our gating strategy, the density estimates are obtained from the estimated mixture models.

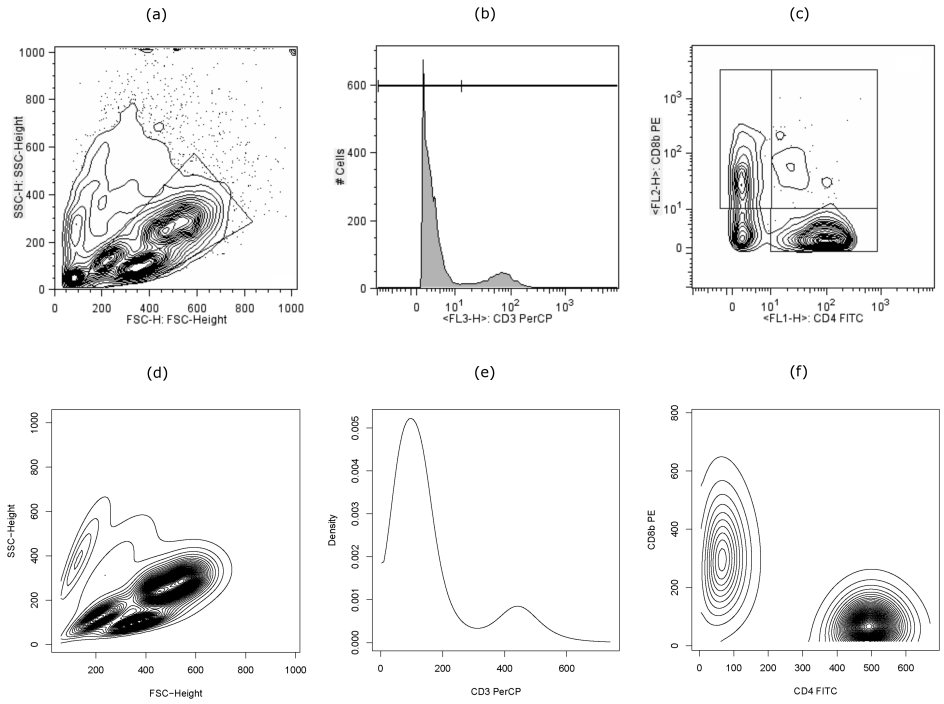


Figure 4.3: Strategy for clustering the GvHD control sample. (a–c) The same manual gating strategy was applied by the expert researcher. (c) The upper right gate corresponding to the $CD3^{+}CD4^{+}CD8\beta^{+}$ population contains very few cells, a distinct difference from the positive sample. (d–f) A t mixture model with the Box-Cox transformation was used to mimic this manual selection process; here we display the corresponding density estimates.

4.3 Results

4.3.1 Application to Real Datasets

The Rituximab dataset

We have re-analyzed a part of the Rituximab dataset using our sequential clustering approach. This data contains 1545 cells and four variables: FSC, SSC and two fluorescent markers, namely, 7-AAD and anti-BrdU. We compared the different models, namely, t mixture with Box-Cox, t mixture, Gaussian mixture with Box-Cox, and Gaussian mixture, with the results obtained through expert manual analysis using the commercial gating software FlowJo (Tree star, Ashland, Oregon) and the K-means clustering algorithm (MacQueen, 1967). As mentioned in Section 4.2.4, we use a sequential approach where we first cluster the FSC vs. SSC variables to select basic cell populations (first stage), and then cluster the selected population(s) using all remaining variables (second stage).

Figure 4.4(a) shows the initial gating performed by a researcher using FlowJo on the FSC and SSC variables. To facilitate the comparison of our clustering approach with manual analysis at the second stage, we tried to mimic this analysis. In order to do so, we used a t mixture model with Box-Cox transformation, fixing the number of components at one, and removed points with weights $\tilde{u}$ (please refer to Section 3.2.3 for details) less than 0.5, corresponding to outliers. As shown in Figure 4.4, the selected cells are not exactly the same but close enough to allow us to compare our clustering approach to manual gating results when using the two fluorescent markers.

At the second stage, we compare the different clustering models on the selected cells. Since the number of clusters is unknown in advance, we make use of the BIC. The BIC curves shown in Figure 4.5, corresponding to the different models, peak around three to four clusters, motivating us to examine the results obtained using three (Figure 4.6) and four (Figure 4.7) clusters respectively. As expected, K-means performs poorly as spherical clusters do not provide a good fit. Similarly, untransformed mixture models (t and Gaussian), constrained by the assumption of elliptical clusters,

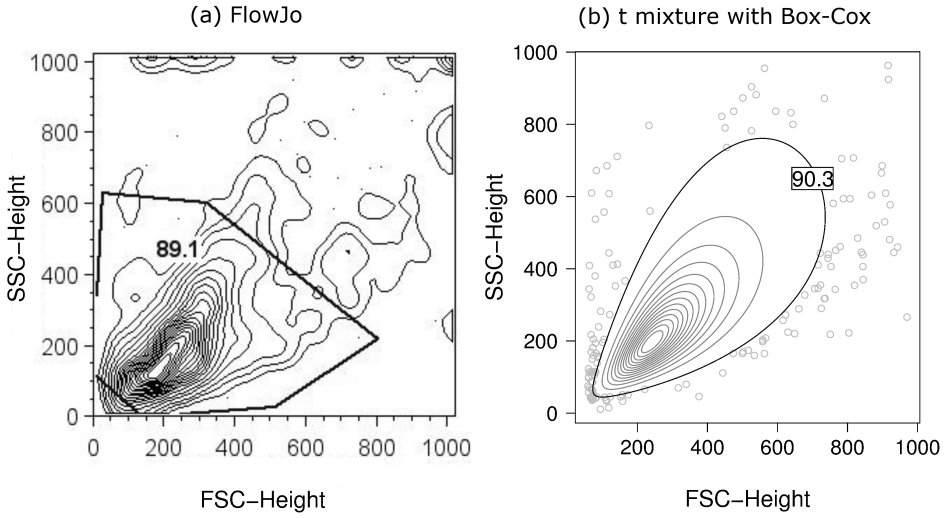


Figure 4.4: Initial clustering of the Rituximab data using the FSC and SSC variables. (a) In typical analysis a gate was manually drawn to select a group of cells for further investigation. (b) A t mixture model with Box-Cox transformation was used to mimic this manual selection process. In (b) points (shown in gray) outside the boundary drawn in black have weights $\tilde{u}$ less than 0.5 and will be removed from the next stage. It can be shown that this boundary corresponds approximately to the 90th percentile region for the t distribution transformed back on the original scale using the Box-Cox parameter. The numbers shown in both plots are the percentages of points within the boundaries which are extracted for the next stage. Both gates capture the highest density region, as shown by the two density estimates.

are not flexible enough to capture the top cluster. Furthermore, Gaussian mixture models (even with the Box-Cox transformation) are very sensitive to outliers, which can result in poor classification. For example, when four clusters are used, the Gaussian mixture model breaks the larger cluster into two to accommodate outliers, while the Gaussian mixture model with the Box-Cox transformation also has a large spread out cluster to accommodate outliers. Finally, Figures 4.6(b) and 4.7(b) show that our t mixture model-based clustering approach with the Box-Cox transformation can provide comparable results with the manual gating analysis by identifying three

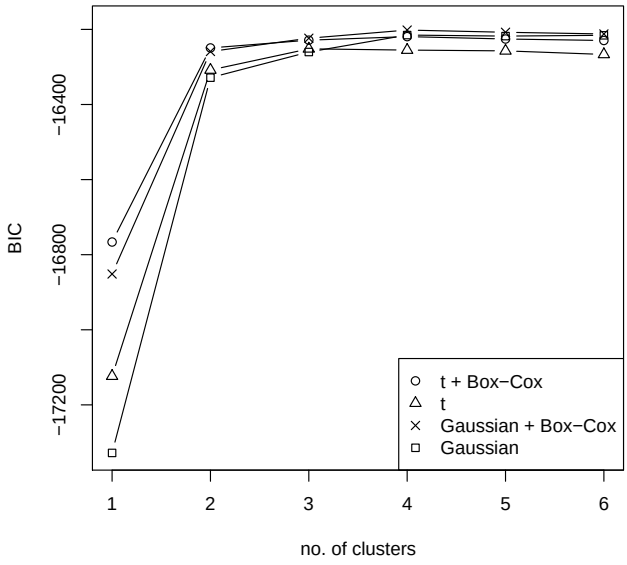


Figure 4.5: BIC as a function of the number of clusters for different models applied to the Rituximab data. All models have a maximum BIC value around three to four clusters, though there is some uncertainty as the BIC values are relatively close.

of the four clusters with well-fit boundaries. Note, however, that none of the four clustering methods detect the left rectangular gate seen on Figure 4.6(a), which is most likely because of its lower cell density compared to the other gates and the lack of clear separation along the "7-AAD" dimension. This gate, which corresponds to apoptotic cells (Gasparetto *et al.*, 2004), contains a loose assemblage of cells located at the left of the three far right gates. Our methodology permits the identification of the three right clusters with well-fit boundaries, and thus could be combined with expert knowledge in order to identify apoptotic cells. For example, one could compute a one dimensional boundary at the left-end border of the two largest clusters, and automatically label cells on the left of that line apoptotic.

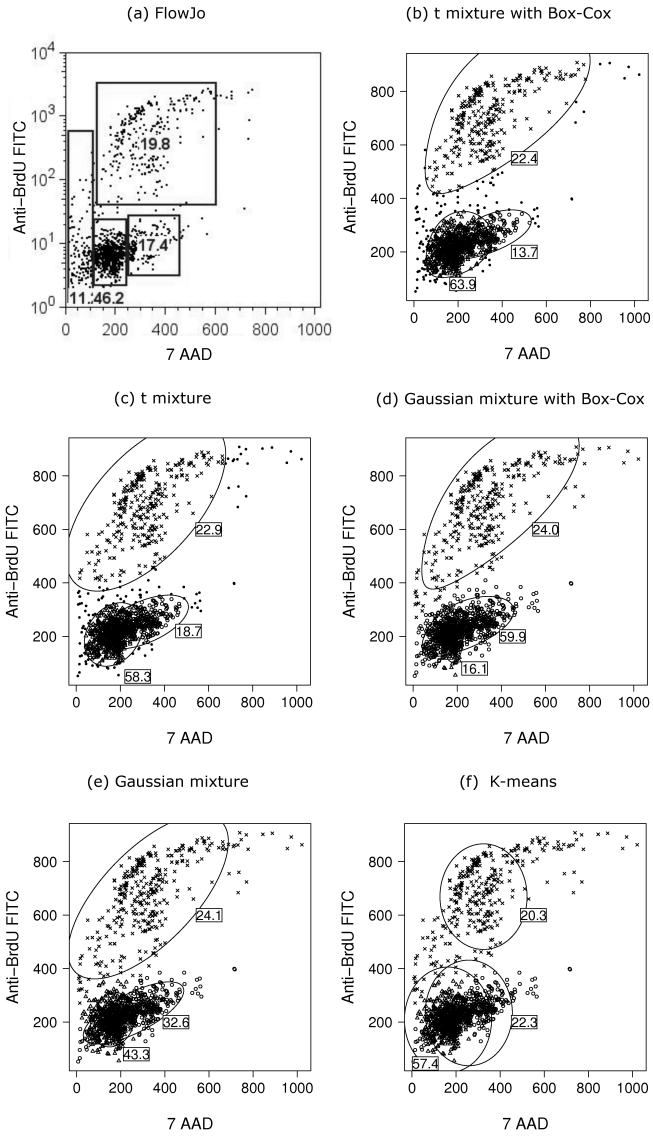


Figure 4.6: Second-stage clustering of the Rituximab data using all the fluorescent markers (three clusters). (a) Four gates were drawn by a researcher to define four populations of interest. (b–f) Clustering was performed on the cells preselected from the first stage as shown in Figure 4.4(b). The number of clusters was set to be three. (b–c) Points outside the boundary drawn in black have weights less than 0.5 and are labeled with "·" when t distributions were used. (d–f) For clustering performed without using t distributions, for comparison sake, boundaries are drawn in a way such that they correspond to the region of the same percentile which the boundaries drawn in (b–c) represent. Different symbols are used for the different clusters. The numbers shown in all plots are the percentages of cells assigned to each cluster. The K-means algorithm is equivalent to the classification EM algorithm (Celeux and Govaert, 1992, 1995) for a Gaussian mixture model assuming equal proportions and a common covariance matrix being a scalar multiple of the identity matrix. The spherical clusters with equal volumes drawn in (f) correspond to such a constrained model.

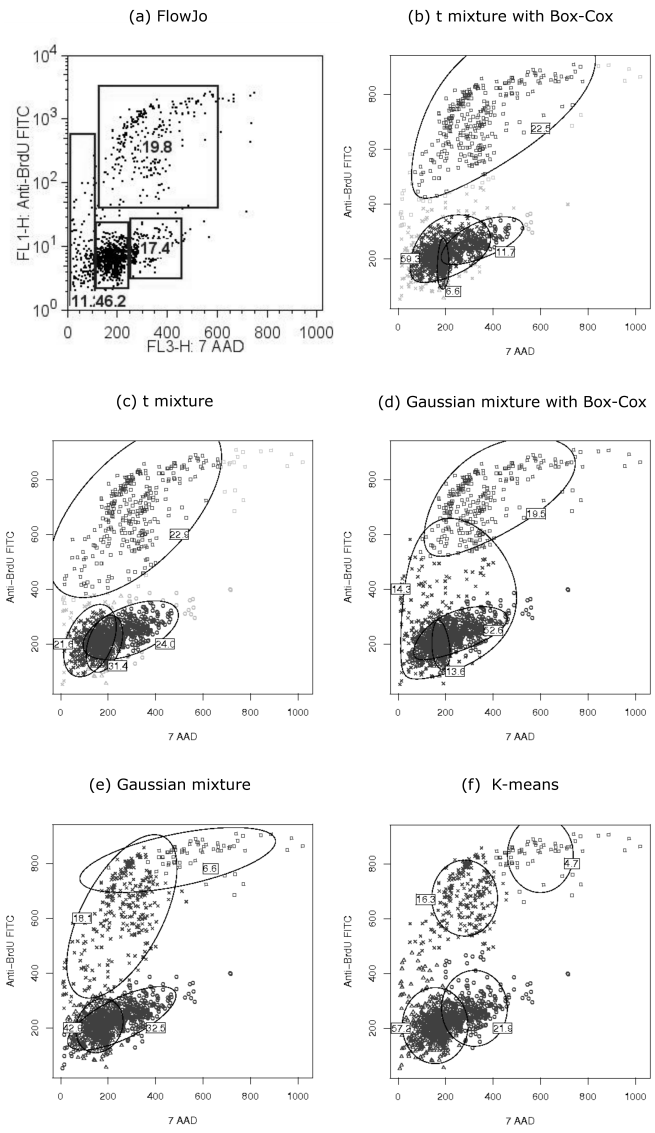


Figure 4.7: Second-stage clustering of the Rituximab data using all the fluorescent markers (four clusters). (a) Four gates were drawn by a researcher to define four populations of interest. (b–f) Clustering was performed on cells preselected from the first stage. The number of clusters was set to be four. (b–c) Points outside the boundary drawn in black have weights less than 0.5 and are shown in gray when t distributions were used. (d–f) For clustering performed without using t distributions, for comparison sake, boundaries are drawn in a way such that they correspond to the region of the same percentile which the boundaries drawn in (b–c) represent. Different symbols are used for the different clusters. The numbers shown in all plots are the percentages of cells assigned to each cluster.

Having shown the superiority of our clustering framework in terms of flexibility and robustness compared to common approaches, we now turn to a larger dataset to demonstrate further its capability.

The GvHD Dataset

Two samples of the GvHD dataset (Brinkman *et al.*, 2007) have been re-analyzed, one from a patient who eventually developed acute GvHD, and one from a control. Both datasets consist of more than 12,000 cells and four markers, namely, anti-CD4, anti-CD8β, anti-CD3 and anti-CD8, in addition to the FSC and SSC variables. One objective of the analysis is to look for the CD3$^+$CD4$^+$CD8β^+ cells. To demonstrate the capability of our proposed automated clustering approach, we try to mimic the gating strategy stated in Brinkman *et al.* (2007). Figures 4.2(a–c) and 4.3(a–c) show the gating performed by an expert researcher using FlowJo.

In the initial gating, we first extracted the lymphocyte population using the FSC and SSC variables by applying a t mixture model with the Box-Cox transformation, fixing the number of clusters from one to eight in turn. Figure 4.8(a) shows that the BIC for the positive sample has a large increase from three to four clusters and remains relatively constant afterwards, suggesting a model fit using four clusters is appropriate. Figure 4.8(b) is the corresponding scatterplot showing the cluster assignment of the points on removing those with weights less than 0.5, regarded as outliers. It is clear that the region combining three of the clusters formed matches closely with the gate drawn by the researcher as shown in Figure 4.2(a), corresponding to the lymphocyte population.

The next two stages in the manual gating strategy consist of locating the CD3$^+$ cells by placing an interval gate in the CD3 density plot (Figure 4.2(b)), and then identifying the CD3$^+$CD4$^+$CD8β^+ cells through the upper right gate in the CD4 vs CD8β contour plot (Figure 4.2(c)). When applying our proposed clustering approach, we can combine these two stages by handling all the variables consisting of fluorescent markers at once, fully utilizing the multidimensionality of FCM data.

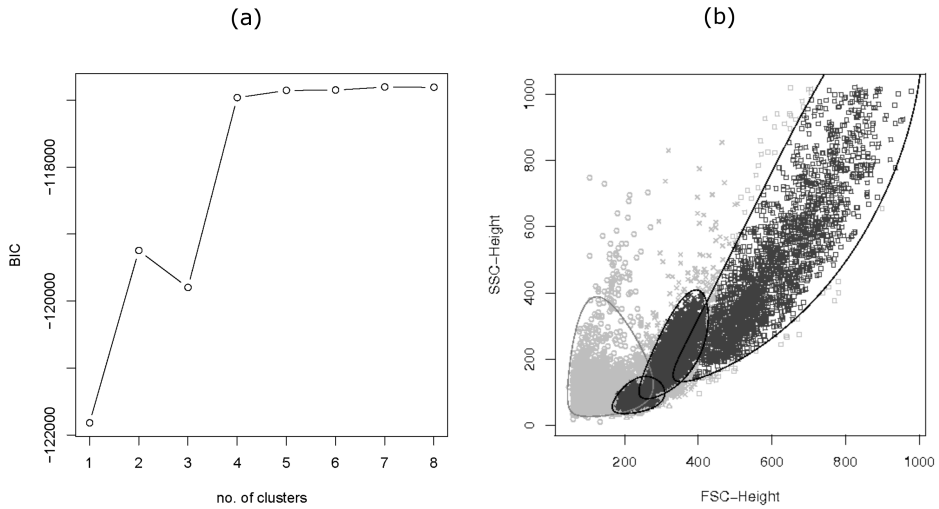


Figure 4.8: Initial clustering of the GvHD positive sample using the FSC and SSC variables. (a) The BIC curve remains constant beyond four clusters. (b) The scatterplot reveals the use of three clusters to represent the lymphocyte population and the remaining cluster (shown in gray) for dead cells. Points shown in gray have weights less than 0.5 and will be removed from the next stage.

The fitted model with 12 clusters seems to provide a good fit as suggested by the BIC (Figure 4.9(a)). We compared our results with those obtained through the manual gating approach by first examining the estimated density projected on the CD3 dimension. The unimodal, yet skewed, density curve suggests that it is composed of two populations with substantially different proportions superimposed on each other (Figure 4.2(e)). At a level of around 280, we can well separate the 12 cluster means along the CD3 dimension into two groups, and use the group with high cluster means in the CD3 dimension to represent the $CD3^+$ population. The unimodal nature of the density curve (Figures 4.2(b,e)) implies that the two underlying populations somewhat mix together, and therefore setting a fixed cutoff to classify the cells is likely inappropriate. The merit of our automated clustering approach is shown here, that, instead of setting a cutoff, it makes use of

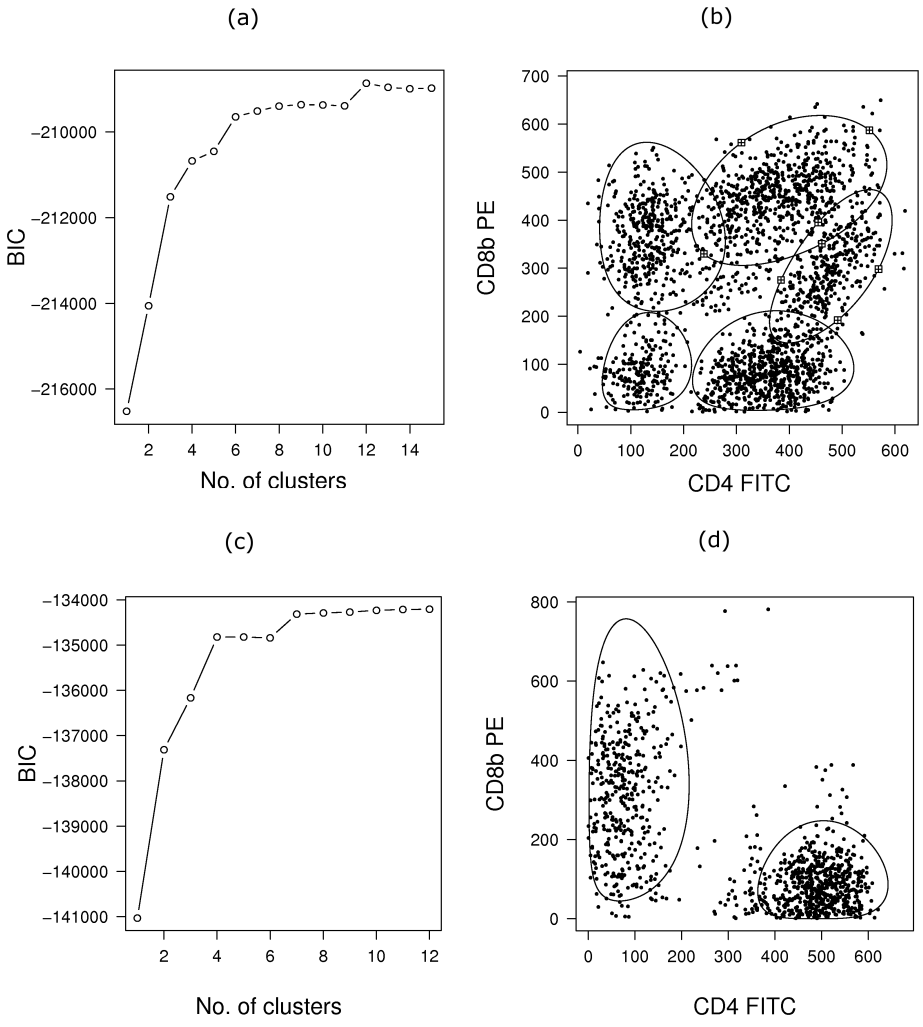


Figure 4.9: Second-stage clustering of the GvHD positive sample (a–b) and control sample (c–d) using all the fluorescent markers. Clustering was performed on the cells preselected from the first stage. For the positive sample, (a) the BIC reaches a maximum at 12 clusters; (b) the scatterplot reveals the cluster assignment of the cells. Points which are assigned to the five clusters with high CD3 means are classified as $CD3^+$ cells. The five regions drawn in solid lines form the $CD3^+$ population. The two regions in the upper right marked with the $\boxplus$ symbols are identified as the $CD3^+CD4^+CD8\beta^+$ population. For the control sample, (c) little increment is observed in the BIC beyond seven clusters, suggesting that seven clusters, much fewer than for the positive sample, are enough to model the data in the second stage; (d) the scatterplot reveals the cluster assignment of the cells. Only two clusters have been used to model the $CD3^+$ population.

the information provided by the other dimensions to help classify the cells into $CD3^+/CD3^-$ populations. The group with high cluster means in the CD3 dimension consists of five clusters, and among these five clusters, we can easily identify the two clusters at the upper right in the CD4 vs CD8β scatterplot (Figure 4.9(b)) as the $CD3^+CD4^+CD8\beta^+$ population.

We have applied the same strategy to the control sample; see Figures 4.3 and 4.9(c–d). Figure 4.9(c) suggests that, this time, only seven clusters are necessary as the BIC is relatively flat after that. The associated gating results for the control sample is characterized by an absence of the $CD3^+CD4^+CD8\beta^+$ cells, a distinct difference from the positive sample. This feature is also captured using our automated clustering approach; the fitted model contains no clusters at the upper right of the CD4 vs CD8β scatterplot (Figure 4.9(d)). This cell population was of specific interest, as it was identified as one possibly predictive of GvHD, based on the manual gating analysis in Brinkman *et al.* (2007).

4.3.2 Simulation studies

We have conducted a series of simulations to study the performance of different model-based clustering approaches under different model specifications. Model performance is compared using the following two criteria: (a) the accuracy in cluster assignment; (b) the accuracy in selecting the number of clusters. We performed two simulation studies, one where we set the dimension to two resembling the Rituximab dataset, and one where the dimension was set to four resembling the GvHD dataset. In each case, we generated data from each of the following models: t mixture with Box-Cox, t mixture, Gaussian mixture with Box-Cox, and Gaussian mixture, using the parameter estimates obtained at the second stage in the Rituximab and GvHD (positive sample) analyses. For the GvHD, to reduce computational burden, we only selected the five clusters with the largest means in the CD3 dimension, corresponding to the $CD3^+$ population. We refer to the simulation experiments as the Rituximab and the GvHD settings, respectively. We fixed the number of cells at 500 and generated 1000 datasets under each

Table 4.1: Average misclassification rates for different models applied to data generated under the Rituximab or GvHD setting.

		Model used to fit data			
		t+Box-Cox	t	G.+Box-Cox	Gaussian
Model used to generate data under the Rituximab setting	t+Box-Cox	**0.187**	0.211	0.279	0.251
	t	**0.255**	0.263	0.339	0.315
	G.+Box-Cox	0.321	0.400	**0.251**	0.352
	Gaussian	0.344	0.329	0.317	**0.301**
Model used to generate data under the GvHD setting	t+Box-Cox	**0.112**	0.116	0.205	0.230
	t	**0.107**	0.111	0.191	0.221
	G.+Box-Cox	**0.135**	0.143	0.139	0.152
	Gaussian	0.134	0.132	0.132	**0.126**

G. = Gaussian; the best results are shown in bold.

of the aforementioned models. To study the accuracy in selecting the number of clusters using BIC, we generated 100 datasets from the same GvHD setting with 1000 cells. Here, we used 1000 cells to avoid numerical problems with small clusters when the number of clusters used is significantly larger than the true number, while we decreased the number of datasets to 100 because of the increase in computation when estimating the number of clusters.

Classification Results

The four clustering methods in comparison were applied to each of the 1000 datasets generated from each model. Model fitting was done by presuming that the number of clusters is known, i.e., four clusters for the Rituximab setting and five for GvHD. We compared the models via misclassification rates, i.e., the proportions of cells assigned to incorrect clusters. When computing the misclassification rates, all permutations of the cluster labels were considered, and the lowest misclassification rate was determined.

The scatterplot of one of the datasets (GvHD setting) generated from the t mixture model with Box-Cox transformation can be found in Figure 4.10. Overall results are shown in Table 4.1. As expected, the Gaussian mixture models perform poorly when data were generated from the t mixture models because of a lack of mechanisms to handle outliers. When a transformation

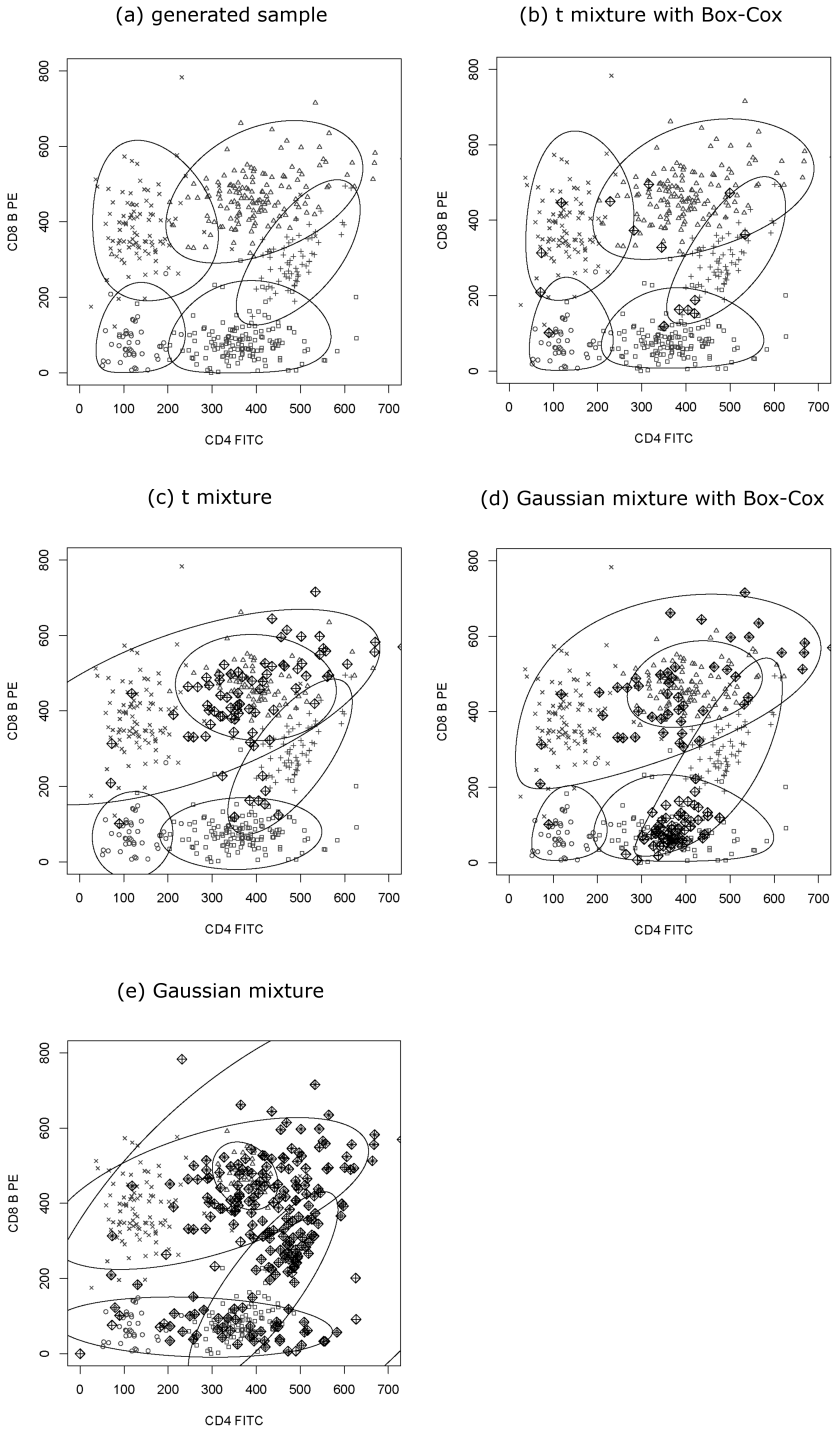


Figure 4.10: A representative sample generated from the t mixture model with the Box-Cox transformation under the GvHD setting. (a) The sample is displayed through the CD4 and CD8β dimensions. (b–e) Classification results are shown for the four clustering methods. Different plotting symbols are used for different clusters. Misclassified points are marked with the $\boxtimes$ symbols.

was applied during data generation, the mixture models without the Box-Cox transformation fail to perform well. On the contrary, the flexibility of the t mixture model with the Box-Cox transformation does not penalize too much for model misspecification. This is illustrated by the results from the GvHD setting: the t mixture model with the Box-Cox transformation gives the lowest misclassification rates when the true model is instead the t mixture model without transformation or the Gaussian mixture model with the Box-Cox transformation.

Selecting the Number of Clusters

In this part of the study, the four models in comparison were applied to each of the 100 datasets generated, setting the number of clusters from one to ten in turn. The number of clusters that delivered the highest BIC was selected. We compared the models via the mode and the 80% coverage interval of the number of clusters selected out of the 100 repetitions. As shown in Table 4.2, the t mixture models can select the correct number of clusters in the majority of repetitions, even in case of model misspecification. In addition, they deliver the same 80% coverage intervals as the Gaussian mixture models do when data were generated from Gaussian mixtures, suggesting that the robustness against outliers of the t mixture models provides satisfactory protection against model misspecification. On the contrary, the Gaussian mixture models tend to overestimate the number of clusters when an excess of outliers is present in the data generated from t mixtures, and in most instances in which overestimation happens, six clusters are selected.

4.4 Discussion

The experimental data and the simulation studies have demonstrated the importance of handling transformation selection, outlier identification and clustering simultaneously. While a stepwise approach in which transformation is preselected ahead of outlier detection (or vice versa) may be considered, it is unlikely to tackle the problem well in general, as the preselected

Table 4.2: Modes and 80% coverage intervals of the number of clusters selected by the BIC for different models applied to data generated under the GvHD setting.

		Model used to fit data							
		t+Box-Cox		t		G.+Box-Cox		Gaussian	
		M.	Int.	M.	Int.	M.	Int.	M.	Int.
Model	t+Box-Cox	5	(5, 6)	5	(5, 6)	6	(6, 7)	6	(6, 8)
used to	t	5	(5, 7)	5	(5, 6)	6	(6, 7)	6	(6, 8)
generat	G.+Box-Cox	5	(5, 6)	5	(5, 6)	5	(5, 6)	5	(5, 6)
data	Gaussian	5	(5, 6)	5	(5, 6)	5	(5, 6)	5	(5, 6)

M. = mode; Int. = 80% coverage interval; G. = Gaussian.

transformation may be influenced by the presence of outliers. This is shown in the analysis of the Rituximab dataset. Without outlier removal the use of Gaussian mixture models led to inappropriate transformation and poor classification in order to accommodate outliers (Figures 4.6(d) and 4.7(d)). Conversely, without transformation, the t mixture model could not model the shape of the top cluster well (Figures 4.6(c) and 4.7(c)). Similarly, it is necessary to perform transformation selection and clustering simultaneously (Gutierrez *et al.*, 1995; Schork and Schork, 1988) as opposed to a stepwise approach. It is difficult to know what transformation to select beforehand as one only observes the mixture distribution, and the classification labels are unknown. A skewed distribution could be the result of one dominant cluster and one (or more) smaller cluster. As shown by our analysis with the experimental data and the simulation studies, our proposed approach based on t mixture models with Box-Cox transformation benefits from handling these issues, which have mutual influence, simultaneously. Furthermore, confirmed by results of our simulation studies, our proposed approach is robust against model misspecification and can avoid the problem of Gaussian mixture models that excessive clusters are often needed to provide a reasonable fit in case of model misspecification (Yeung *et al.*, 2001).

One of the benefits of model-based clustering is that it provides mechanism for both "hard" clustering (i.e., the partitioning of the whole data into separate clusters) and fuzzy clustering (i.e., a "soft" clustering approach in

which each event may be associated with more than one cluster). The latter approach is in line with the rationale that there exists uncertainty about to which cluster an event should be assigned. The overlaps between clusters as seen in Figures 4.6 and 4.9 reveal such uncertainty in the cluster assignment.

It is well known that the convergence of the EM algorithm depends on the initial conditions used. A bad initialization may incur slow convergence or convergence to a local minimum. In the real-data examples and the simulation studies, we used a deterministic approach called hierarchical clustering (Banfield and Raftery, 1993; Fraley, 1998) for initialization. We have found this approach to perform well in the datasets explored here. However, better initialization, perhaps incorporating expert knowledge, might be needed for more complex datasets. For example, if there is a high level of noise in the data, it might be necessary to use an initialization method that accounts for such outliers; see Fraley and Raftery (2002) for an example.

To estimate how long it takes to analyze a sample of size typical for an FCM dataset, we have carried out a test run on a synthetic dataset, which consists of one million events and 10 dimensions. To complete an analysis with 10 clusters, it took about 20 minutes on a 3GHz Intel Xeon processor with 2GB of RAM. This illustrates that the algorithm should be quick enough for analyzing a large flow dataset. In general, the computational time increases linearly with the number of events and increases in the order of p^2 with the number of variables, p, per EM iteration. This is an advantage over hierarchical clustering in which the computational time and memory space required increase in the order of n^2 with the number of events, making a hierarchical approach impractical when a sample of a moderate size, say, $>$5000, is investigated.

Like all clustering approaches, the methodology we have developed includes assumptions which may limit the applicability of this approach, and it will not identify every cell population in every sample. If the distribution of the underlying population is highly sparse without a well-defined core, our approach may not properly identify all sub-populations. This is illustrated in the Rituximab analysis where the loosely structured group of apoptotic cells was left undetected. This in turn has hindered the capa-

bility of the approach from giving satisfactory esimates of the G1 and S frequences for the identified clusters that would be desired for normal analysis of a 7-AAD DNA distribution for cultured cells. On the other hand identification of every cluster may not always be important. The Rituximab study was designed as a high throughput drug screen to identify compounds that caused a >50% reduction in S-phase cells (Gasparetto *et al.*, 2004), as would be captured by both the manual gates and our automated analysis should it occur. Furthermore, the exact identification of every cluster through careful manual analysis may not always be possible, especially in high throughput experiments. For instance, in the manual analysis of the GvHD dataset, a quadrant gate was set in Figure 4.2(c) in order to identify the $CD3^+CD4^+CD8\beta^+$ population which was of primary interest. For convenience sake, this gate was set at the same level across all the samples being investigated. While five clusters can be clearly identified on the graph, it would be time-consuming to manually adjust the positions of each of the gates for all the samples in a high-throughput environment as well as identify all novel populations. Contrariwise, our automated approach can identify these clusters in short order without the need for manual adjustment. To complete the analysis of the GvHD dataset (>12,000 cells, six dimensions) to identify the $CD3^+CD4^+CD8\beta^+$ population (Figure 4.2), it took less than five minutes, using the aforementioned sequential approach to clustering, on an Intel Core 2 Duo with 2GB of RAM running Mac OS X 10.4.10.

A rigorous quantitative assessment is important before implementing this, or any approach, as a replacement for expert manual analysis. The availability of a wide variety of example data would aid in the development and evaluation of automated analysis methodologies. We are therefore developing such a public resource, and would welcome contributions from the wider FCM community.

Bibliography

Abraham, V. C., Taylor, D. L., and Haskins, J. R. (2004). High content screening applied to large-scale cell biology. *Trends in Biotechnology*, 22(1):15–22.

Atkinson, A. C. (1988). Transformations unmasked. *Technometrics*, 30:311–318.

Bagwell, C. B. (2004). DNA histogram analysis for node-negative breast cancer. *Cytometry Part A*, 58A(1):76–78.

Bakker Schut, T. C., de Grooth, B. G., and Greve, J. (1993). Cluster analysis of flow cytometric list mode data on a personal computer. *Cytometry*, 14(6):649–659.

Banfield, J. D. and Raftery, A. E. (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 49:803–821.

Beckman, R. J., Salzman, G. C., and Stewart, C. C. (1995). Classification and regression trees for bone marrow immunophenotyping. *Cytometry*, 20(3):210–217.

Bickel, P. J. and Doksum, K. A. (1981). An analysis of transformations revisited. *Journal of the American Statistical Association*, 76(374):296–311.

Boddy, L. and Morris, C. W. (1999). Artificial neural networks for pattern recognition. In Fielding, A. H., editor, *Machine Learning Methods for Ecological Applications*, pages 37–87. Kluwer, Boston.

Boddy, L., Morris, C. W., Wilkins, M. F., Al-Haddad, L., Tarran, G. A., Jonker, R. R., and Burkill, P. H. (2000). Identification of 72 phytoplankton species by radial basis function neural network analysis of flow cytometric data. *Marine Ecology Progress Series*, 195:47–59.

Braylan, R. C. (2004). Impact of flow cytometry on the diagnosis and characterization of lymphomas, chronic lymphoproliferative disorders and plasma cell neoplasias. *Cytometry Part A*, 58A(1):57–61.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth & Brooks, Monterey, CA.

Brinkman, R. R., Gasparetto, M., Lee, S. J. J., Ribickas, A., Perkins, J., Janssen, W., Smiley, R., and Smith, C. (2007). High-content flow cytometry and temporal data analysis for defining a cellular signature of Graft-versus-Host disease. *Biology of Blood and Marrow Transplantation*, 13(6):691–700.

Burges, C. J. C. (1998). *A Tutorial on Support Vector Machines for Pattern Recognition*. Kluwer, Boston.

Carroll, R. J. (1982). Prediction and power transformations when the choice of power is restricted to a finite set. *Journal of the American Statistical Association*, 77(380):908–915.

Celeux, G. and Govaert, G. (1992). A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics and Data Analysis*, 14(3):315–332.

Celeux, G. and Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern Recognition*, 28(5):781–793.

de Rosa, S. C., Brenchley, J. M., and Roederer, M. (2003). Beyond six colors: a new era in flow cytometry. *Nature Medicine*, 9(1):112–117.

Demers, S., Kim, J., Legendre, P., and Legendre, L. (1992). Analyzing multivariate flow cytometric data in aquatic sciences. *Cytometry*, 13(3):291–298.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39:1–38.

Fraley, C. (1998). Algorithms for model-based Gaussian hierarchical clustering. *SIAM Journal on Scientific Computing*, 20(1):270–281.

Fraley, C. and Raftery, A. E. (1998). How many clusters? Which clustering method? Answers via model-based cluster analysis. *The Computer Journal*, 41(8):578–588.

Fraley, C. and Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631.

Fraley, C. and Raftery, A. E. (2006). MCLUST version 3 for R: Normal mixture modeling and model-based clustering. Technical Report 504, University of Washington, Department of Statistics.

Gasparetto, M., Gentry, T., Sebti, S., O'Bryan, E., Nimmanapalli, R., Blaskovich, M. A., Bhalla, K., Rizzieri, D., Haaland, P., Dunne, J., and Smith, C. (2004). Identification of compounds that enhance the anti-lymphoma activity of rituximab using flow cytometric high-content screening. *Journal of Immunological Methods*, 292(1–2):59–71.

Gutierrez, R. G., Carroll, R. J., Wang, N., Lee, G.-H., and Taylor, B. H. (1995). Analysis of tomato root initiation using a normal mixture distribution. *Biometrics*, 51:1461–1468.

Hahne, F., Arlt, D., Sauermann, M., Majety, M., Poustka, A., Wiemann, S., and Huber, W. (2006). Statistical methods and software for the analysis of high throughput reverse genetic assays using flow cytometry readouts. *Genome Biology*, 7(8):R77.

Kothari, R., Cualing, H., and Balachander, T. (1996). Neural network analysis of flow cytometry immunophenotype data. *IEEE Transactions on Biomedical Engineering*, 43(8):803–810.

Lange, K. L., Little, R. J. A., and Taylor, J. M. G. (1989). Robust statistical modeling using the t-distribution. *Journal of the American Statistical Association*, 84:881–896.

Lizard, G. (2007). Flow cytometry analyses and bioinformatics: interest in new softwares to optimize novel technologies and to favor the emergence of innovative concepts in cell research. *Cytometry Part A*, 71A:646–647.

Lugli, E., Pinti, M., Nasi, M., Troiano, L., Ferraresi, R., Mussi, C., Salvioli, G., Patsekin, V., Robinson, J. P., Durante, C., Cocchi, M., and Cossarizza, A. (2007). Subject classification obtained by cluster analysis and principal component analysis applied to flow cytometric data. *Cytometry Part A*, 71A:334–344.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In LeCam, L. and Neyman, J., editors, *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, Berkeley. University of California Press.

Maynadié, M., Picard, F., Husson, B., Chatelain, B., Cornet, Y., Le Roux, G., Campos, L., Dromelet, A., Lepelley, P., Jouault, H., Imbert, M., Rosenwadj, M., Vergé, V., Bissières, P., Raphaël, M., Béné, M. C., Feuillard, J., and GEIL (2002). Immunophenotypic clustering of myelodysplastic syndromes. *Blood*, 100(7):2349–2356.

McLachlan, G. and Peel, D. (2000). *Finite Mixture Models*. Wiley-Interscience, New York.

McLachlan, G. J. and Basford, K. E. (1988). *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker Inc, New York, NY.

Morris, C. W., Autret, A., and Boddy, L. (2001). Support vector machines for identifying organisms – a comparison with strongly partitioned radial basis function networks. *Ecological Modelling*, 146(1–3):57–67.

Murphy, R. F. (1985). Automated identification of subpopulations in flow cytometric list mode data using cluster analysis. *Cytometry*, 6(4):302–309.

Pan, W., Lin, J., and Le, C. T. (2002). Model-based cluster analysis of microarray gene-expression data. *Genome Biology*, 3(2):R9.

Parks, D. R. (1997). Data processing and analysis: data management. In Robinson, J. P., editor, *Current Protocols in Cytometry*, chapter 10. John Wiley & Sons, Inc, New York.

Peel, D. and McLachlan, G. J. (2000). Robust mixture modelling using the t distribution. *Statistics and Computing*, 10(4):339–348.

Redelman, D. (2004). CytometryML. *Cytometry Part A*, 62A(1):70–73.

Roederer, M. and Hardy, R. R. (2001). Frequency difference gating: a multivariate method for identifying subsets that differ between samples. *Cytometry*, 45(1):56–64.

Roederer, M., Moore, W., Treister, A., Hardy, R. R., and Herzenberg, L. A. (2001a). Probability binning comparison: a metric for quantitating multivariate distribution differences. *Cytometry*, 45(1):47–55.

Roederer, M., Treister, A., Moore, W., and Herzenberg, L. A. (2001b). Probability binning comparison: a metric for quantitating univariate distribution differences. *Cytometry*, 45(1):37–46.

Rousseeuw, P. J., Kaufman, L., and Trauwaert, E. (1996). Fuzzy clustering using scatter matrices. *Computational Statistics and Data Analysis*, 23(1):135–151.

Schölkopf, B. and Smola, A. J. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. The MIT Press, Cambridge, Massachusetts.

Schork, N. J. and Schork, M. A. (1988). Skewness and mixtures of normal distributions. *Communications in Statistics: Theory and Methods*, 17:3951–3969.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6:461–464.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*. Chapman-Hall, New York.

Spidlen, J., Gentleman, R. C., Haaland, P. D., Langille, M., Le Meur, N., Ochs, M. F., Schmitt, C., Smith, C. A., Treister, A. S., and Brinkman, R. R. (2006). Data standards for flow cytometry. *OMICS*, 10(2):209–214.

Suni, M. A., Dunn, H. S., Orr, P. L., de Laat, R., Sinclair, E., Ghanekar, S. A., Bredt, B. M., Dunne, J. F., Maino, V. C., and Maecker, H. T. (2003). Performance of plate-based cytokine flow cytometry with automated data analysis. *BMC Immunology*, 4:9.

Titterington, D. M., Smith, A. F. M., and Makov, U. E. (1985). *Statistical Analysis of Finite Mixture Distributions*. Wiley, Chichester, UK.

Tzircotis, G., Thorne, R. F., and Isacke, C. M. (2004). A new spreadsheet method for the analysis of bivariate flow cytometric data. *BMC Cell Biology*, 5:10.

Wilkins, M. F., Hardy, S. A., Boddy, L., and Morris, C. W. (2001). Comparison of five clustering algorithms to classify phytoplankton from flow cytometry data. *Cytometry*, 44(3):210–217.

Yeung, K. Y., Fraley, C., Murua, A., Raftery, A. E., and Ruzzo, W. L. (2001). Model-based clustering and data transformations for gene expression data. *Bioinformatics*, 17(10):977–987.

Chapter 5

flowClust: a Bioconductor package for automated gating of flow cytometry data*

5.1 Introduction

In Chapter 4, we mentioned the lack of an automated analysis platform to parallel the high-throughput data-generation platform in flow cytometry (FCM). How to resolve this current bottleneck has become an open question among the FCM community. Recently, a suite of several **R** packages providing infrastructure for FCM analysis have been released though Bioconductor (Gentleman *et al.*, 2004), an open source software development project for the analysis of genomic data. **flowCore** (Hahne *et al.*, 2009), the core package among them, provides data structures and basic manipulation of FCM data. **flowViz** (Sarkar *et al.*, 2008) offers visualization tools, while **flowQ** provides quality control and quality assessment tools for FCM data. Finally, **flowUtils** provides utilities to deal with data import/export for **flowCore**. In spite of these low-level tools, there is still a dearth of software that helps automate FCM gating analysis with a sound theoretical foundation (Lizard, 2007).

In view of the aforementioned issues, based on a formal statistical clustering approach, we have developed the **flowClust** package to help resolve the current bottleneck. **flowClust** implements a robust model-based cluster-

* A version of this chapter has been published. Lo, K., Hahne, F., Brinkman, R. R. and Gottardo, R. (2009). flowClust: a Bioconductor package for automated gating of flow cytometry data. *BMC Bioinformatics*, 10:145.

ing approach (Peel and McLachlan, 2000; McLachlan and Peel, 2000; Fraley and Raftery, 2002) which extends the multivariate t mixture model with the Box-Cox transformation proposed in Chapter 4. As a result of the extensions made, **flowClust** has included options allowing for a cluster-specific estimation of the Box-Cox transformation parameter and/or the degrees of freedom parameter.

5.2 Implementation

With the robust model-based clustering approach described in Chapter 4 as the theoretical basis, we have developed **flowClust**, an **R** package to conduct an automated FCM gating analysis and produce visualizations for the results. **flowClust** is released through Bioconductor (Gentleman *et al.*, 2004), along with those **R** packages mentioned in Section 5.1. The GNU Scientific Library (GSL) is needed for successful installation of **flowClust**. We have provided a vignette (Appendix B) that comes with **flowClust** to enunciate details about installation, and procedures of linking GSL to **R**, especially for Windows users.

In recognition of the potential need for analyzing a large number of FCM samples in parallel, during the process of package development, tremendous effort has been put into code optimization and automation. The source code for the entire model-fitting process via the EM algorithm is written in C for optimal utilization of system resources, and makes use of the Basic Linear Algebra Subprograms (BLAS) library, which facilitates multithreaded processes when an optimized library is provided. To ensure that code is developed in an efficient manner, vectorization is administered wherever possible in order to attain minimal explicit looping, one of the major factors leading to sub-optimal efficiency in programming with **R**. In addition, instead of straightforward conversion of mathematical formulae into programming code, a comprehensive account of the EM algorithm has been taken and the code has been developed in a fashion such that redundant computation of the same or uncalled-for quantities is avoided. On the other hand, the encounter with undesirable execution halt at runtime due to computational

errors would undermine the level of automation achieved. This is critical especially when a user needs to analyze a large number of samples. On this account, we have developed substantial error handling strategies to cope with various scenarios such as poor initialization of the EM algorithm, and failure of root-finding for the transformation parameter. Another important measure taken towards automation is the provision of a good default setting for parameters (e.g., search interval for the root-finding problem, and tolerance level for the convergence of EM) involved at different steps of the model-fitting process, or for arguments (e.g., colors for representing individual clusters, and cutoffs for defining outliers) used in filtering or visualizing the clustering result. Whilst parameter tuning for individual samples may still be feasible in a small-scale study, it becomes impractical when hundreds of samples need to be processed in parallel. We have undergone an extensive tuning process to test against a large number of real FCM samples such that sensible results or visualization would be delivered within a reasonable timeframe for the majority of cases upon which the default setting is applied. Finally, in consideration of convenience from users' perspective, many functions or methods in **flowClust** have been specifically adapted to cater for various input data structures. Effort has also been made to adapt to the custom of FCM researchers whilst developing tools of visualization and for constructing data filters.

A formal object-oriented programming discipline, the S4 system (Chambers, 2004), has been adopted to build the **flowClust** package. Two key features of the S4 system, namely, multiple dispatch and multiple inheritance, have been essential for defining classes and methods. For most generic functions defined or utilized in **flowClust** (e.g., `Subset`, `split` and `plot`), method dispatch relies on the multiple dispatch capabilities and is done in accordance with a signature taking more than one argument. Incidentally, inheritance is employed to extend classes defined in other packages; see Section 5.3.2 for details about integration with other Bioconductor packages dedicated to FCM analysis. In particular, for the sake of organization, multiple inheritance is exploited such that multiple classes can be extended simultaneously.

The core function, `flowClust`, of the package implements the clustering methodology and returns an object of class `flowClust`. A `flowClust` object stores essential information related to the clustering result which can be retrieved through various methods such as `summary`, `Map`, `getEstimates`, etc. To visualize the clustering results, the `plot` and `hist` methods can be applied to produce scatterplots, contour or image plots and histograms.

To enhance communications with other Bioconductor packages designed for the cytometry community, **flowClust** has been built with the aim of being highly integrated with **flowCore**. Methods in **flowClust** can be directly applied on a `flowFrame`, the standard **R** implementation of a Flow Cytometry Standard (FCS) file defined in **flowCore**; FCS is the typical storage mode for FCM data. Another step towards integration is to overload basic filtering methods defined in **flowCore** (e.g., `filter`, `%in%`, `Subset` and `split`) in order to provide similar functionality for classes defined in **flowClust**.

5.3 Results and Discussion

5.3.1 Analysis of Real FCM Data

In this section, we illustrate how to use **flowClust** to conduct an automated gating analysis of real FCM data. For demonstration, we use the graft-versus-host disease (GvHD) data (Brinkman *et al.*, 2007). The data are stored in FCS files, and consist of measurements of four fluorescently conjugated antibodies, namely, anti-CD4, anti-CD8β, anti-CD3 and anti-CD8, in addition to the forward scatter and sideward scatter parameters. One objective of the gating analysis is to look for the $CD3^+CD4^+CD8\beta^+$ cell population, a distinctive feature found in GvHD-positive samples. We have adopted a two-stage strategy (Section 4.2.4): we first cluster the data by using the two scatter parameters to identify basic cell populations, and then perform clustering on the population of interest using all fluorescence parameters.

At the initial stage, we extract the lymphocyte population using the

forward scatter (FSC-H) and sideward scatter (SSC-H) parameters:

```
GvHD <- read.FCS("B07", trans=FALSE)
res1 <- flowClust(GvHD, varNames=c("FSC-H", "SSC-H"), K=1:8)
```

To estimate the number of clusters, we run `flowClust` on the data repetitively with `K=1` up to `K=8` clusters in turn, and apply the Bayesian Information Criterion (BIC) (Schwarz, 1978) to guide the choice. Values of the BIC can be retrieved through the `criterion` method. Figure 5.1 shows that the BIC curve remains relatively flat beyond four clusters. We therefore choose the model with four clusters. Below is a summary of the corresponding clustering result:

```
** Experiment Information **
Experiment name: Flow Experiment
Variables used: FSC-H SSC-H
** Clustering Summary **
Number of clusters: 4
Proportions: 0.1779686 0.1622115 0.3882043 0.2716157
** Transformation Parameter **
lambda: 0.1126388
** Information Criteria **
Log likelihood: -146769.5
BIC: -293765.9
ICL: -300546.2
** Data Quality **
Number of points filtered from above: 168 (1.31%)
Number of points filtered from below: 0 (0%)
Rule of identifying outliers: 90% quantile
Number of outliers: 506 (3.93%)
Uncertainty summary:
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's
9.941e-04 1.211e-02 3.512e-02 8.787e-02 1.070e-01 6.531e-01 1.680e+02
```

The estimate of the Box-Cox parameter λ is 0.11, implying a transformation close to a logarithmic one ($\lambda = 0$).

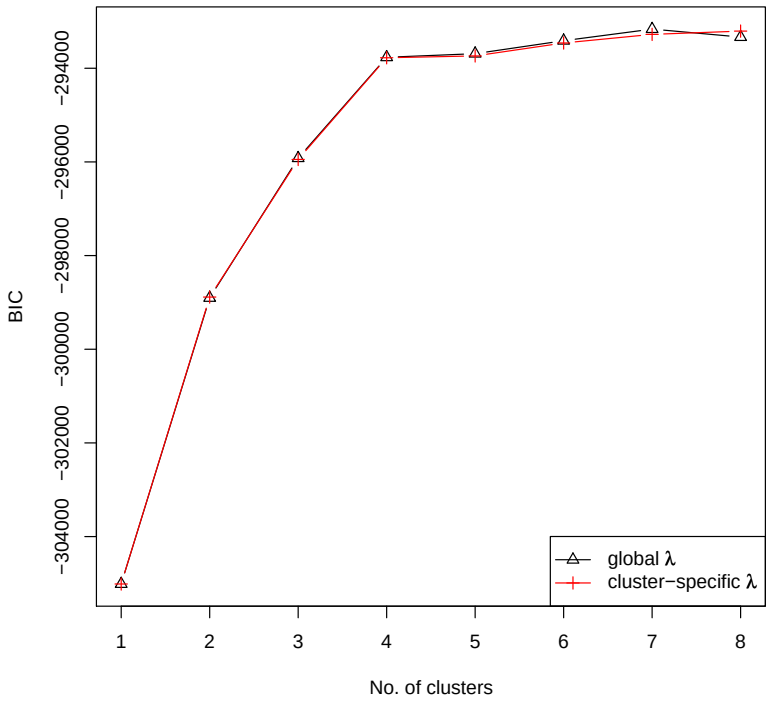


Figure 5.1: A plot of BIC against the number of clusters for the first-stage cluster analysis. The two curves correspond to the settings with a common λ and cluster-specific λ respectively for the first-stage cluster analysis. Little difference in the BIC values between the two settings is observed. The BIC curves remain relatively flat beyond four clusters, suggesting that the model fit using four clusters is appropriate.

Note that, by default, `flowClust` selects the same transformation for all clusters. We have also enabled the option of estimating the Box-Cox parameter λ for each cluster. For instance, if a user finds the shapes of the clusters significantly deviate from one another and opts for a different transformation for each cluster, he may write the following line of code:

```
res1s <- flowClust(GvHD, varNames=c("FSC-H", "SSC-H"), K=1:8,
  trans=2)
```

The `trans` argument acts as a switch to govern how λ is handled: fixed at a predetermined value (`trans=0`), estimated and set common to all clusters (`trans=1`), or estimated for each cluster (`trans=2`). Incidentally, the option of estimating the degrees of freedom parameter ν has also been made available, either common to all clusters or specific to each of them. The `nu.est` argument is the corresponding switch and takes a similar interpretation to `trans`. Such an option of estimating ν further fine-tunes the model-fitting process such that the fitted model can reflect the data-specific level of abundance of outliers. To compare the models adopting a different combination of these options, one may make use of the BIC again. Figure 5.1 shows that little difference in the two BIC curves corresponding to the default setting (common λ) and the setting with cluster-specific λ respectively can be observed. In accordance with the principle of parsimony in statistics which favors a simpler model, we opt for the default setting here.

Graphical functionalities are available to users for visualizing a wealth of features of the clustering results, including the cluster assignment, outliers, and the size and shape of the clusters. Figure 5.2 is a scatterplot showing the cluster assignment of points upon the removal of outliers. Outliers are shown in grey with the + symbols. The black solid lines represent the 90% quantile region of the clusters which defines the cluster boundaries. The summary shown above states that the default rule used to identify outliers is `90% quantile`, which means that a point outside the 90% quantile region of the cluster to which it is assigned will be called an outlier. In most applications, the default rule should be appropriate for identifying outliers. In case a user wants finer control and would like to specify a different rule,

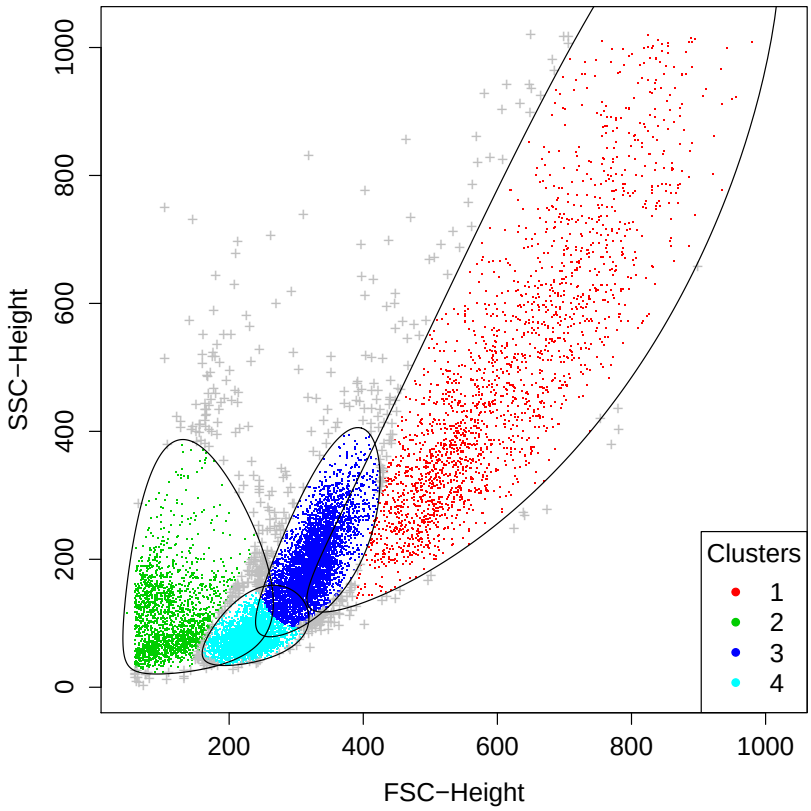


Figure 5.2: A scatterplot revealing the cluster assignment in the first-stage analysis. Clusters 1, 3 and 4 correspond to the lymphocyte population, while cluster 2 is referred to as the dead cell population. The black solid lines represent the 90% quantile region of the clusters which define the cluster boundaries. Points outside the boundary of the cluster to which they are assigned are called outliers and marked with "+".

he may apply the `ruleOutliers` replacement method:

```
ruleOutliers(res1[[4]]) <- list(level=0.95)
```

An excerpt of the corresponding summary is shown below:

```
** Data Quality **
Number of points filtered from above: 168 (1.31%)
Number of points filtered from below: 0 (0%)
Rule of identifying outliers: 95% quantile
Number of outliers: 133 (1.03%)
```

As shown in the summary, this rule is more stringent than the `90% quantile` rule: 133 points (1.03%) are now called outliers, as opposed to 506 points (3.93%) in the default rule.

Clusters 1, 3 and 4 in Figure 5.2 correspond to the lymphocyte population defined with a manual gating strategy adopted in Brinkman *et al.* (2007). We then extract these three clusters to proceed with the second-stage analysis:

```
GvHD2 <- split(GvHD, res1[[4]], population=list(lymphocyte=
  c(1,3,4), deadcells=2))
```

The subsetting method `split` allows us to split the data into several `flowFrame`'s representing the different cell populations. To extract the lymphocyte population (clusters 1, 3 and 4), we may type `GvHD2$lymphocyte` or `GvHD2[[1]]`, which is a `flowFrame`. By default, `split` removes outliers upon extraction. The `deadcells=2` list element is included above for demonstration purpose; it is needed only if we want to extract the dead cell population (cluster 2), too.

In the second-stage analysis, in order to fully utilize the multidimensionality of FCM data we cluster the lymphocyte population using all the four fluorescence parameters, namely, anti-CD4 (`FL1-H`), anti-CD8β (`FL2-H`), anti-CD3 (`FL3-H`) and anti-CD8 (`FL4-H`), at once:

```
res2 <- flowClust(GvHD2$lymphocyte, varNames=c("FL1-H",
  "FL2-H", "FL3-H", "FL4-H"), K=1:15)
```

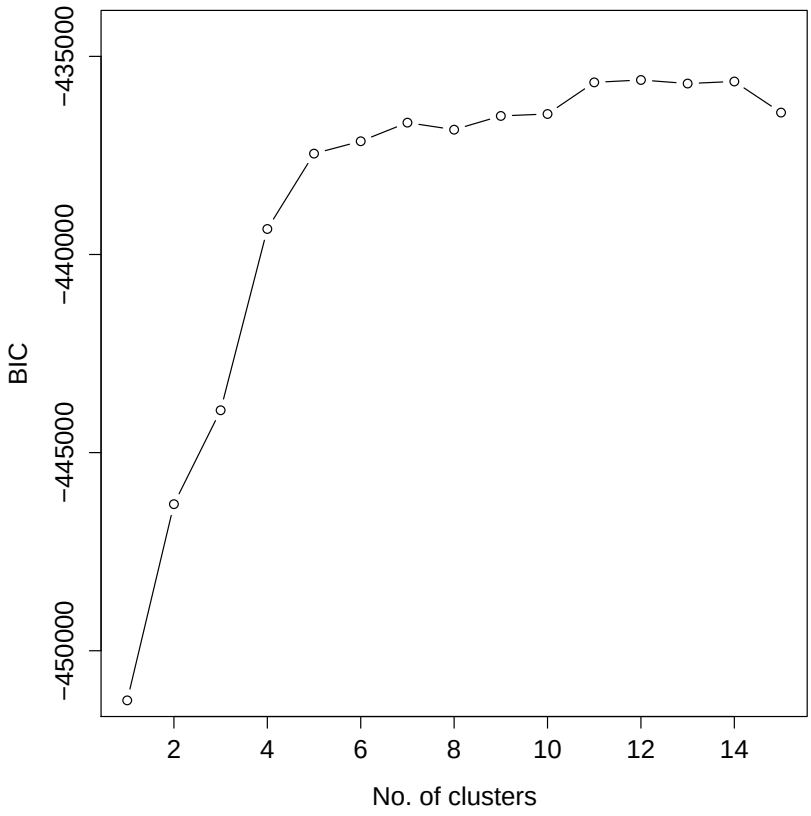
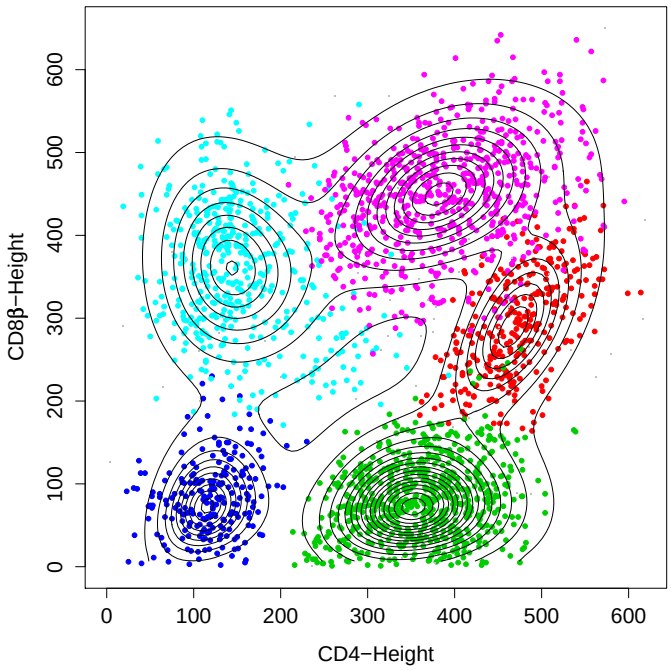


Figure 5.3: A plot of BIC against the number of clusters for the second-stage cluster analysis. The BIC curve remains relatively flat beyond 11 clusters, suggesting that the model fit using 11 clusters is appropriate.

The BIC curve remains relatively flat beyond 11 clusters (Figure 5.3), suggesting that the model with 11 clusters provides a good fit. Figure 5.4(a) shows a contour plot superimposed on a scatterplot of CD8β against CD4 for the sub-population of CD3-stained cells, which were selected based on a threshold obtained from a negative control sample (Brinkman *et al.*, 2007). We can easily identify from it the red and purple clusters at the upper right as the $\text{CD3}^{+}\text{CD4}^{+}\text{CD8}\beta^{+}$ cell population. A corresponding image plot is given by Figure 5.4(b). The code used to produce all the plots shown in this chapter can be found in Appendix C.

(a)



(b)

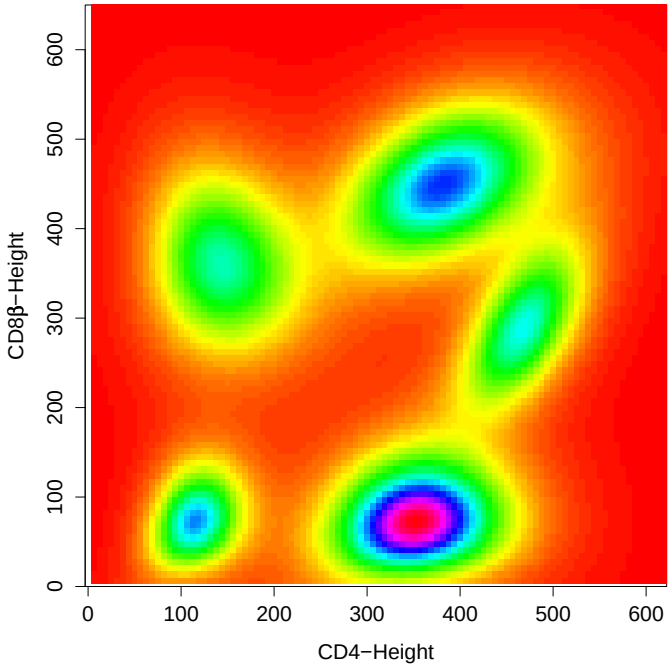


Figure 5.4: Plots of CD8β against CD4 for the CD3$^+$ population. (a) A contour plot is superimposed on a scatterplot. The red and purple clusters at the upper right correspond to the CD3$^+$CD4$^+$CD8β^+ cell population, indicative of the GvHD. (b) The five clusters corresponding to the CD3$^+$ population can also be identified clearly on an image plot.

The example above shows how an FCM analysis is conducted with the aid of **flowClust**. When the number of cell populations is not known in advance, and the BIC values are relatively close over a range of the possible number of clusters, the researcher may be presented with a set of possible solutions instead of a clear-cut single one. In such a case, the level of automation may be undermined as the researcher may need to select the best one based on his expertise. We acknowledge that more effort is needed to extend our proposed methodology towards a higher level of automation. Currently, we are working on an approach which successively merges the clusters in the solution as suggested by the BIC using some entropy criterion to give a more reasonable estimate of the number of clusters; see Section 6.2.3 for more details.

5.3.2 Integration with flowCore

As introduced in Section 5.1, **flowClust** has been built in a way such that it is highly integrated with the **flowCore** package. The core function `flowClust` which performs the clustering operation may be replaced by a call to the constructor `tmixFilter` creating a `filter` object similar to the ones used in other gating or filtering operations found in **flowCore** (e.g., `rectangleGate`, `norm2Filter`, `kmeansFilter`). As an example, the code

```
res1 <- flowClust(GvHD, varNames=c("FSC-H", "SSC-H"), K=1:8)
```

used in the first-stage analysis of the GvHD data may be replaced by:

```
s1filter <- tmixFilter("lymphocyte", c("FSC-H", "SSC-H"), K=1:8)
res1f <- filter(GvHD, s1filter)
```

The use of a dedicated `tmixFilter`-class object separates the task of specifying the settings (`tmixFilter`) from the actual filtering operation (`filter`), facilitating the common scenario in FCM gating analysis that filtering with the same settings is performed upon a large number of data files. The `filter` method returns a list object `res1f` with elements each of class `tmixFilterResult`, which directly extends the `filterResult` class defined

in **flowCore**. Users may apply various subsetting operations defined for the `filterResult` class in a similar fashion on a `tmixFilterResult` object. For instance,

```
Subset(GvHD[,c("FSC-H", "SSC-H")], res1f[[4]])
```

outputs a `flowFrame` that is the subset of the GvHD data upon the removal of outliers, consisting of the two selected parameters, `FSC-H` and `SSC-H`, only. Another example is given by the `split` method introduced earlier in Section 5.3.1.

We realize that occasionally a researcher may opt to combine the use of **flowClust** with filtering operations in **flowCore** to define the whole sequence of an FCM gating analysis. To enable the exchange of results between the two packages, filters created by `tmixFilter` may be treated like those from **flowCore**; users of **flowCore** will find that filter operators, namely, `&`, `|`, `!` and `%subset%`, also work in the **flowClust** package. For instance, suppose the researcher is interested in clustering the $CD3^+$ cell population which he defines by constructing an interval gate with the lower end-point at 280 on the CD3 parameter. He may use the following code to perform the analysis:

```
rectGate <- rectangleGate(filterId="CD3+", "FL3-H"=c(280, Inf))
s2filter <- tmixFilter("s2filter", c("FL1-H", "FL2-H", "FL3-H",
  "FL4-H"), K=5)
res2f <- filter(GvHD2$lymphocyte, s2filter %subset% rectGate)
```

The constructors `rectangleGate` and `tmixFilter` create two `filter` objects storing the settings of the interval gate and `flowClust`, respectively. When the last line of code is run, the interval gate will first be applied to the GvHD data. `flowClust` is then performed on a subset of the GvHD data contained by the interval gate.

5.4 Conclusion

flowClust is an **R** package dedicated to FCM gating analysis, addressing the increasing demand for software capable of processing and analyzing the

voluminous amount of FCM data efficiently via an objective, reproducible and automated means. The package implements a statistical clustering approach using multivariate t mixture models with the Box-Cox transformation introduced in Chapter 4, and provides tools to summarize and visualize results of the analysis. The package contributes to the cytometry community by offering an efficient, automated analysis platform which facilitates the active, ongoing technological advancement.

Bibliography

Brinkman, R. R., Gasparetto, M., Lee, S. J. J., Ribickas, A., Perkins, J., Janssen, W., Smiley, R., and Smith, C. (2007). High-content flow cytometry and temporal data analysis for defining a cellular signature of Graft-versus-Host disease. *Biology of Blood and Marrow Transplantation*, 13(6):691–700.

Chambers, J. M. (2004). *Programming with Data: a Guide to the S Language*. Springer, New York, NY.

Fraley, C. and Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631.

Gentleman, R. C., Carey, V. J., Bates, D. M., Bolstad, B., Dettling, M., Dudoit, S., Ellis, B., Gautier, L., Ge, Y., Gentry, J., Hornik, K., Hothorn, T., Huber, W., Iacus, S., Irizarry, R., Leisch, F., Li, C., Maechler, M., Rossini, A. J., Sawitzki, G., Smith, C., Smyth, G., Tierney, L., Yang, J. Y. H., and Zhang, J. (2004). Bioconductor: open software development for computational biology and bioinformatics. *Genome Biology*, 5(10):R80.

Hahne, F., Le Meur, N., Brinkman, R. R., Ellis, B., Haaland, P., Sarkar, D., Spidlen, J., Strain, E., and Gentleman, R. (2009). flowCore: a Bioconductor package for high throughput flow cytometry. *BMC Bioinformatics*, 10:106.

Lizard, G. (2007). Flow cytometry analyses and bioinformatics: interest in new softwares to optimize novel technologies and to favor the emergence of innovative concepts in cell research. *Cytometry Part A*, 71A:646–647.

McLachlan, G. and Peel, D. (2000). *Finite Mixture Models*. Wiley-Interscience, New York.

Peel, D. and McLachlan, G. J. (2000). Robust mixture modelling using the t distribution. *Statistics and Computing*, 10(4):339–348.

Sarkar, D., Le Meur, N., and Gentleman, R. (2008). Using flowViz to visualize flow cytometry data. *Bioinformatics*, 24(6):878–879.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6:461–464.

Chapter 6

Conclusion and Future Directions

6.1 Summary and Discussion

The intent of this thesis is to develop statistical methodology based on flexible forms of finite mixture models to address issues arising from high-throughput biological data sources. In Chapter 2, we introduced an empirical Bayes approach to detect differentially expressed genes from microarray data, extending the hierarchical Gamma-Gamma and Lognormal-Normal models (Newton *et al.*, 2001; Kendziorski *et al.*, 2003). The extension results in a release of the unreasonable assumption of a constant coefficient of variation for all genes, and has been shown to remarkably improve the original framework. Next, in Chapter 3, we proposed a mixture modeling framework based on a new class of skewed distributions, the multivariate t distribution with the Box-Cox transformation. We emphasize on a concurrent treatment of data transformation and outlier identification, instead of tackling the two issues of mutual impact in a sequential manner. The approach is robust to both asymmetric components and outliers, and remains to be highly competitive in comparisons made with the computationally much more complicated approach using the skew t mixture model. In Chapter 4, we reframed the gating analysis in flow cytometry (FCM) as a clustering problem, and applied the approach proposed in Chapter 3 to automate the identification of cell populations. The result shows that our approach is well adapted to FCM data, in which a high abundance of outliers is often observed. Moreover, our approach has an appeal of being

computationally competitive, which is crucial for FCM analysis considering the high dimensionality of data and the large number of samples usually involved. In recognition of concern of the FCM community which has been seeking an automated analysis platform with a solid theoretical foundation, we have developed an open-source software package called `flowClust`, details of which are delineated in Chapter 5. While `flowClust` is publicly released as an **R** package, its core part that implements the model-fitting process is coded in C to ensure computational efficiency at users' end. To facilitate the convenience of use, specific efforts have been made to adapt to the custom of FCM researchers, such as developing tools of visualization and for constructing data filters, or "gates". `flowClust` has been built in a way such that it directly accepts data in FCM-dedicated format, and is well integrated with other Bioconductor FCM packages. The package is under active maintenance for further enrichment in the modeling aspect, feature enhancement for presentation and dissemination of analysis results, and integration to existing and upcoming FCM analysis tools.

From a monochromatic technology dated back to the late 1960's, FCM has evolved into a technology that can simultaneously measure nearly 20 parameters for each cell (Perfetto *et al.*, 2004). To date, the LSR II flow cytometer from BD Biosciences (San Jose, California) can detect up to 18 colors (corresponding to biomarkers such as antigens) in one experiment. To the accompaniment of technological advancement, the impact of FCM on a wealth of fields of biology and medicine has undergone tremendous growth in the last few years; see, for example, Valet and Tárnok (2003), Valet (2005), Herrera *et al.* (2007) and Lizard (2007). We believe that FCM in the next few years will reach a level of prominence that microarray has attained in the last decade. Along with the increase in dimensionality of FCM data, it becomes apparent that the traditional way of gating analysis by relying on expertise in defining a gating sequence and positioning the gates is inefficient. How to resolve the bottleneck of a lack of an analysis platform to parallel such a high-throughput data generation platform has become an open question among the FCM community. A pleasant trend has been observed over the past one or two years, when more research work of

statistical methodology dedicated to FCM comes to light (e.g., Lugli *et al.*, 2007; Boedigheimer and Ferbas, 2008; Chan *et al.*, 2008; Lo *et al.*, 2008; Pyne *et al.*, 2009). Such an accelerating trend can also be observed from regular meetings of the Flow Informatics and Computational Cytometry Society (FICCS) and other conferences. Since published in April 2008, our article (corresponding to Chapter 4 of this thesis) has been cited 18 times to date according to the search result of Web of Science and Google Scholar. Meanwhile, a steady overall increase in the download statistics for `flowClust` has been observed from the Package Downloads Report at Bioconductor. These evidences provide a positive sign that our proposed methodology has the potential for being a mainstream automated gating approach in an FCM analysis pipeline.

6.2 Future Directions

In the remainder of this chapter, we briefly describe a few possible directions for future research, and report preliminary results therein.

6.2.1 Robustification of the Empirical Bayes Model for Differential Gene Expression

The extension we proposed in Chapter 2 allows for a gene-specific coefficient of variation in the hierarchical empirical Bayes models originated from Newton *et al.* (2001) and Kendziorski *et al.* (2003) for microarray data. Such an enhanced flexibility does not effectively constitute a mechanism to accommodate outliers, though. An outlying data value could occur because of scratches or dust on the surface, imperfections in the glass slide, or imperfections in the array production. As a possible way to robustify the empirical Bayes approach, we may consider the eLNN formulation and replace the lognormal sampling distribution with a log t distribution. In other words,

we build a model with the following hierarchical representation:

$$\begin{aligned}
\log x_{gr} &= \mu_{gx} + \frac{\epsilon_{gxr}}{\sqrt{w_{gr}}} \\
\mu_{gx}|\tau_{gx} &\sim \mathrm{N}(m, k\tau_{gx}^{-1}) \\
\epsilon_{gxr}|\tau_{gx} &\sim \mathrm{N}(0, \tau_{gx}^{-1}) \\
\tau_{gx} &\sim \mathrm{Gamma}(\alpha, \beta) \\
w_{gr} &\sim \mathrm{Gamma}\left(\frac{\nu_r}{2}, \frac{\nu_r}{2}\right)
\end{aligned} \tag{6.1}$$

where w_{gr} and ϵ_{gxr} are independent and therefore $\epsilon_{gxr}/\sqrt{w_{gr}}$ follows a central t distribution with scale matrix τ_{gx}^{-1} and degrees of freedom ν_r. All other notations in (6.1) follow the convention used in Chapter 2, and the model specification for y_{gr} can be derived accordingly. If we fix $w_{gr} = 1$ for all g and r, the aforementioned model reduces to the eLNN model introduced in Chapter 2.

The joint prior on μ_{gx}, τ_{gx} and w_{gr} is not conjugate to the sampling distribution, and the marginal density cannot be derived in closed form. However, the marginal density is analytically available conditional on w_{gr}. As a result, it is possible to proceed in a way similar to what we described in Section 2.2.3 for the eGG model in which a closed-form marginal density is available conditional on the gene-specific shape parameter. We may take accordingly the log prior density of $\mathbf{w}_g = (w_{g1}, w_{g2}, \ldots, w_{gR})'$ as the penalty term, and consider the modified complete-data log-likelihood

$$\begin{aligned}
\tilde{l}_c(\boldsymbol{\Phi}|\mathbf{x}, \mathbf{y}, \mathbf{z}) = \sum_g \Big\{ & z_g \log\ p_A(\mathbf{x}_g, \mathbf{y}_g|\boldsymbol{\psi}, w_{gr}) + (1 - z_g) \log\ p_0(\mathbf{x}_g, \mathbf{y}_g|\boldsymbol{\psi}, w_{gr}) \\
& + (1 + z_g)\log(p) + (2 - z_g)\log(1 - p) + \sum_r \log \pi(w_{gr}|\nu_r) \Big\},
\end{aligned} \tag{6.2}$$

where $\boldsymbol{\psi} = (m, k, \alpha, \beta)'$ and $\boldsymbol{\Phi} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_G, \boldsymbol{\psi}', p)'$. Parameter estimation may then be handled by the EM-type algorithm described in Section 2.2.3 in which the M-step is split into two constrained maximization steps. This robust approach provides a favorable alternative to the fully

Bayesian approach BRIDGE (Gottardo *et al.*, 2006), which takes a similar model specification but relies on MCMC techniques and is computationally intensive.

6.2.2 Development of an Automated FCM Analysis Pipeline

The analysis of FCM data usually involves two major components: (1) identifying homogeneous cell populations (commonly referred to as gating), each of which displays a particular biological function, and (2) finding correlations between identified cell populations and clinical diagnosis. We presented in Chapter 4 the statistical methodology based on robust model-based clustering to automate the gating process. An ensuing research focus would be devising a methodology that extracts features from the result of the automated gating analysis to facilitate disease diagnosis, and identifies biomarkers that correlate with the disease. Essentially, we would like to develop a pipeline, with minimal manual intervention, for the different stages of FCM data analysis, including identification of cell populations, extraction of useful features (biomarkers) correlated with a target disease, and classification of samples. Figure 6.1 shows the overall flow of the proposed data analysis pipeline (Bashashati *et al.*, 2009).

As a motivational example of FCM analysis which fits into such a pipeline, here we present our preliminary study on paroxysmal nocturnal hemoglobinuria (PNH), a disease of red blood cell breakdown with release of hemoglobin into the urine. The objective of the study is to build a classification rule that separates subjects according to their disease status (positive or negative). A series of FCM samples were obtained from 17 PNH patients and 15 controls. A complete set of samples for one subject includes two red blood cell samples and three white blood cell samples. Each sample consists of a distinct antigenic marker. Figure 6.2 shows two histograms from the red blood cell samples of a PNH patient and a control respectively. A distinctive subpopulation of low intensities is found in the positive sample. This serves as the discriminating information for subject classification.

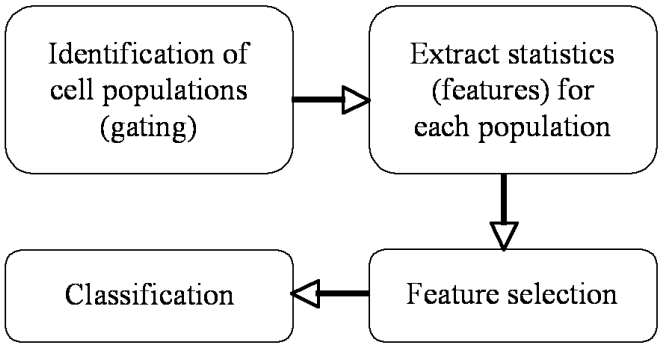


Figure 6.1: The overall flow of the proposed automated FCM analysis pipeline.

To quantify the discriminating information, we applied the methodology described in Chapter 4 to cluster each red blood cell sample into two subpopulations. The separation between the two cluster means, that is expected to be large for a positive sample, provides the basis of discriminating the two groups of subjects. We proceeded in a similar manner for each cell type, namely, granulocytes, lymphocytes and monocytes, identified in the white blood cell samples. At the next-stage analysis, subjects were represented with the features of interest (i.e., the separation between two cluster means), or a subset of them, extracted from the clustering stage. We built classifiers using support vector machines (SVM) (Schölkopf and Smola, 2002) with a linear kernel. Leave-one-out cross-validation was used to assess the accuracy of the classifiers built. Classifiers with $> 97\%$ accuracy have been found, with a few features consistently found among them.

The preliminary study on PNH presented a simplified scenario of typical FCM analysis. Very often, we do not know the number of cell populations in advance, and multiple colors are used in each sample. In such a case, a better example is given by our current study in which we attempt to devise an analysis pipeline to discriminate subtypes of lymphoma and identify biomarkers that contribute to such a classification (Bashashati *et al.*, 2009). Data in this study were generated at the British Columbia Cancer Agency

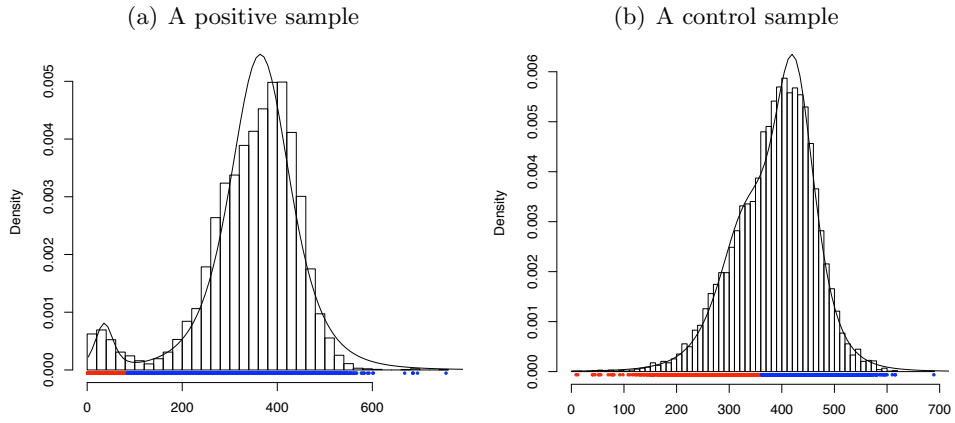


Figure 6.2: Clustering of red blood cell samples from the PNH data. The graphs shown are the histograms of CD55 from (a) a positive sample, and (b) a control sample. The presence of the distinctive subpopulation of low intensities in the positive sample is also expected to be observed on some/all of the three basic cell types from positive white blood cell samples. A clinical diagnosis would determine a subject to be PNH positive if the distinctive subpopulation is observed from at least two cell types.

in 2002–2007. FCM samples were obtained from biopsies of lymph nodes from 438 lymphoma patients of different subtypes. Samples were divided into seven tubes, each of which was stained with a distinct set of three fluorescently conjugated antibodies.

To proceed, we first apply the robust model-based clustering methodology to identify cell populations, and use the BIC to determine the number of cell populations in each sample. Statistics such as the proportion, mean and variance for each cluster are extracted, resulting in a long list of candidate features with discriminating information. The majority of features are expected to be uninformative, and in order to discard them we apply the minimum redundancy maximum relevance (mRMR) feature selection technique (Peng *et al.*, 2005; Ding and Peng, 2005). The mRMR technique aims at selecting features that are relevant to the class label (i.e., subtype of lymphoma) whilst minimizing the redundancy of the selected features; the

Euclidean distance or, more effectively, the Pearson correlation coefficient between features may be taken as the redundancy measure. Based on the selected features, we build a classifier using SVM or random forest (Breiman, 2001) to classify the samples, and make predictions about future incoming samples.

In our current attempt, we randomly split the samples into the training and testing sets. Samples in the training set are used to select informative features and build the classifiers, while the training samples are used for performance evaluation. To date, the devised analysis pipeline is confined to a binary classification, i.e., discrimination between two subtypes of lymphoma. Our preliminary result shows that 80%–96% accuracy has been achieved in a few binary classifications performed. Features (in terms of markers) identified to be informative are in line with previous biological findings (Dogan, 2005), providing promising evidence that the proposed analysis pipeline can extract biologically meaningful features from FCM data. Subsequent work would be refining the various components of the pipeline in order to achieve higher discriminating accuracy, and extending the methodology to facilitate multi-class discriminations.

6.2.3 Combining Mixture Components in Clustering

In clustering analysis, very often the number of clusters is unknown and requires estimation. There are several approaches for selecting the number of components in model-based clustering, such as resampling, cross validation, and various information criteria; see McLachlan and Peel (2000) for a review. In this thesis, our approach to the problem is based on the BIC. Model selection based on the BIC has been shown not to underestimate the number of clusters asymptotically (Leroux, 1992). Moreover, the BIC is computationally cheap to compute once maximum likelihood estimation of the model parameters has been completed, an advantage over other approaches, especially in the context of FCM where datasets tend to be very large. Nevertheless, if the correct model is not in the family of models being considered, more than one mixture component may be needed to provide a

reasonable representation of an individual cluster of data. In such a case, the BIC tends to select an excessive number of components relative to the correct number of clusters (Biernacki and Govaert, 1997; Biernacki *et al.*, 2000). Biernacki *et al.* (2000) attempted to rectify this problem by proposing an alternative to the BIC based on the integrated completed likelihood (ICL). The ICL criterion turns out to be equivalent to the BIC penalized by the entropy of the corresponding clustering:

$$\mathrm{ICL}_G = \mathrm{BIC}_G - 2\ \mathrm{ENT}_G, \tag{6.3}$$

where

$$\mathrm{ENT}_G = -\sum_{i=1}^{n}\sum_{g=1}^{G} \hat{z}_{ig} \log \hat{z}_{ig} \tag{6.4}$$

is the entropy for the corresponding G-component mixture model, and $\hat{z}_{ig}$ is the conditional probability that the i-th observation arises from the g-th component. The entropy ENT_G is a measure of relevance of the G components from a mixture model to the partition of data. Conceptually, it increases with the scale of overlap between the components. In consequence, the ICL favors models with well-separated mixture components. In practice, however, the ICL tends to underestimate the correct number of clusters (Murphy and Martin, 2003). Such a tendency was also observed when we attempted to apply the ICL to FCM data.

In a current attempt, we propose an approach for selecting the number of clusters by combining the ideas underlying the BIC and ICL (Baudry *et al.*, 2008). The BIC is used to select the number of components in the mixture model in order to provide a good representation of data. We then define a sequence of possible solutions by hierarchical merging of the components identified by the BIC. The decision about which components to merge is based on the same entropy criterion given by Eq.(6.4) that the ICL uses. In this way, we propose a way of interpreting the mixture model by identifying the set of merged components as one cluster.

In the following, we describe in details the hierarchical merging scheme. At each stage, we choose two mixture components to be merged so as to

minimize the entropy of the resulting clustering. If components k and k' from a G-component solution are merged, the conditional probability $\hat{z}_{ig}$ will remain the same for every g except for k and k'. The new cluster $k \cup k'$ then has the following conditional probability:

$$\hat{z}_{i,k\cup k'} = \hat{z}_{ik} + \hat{z}_{ik'}. \tag{6.5}$$

The entropy for the resulting $(G-1)$-cluster solution is

$$-\sum_{i=1}^{n} \left\{ \sum_{g\neq k,k'} \hat{z}_{ig} \log \hat{z}_{ig} + \hat{z}_{i,k\cup k'} \log \hat{z}_{i,k\cup k'} \right\}. \tag{6.6}$$

The two components k and k' to be merged are those minimizing the criterion

$$\sum_{i=1}^{n} \left\{ \hat{z}_{ik} \log \hat{z}_{ik} + \hat{z}_{ik'} \log \hat{z}_{ik'} - \hat{z}_{i,k\cup k'} \log \hat{z}_{i,k\cup k'} \right\}$$

among all possible pairs of components. Components in the model selected by the BIC are successively merged by repeating the aforementioned procedure, until the data are reduced to one single cluster.

The proposed approach yields one solution for each value of $g = 1, 2, \ldots, G$, and the user can choose between them on substantive grounds. If a more automated procedure is desired for choosing a single solution, one possibility is to select, among the possible solutions, the solution providing the number of clusters selected by the ICL. An alternative is to detect an "elbow" on the entropy curve, i.e., the graph of entropy against the number of clusters. Intuitively, when mixture components overlap significantly, the corresponding entropy will be large. As overlapping components are combined in subsequent stages of the hierarchical merging scheme, the entropy will decrease. When only well-separated components are left in the clustering solution, further merging will incur little reduction in the resultant entropy. This idea has been formalized by Finak *et al.* (2009) in which a changepoint analysis is performed. On setting the changepoint at $g = 2, 3, \ldots, G-1$ in turn, a series of two-segment piecewise linear regression models is used to fit the

entropy curve. The optimal location $\tilde{g}$ of the changepoint is determined by the regression model with the minimum residual sum of squares. Finally, the presence or absence of such a changepoint may be determined by comparing the two-segment piecewise regression model with a simple linear regression model via the BIC or ANOVA. If the result is in favor of the two-segment piecewise regression model, the proposed hierarchical merging scheme is able to select a $\tilde{g}$-cluster solution as the optimal.

Bibliography

Bashashati, A., Lo, K., Gottardo, R., Gascoyne, R. D., Weng, A., and Brinkman, R. (2009). A pipeline for automated analysis of flow cytometry data: Preliminary results on lymphoma sub-type diagnosis. *Conference Proceedings of the IEEE Engineering in Medicine and Biology Society*, (In press).

Baudry, J.-P., Raftery, A. E., Celeux, G., Lo, K., and Gottardo, R. (2008). Combining mixture components for clustering. Submitted to *Journal of Computational and Graphical Statistics*.

Biernacki, C., Celeux, G., and Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):719–725.

Biernacki, C. and Govaert, G. (1997). Using the classification likelihood to choose the number of clusters. *Computing Science and Statistics*, 29:451–457.

Boedigheimer, M. J. and Ferbas, J. (2008). Mixture modeling approach to flow cytometry data. *Cytometry Part A*, 73A(5):421–429.

Breiman, L. (2001). Random forests. *Machine Learning*, 45:5–32.

Chan, C., Feng, F., Ottinger, J., Foster, D., West, M., and Kepler, T. B. (2008). Statistical mixture modeling for cell subtype identification in flow cytometry. *Cytometry Part A*, 73A:693–701.

Ding, C. and Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology*, 3(2):185–205.

Dogan, A. (2005). Modern histological classification of low grade B-cell lymphomas. *Best Practice and Research: Clinical Haematology*, 18(1):11–26.

Finak, G., Bashashati, A., Brinkman, R., and Gottardo, R. (2009). Merging mixture components for cell population identification in flow cytometry. Submitted to *Advances in Bioinformatics*.

Gottardo, R., Raftery, A. E., Yeung, K. Y., and Bumgarner, R. E. (2006). Bayesian robust inference for differential gene expression in microarrays with multiple samples. *Biometrics*, 62(1):10–18.

Herrera, G., Diaz, L., Martinez-Romero, A., Gomes, A., Villamon, E., Callaghan, R. C., and O'Connor, J. E. (2007). Cytomics: a multiparametric, dynamic approach to cell research. *Toxicology In Vitro*, 21(2):176–182.

Kendziorski, C., Newton, M., Lan, H., and Gould, M. N. (2003). On parametric empirical Bayes methods for comparing multiple groups using replicated gene expression profiles. *Statistics in Medicine*, 22:3899–3914.

Leroux, M. (1992). Consistent estimation of a mixing distribution. *Annals of Statistics*, 20:1350–1360.

Lizard, G. (2007). Flow cytometry analyses and bioinformatics: interest in new softwares to optimize novel technologies and to favor the emergence of innovative concepts in cell research. *Cytometry Part A*, 71A:646–647.

Lo, K., Brinkman, R. R., and Gottardo, R. (2008). Automated gating of flow cytometry data via robust model-based clustering. *Cytometry Part A*, 73A(4):321–332.

Lugli, E., Pinti, M., Nasi, M., Troiano, L., Ferraresi, R., Mussi, C., Salvioli, G., Patsekin, V., Robinson, J. P., Durante, C., Cocchi, M., and Cossarizza, A. (2007). Subject classification obtained by cluster analysis and principal component analysis applied to flow cytometric data. *Cytometry Part A*, 71A:334–344.

McLachlan, G. and Peel, D. (2000). *Finite Mixture Models*. Wiley-Interscience, New York.

Murphy, T. B. and Martin, D. (2003). Mixtures of distance-based models for ranking data. *Computational Statistics and Data Analysis*, 41(3–4):645–655.

Newton, M. C., Kendziorski, C. M., Richmond, C. S., Blattner, F. R., and Tsui, K. W. (2001). On differential variability of expression ratios: improving statistical inference about gene expression changes from microarray data. *Journal of Computational Biology*, 8:37–52.

Peng, H., Long, F., and Ding, C. (2005). Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1338.

Perfetto, S. P., Chattopadhyay, P. K., and Roederer, M. (2004). Seventeen-colour flow cytometry: unravelling the immune system. *Nature Reviews Immunology*, 4(8):648–655.

Pyne, S., Hu, X., Wang, K., Rossin, E., Lin, T.-I., Maier, L. M., Baecher-Allan, C., McLachlan, G. J., Tamayo, P., Hafler, D. A., De Jager, P. L., and Mesirov, J. P. (2009). Automated high-dimensional flow cytometric data analysis. *Proceedings of the National Academy of Sciences of the United States of America*, 106(21):8519–8524.

Schölkopf, B. and Smola, A. J. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. The MIT Press, Cambridge, Massachusetts.

Valet, G. (2005). Human cytome project, cytomics, and systems biology: the incentive for new horizons in cytometry. *Cytometry Part A*, 64A:1–2.

Valet, G. K. and Tárnok, A. (2003). Cytomics in predictive medicine. *Cytometry Part B: Clinical Cytometry*, 53B(1):1–3.

Appendix A

Additional Material for Chapter 2

A.1 Marginal Densities of Measured Intensities

Under the extended GG model, the joint marginal densities of measured intensities of a given gene g are developed without integrating a_g away, i.e., they are conditional on a_g. Denote by $\mathrm{G}(x; a, b)$ the Gamma density function with shape a and rate b. The explicit forms of the conditional marginal densities are given by

$$\begin{aligned} p_A(\mathbf{x}_g, \mathbf{y}_g|\boldsymbol{\psi}, a_g) &= \int_0^\infty \prod_{r=1}^{R} \mathrm{G}(x_{gr}; a_g, \theta_{gx})\ \mathrm{G}(\theta_{gx}; \boldsymbol{\psi})\ d\theta_{gx} \\ &\quad \times \int_0^\infty \prod_{r=1}^{R} \mathrm{G}(y_{gr}; a_g, \theta_{gy})\ \mathrm{G}(\theta_{gy}; \boldsymbol{\psi})\ d\theta_{gy} \\ &= \left\{ \frac{\Gamma(Ra_g + a_0)}{\Gamma^R(a_g)\ \Gamma(a_0)} \right\}^2 \frac{\nu^{2a_0} \big(\prod_r x_{gr}\ y_{gr} \big)^{a_g - 1}}{\big[(\nu + \sum_r x_{gr})(\nu + \sum_r y_{gr}) \big]^{Ra_g + a_0}} \end{aligned} \tag{A.1}$$

and

$$\begin{aligned} p_0(\mathbf{x}_g, \mathbf{y}_g|\boldsymbol{\psi}, a_g) &= \int_0^\infty \prod_{r=1}^{R} \mathrm{G}(x_{gr}; a_g, \theta_g) \prod_{r=1}^{R} \mathrm{G}(y_{gr}; a_g, \theta_g) \cdot \mathrm{G}(\theta_g; \boldsymbol{\psi})\ d\theta_g \\ &= \frac{\Gamma(2Ra_g + a_0)}{\Gamma^{2R}(a_g)\ \Gamma(a_0)} \frac{\nu^{a_0} \big(\prod_r x_{gr}\ y_{gr} \big)^{a_g - 1}}{\big(\nu + \sum_r x_{gr} + \sum_r y_{gr} \big)^{2Ra_g + a_0}}, \end{aligned} \tag{A.2}$$

where $\boldsymbol{\psi} = (a_0, \nu)'$.

The joint marginal densities of measured intensities under the extended LNN model are developed in a similar fashion, this time by integrating μ_g and τ_g away. Denote by $\mathrm{LN}(x; a, b)$ the Lognormal density function with mean and variance parameters a and b respectively, and by $\mathrm{N}(x; a, b)$ the normal density function. The marginal densities are developed as follows:

$$
\begin{aligned}
&p_A(\mathbf{x}_g, \mathbf{y}_g|\boldsymbol{\psi}) \\
&\quad = \int_0^\infty\!\!\int_{-\infty}^\infty \prod_r \mathrm{LN}(x_{gr}; \mu_{gx}, \tau_{gx}^{-1})\ \mathrm{N}(\mu_{gx}; m, k\tau_{gx}^{-1})\mathrm{G}(\tau_{gx}; \alpha, \beta)\ d\mu_{gx}d\tau_{gx} \\
&\qquad \times \int_0^\infty\!\!\int_{-\infty}^\infty \prod_r \mathrm{LN}(y_{gr}; \mu_{gy}, \tau_{gy}^{-1})\ \mathrm{N}(\mu_{gy}; m, k\tau_{gy}^{-1})\mathrm{G}(\tau_{gy}; \alpha, \beta)\ d\mu_{gy}d\tau_{gy} \\
&\quad = \frac{\beta^{2\alpha}\ \Gamma^2\big(\frac{R}{2}+\alpha\big)}{\big(\prod_r x_{gr}\ y_{gr}\big)(2\pi)^R\ (kR+1)\ \Gamma^2(\alpha)} \\
&\qquad \times \left\{\beta + \frac{1}{2k}\left[\frac{-\big(k\sum_r \log x_{gr} + m\big)^2}{kR+1} + k\textstyle\sum_r(\log x_{gr})^2 + m^2\right]\right\}^{-(\frac{R}{2}+\alpha)} \\
&\qquad \times \left\{\beta + \frac{1}{2k}\left[\frac{-\big(k\sum_r \log y_{gr} + m\big)^2}{kR+1} + k\textstyle\sum_r(\log y_{gr})^2 + m^2\right]\right\}^{-(\frac{R}{2}+\alpha)}
\end{aligned}
\tag{A.3}
$$

and

$$
\begin{aligned}
&p_0(\mathbf{x}_g, \mathbf{y}_g|\boldsymbol{\psi}) \\
&\qquad = \int_0^\infty\!\!\int_{-\infty}^\infty \prod_r \mathrm{LN}(x_{gr}; \mu_g, \tau_g^{-1}) \prod_r \mathrm{LN}(y_{gr}; \mu_g, \tau_g^{-1}) \\
&\qquad\quad \times \mathrm{N}(\mu_g; m, k\tau_g^{-1})\ \mathrm{G}(\tau_g; \alpha, \beta)\ d\mu_g d\tau_g \\
&\qquad = \frac{\beta^\alpha\ \Gamma(R+\alpha)}{\big(\prod_r x_{gr}\ y_{gr}\big)(2\pi)^R(2kR+1)^{\frac{1}{2}}\ \Gamma(\alpha)} \\
&\qquad\quad \times \left\{\beta + \frac{1}{2k}\left(\frac{-\big[k\big(\sum_r \log x_{gr} + \sum_r \log y_{gr}\big) + m\big]^2}{2kR+1}\right.\right. \\
&\qquad\qquad \left.\left. + k\Big[\textstyle\sum_r(\log x_{gr})^2 + \sum_r(\log y_{gr})^2\Big] + m^2\right)\right\}^{-(R+\alpha)},
\end{aligned}
\tag{A.4}
$$

where $\boldsymbol{\psi} = (m, k, \alpha, \beta)'$.

A.2 Estimation of η and ξ for the Prior of a_g

As mentioned in Section 2.2.3, to make use of the modified complete-data log-likelihood given by Eq.(2.4) in the extended GG model we need to provide estimates of the hyperparameters for the Lognormal(η, ξ) prior of a_g beforehand. Here we propose to use the method of moments (MM) to estimate η and ξ. First we would like to come up with simple estimates of the a_g's. On noting that the coefficient of variation is given by $1/\sqrt{a_g}$ for each gene, a robust empirical estimate of a_g may be provided by

$$\tilde{a}_g = \frac{\text{med}(\mathbf{x}_g, \mathbf{y}_g)^2}{\text{mad}(\mathbf{x}_g, \mathbf{y}_g)^2},$$

where med and mad stand for median and median absolute deviation respectively. Note that a robust counterpart to mean and standard deviation is adopted since there are usually relatively few replicates. With these crude estimates of a_g's, we can then obtain the estimates of η and ξ:

$$\hat{\eta} = \text{med}(\{\log\ \tilde{a}_g\}) \quad \text{and} \quad \hat{\xi} = \text{mad}(\{\log\ \tilde{a}_g\})^2.$$

Again, a robust version of MM is proposed here.

A.3 Initialization of the EM Algorithm

We need to initialize the parameters to be estimated before the EM-type algorithm described in Section 2.2.3 can be applied. Similar to the estimation for η and ξ above, robust MM estimates of (a, a_0, ν) are obtained for the extended GG model. Similar measure is taken for (m, α, β) if the data are modeled under the extended LNN framework, while k is empirically chosen to be 30. After the crude estimation step, updated estimates of the aforementioned parameters are obtained on maximizing the corresponding marginal null log-likelihood under either model formulation. This step is

taken in order to bring the initial estimates closer to the estimates returned by the EM algorithm. Using these initial estimates together with p set as 0.5, the most likely value under the Beta$(2, 2)$ prior, initial estimates of z_g's are obtained, which are then used to update the parameter estimates in the EM algorithm.

Appendix B

Vignette of the flowClust Package

B.1 Licensing

Under the Artistic License, you are free to use and redistribute this software. However, we ask you to cite the following papers if you use this software for publication.

1. Lo, K., Brinkman, R. R., and Gottardo, R. (2008). Automated gating of flow cytometry data via robust model-based clustering. *Cytometry Part A*, 73A(4):321–332.

2. Lo, K., Hahne, F., Brinkman, R. R., and Gottardo, R. (2009). flowClust: a Bioconductor package for automated gating of flow cytometry data. *BMC Bioinformatics*, 10:145.

B.2 Overview

We apply a robust model-based clustering approach proposed by Lo *et al.* (2008) to identify cell populations in flow cytometry data. The proposed approach is based on multivariate t mixture models with the Box-Cox transformation. This approach generalizes Gaussian mixture models by modeling outliers using t distributions and allowing for clusters taking non-ellipsoidal shapes upon proper data transformation. Parameter estimation is carried out using an Expectation-Maximization (EM) algorithm which simultaneously handles outlier identification and transformation selection. Please refer to Lo *et al.* (2008) for more details.

This **flowClust** package consists of a core function to implement the aforementioned clustering methodology. Its source code is built in C for optimal utilization of system resources. Graphical functionalities are available to users for visualizing a wealth of features of the clustering results, including the cluster assignment, outliers, and the size and shape of the clusters. The fitted mixture model may be projected onto any one/two dimensions and displayed by means of a contour or image plot. Currently, **flowClust** provides two options for estimating the number of clusters when it is unknown, namely, the Bayesian Information Criterion (BIC) and the Integrated Completed Likelihood (ICL).

flowClust is built in a way such that it is highly integrated with **flowCore**, the core package for flow cytometry that provides data structures and basic manipulation of flow cytometry data. Please read Section B.4.3 for details about actual implementation.

B.3 Installation

B.3.1 Unix/Linux/Mac Users

To build the **flowClust** package from source, make sure that the following is present on your system:

- a C compiler
- GNU Scientific Library (GSL)
- Basic Linear Algebra Subprograms (BLAS)

A C compiler is needed to build the package as the core function is coded in C. GSL can be downloaded at `http://www.gnu.org/software/gsl/`. In addition, the package uses BLAS to perform basic vector and matrix operations. Please go to `http://www.netlib.org/blas/faq.html#5` for a list of optimized BLAS libraries for a variety of computer architectures. For instance, Mac users may use the built-in vecLib framework, while users of Intel machines may use the Math Kernel Library (MKL).

For the package to be installed properly you may have to type the following command before installation:

```
export LD_LIBRARY_PATH='/path/to/GSL/:/path/to/BLAS/':
  $LD_LIBRARY_PATH
```

which will tell **R** where your GSL and BLAS libraries are. Note that this may have already been configured on your system, so you may not have to do so. In case you need to do it, you may consider including this line in your `.bashrc` such that you do not have to type it every time.

If GSL is installed to some non-standard location such that it cannot be found when installing **flowClust**, you may set the environment variable `GSL_CONFIG` to point to the correct copy of `gsl-config`, for example,

```
export GSL_CONFIG='/global/home/username/gsl-1.12/bin/gsl-config'
```

For convenience sake, this line may also be added to `.bashrc`.

Now you are ready to install the package:

```
R CMD INSTALL flowClust_x.y.z.tar.gz
```

The package will look for a BLAS library on your system, and by default it will choose gslcblas, which is not optimized for your system. To use an optimized BLAS library, you can use the `--with-blas` argument which will be passed to the `configure.ac` file. For example, on a Mac with vecLib pre-installed the package may be installed via:

```
R CMD INSTALL flowClust_x.y.z.tar.gz --configure-args=
  "--with-blas='-framework vecLib'"
```

On a 64-bit Intel machine which has MKL as the optimized BLAS library, the command may look like:

```
R CMD INSTALL flowClust\_x.y.z.tar.gz --configure-args="--with-
  blas='-L/usr/local/mkl/lib/em64t/ -lmkl -lguide -lpthread'"
```

where `/usr/local/mkl/lib/em64t/` is the path to MKL.

If you prefer to install a prebuilt binary, you need GSL for successful installation.

B.3.2 Windows Users

You need the GNU Scientific Library (GSL) for the **flowClust** package. GSL is freely available at `http://gnuwin32.sourceforge.net/packages/gsl.htm` for Windows distributions.

To install a prebuilt binary of **flowClust** and to load the package successfully you need to tell **R** where to link GSL. You can do that by adding `/path/to/libgsl.dll` to the `Path` environment variable. To add this you may right click on "My Computer", choose "Properties", select the "Advanced" tab, and click the button "Environment Variables". In the dialog box that opens, click "Path" in the variable list, and then click "Edit". Add `/path/to/libgsl.dll` to the "Variable value" field. It is important that the file path does not contain any space characters; to avoid this you may simply use the short forms (8.3 DOS file names) found by typing `dir /x` at the Windows command line. For example, the following may be added to the `Path` environment variable:

```
C:/PROGRA~1/GNUWIN32/bin
```

and the symbol `;` is used to separate it from existing paths.

To build **flowClust** from source (using Rtools), in addition to adding `/path/to/libgsl.dll` to `Path`, you need to tell **flowClust** where your GSL library and header files are. You can do that by setting up two environment variables `GSL_LIB` and `GSL_INC` with the correct path to the library files and header files respectively. You can do this by going to the "Environment Variables" dialog box as instructed above and then clicking the "New" button. Enter `GSL_LIB` in the "Variable name" field, and `/path/to/your/gsl/lib/directory` in the "Variable value" field. Likewise, do this for `GSL_INC` and `/path/to/your/gsl/include/directory`. Remember to use "/" instead of "\" as the directory delimiter.

You can download Rtools at `http://www.murdoch-sutherland.com/Rtools/` which provides the resources for building **R** and **R** packages. You should add to the `Path` variable the paths to the various components of Rtools. Please read the "Windows Toolset" appendix at `http://cran.`

r-project.org/doc/manuals/R-admin.html#The-Windows-toolset for more details.

B.4 Example: Clustering of the Rituximab Dataset

B.4.1 The Core Function

To demonstrate the functionality we use a flow cytometry dataset from a drug-screening project to identify agents that would enhance the anti-lymphoma activity of Rituximab, a therapeutic monoclonal antibody. The dataset is an object of class `flowFrame`; it consists of eight parameters, among them only the two scattering parameters (`FSC.H`, `SSC.H`) and two fluorescence parameters (`FL1.H`, `FL3.H`) are of interest in this experiment. Note that, apart from a typical `matrix` or `data.frame` object, **flowClust** may directly take a `flowFrame`, the standard **R** implementation of an FCS file, which may be returned from the `read.FCS` function in the **flowCore** package, as data input. The following code performs an analysis with one cluster using the two scattering parameters:

```
> library(flowClust)
> data(rituximab)
> summary(rituximab)

          FSC.H  SSC.H FL1.H FL2.H  FL3.H   FL1.A FL1.W Time
Min.       59.0   11.0   0.0   0.0    1.0    0.00   0.0    2
1st Qu.   178.0  130.0 197.0  55.0  150.0    0.00   0.0  140
Median    249.0  199.0 244.0 116.0  203.0    0.00   0.0  285
Mean      287.1  251.8 349.2 126.4  258.3   73.46  17.6  294
3rd Qu.   331.0  307.0 445.0 185.0  315.0    8.00   0.0  451
Max.     1023.0 1023.0 974.0 705.0 1023.0 1023.00 444.0  598

> res1 <- flowClust(rituximab, varNames=c("FSC.H", "SSC.H"),
  K=1, B=100)
```

`B` is the maximum number of EM iterations; for demonstration purpose here we set a small value for `B`. The main purpose of performing an analysis with one cluster here is to identify outliers, which will be removed from subsequent analysis.

Next, we would like to proceed with an analysis using the two fluorescence parameters on cells selected from the first stage. The following code performs the analysis with the number of clusters being fixed from one to six in turn:

```
> rituximab2 <- rituximab[rituximab %in% res1,]
> res2 <- flowClust(rituximab2, varNames=c("FL1.H", "FL3.H"),
  K=1:6, B=100)
```

We select the best model based on the BIC. Values of the BIC can be retrieved through the `criterion` method. By inspection, the BIC values stay relatively constant beyond three clusters. We therefore choose the model with three clusters and print a summary of the corresponding clustering result:

```
> summary(res2[[3]])

** Experiment Information **
Experiment name: Flow Experiment
Variables used: FL1.H FL3.H
** Clustering Summary **
Number of clusters: 3
Proportions: 0.2658702 0.5091045 0.2250253
** Transformation Parameter **
lambda: 0.4312673
** Information Criteria **
Log likelihood: -16475.41
BIC: -33080.88
ICL: -34180.67
** Data Quality **
Number of points filtered from above: 0 (0%)
```

```
Number of points filtered from below: 0 (0%)
Rule of identifying outliers: 90% quantile
Number of outliers: 96 (6.99%)
Uncertainty summary:
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
0.0005699 0.0153000 0.1669000 0.2019000 0.3738000 0.5804000
```

The summary states that the rule used to identify outliers is `90% quantile`, which means that a point outside the 90% quantile region of the cluster to which it is assigned will be called an outlier. To specify a different rule, we make use of the `ruleOutliers` replacement method. The example below applies the more conservative `95% quantile` rule to identify outliers:

```
> ruleOutliers(res2[[3]]) <- list(level=0.95)
Rule of identifying outliers: 95% quantile

> summary(res2[[3]])
...
** Data Quality **
Number of points filtered from above: 0 (0%)
Number of points filtered from below: 0 (0%)
Rule of identifying outliers: 95% quantile
Number of outliers: 35 (2.55%)
```

We can also combine the rule set by the `z.cutoff` argument to identify outliers. Suppose we would like to assign an observation to a cluster only if the associated posterior probability is greater than 0.6. We can add this rule with the following command:

```
> ruleOutliers(res2[[3]]) <- list(z.cutoff=0.6)

Rule of identifying outliers: 95% quantile,
                              probability of assignment < 0.6

> summary(res2[[3]])
```

```
...
** Data Quality **
Number of points filtered from above: 0 (0%)
Number of points filtered from below: 0 (0%)
Rule of identifying outliers: 95% quantile,
                              probability of assignment < 0.6
Number of outliers: 317 (23.07%)
```

This time more points are called outliers. Note that such a change of the rule will not incur a change of the model-fitting process. The information about which points are called outliers is conveyed through the `flagOutliers` slot, a logical vector in which the positions of `TRUE` correspond to points being called outliers.

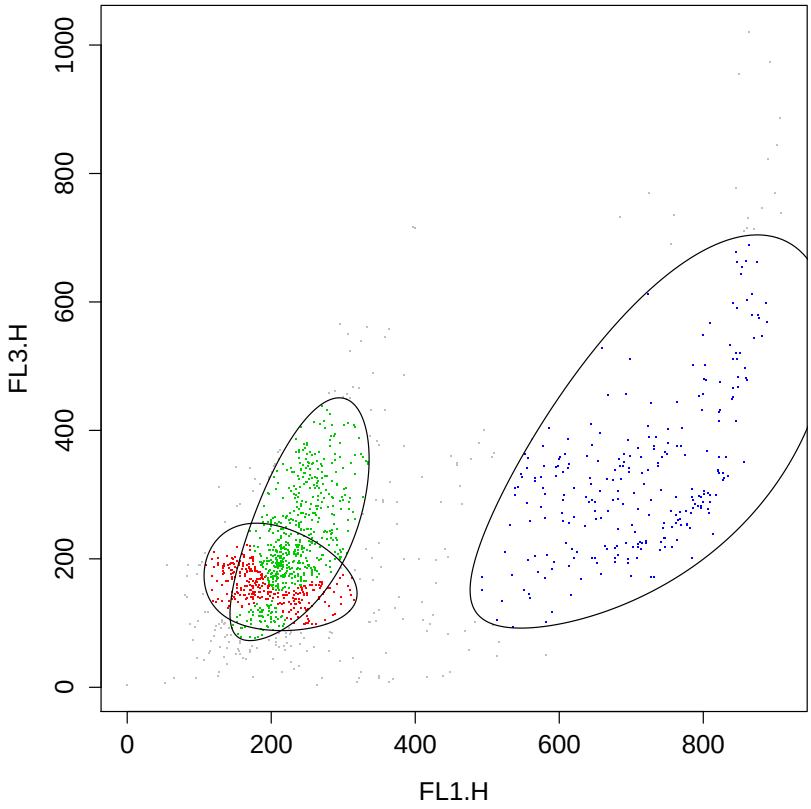
By default, when 10 or more points accumulate on the upper or lower boundary of any parameter, the `flowClust` function will filter those points. To change the threshold count from the default, users may specify `max.count` and `min.count` when running `flowClust`. To suppress filtering at the upper and/or the lower boundaries, set `max.count` and/or `min.count` as -1. We can also use the `max` and `min` arguments to control filtering of points, but from a different perspective. For instance, if we are only interested in cells which have a `FL1.H` measurement within $(0, 400)$ and `FL3.H` within $(0, 800)$, we may use the following code to perform the cluster analysis:

```
> flowClust(rituximab2, varNames=c("FL1.H", "FL3.H"), K=2,
  B=100, min=c(0,0), max=c(400,800))
```

B.4.2 Visualization of Clustering Results

Information such as the cluster assignment, cluster shape and outliers may be visualized by calling the `plot` method to make a scatterplot:

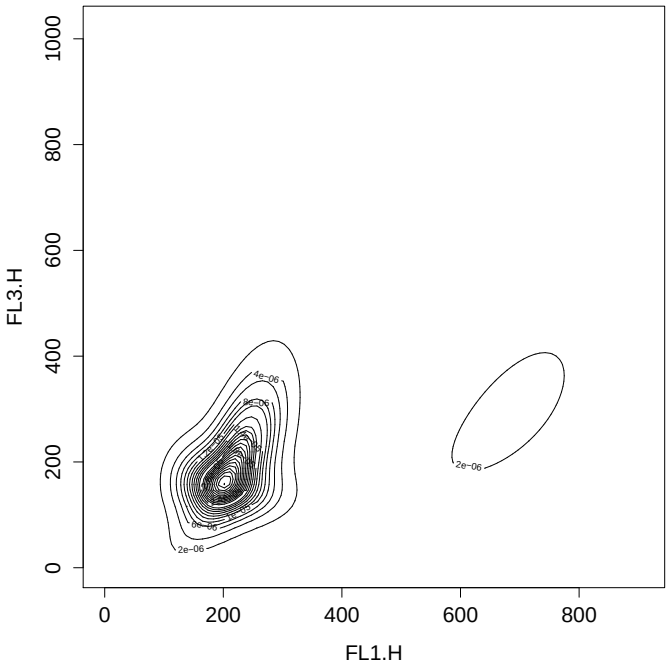
```
> plot(res2[[3]], data=rituximab2, level=0.8, z.cutoff=0)

Rule of identifying outliers: 80% quantile
```



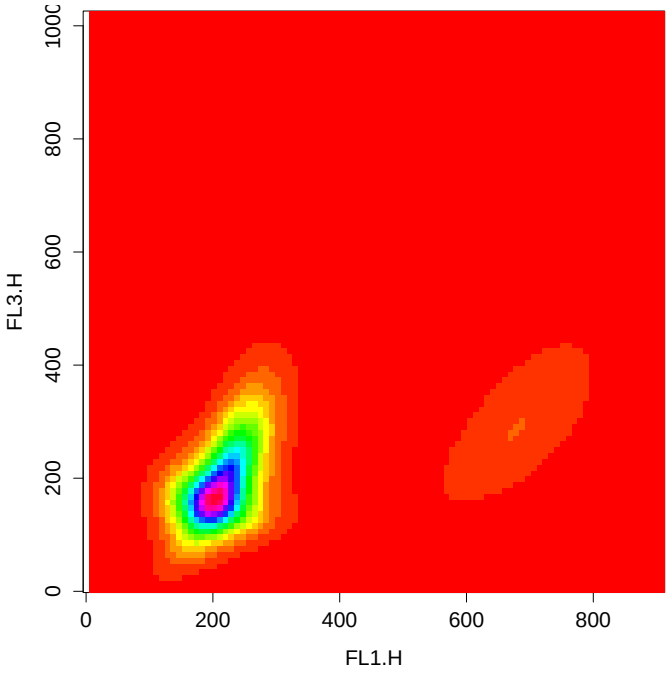
The `level` and/or `z.cutoff` arguments are needed when we want to apply a rule different from that stored in the `ruleOutliers` slot of the `flowClust` object to identify outliers.

To look for densely populated regions, a contour/image plot can be made:

```
> res2.den <- density(res2[[3]], data=rituximab2)
> plot(res2.den)
```

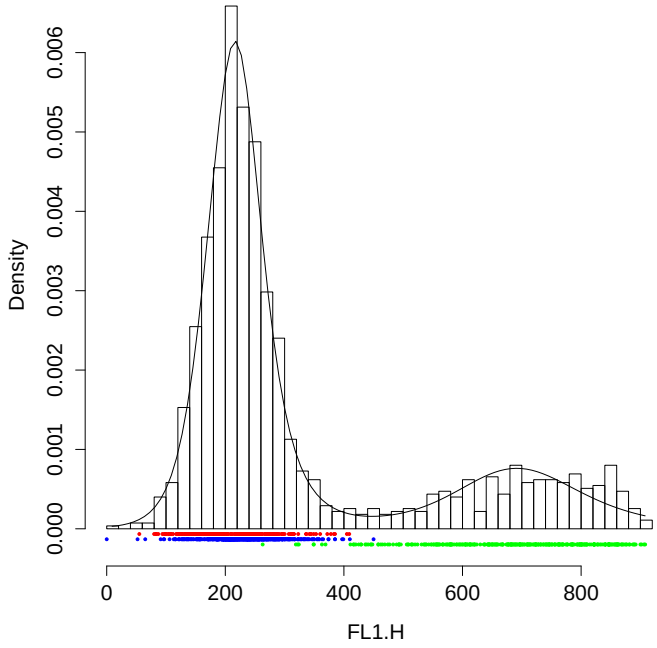



```
> plot(res2.den, type="image")
```



When we want to examine how the fitted model and/or the data are distributed along one chosen dimension, we can use the `hist` method:

```
> hist(res2[[3]], data=rituximab2, subset="FL1.H")
```



The `subset` argument may also take a numeric value:

```
> hist(res2[[3]], data=rituximab2, subset=1)
```

Since `FL1.H` is the first element of `res2[[3]]@varNames`, this line produces exactly the same histogram as the one generated by the line taking `subset="FL1.H"`. Likewise, the `subset` argument of both `plot` methods accepts either a numeric or a character vector to specify which two variables are to be shown on the plot.

B.4.3 Integration with flowCore

As mentioned in Overview, effort has been made to integrate **flowClust** with the **flowCore** package. Users will find that most methods defined in **flowCore** also work in the context of **flowClust**.

The very first step of integration is to replace the core function `flowClust` with a call to the constructor `tmixFilter` followed by the `filter` method. The aim is to wrap clustering in a filtering operation like those found in **flowCore**. The `tmixFilter` function creates a `filter` object to store all settings required for the filtering operation. The object created is then passed to the actual filtering operation implemented by the `filter` method. The use of a dedicated `tmixFilter`-class object separates the task of specifying the settings (`tmixFilter`) from the actual filtering operation (`filter`), facilitating the common scenario in FCM gating analysis that filtering with the same settings is performed upon a set of data files.

As an example, the filtering operation that resembles the second-stage clustering using `FL1.H` and `FL3.H` with three clusters (see Section B.4.1) is implemented by the following code:

```
> s2filter <- tmixFilter("s2filter", c("FL1.H", "FL3.H"), K=3,
  B=100)
> res2f <- filter(rituximab2, s2filter)
```

The object `res2f` is of class `tmixFilterResult`, which extends the `multipleFilterResult` class defined in **flowCore**. Users may apply various subsetting operations defined for the `multipleFilterResult` class in a similar fashion on a `tmixFilterResult` object:

```
> Subset(rituximab2, res2f)

flowFrame object 'A02'
with 1267 cells and 8 observables:
     name                desc range
$P1 FSC.H          FSC-Height  1024
$P2 SSC.H        Side Scatter  1024
$P3 FL1.H     Anti-BrdU FITC   1024
$P4 FL2.H                <NA>  1024
$P5 FL3.H              7 AAD   1024
$P6 FL1.A                <NA>  1024
$P7 FL1.W                <NA>  1024
```

```
$P8  Time Time (204.80 sec.)  1024
135 keywords are stored in the 'descripton' slot

> split(rituximab2, res2f, population=list(sc1=1:2, sc2=3))

$sc1
flowFrame object 'A02 (1,2)'
with 976 cells and 8 observables:
     name               desc range
$P1 FSC.H         FSC-Height  1024
$P2 SSC.H       Side Scatter  1024
$P3 FL1.H     Anti-BrdU FITC  1024
$P4 FL2.H               <NA>  1024
$P5 FL3.H             7 AAD   1024
$P6 FL1.A               <NA>  1024
$P7 FL1.W               <NA>  1024
$P8  Time Time (204.80 sec.)  1024
3 keywords are stored in the 'descripton' slot

$sc2
flowFrame object 'A02 (3)'
with 291 cells and 8 observables:
     name               desc range
$P1 FSC.H         FSC-Height  1024
$P2 SSC.H       Side Scatter  1024
$P3 FL1.H     Anti-BrdU FITC  1024
$P4 FL2.H               <NA>  1024
$P5 FL3.H             7 AAD   1024
$P6 FL1.A               <NA>  1024
$P7 FL1.W               <NA>  1024
$P8  Time Time (204.80 sec.)  1024
136 keywords are stored in the 'descripton' slot
```

The `Subset` method above outputs a `flowFrame` consisting of observations within the data-driven filter constructed. The `split` method separates the data into two populations upon the removal of outliers: the first population is formed by observations assigned to clusters 1 and 2 constructed by the filter, and the other population consists of observations assigned to cluster 3. The two populations are returned as two separate `flowFrame`'s, which are stored inside a list and labelled with `sc1` and `sc2` respectively.

The `%in%` operator from **flowCore** is also defined for a `tmixFilterResult` object. A logical vector will be returned in which a `TRUE` value means that the corresponding observation is accepted by the filter. In fact, the implementation of the `Subset` method needs to call `%in%`.

The object returned by `tmixFilter` is of class `tmixFilter`, which extends the `filter` class in **flowCore**. Various operators, namely, `&`, `|`, `!` and `%subset%`, which have been constructed for `filter` objects in **flowCore**, also produce similar outcomes when applied to a `tmixFilter` object. For example, to perform clustering on a subset of data enclosed by a rectangle gate, we may apply the following code:

```
> rectGate <- rectangleGate(filterId="rectRegion",
  "FL1.H"=c(0, 400), "FL3.H"=c(0, 800))
> MBCfilter <- tmixFilter("MBCfilter", c("FL1.H", "FL3.H"),
  K=2, B=100)
> filter(rituximab2, MBCfilter %subset% rectGate)

A filterResult produced by the filter named 'MBCfilter in
rectRegion'
```

Appendix C

Code to Produce the Plots in Chapter 5

```
# Figure 5.1
plot(criterion(res1, "BIC"), xlab="No. of clusters", ylab="BIC",
  type="b", pch=2)
points(criterion(res1s, "BIC"), type="b", pch=3, col=2)
legend("bottomright", col=1:2, pch=2:3, lty=1,
  legend=c(expression(paste("global ", lambda)),
  expression(paste("cluster-specific ", lambda))))

# Figure 5.2
plot(res1[[4]], data=GvHD, pch.outliers="+", xlab="FSC-Height",
  ylab="SSC-Height")
legend("bottomright", col=2:5, legend=1:4, title="Clusters",
  pch=20)

# Figure 5.3
plot(criterion(res2, "BIC"), xlab="No. of clusters", ylab="BIC",
  type="b")

# Figure 5.4(a)
CD3p <- which(getEstimates(res2[[11]])$locations[,3] > 280)
plot(res2[[11]], data=GvHD2$lymphocyte, include=CD3p, ellipse=F,
  pch=20, xlab="CD4-Height", ylab=expression(paste("CD8", beta,
  "-Height")))
```

```
res2d <- density(res2[[11]], data=GvHD2$lymphocyte, include=CD3p)
plot(res2d, drawlabels=F, add=T, nlevels=20)

# Figure 5.4(b)
plot(res2d, type="image", nlevels=100, xlab="CD4-Height", ylab=
  expression(paste("CD8", beta, "-Height")))
```