# Detection and Characterization of Multilevel Genomic Patterns

Yuanjian Feng

Dissertation submitted to

the faculty of the Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

In

Computer Engineering

Yue Wang, Committee Chair

Jianhua Xuan

Chang-Tien Lu

Christopher L. Wyatt

Luiz A. DaSilva

May 26, 2010

Arlington, Virginia

Keywords: Regression Analysis, Tree of Phenotypes, Stability Analysis, DNA Copy Number Changes, Gene Expressions.

# Detection and Characterization of Multilevel Genomic Patterns

**Yuanjian Feng**

## ABSTRACT

DNA microarray has become a powerful tool in genetics, molecular biology, and biomedical research. DNA microarray can be used for measuring the genotypes, structural changes, and gene expressions of human genomes. Detection and characterization of multilevel, high-throughput microarray genomic data pose new challenges to statistical pattern recognition and machine learning research. In this dissertation, we propose novel computational methods for analyzing DNA copy number changes and learning the trees of phenotypes using DNA microarray data.

DNA copy number change is an important form of structural variations in human genomes. The copy number signals measured by high-density DNA microarrays usually have low signal-to-noise ratios and complex patterns due to inhomogeneous composition of tissue samples. We propose a robust detection method for extracting copy number changes in a single signal profile and consensus copy number changes in the signal profiles of a population. We adapt a solution-path algorithm to efficiently solve the optimization problems associated with the proposed method. We tested the proposed method on both simulation and real CGH and SNP microarray datasets, and observed competitively improved performance as compared to several widely-adopted copy number change detection methods. We also propose a chromosome instability measure to summarize the extracted copy number changes for assessing chromosomal instabilities of tumor genomes. The proposed measure demonstrates distinct patterns between different subtypes of ovarian serous carcinomas and normal samples.

Among active research on complex human diseases using genomic data, little effort and progress have been made in discovering the relational structural information embedded in the molecular data. We propose two stability analysis based methods to learn stable and highly resolved trees of phenotypes using microarray gene expression data of heterogeneous diseases. In the first method, we use a hierarchical, divisive visualization approach to explore the tree of phenotypes and a leave-one-out cross validation to select stable tree structures. In the second method, we propose a node bandwidth constraint to construct stable trees that can balance the descriptive power and reproducibility of tree structures. Using a top-down merging procedure, we modify the binary tree structures learned by hierarchical group clustering methods to achieve a given node bandwidth. We use a bootstrap based stability analysis to select stable tree structures under different node bandwidth constraints. The experimental results on two microarray gene expression datasets of human diseases show that the proposed methods can discover stable trees of phenotypes that reveal the relationships between multiple diseases with biological plausibility.

# Acknowledgements

I am grateful to my Ph.D. advisor, Dr. Yue Wang, for his guidance and support throughout these years. Dr. Wang has been giving me invaluable advice and inspiration in my research work. I am grateful for his confidence in my capability as a researcher, and for providing opportunities to strengthen my research capability. I am grateful for his encouragement and for being patient with me when I meet difficulties.

I am grateful to my colleagues and former members in the Computational Bioinformatics and Bio-imaging Laboratory, Chen Wang, Bai Zhang, Lei Song, Guoqiang Yu, Jianghui Xiong, Dr. Zuyi Wang, and Dr. Yitan Zhu, for sharing their knowledge, experience and insight, and for the exciting discussions.

I am grateful to Dr. Jianhua Xuan for giving me many suggestions in my research work and projects. I am also grateful to the other members of my dissertation committee, Dr. Chang-Tien Lu, Dr. Christopher Wyatt, and Dr. Luiz DaSilva, for their time and suggestions for my dissertation work. I want to thank Dr. Ie-Ming Shih, Dr. Tian-Li Wang, Dr. Bin Guan, and Dr. Kuan-Ting Kuo at Johns Hopkins Medical Institutions, for their guidance and suggestions in our collaborative research.

Finally, none of my achievements would be possible without the unconditional love and support from my wife, Jiajing Wang, my mother, and my brother. I am indebted to them for their endless love and confidence in me.

# Table of Contents

# List of Figures

# List of Tables

# List of Acronyms

| | |
|---|---|
| AUC | Area Under the Curve |
| bp | Base Pair |
| CIN | Chromosome INstability |
| CNA | Copy Number Alteration |
| CNV | Copy Number Variation |
| EM | Expectation Maximization |
| FMR | Fused Margin Regression |
| FPR | False Positive Rate |
| HC | Hierarchical Clustering |
| HMM | Hidden Markov Model |
| i.i.d. | Independently and Identically-Distributed |
| kb | Kilo Base pairs |
| LOO | Leave-One-Out |
| LP | Linear Programming |
| Mb | Mega Base pairs |
| MDL | Minimum Description Length |
| OVR | One-Versus-Rest |
| pdf | Probability Density Function |
| pmf | Probability Mass Function |
| ROC | Receiver Operating Characteristic |
| SD | Standard Deviation |
| SFNM | Standard Finite Normal Mixture |

| | |
|---|---|
| SIC | Schwartz Information Criterion |
| SNP | Single Nucleotide Polymorphism |
| SNR | Signal-to-Noise Ratio |
| SSE | Sum of Squared Errors |
| SVM | Support Vector Machine |
| SVR | Support Vector Regression |
| TOP | Tree Of Phenotypes |
| TPR | True Positive Rate |

# 1 Introduction

The advances of biotechnology have been changing the practice of biomedical research in the past few decades. New methods for genotyping/sequencing DNA and measuring gene activities, represented by DNA microarrays, have become powerful tools for molecular and genome biology research. DNA microarrays can generate whole genome, high-throughput, and quantitative measurements of human genomes and their dynamic gene expression products. These molecular level data allows researchers to use mathematical and statistical methods to analyze the causes, initiations, progressions, and outcomes of human diseases and their responses to treatments in an unprecedentedly fine yet broad scope.

## 1.1 Background

DNA (Deoxyribonucleic Acid) is a polymeric molecule carrying the genetic instructions for the development and maintaining of a living organism (Brown, 2007). Conceptually, DNA has a double helix structure – it consists of two intertwisted strands of nucleotide bases, including adenine (A), cytosine (C), guanine (G), and thymine (T). The two strands are complementary to each other and contain essentially the same genetic information: adenine/guanine in one strand is always paired with thymine/cytosine in the other. Therefore, DNA can be conveniently considered as a sequence of the four symbols, A, T, C, and G. The DNA sequence contains protein- and RNA (Ribonucleic Acid)-coding genes and other non-coding sequences. In human somatic cells, DNA molecules are packed into 46 chromosomes constituting 23 homologous pairs. Each pair of chromosomes contains genetic information inherited from the parents. The DNA sequences of the 23 pairs of chromosomes form a human genome, which consists of about $3\times10^9$ nucleotide base pairs.

### 1.1.1 DNA Copy Number Change

There are considerable amount of differences between the genomes of different individuals. Single Nucleotide Polymorphisms (SNPs) and structural variations in genomes are the two major genetic variations currently being intensively investigated. A SNP is a single base variation, usually having two alleles (variants), at a particular locus of the human genomes in the population – some individuals have one allele (a particular nucleotide base) and others have the other allele. It was once believed that 99.9% of the human genomes are the same, and SNPs account for majority of the 0.1% differences (Check, 2005); but recent studies show that structural variations, including DNA copy number variations (CNVs), insertions, inversions, and translocations of DNA segments, present with a much larger scale in human genomes compared with SNPs (Feuk, et al., 2006), and CNV alone covers more than 12% of the genetic content (Redon, et al., 2006). CNV, or germline copy number change, refers to the change in the number of copies of a DNA segment in a target genome compared with a reference genome. CNVs are potentially associated with phenotypic variations (White, et al., 2007) and disease susceptibilities (The Wellcome Trust Case Control Consortium, 2010).

Besides inherited genomic variations, *de novo* mutations can be introduced in the DNA sequence in many ways, such as cell division, DNA replication, environmental factors, etc. *De novo* deletions and duplications of DNA segments, which are called Copy Number Alterations (CNAs), can alter the gene dosage and interfere with the functioning of genes. In cancer research, CNAs are considered hallmarks of tumorigenesis (Kuo, et al., 2009). Deletions of DNA segments can cause the losses of tumor suppressor genes, while duplications can increase the dosages of oncogenes and induce drug resistance.

During the past few decades, technologies for measuring DNA copy number changes have been quickly evolving and the resolutions of measurements and coverage of the genome have been greatly improved. DNA copy number changes have long been known associated with diseases; but limited by the availability of technology, researchers were only able to study microscopic level, chromosome-scale structural changes (Carter, 2007). Fluorescence *in situ* Hybridization (FISH) improves the resolution of detectable copy number changes to a few megabase-long (~$10^6$ to $10^7$ base pairs). In FISH, the copy number of a particular DNA segment is detected by hybridizing fluorescently labeled probes directly to the chromosome and observing the fluorescence intensity under microscope. Clone based Comparative Genomic Hybridization (CGH) uses large-insert DNA clones as the probes, whose lengths are between $10^5$ to $2\times10^5$ base pairs (Ylstra, et al., 2006). In clone based CGH, a reference DNA sample and a target DNA sample are labeled with different fluorophores and competitively hybridized to a substrate spotted with probes. The relative fluorescence intensity of a probe is, ideally, proportional to the relative copy number between the inspected DNA segment in the target sample and that in the reference sample. Based on the same comparative hybridization mechanism, oligonucleotide based CGH microarrays use 25 to 50 base-long probes to measure copy numbers at preselected loci in the genome. Compared with clone based CGH, oligonucleotide based CGH array can generate 1- to 2-fold more measurements. In addition to genotyping SNPs, SNP microarray, which uses oligonucleotides as the probes as well, can also be used to measure DNA copy numbers. The most recent SNP array platforms incorporate copy number probes uniformly distributed along the chromosomes to further increase the resolution of measurements (McCarroll, et al., 2008). Commercial SNP/CNV arrays and oligonucleotide based CGH arrays are the most popular platforms for analyzing CNV/CNA not only due to their higher resolutions and whole-genome coverage, but also because of the more mature array manufacturing technologies and widely practiced standard experiment protocols. In Chapter 2, we will further elucidate the differences between CGH and SNP arrays and the implications of those differences in computational analysis of DNA copy numbers.

Due to the importance of CNV/CNA in biomedical research and the availability of DNA microarray technologies, many large-scale database projects have been established or initiated for various purposes. Based on the HapMap sample collection, Redon et al (2006) created a copy number dataset of 270 individuals from four ethnic groups using CGH and SNP arrays. They proposed the first whole-genome human CNV map based on the dataset. The Centre for Applied Genomics at the Hospital for Sick Children, Canada, hosts the Database of Genomic Variants (Iafrate, et al., 2004; Zhang, et al., 2006) that catalogs structural variations in the human genomes published in literatures. There are about 58,000 entries of CNVs among the total 90,000 entries of structural variations in the database. The Cancer Genome Atlas (TCGA) project of NCI and NHGRI (The Cancer Genome Atlas Research Network, 2008) is a comprehensive collection of genomic data for studying human cancers, where the DNA copy number alteration dataset is an important part of the collection. The copy number data is generated using an oligonucleotide based CGH array platform and two SNP microarray platforms. These database projects provide invaluable resources for studying DNA copy number changes in genetics, epidemiology, and complex diseases.

From the measurements of DNA abundance in a sample, we can estimate the copy numbers of DNA segments through a series of computational analyses. After hybridization, the fluorescence intensities of the probes on a microarray are captured as a digital image, which is then converted to intensity signals of probes by image processing algorithms. For CGH arrays, the comparative hybridization signals effectively eliminate experiment-related biases across different arrays and can be analyzed directly to detect copy number changes. For SNP microarrays, the pre-processing step is more complicated. To measure a particular genomic locus, a probe set consisting of multiple probes is designed and manufactured on the array. The intensity signals of the probes in a set are summarized as a single intensity value for the targeted locus (Li and Wong, 2001). These probes are designed to be slightly different in their sequences of nucleotide bases in order to provide contrast in the hybridization signals and enhance the reliability of the summarized intensity. In SNP array assays, a single DNA sample is hybridized to an array; the systematic difference between the intensity signal levels of different arrays has to be removed before analyzing the arrays as a batch (Li and Wong, 2001; Zhao, et al., 2004). For both CGH and SNP arrays, the core step of analyses after pre-processing is to estimate the copy numbers of DNA segments and extract frequently altered DNA regions in multiple genomes, which are the main focuses of this dissertation. We will provide a thorough discussion of the topics and propose novel algorithms in Chapter 2.

### 1.1.2 Gene Expression Analysis

The instructions for maintaining the activity of a living organism are stored in DNA sequences. One of the most important processes for utilizing these instructions is gene expression: protein-coding genes are transcribed to messenger RNAs (mRNAs), and mRNAs are then translated to proteins, which are the essential components of all biological processes. Gene expression levels can be quantitatively measured by DNA microarrays, which can monitor hundreds of to tens of thousands of genes simultaneously. cDNA (complementary DNA) and oligonucleotide based microarrays are the two most widely used platforms. Both platforms measure the abundance of mRNAs in the sample, which may indicate the amount of the final products of gene expressions, i.e. proteins. cDNA array is usually used to measure the relative expression levels between differentially labeled target and reference samples (two-channel signals), while oligonucleotide array is more often used to measure the expression levels of a single fluorescently labeled sample (one-channel signals). The signal intensities of one-channel microarrays need to be normalized in order to compare gene expression levels between multiple samples or experiments.

The expression levels of multiple genes measured by DNA microarrays can be considered collectively as a snapshot of the activities of the genes. Computational analyses of gene expression data include phenotype prognosis/diagnosis, drug response prediction, discovery of new disease subtypes, biomarker selection, etc (Wang, et al., 2008). Although all these problems can find their corresponding fields in pattern recognition and machine learning research, the unique properties of small sample size and high dimensionality in microarray gene expression data pose new challenges to the biomedical research community.

Gene expression analyses can be roughly categorized into supervised and unsupervised problems depending on the availability of phenotypic information (Jain, et al., 2000). For supervised learning, in addition to the expression levels of $m$ genes measured on $n$ samples, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^{\mathrm{T}}$ , we have an outcome or response, $y_i$ , associated with each sample. The outcome is categorical/continuous in classification/regression problems, respectively. The purpose of supervised learning is to estimate the dependency $f$ between the outcome and the genes based on the data $\mathbf{y} = \{y_1, \ldots, y_n\}$ and $\mathbf{X}$ in order to make generalizable predictions of the outcomes using $f$ on unseen data, i.e. the gene expression levels of new samples. In biomedical research or clinical applications, the phenotypic information can be disease/control labels, disease phenotypes, surviving time, etc. The purpose of unsupervised learning, where the phenotypic information is not available, is to discover cluster structures (e.g., disease subtypes) in the gene expression data using clustering algorithms. The general idea of clustering is to find groups in the given data such that samples within the same group are more similar to each other than those in different groups (Jain, et al., 2000). For gene expression data, clustering can be performed on samples or on genes, where sample clustering is a more challenging task due to the small sample size and high dimensionality of the data.

With the decreasing cost of DNA microarray assays, more and more human diseases, especially cancers, are being studied using gene expression data. Although many classification methods have been proposed for diagnosis/prognosis of multi-class, heterogeneous diseases (Ramaswamy, et al., 2001; Statnikov, et al., 2005; Tibshirani, et al., 2002), little attention has been paid to study the relationships between multiple diseases (e.g., phenotypic or mechanistic relationships between different cancers) that remains a very challenging task in machine learning research. Some classification methods utilize tree-structured hierarchical classification schemes (Park and Hastie, 2005; Tibshirani and Hastie, 2007), where the trees are only the by-product of classifier training and they often have highly unstable structures on small sample size gene expression data. On the other hand, clustering methods that generate coarse-to-fine hierarchies of samples have long been used in pattern recognition (Duda, et al., 2001) and phylogenetic studies (Felsenstein, 2004). The importance of analyzing hierarchical cluster structures embedded in data is further emphasized in (Wang, et al., 2000; Wang, et al., 2003; Zhu, et al., 2008). In Chapter 3 of the dissertation, we extend the idea of structure learning to discover reproducible and highly resolved trees of phenotypes (TOP) using labeled microarray gene expression data.

## 1.2 Research Topics and Major Contributions

In this dissertation, our research work mainly focuses on the following two problems: (1) computational analysis of DNA copy number changes and (2) learning tree of phenotypes using microarray gene expression data. We will report the detailed research considerations and outcomes in Chapter 2 and Chapter 3 respectively. Specifically, since the two topics have their unique biological motivations and challenges, we will give more detailed discussions on the biological background, existing methodologies, technical challenges, and our approaches and contributions in the main contents of these dedicated chapters. Here we highlight our major contributions on these two topics.

For the computational analysis of DNA copy number changes:

1. We propose a unified framework, namely Fused Margin Regression (FMR), for detecting (a) copy number changes in a single signal profile and (b) consensus copy number changes in population data;
2. We adapt a solution-path algorithm (Li and Zhu, 2007; Zhu, et al., 2004) to efficiently solve the FMR optimization problems;
3. We propose a Chromosome Instability (CIN) index that summarizes DNA copy number changes as a chromosomal instability measure.

For learning tree of phenotypes using microarray gene expression data:

1. We propose a leave-one-out stability analysis based human interactive visualization method, namely Color-Coded Supervised Mode VIsual Statistical Data Analyzer (SA-ccsmVISDA), for learning tree of phenotypes;
2. We introduce the concept of node bandwidth of trees as a parameter for controlling the trade-off between the reproducibility and complexity of tree structures;
3. We propose a bootstrap stability analysis based automatic tree learning method for generating a spectrum of stable tree structures under different node bandwidth constraints.

In addition to the methodology development in these two main topics, we have also applied the proposed methods/algorithms to support biomedical research performed in the laboratories of our biomedical collaborators. The related activities and contributions include:

1. We analyzed DNA copy number alterations in three subtypes of ovarian serous carcinomas (serous borderline tumor, low-grade and high-grade serous carcinomas) and reference normal samples, and measured their chromosome instabilities using the proposed CIN index, which shows distinct patterns between the four phenotypes. The genome-wide CIN indices also demonstrate significant statistical differences between the phenotypes. The results are published in (Kuo, et al., 2009);
2. We analyzed DNA copy number alterations in ovarian clear cell carcinomas and measured their CIN indices, which show a distinct pattern from ovarian serous carcinomas as published in the previous study. The results are reported in (Kuo, et al., 2010);
3. We implemented the FMR algorithm and CIN index in a caBIG (cancer Biomedical Informatics Grid) adopted analytical tool, CNSuite (Copy Number Suite), for analyzing germline CNVs and somatic CNAs for the studies of population genetics and tumor genomics, respectively. CNSuite is intended for facilitating the translation of methodology research to biomedical applications.

## 1.3 Outline of the Dissertation

In this chapter, we have briefly introduced the background knowledge of DNA copy number change, gene expression, and the microarray technologies for measuring them.

In Chapter 2, we will discuss our research work on the computational analysis of DNA copy number changes. We first provide more background introduction on this topic. We then introduce the model of Fused Margin Regression (FMR) for detecting copy number changes in a single signal profile. We adopt and improve a solution-path algorithm (Li and Zhu, 2007; Zhu, et al., 2004) to efficiently solve the FMR optimization problem and derive the algorithm in details, where we also propose a method to estimate the parameter in FMR. Then, by modifying the original formulation of the FMR optimization model, we extend FMR for detecting consensus copy number changes. Lastly, we introduce a CIN index for summarizing copy number changes as a chromosomal instability measure. To evaluate the performance of FMR for detecting copy number changes in a single profile, we compare FMR with several existing detection methods on two systematically constructed simulation datasets. The results show that FMR has competitively improved the performance of copy number change detection compared with the existing methods. We also visually demonstrate FMR on real CGH and SNP microarray data. We further compare FMR with two recently published consensus copy number change detection methods on a simulation scheme. The results show that FMR has better sensitivity and specificity compared with the two existing benchmark methods. We apply FMR on a public CNV dataset and compare some of the results with the first generation human CNV map (Redon, et al., 2006). Finally, we show the results of CIN index on an ovarian cancer copy number dataset.

In Chapter 3, we introduce two novel methods for learning the trees of phenotypes using microarray gene expression data. The first method, SA-ccsmVISDA, uses a human interactive visualization approach to generate a coarse-to-fine hierarchy of phenotypes. A leave-one-out cross validation is used to select a stable tree structure. In the second method, we first explain the concept of node bandwidth as a parameter for controlling the trade-off between the reproducibility and descriptive power of the learned trees. We then propose an effective procedure for controlling the bandwidth of a binary tree. Finally, we explain a bootstrap based stability analysis approach to derive stable tree structures under different node bandwidth constraints. We demonstrate the two methods on a muscular dystrophy dataset and a human cancer dataset.

In the last chapter, we conclude our research work and discuss several possible extensions to the proposed methods.

# 2 Computational Analysis of DNA Copy Number Changes

## 2.1 Introduction

As introduced in Chapter 1, DNA copy number change is an important form of structural variations in human genomes. It refers to the variation of copy number of a DNA segment in one genome compared with that in a reference genome. Germline Copy Number Variations (CNVs) are associated with phenotypic diversities and disease susceptibilities (Redon, et al., 2006). Somatic Copy Number Alterations (CNAs) can cause the acquisition of oncogenes and loss of tumor suppressor genes that play important roles in tumorigenesis (Pollack, et al., 2002). The lengths of copy number altered DNA segments vary from a few hundred to several million nucleotide bases (Carter, 2007). Recent progress in microarray technologies allows researchers to study copy number changes in the sub-microscopic level of a few thousand to three million bases (Feuk, et al., 2006).

### 2.1.1 Microarray Technologies

Major microarray platforms for measuring DNA copy numbers include clone based array Comparative Genomic Hybridization (CGH), oligonucleotide based array CGH, and Single Nucleotide Polymorphism (SNP) arrays. For CGH arrays, paired test and reference DNA samples, for example a tumor sample and a normal tissue sample from the same patient, are labeled with different fluorophores and competitively hybridized to the arrays. The fluorescent intensity ratios quantitatively indicate the relative copy numbers between the two samples (Pinkel and Albertson, 2005). Germline copy number changes are suppressed in the measurements if the two samples are from the same subject (van de Wiel, et al., 2010). SNP microarrays are originally designed for genotyping genomes at pre-selected loci of single-base mutations (Kim and Misra, 2007). In copy number analysis, a digested and labeled DNA sample is hybridized to the SNP array and the fluorescent intensities of the oligonucleotide probes are used as raw copy number signals. A unique advantage of SNP arrays over CGH arrays is that it can be used to detect copy number neutral loss of heterozygosity, which requires both genotype and copy number information to be revealed. It is important to note that in the most popular Affymetrix SNP arrays, each SNP is measured by a set of probes in order to increase the confidence of genotype calls. The intensities of individual probes in a set are summarized as a single intensity value by normalization software such as dChip (Li and Wong, 2001; Li and Wong, 2001; Zhao, et al., 2004). The summary intensity associated with a SNP is used as the raw copy number measure at the SNP locus.

The spatial resolutions of microarrays are determined mainly by the lengths of probes. Shorter probes usually suggest higher densities of measurements in the same genomic region, finer scales of detectable copy number changes, and more accurate estimations of breakpoints. For example, in Affymetrix SNP arrays where the probes are 25 bp-long, there can be hundreds of thousands of probes designed for a single chromosome; copy number changes as short as a few kb can be detected. In comparison, the probes of BAC clone based CGH arrays are 100 to 200 kb in length (Pinkel and Albertson, 2005) and there are only a few hundred to several thousand probes in a chromosome; consequently, detectable copy number changes are usually longer than 1 Mb. Both oligonucleotide base CGH and SNP arrays can provide high-resolution, whole-genome copy number measurements, hence they are becoming increasingly popular in copy number change studies. The most recent generation of SNP arrays (McCarroll, et al., 2008) incorporate copy number probes to further improve the spatial resolutions.

The quality of microarray copy number signals is affected by multiple factors related to array design in addition to various experimental conditions. Though oligonucleotide probes enhance the spatial resolutions, their hybridization intensities are usually less stable than those of clone based probes (Pinkel and Albertson, 2005). Furthermore, compared with SNP arrays where a single sample is hybridized, the ratio signals measured by CGH arrays usually have better contrasts, or signal-to-noise ratios (SNRs), especially for somatic deletions in the test samples.

In summary, oligonucleotide based CGH and SNP arrays can provide high-resolution copy number measurements. The relative copy numbers of paired test/reference samples measured by CGH arrays have better signal quality and naturally suppress germline mutations in the data. SNP arrays can provide both genotype and copy number data, which makes them particularly useful in certain applications requiring both types of information. It is ideal to use both platforms in rigorous analysis of copy number changes, such as in The Cancer Genome Atlas (TCGA) project (The Cancer Genome Atlas Research Network, 2008) where both oligonucleotide-based Aglient CGH arrays and Affymetrix SNP 6.0 arrays are used to acquire copy number data.

### 2.1.2 Detecting Copy Number Changes in Microarray Data

Accurate and efficient detection of copy number changes in CGH and SNP array data remains a challenging task mainly due to large numbers of measurements and low signal-to-noise ratios in the raw copy number signals, as well as heterogeneity in the tissue samples. As aforementioned in 2.1.1, oligonucleotide-based microarrays usually have $10^4$~$10^6$ probes for each chromosome, which requires computationally efficient and memory conserving detection algorithms in order to perform routine analysis on common desktop computers and workstations. Besides various experimental and array design factors that affect the characteristics of microarray data, normal tissue contamination and inhomogeneous cell populations in tissue samples introduce intermediate copy number states and reduce SNRs in the signals. Complex copy number patterns caused by heterogeneity are observed more often in tumor samples (CNAs) than in normal tissue samples (CNVs), and a detection algorithm should have both high sensitivity and specificity to detect those complex patterns.

Detecting copy number changes in a single signal profile and detecting consensus or recurrent copy number changes in the signal profiles of a population are two different computational tasks. The purpose of the first task is to estimate the copy numbers at individual loci and find breakpoints between DNA segments with different estimated copy numbers. The second task aims to find common regions of copy number changes occurring in a significant portion of subjects in a population, where the population of interest can be defined based on ethnic groups, phenotypic traits, disease types, etc.

Various computational methods have been proposed to detect copy number changes in a single signal profile. Representative methods include Circular Binary Segmentation (CBS) (Olshen, et al., 2004), Hidden Markov Models (HMM) (Andersson, et al., 2008; Marioni, et al., 2006; Stjernqvist, et al., 2007), wavelets (Ben-Yaacov and Eldar, 2008), etc. We refer interested readers to the comprehensive review by Lai et al. (2005) that compared 11 widely-adopted detection methods using simulated data and real array CGH data. New detection methods have been constantly emerging in the recent years. For the family of HMM based methods, Stjernqvist et al (2007) use continuous-index HMM to detect copy number changes in array CGH data where the BAC clone probes have substantial overlaps. The method can solve the ambiguity of assigning different copy number states to the same DNA segment contained in different BAC clones. Andersson et al (2008) proposed a HMM based method that uses the maximum *a posteriori* approach to incorporate prior knowledge. They use a six-state model to represent discrete copy numbers 0 to 4 and all those above 4. It is a common practice in HMM based methods to assume a limited number of hidden states and homogeneous Gaussian noises in the signal profile. Such assumptions may not be appropriate for real microarray data, especially those of tumor samples with complex CNA patterns. Pique-Regi et al (2008) proposed a decomposition based method to detect breakpoints in a signal profile. The signal profile is approximated by the weighted sum of a set of step sequences, each of which has the same length as the signal profile and contains a single breakpoint at a particular locus. A sparse solution of the weights is derived by the sparse Bayesian learning method, and accordingly, the estimated copy number profile contains only a limited number of breaks. Ben-Yaacov and Eldar (2008) proposed the HaarSeg method where multi-resolution Haar wavelets are used to extract the magnitudes (coefficients) of breakpoints. Each coefficient is assigned with a *p*-value based on a theoretical null-distribution. Significant breaks are then selected by controlling the False Discovery Rate (FDR) under a pre-determined level. The estimated copy numbers are recovered by merging segments detected at all resolutions. A family of regression based methods has recently been proposed (Eilers and de Menezes, 2005; Huang, et al., 2005; Li and Zhu, 2007; Tibshirani and Wang, 2008). These methods model copy number change detection as a curve fitting or regression problem, where the approximating curve is controlled to be piecewise constant by the first order variable fusion rule (Land and Friedman, 1996). The major difference between these methods is the selection of error penalty functions.

Detecting consensus copy number changes is of great importance for studying genomic structural variations. Consensus CNVs in normal human genomes are potentially associated with phenotypic diversities and disease susceptibilities. Consensus CNAs in tumor genomes may harbor important driver oncogenes or tumor suppressors. Methods for detecting consensus CNVs/CNAs can be conceptually divided into one-stage and two-stage methods. Representative two-stage methods include Genomic Identification of Significant Targets in Cancer (GISTIC) (Beroukhim, et al., 2007), Significance Testing for Aberrant Copy numbers (STAC) (Diskin, et al., 2006), Minimal Alteration Region (MAR) (Rouveirol, et al., 2006), etc. A two-stage method first detects CNVs/CNAs in individual profiles and then performs statistical analysis to find overlaps of CNVs/CNAs with significant frequencies across different samples. Apparently, the performance of a two-stage method heavily relies on the single-profile detection method being used. For example, GISTIC uses Gain and Loss Analysis of DNA (GLAD) (Hupé, et al., 2004) to detect CNAs in each signal profile, where the null hypothesis assumes that the amplitudes of amplified (deleted) loci in each profile are independent and follow the same distribution. The sum of the magnitudes, namely the G-score, of amplifications (or deletions) at a particular locus in different samples is used to test whether there is a consensus gain (or loss) at the locus. The null distribution of the G-score is derived from the histograms of amplifications (or deletions) of individual samples. Finally, adjacent loci with significantly large G-scores are connected as consensus regions. Representative one-stage methods include the Bayesian Segmentation Approach (BSA) (Wu, et al., 2009), Correlation Matrix Diagonal Segmentation (CMDS) (Zhang, et al., 2009), etc. BSA detects consensus copy number changes by iteratively applying a single-segment detection step and a peeling step on a set of signal profiles. In each iteration, a copy number altered segment is selected from a group of candidate segments by comparing their likelihood ratios $P(S|H_1)/P(S|H_0)$, where $H_0$ denotes the null hypothesis that the segment $S$ belongs to the background (no copy number changes) and $H_1$ denotes the alternative hypothesis that $S$ is a copy number altered segment. The posterior probabilities are estimated using the Bayes approach. The observations in the selected segment are then removed from the signal profiles from further analytic iterations. The selection and peeling steps are repeated until there is no more consensus segment can be detected. CMDS is a hypothesis testing based consensus copy number change detection method. CMDS first calculates the Pearson's correlation coefficient of the copy number signals for each pair of loci. A consensus copy number altered region corresponds to a square submatrix along the diagonal of the correlation coefficient matrix. CMDS uses a window of fixed size to scan the diagonal of the correlation coefficient matrix in order to determine the locations of those submatrices. A scanned locus (center of the window) is considered to be in a consensus region if the average correlation coefficient in the window is significantly large. Finally, the selected loci are combined into consensus regions. In 2.10.3, we will compare our proposed one-stage method with BSA and CMDS on simulated datasets.

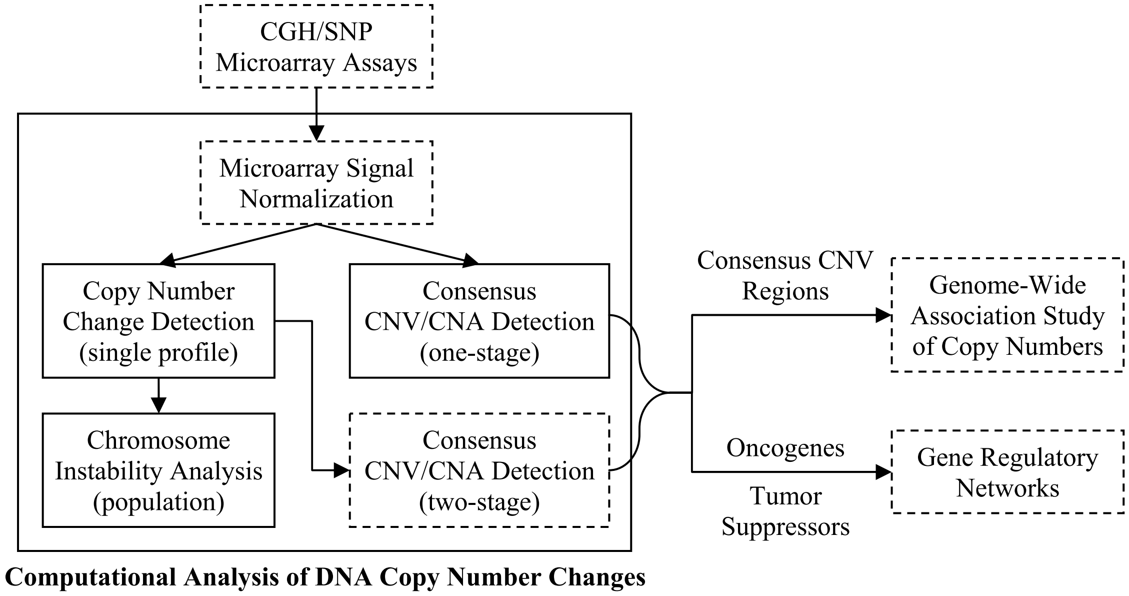

CGH/SNP Microarray Assays
Microarray Signal Normalization
Copy Number Change Detection (single profile)
Consensus CNV/CNA Detection (one-stage)
Chromosome Instability Analysis (population)
Consensus CNV/CNA Detection (two-stage)
Consensus CNV Regions
Genome-Wide Association Study of Copy Numbers
Oncogenes
Tumor Suppressors
Gene Regulatory Networks
Computational Analysis of DNA Copy Number Changes

Figure 2.1. Critical tasks and outcomes of computational analysis of DNA copy number changes using microarray data. The tasks in solid boxes are those to be addressed in this dissertation.

### 2.1.3 Chromosome Instability Index

Copy number alterations in tumor genomes are considered hallmarks of tumorigenesis. Amplifications of DNA segments can lead to increased dosage of oncogenes and development of drug resistance (Albertson, et al., 2003), while allelic deletions may deactivate tumor suppressors. Quantitative summaries of chromosome instabilities can provide global views of the rates and severity of structural mutations and can be used for prognosis/diagnosis of the stages/subtypes of tumors.

## 2.2 Objectives and Contributions

In Figure 2.1 we show a diagram of major tasks in the computational analysis of DNA copy number changes using microarray data. Among these tasks, this dissertation research addresses the following three core tasks: (1) copy number change detection in a single signal profile, (2) one-stage consensus CNV/CNA detection in population data, and (3) chromosome instability analysis based on copy number changes. Normalization of microarray data is a critical step in the workflow of copy number data analysis, especially for SNP microarrays. Many normalization methods have been proposed and proved to be effective in real world applications; we will base our work on the output of existing normalization methods instead of proposing new solutions.

The major contributions of the dissertation research on DNA copy number data analysis can be summarized as follows:

1. To detect both copy number changes in a single profile and consensus copy number changes in population data, we propose a unified framework, namely Fused Margin Regression (FMR), which models the detection problems as a constrained fitting error minimization problem. FMR is flexible and robust for detecting both germline copy number changes in normal genomes and somatic copy number changes with complex patterns in tumor genomes;
2. We modify and improve the solution-path algorithm (Gunter and Zhu, 2007; Li and Zhu, 2007; Zhu, et al., 2004) for FMR to effectively solve the associated optimization problem. The computational efficiency allows our method to be applicable to all existing high-density microarrays and potentially future high-throughput data types, such as short-read data generated by next generation sequencing platforms;
3. We propose a biologically-plausible definition of chromosome instability index that can summarize and visualize the global trend of copy number alterations in tumor samples;
4. We implement the aforementioned methods in an open source software package CNSuite, which can be used to analyze DNA copy number data in broad medical research.

The rest of the chapter is organized as follows. We first introduce in 2.3 the theoretical background of FMR and other regression based copy number change detection methods closely related to FMR. In 2.4, we explain the model of FMR for detecting copy number changes in a single signal profile. In 2.5, we derive the path algorithm for solving the linear programming problem associated with the FMR model proposed in 2.4. In 2.6, we relate the parameter $\epsilon$ of the loss function employed in FMR with the noise variance of copy number signals and provide methods to estimate the noise variance. In 2.7, we introduce a two-tier FMR for copy number signals with inhomogeneous noise distributions. In 2.8, we extend the original FMR formulation for detecting consensus copy number changes in a set of signal profiles. In 2.9, we introduce the definition of chromosome instability index for analyzing structural stabilities of genomes. In 2.10, we test the proposed methods using systematic simulations and real CGH and SNP microarray data. Finally, we conclude our work on computational analysis of DNA copy number changes in 2.11.

## 2.3 Regression Models for Copy Number Change Detection

Given a sequence of $n$ observed copy number signals $\mathbf{y} = \{y_i\}_{i=1}^{n}$ with their genomic positions $\mathbf{x} = \{x_i\}_{i=1}^{n}$, $x_1 < \ldots < x_i < \ldots < x_n$ in a chromosome, we model the signal at each locus by

$$Y_i = \eta_i + N_i, \quad i = 1, \ldots, n, \tag{2.1}$$

where $\eta_i$ is the true copy number and $\{N_i\}_{i=1}^{n}$ are independently distributed noises with zero means. Copy number change detection aims to estimate the true copy number for each locus and infer the locations and copy numbers of abnormal DNA segments. Denote the estimates of the true copy numbers $\boldsymbol{\eta} = \{\eta_i\}_{i=1}^{n}$ by $\hat{\boldsymbol{\eta}} = \{\hat{\eta}_i\}_{i=1}^{n}$. Since a copy number altered DNA segment is usually measured by multiple probe sets, the true copy numbers at adjacent probe sets are highly likely to be same. Consequently, the profiles of $\boldsymbol{\eta}$ and $\hat{\boldsymbol{\eta}}$ should both be piecewise constant. Different detection methods encode such spatial correlation of true copy numbers in various ways. For example, HMM based methods use a predetermined, finite number of hidden states to represent possible copy number states in a signal profile. The likelihood of transition from one copy number to the other is estimated from the signal profile and encoded in the state transition probability matrix. Different from HMM, CBS and HaarSeg directly model the piecewise constancy of $\boldsymbol{\eta}$ rather than the transitions between copy number states. The advantage is that there is no constraint imposed on the number and magnitudes of copy number states. In CBS, a signal profile is recursively partitioned and in each of the iterations two breakpoints are selected that enclose a segment with a different copy number from the flanking segments. In HaarSeg, a signal profile is decomposed into a family of segments with different locations and lengths. Those segments with insignificant coefficients are discarded.

### 2.3.1 Regularization by Variable Fusion

To model the spatial pattern that the true copy numbers of adjacent measurements are highly likely to be same, the regression based methods (Eilers and de Menezes, 2005; Huang, et al., 2005; Li and Zhu, 2007; Tibshirani and Wang, 2008) employ the first-order variable fusion constraint, which is proposed by Land and Friedman (1996) in the context of linear squares regression

$$\min_{\boldsymbol{\beta},\beta_0} \left\| \mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \beta_0 \mathbf{1} \right\|^2 + \lambda C(\boldsymbol{\beta}), \quad \lambda \ge 0, \tag{2.2}$$

where $\mathbf{y}$ is a vector of $n$ responses; $\mathbf{X}$ is a $n$ by $p$ matrix of the realizations of predictors $X_1, \ldots, X_p$; $\boldsymbol{\beta}$ and $\beta_0$ are the coefficients of the linear regression model; $\mathbf{1}$ is a $n$-vector of all ones; $\lambda$ is the non-negative weight of the penalty function $C(\boldsymbol{\beta})$. Land and Friedman proposed two forms of penalties,

$$C_0(\boldsymbol{\beta}) = \lim_{q \to 0} \sum_{i=2}^{p} |\beta_i - \beta_{i-1}|^q \quad \text{and} \quad C_1(\boldsymbol{\beta}) = \sum_{i=2}^{p} |\beta_i - \beta_{i-1}|,$$

which are called the zero-order and first-order variable fusion penalties, respectively. Both penalty functions embody the constraint that spatially adjacent predictors should have similar values in their corresponding coefficients, which leads to a piecewise constant profile of the coefficient values in the solutions. The difference between $C_0(\boldsymbol{\beta})$ and $C_1(\boldsymbol{\beta})$ is that, $C_0(\boldsymbol{\beta})$ penalizes any non-zero breaks in the coefficient profile and hence yields fewer, but larger breaks; $C_1(\boldsymbol{\beta})$ penalizes the total magnitude of breaks, which allows more breaks with smaller magnitudes to be included. Furthermore, objective functions with constraint $C_0(\boldsymbol{\beta})$ cannot be solved exactly without exhaustive search (Land and Friedman, 1996), while $C_1(\boldsymbol{\beta})$ can be converted to a set of linear constraints, which makes (2.2) a quadratic programming problem that can be solved using various convex optimization methods. Therefore, the regression based copy number change detection methods often adopt the first-order variable fusion constraint.

### 2.3.2 Regression-based Copy Number Change Detection Methods

In the context of copy number change detection, the first-order variable fusion rule becomes

$$C(\hat{\boldsymbol{\eta}}) = \sum_{i=2}^{n} \left|\hat{\eta}_i - \hat{\eta}_{i-1}\right| . \tag{2.3}$$

At each locus of the probe sets, the fitting error of using the estimate $\hat{\eta}_i$ to approximate observations $y_i$ is measured by a loss function $L(y_i, \hat{\eta}_i)$, whose particular form determines the varying properties of the regression based methods. Given the variable fusion rule and the loss function, the family of regression based copy number change detection methods pose optimization problems in the following generic form

$$\min_{\hat{\boldsymbol{\eta}}} \sum_{i=1}^{n} L(y_i, \hat{\eta}_i), \quad \text{subject to} \quad \sum_{i=2}^{n} \left|\hat{\eta}_i - \hat{\eta}_{i-1}\right| \le s, \quad s \ge 0 . \tag{2.4}$$

An alternative form of (2.4) uses a set of variables to represent the differences between adjacent estimated copy numbers, i.e.

$$\delta_i = \hat{\eta}_i - \hat{\eta}_{i-1}, \quad i = 2, \ldots, n ,$$

where we define $\delta_1 = \hat{\eta}_1$. Therefore the estimated copy numbers are recovered by

$$\hat{\eta}_i = \sum_{k=1}^{i} \delta_k , \quad i = 2, \ldots, n .$$

Denote $\boldsymbol{\delta}_{-1} = \{\delta_i\}_{i=2}^{n}$, the variable fusion constraint can be interpreted as a penalty of the $l_1$-norm $\left\|\boldsymbol{\delta}_{-1}\right\|_1$ of the breaks, which yields a sparse solutions of $\delta_2, \ldots, \delta_n$ , i.e. only a limited number of

$\delta_i$ have non-zero values. Consequently, the profile of $\hat{\boldsymbol{\eta}}$ is piecewise constant. The number of non-zero breaks and their magnitudes depend on the value of $s$ and the particular form of loss function $L(y_i, \hat{\eta}_i)$.

### 2.3.2.1 Lasso

Huang et al. (2005) applied Lasso method to estimate copy numbers. The loss function in Lasso is the squared error $L(y_i, \hat{\eta}_i) = (y_i - \hat{\eta}_i)^2$. The corresponding optimization problem is

$$\min_{\hat{\boldsymbol{\eta}}} \sum_{i=1}^{n} \left(y_i - \hat{\eta}_i\right)^2, \quad \text{s.t. } \sum_{i=2}^{n} \left|\hat{\eta}_i - \hat{\eta}_{i-1}\right| \le s. \tag{2.5}$$

The application of Lasso can be is justified when (2.5) is reformulated as

$$\min_{\mathbf{d}} \left\|\mathbf{y} - \mathbf{X}\mathbf{d}\right\|^2, \text{ s.t. } \left|\boldsymbol{\delta}_{-1}\right| \le s,$$

where $\mathbf{d}$=$[\delta_1, \boldsymbol{\delta}_{-1}]^{\mathrm{T}}$ and $\mathbf{X}$ is a $n$ by $n$ lower triangular matrix with all non-zero elements equal to 1.

### 2.3.2.2 Fused Lasso

Tibshirani et al. (2005) introduced the first-order variable fusion constraint into the original formulation of Lasso for estimating linear regression models with spatially ordered predictors. The fused Lasso method was later applied to detect copy number changes in CGH data (Tibshirani and Wang, 2008). The optimization problem associated with fused Lasso for copy number change detection is

$$\min_{\hat{\boldsymbol{\eta}}} \sum_{i=1}^{n} \left(y_i - \hat{\eta}_i\right)^2, \text{ s.t. } \sum_{i=1}^{n} \left|\hat{\eta}_i\right| \le s_1 \text{ and } \sum_{i=2}^{n} \left|\hat{\eta}_i - \hat{\eta}_{i-1}\right| \le s_2. \tag{2.6}$$

The constraints in (2.6) enforce sparseness in both the estimated copy numbers and the breaks.

#### *2.3.2.3 Fused quantile regression*

Eilers and de Menezes (2005) proposed a fused quantile regression model to detect copy number changes

$$\min_{\hat{\boldsymbol{\eta}}} \sum_{i=1}^{n} |y_i - \hat{\eta}_i|, \text{ s.t. } \sum_{i=2}^{n} |\hat{\eta}_i - \hat{\eta}_{i-1}| \le s . \tag{2.7}$$

The loss function is the $l_1$-norm of the difference $\mathbf{y} - \hat{\boldsymbol{\eta}}$, which can be considered as a special form of the check function

$$L_\tau(y_i, \hat{\eta}_i) = \begin{cases} \tau(y_i - \hat{\eta}_i), & y_i - \hat{\eta}_i \ge 0 \\ -(1-\tau)(y_i - \hat{\eta}_i), & \text{otherwise} \end{cases}, \qquad 0 \le \tau \le 1$$

when $\tau = 0.5$. The check function is the error measure used in quantile regression (Koenker and Bassett, 1978).

Li and Zhu (2007) enhanced the fused quantile regression method by incorporating genomic distances between adjacent loci in the variable fusion constraint

$$\sum_{i=2}^{n} \left| \frac{\hat{\eta}_i - \hat{\eta}_{i-1}}{x_i - x_{i-1}} \right| \le s . \tag{2.8}$$

Intuitively, the modified constraint allows adjacent loci that are far apart on a chromosome to have larger difference in the estimated copy numbers, while enforces smaller difference in the estimated copy numbers for adjacent loci close to each other. It should be noted that in (Eilers and de Menezes, 2005), the authors have pointed out that the incorporation of the adjustment $x_i - x_{i-1}$ as in (2.8) will not actually change the solutions given $x_{i-1} < x_i$. They have suggested that the positions $\mathbf{x}$ are only relevant for the purpose of visualization.

## 2.4 Fused Margin Regression

We propose a robust regression method, namely the fused margin regression (FMR), which uses the variable fusion rule to enforce a piecewise constant solution of the estimated copy numbers and the $\epsilon$-insensitive loss function to penalize the approximation error. The $\epsilon$-insensitive loss function (Smola and Schölkopf, 2004), also called the dead-zone error penalty (Boyd and Vandenberghe, 2004), is defined as

$$L_{\epsilon_i}(y_i,\hat{\eta}_i)=\begin{cases}0, & |y_i-\hat{\eta}_i|\le\epsilon_i\\ |y_i-\hat{\eta}_i|-\epsilon_i, & |y_i-\hat{\eta}_i|>\epsilon_i\end{cases},\quad \epsilon_i\ge 0,\quad i=1,\ldots,n. \tag{2.9}$$

Note that instead of using the same value of $\epsilon$, we can allow different loci to have different sizes of margins in order to make the model of FMR more flexible. $L_{\epsilon_i}$ assigns zero errors to the observations inside the margins $[\hat{\eta}_i-\epsilon_i,\hat{\eta}_i+\epsilon_i]$ and linear errors to those observations outside the margins. Given the definition of $L_{\epsilon_i}$, the optimization problem associated with FMR is

$$\min_{\hat{\boldsymbol{\eta}}}\sum_{i=1}^{n}L_{\epsilon_i}(y_i,\hat{\eta}_i),\ \text{s.t.}\ \sum_{i=2}^{n}|\hat{\eta}_i-\hat{\eta}_{i-1}|\le s,\ s\ge 0. \tag{2.10}$$

We illustrate the difference between FMR and fused Lasso or fused quantile regression using a schematic example in Figure 2.2. The signal profile consists of two observations $y_1=0.3$ and $y_2=0.7$ at loci $x_1$ and $x_2$ (Figure 2.2 (a)). For fused quantile regression (Figure 2.2 (c)) and FMR (Figure 2.2 (d)), the variable fusion constraint is $|\hat{\eta}_2-\hat{\eta}_1|\le 0.2$; for fused Lasso (Figure 2.2 (b)), the two constraints are $|\hat{\eta}_2-\hat{\eta}_1|\le 0.2$ and $|\hat{\eta}_1|+|\hat{\eta}_2|\le 0.5$. The light gray region in each subfigure is the feasible region defined by the constraint(s). The solid diagonal line $\hat{\eta}_1=\hat{\eta}_2$ corresponds to the space of sparse solutions. The contours of the loss function $L(y_1,\hat{\eta}_1)+L(y_2,\hat{\eta}_2)$ for each method are plotted in gray. In Figure 2.2 (d), the dark gray square centered at $(0.3, 0.7)$ is the region in which the loss function equals 0, i.e. $\hat{\eta}_i\in[y_i-\epsilon,y_i+\epsilon]$, $i=1,2$. When $\epsilon_1=\epsilon_2\ge 0.2$, the diagonal line intersects the zero loss plateau, which gives a set of sparse solutions with zero approximation error. It can be seen that given properly chosen values of $\epsilon$, FMR has higher chances to produce sparse solutions.

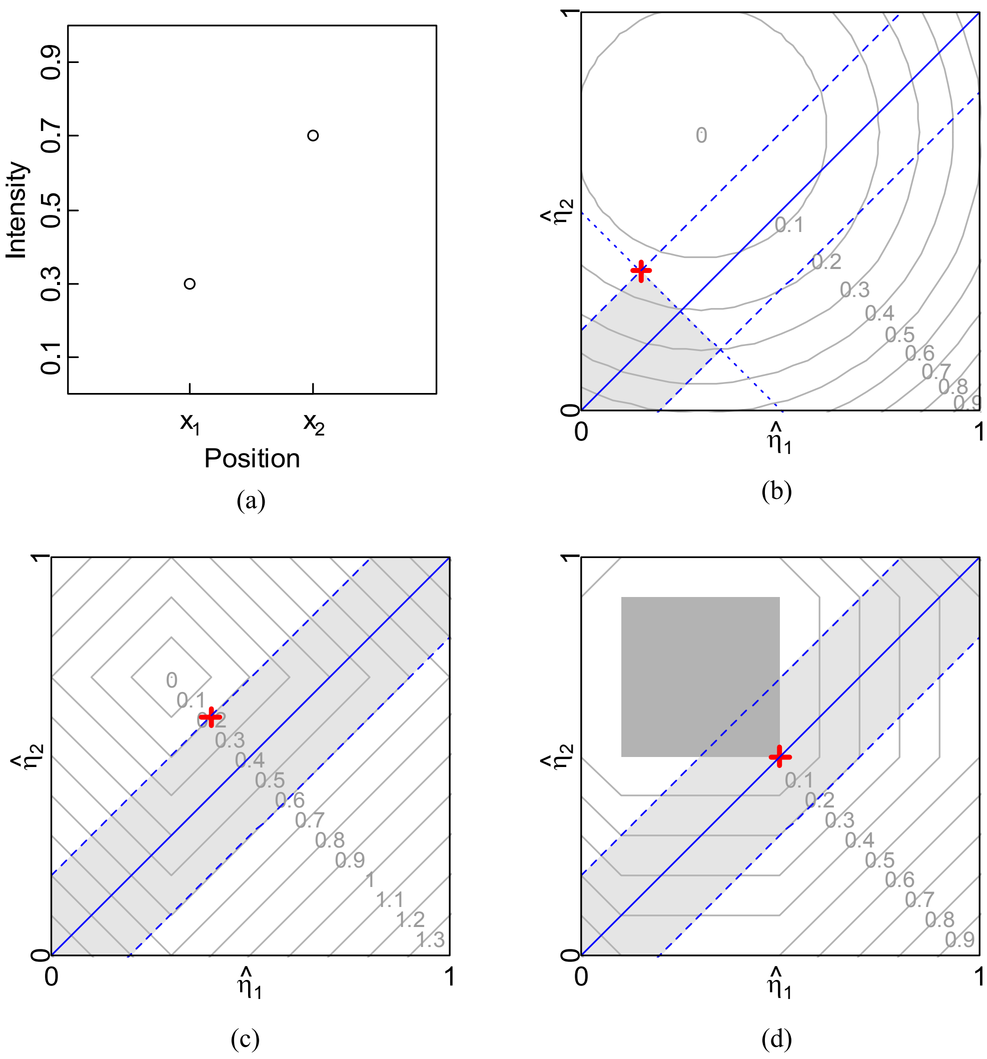


Figure 2.2. A schematic example illustrating the difference between fused Lasso (b), fused quantile regression (c), and FMR (d).

Using the variables $\boldsymbol{\delta} = \{\delta_i\}_{i=1}^{n}$ that correspond to breaks and introducing non-negative slack variables $\boldsymbol{\xi} = \{\xi_i\}_{i=1}^{n}$ and $\boldsymbol{\zeta} = \{\zeta_i\}_{i=1}^{n}$ as in Support Vector Regression (Smola and Schölkopf, 2004), we can convert (2.10) to a linear programming problem (FMR-LP)

$$\min_{\boldsymbol{\delta},\boldsymbol{\xi},\boldsymbol{\zeta}} \sum_{i=1}^{n} \left( \xi_i + \zeta_i \right), \tag{2.11}$$

$$\text{s.t.}\begin{cases}\sum_{i=2}^{n}|\delta_i|\le s \\ y_i-\sum_{k=1}^{i}\delta_k \le \epsilon_i+\xi_i, & i=1,\ldots,n \\ \sum_{k=1}^{i}\delta_k - y_i \le \epsilon_i+\zeta_i, & i=1,\ldots,n \\ \xi_i\ge 0,\ \zeta_i \ge 0, & i=1,\ldots,n\end{cases}. \tag{2.12}$$

The slack variables $\xi_i$ and $\zeta_i$ are interpreted as the distance from $y_i$ to the upper or lower boundaries specified by $\hat{\eta}_i \pm \epsilon_i$ if $y_i$ is outside the margin. The constraint $s$ controls the number of non-zero breaks and their amplitudes. Given the optimal solution $\boldsymbol{\delta}^* = \{\delta_i^*\}_{i=1}^{n}$ of (2.11) and (2.12), the estimated copy numbers are recovered by $\hat{\eta}_i = \sum_{k=1}^{i}\delta_i^*$ , $i=1,\ldots,n$. By introducing non-negative variables $\delta_i^+$ and $\delta_i^-$ that satisfy $\delta_i = \delta_i^+ - \delta_i^-$ and $|\delta_i| = \delta_i^+ + \delta_i^-$ (Fletcher, 2000), (2.11) and (2.12) can be formulated as a standard LP problem and solved by existing simplex or interior-point algorithms. Unfortunately, for high-density CGH and SNP array data, the LP problem has an enormous number of variables and constraints whose solutions could be computationally infeasible. For example, in an Affymetrix 250K Sty-I SNP array, there are about 20,000 probe sets for the first chromosome. To detect copy number changes using FMR, the corresponding LP problem has 80,000 variables and 120,001 linear constraints. Using single precision floating-point numbers, about 36-gigabyte memory is required to store the constraint matrix.

Instead of using conventional linear optimization methods, we adapt the solution-path algorithm (Li and Zhu, 2007; Zhu, et al., 2004) to the FMR-LP problem to detect copy number changes in high-density signal profiles.

## 2.5 Solution-Path Algorithm for FMR

We made two major modifications to the path algorithm proposed in (Li and Zhu, 2007). First, we adapt the rules of iteratively updating constraint $s$ and the corresponding optimal solutions to the $\epsilon$-insensitive loss function of FMR; second, we use the original variable fusion constraint

(2.3) instead of the weighted version in (2.8), which eliminates matrix-inversion operations and consequently makes the path algorithm of FMR much more efficient.

The Lagrange function of the optimization problem (2.11) and (2.12) is

$$\begin{aligned}\mathcal{L}(\boldsymbol{\delta},\boldsymbol{\xi},\boldsymbol{\zeta},\lambda,\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\mu},\mathbf{v}) &= \mathbf{1}^{\mathrm{T}}(\boldsymbol{\xi}+\boldsymbol{\zeta})+\lambda\left(\left\|\boldsymbol{\delta}_{-1}\right\|_1-s\right)+\boldsymbol{\alpha}^{\mathrm{T}}(\mathbf{y}-\mathbf{L}\cdot\boldsymbol{\delta}-\boldsymbol{\varepsilon}-\boldsymbol{\xi}) \\ &\quad +\boldsymbol{\beta}^{\mathrm{T}}(\mathbf{L}\cdot\boldsymbol{\delta}-\mathbf{y}-\boldsymbol{\varepsilon}-\boldsymbol{\zeta})-\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\xi}-\mathbf{v}^{\mathrm{T}}\boldsymbol{\zeta}\end{aligned} \tag{2.13}$$

where $\lambda$, $\boldsymbol{\alpha}=\{\alpha_i\}_{i=1}^n$, $\boldsymbol{\beta}=\{\beta_i\}_{i=1}^n$, $\boldsymbol{\mu}=\{\mu_i\}_{i=1}^n$, and $\mathbf{v}=\{\nu_i\}_{i=1}^n$ are non-negative Lagrange multipliers; $\delta_1$ and $\boldsymbol{\delta}_{-1}=[\delta_2,\ldots,\delta_n]^{\mathrm{T}}$ are the variables corresponding to the breaks; $\boldsymbol{\varepsilon}=\{\epsilon_i\}_{i=1}^n$ are the parameters of the loss functions; $\mathbf{1}$ is a $n$-vector of ones; $\mathbf{L}$ is a $n$ by $n$ lower triangular matrix with its non-zero elements equal to 1. To satisfy the stationarity of the Karush-Kuhn-Tucker (KKT) conditions (Fletcher, 2000), we have

$$\begin{aligned}&\frac{\partial\mathcal{L}}{\partial\delta_1}=0\Rightarrow\mathbf{1}^{\mathrm{T}}(\boldsymbol{\alpha}-\boldsymbol{\beta})=0\\&\frac{\partial\mathcal{L}}{\partial\delta_i}=0\Rightarrow\sum_{k=i}^{n}(\alpha_k-\beta_k)=\lambda\cdot\mathrm{sign}(\delta_i),\quad\forall\delta_i\neq0,\quad i=2,\ldots,n\\&\frac{\partial\mathcal{L}}{\partial\boldsymbol{\xi}}=0\Rightarrow\boldsymbol{\alpha}+\boldsymbol{\mu}=\mathbf{1}\\&\frac{\partial\mathcal{L}}{\partial\boldsymbol{\zeta}}=0\Rightarrow\boldsymbol{\beta}+\mathbf{v}=\mathbf{1}\end{aligned} \tag{2.14}$$

Note that $\mathrm{sign}(\delta_i)$ is undefined at $\delta_i=0$. The complementary condition of the KKT conditions requires that at the optimal solution the primal variables and the Lagrange multipliers must satisfy

$$\begin{cases}\lambda\cdot\left(\left\|\boldsymbol{\delta}_{-1}\right\|_1-s\right)=0\\ \alpha_i(y_i-\sum_{k=1}^{i}\delta_k-\epsilon_i-\xi_i)=0, & i=1,\ldots,n\\ \beta_i(\sum_{k=1}^{i}\delta_k-y_i-\epsilon_i-\zeta_i)=0, & i=1,\ldots,n\\ \mu_i\xi_i=0, & i=1,\ldots,n\\ \nu_i\zeta_i=0, & i=1,\ldots,n\end{cases} \tag{2.15}$$

By introducing variables $\boldsymbol{\gamma} = \{\gamma_i \triangleq \alpha_i - \beta_i\}_{i=1}^{n}$, the stationarity conditions in (2.14) can also be written as

$$\begin{cases} \mathbf{1}^{\mathrm{T}}\boldsymbol{\gamma} = 0 \\ \sum_{k=i}^{n} \gamma_k = \lambda \cdot \operatorname{sign}(\delta_i), \quad \forall \delta_i \neq 0 \\ -\mathbf{1} \leq \boldsymbol{\gamma} \leq \mathbf{1} \end{cases} \tag{2.16}$$

Given a particular value of $s$, combining the conditions of (2.14) and (2.15) yields a partition of the observations that consists of five disjoint index sets

$\mathcal{R}_{\mathrm{U}} = \left\{ i \mid \xi_i = y_i - \sum_{k=1}^{i} \delta_k - \epsilon_i > 0, \zeta_i = 0, \alpha_i = 1, \beta_i = 0, \mu_i = 0, \nu_i = 1 \right\}$ (upper ramp)

$\mathcal{B}_{\mathrm{U}} = \left\{ i \mid \xi_i = y_i - \sum_{k=1}^{i} \delta_k - \epsilon_i = 0, \zeta_i = 0, 0 \leq \alpha_i \leq 1, \beta_i = 0, \mu_i = 1 - \alpha_i, \nu_i = 1 \right\}$ (upper boundary)

$\mathcal{M} = \left\{ i \,\middle|\, \left| y_i - \sum_{k=1}^{i} \delta_k \right| < \epsilon_i, \xi_i = 0, \zeta_i = 0, \alpha_i = 0, \beta_i = 0, \mu_i = 1, \nu_i = 1 \right\}$ (margin)

$\mathcal{B}_{\mathrm{L}} = \left\{ i \mid \xi_i = 0, \zeta_i = \sum_{k=1}^{i} \delta_k - y_i - \varepsilon_i = 0, \alpha_i = 0, 0 \leq \beta_i \leq 1, \mu_i = 1, \nu_i = 1 - \beta_i \right\}$ (lower boundary)

$\mathcal{R}_{\mathrm{L}} = \left\{ i \mid \xi_i = 0, \zeta_i = \sum_{k=1}^{i} \delta_k - y_i - \varepsilon_i > 0, \alpha_i = 0, \beta_i = 1, \mu_i = 1, \nu_i = 0 \right\}$ (lower ramp)

where $\mathcal{R}_{\mathrm{U}} \cup \mathcal{B}_{\mathrm{U}} \cup \mathcal{M} \cup \mathcal{B}_{\mathrm{L}} \cup \mathcal{R}_{\mathrm{L}} = \{1, \ldots, n\}$. We use $\mathcal{R}_{\mathrm{U}}$ as an example to show how these index sets are derived. If $y_i - \sum_{k=1}^{i} \delta_k > \epsilon_i$, we have $\sum_{k=1}^{i} \delta_k - y_i < -\epsilon_i < \epsilon_i + \zeta_i$ due to $\zeta_i \geq 0$. Since this inequality constraint is active, the corresponding Lagrange multiplier $\beta_i = 0$ according to the complementary condition in (2.15). Consequently we have $\nu_i = 1 - \beta_i = 1$ and $\zeta_i = 0$ due to $\nu_i \zeta_i = 0$. According to the feasibility in (2.12) and the prerequisite $y_i - \sum_{k=1}^{i} \delta_k - \epsilon_i > 0$, we have $\xi_i \geq y_i - \sum_{k=1}^{i} \delta_k - \epsilon_i > 0$. Therefore $\mu_i = 0$ since $\mu_i \xi_i = 0$ and hence $\alpha_i = 1 - \mu_i = 1$, which in turn implies that $\xi_i = y_i - \sum_{k=1}^{i} \delta_k - \epsilon_i$. The other index sets can be derived similarly.

An observation $y_i$ can be inside the margin ( $\hat{\eta}_i - \epsilon_i < y_i < \hat{\eta}_i + \epsilon_i$ ), on the margin boundaries ( $y_i = \hat{\eta}_i \pm \epsilon_i$ ), or outside the margin ( $y_i < \hat{\eta}_i - \epsilon_i$ or $y_i > \hat{\eta}_i + \epsilon_i$ ). We call those observations on the upper and lower margin boundaries as anchors, which satisfy

$$y_i - \epsilon_i = \sum_{k=1}^{i} \delta_k \text{ , } \forall i \in \mathcal{B}_{\mathrm{U}} \text{ and } y_i + \epsilon_i = \sum_{k=1}^{i} \delta_k \text{ , } \forall i \in \mathcal{B}_{\mathrm{L}} \text{ .}$$

Denote the index set of anchors by $\mathcal{B} = \mathcal{B}_{\mathrm{U}} \cup \mathcal{B}_{\mathrm{L}}$ and the index set of non-zero breaks by $\mathcal{P} = \{i \mid \delta_i \neq 0, i = 2, \ldots, n\}$. We further denote the $i$th elements in the ascendingly ordered sets $\mathcal{P}$ and $\mathcal{B}$ by $\mathcal{P}(i)$ and $\mathcal{B}[i]$, respectively. When $\mathcal{P}(i)$ and $\mathcal{B}[i]$ are used in subscripts, we simply use $(i)$ and $[i]$ to represent them. For example, $\delta_{(i)}$ and $\delta_{\mathcal{P}(i)}$ refer to the same variable. In the following sections we will show that the inductive steps of the path algorithm guarantee that the indices of anchors and breaks of an optimal solution are interleaved, i.e. $1 \leq \mathcal{B}[1] < \mathcal{P}(1) \leq \mathcal{B}[2] < \ldots \leq \mathcal{B}[i] < \mathcal{P}(i) \leq \ldots < \mathcal{P}(m) \leq \mathcal{B}[m+1] \leq n$ , $|\mathcal{P}| = m$ . Consequently, we have $|\mathcal{B}| = |\mathcal{P}| + 1 = m + 1$ for each step.

### 2.5.1 Initialization

The path algorithm starts from $s = 0$ where only $\delta_1$ is freely adjustable and $\delta_i = 0$, $i = 2, \ldots, n$. The objective function $L(\delta_1; \mathbf{y}, \boldsymbol{\varepsilon}) = \sum_{i=1}^{n} L_{\epsilon_i}(y_i, \delta_1)$ is a convex function of $\delta_1$. To find the minimum value of the objective function, we can simply find the point (or a plateau) where the slope of $L(\delta_1; \mathbf{y}, \boldsymbol{\varepsilon})$ changes from negative to positive (or from negative to 0 at the lower boundary of a plateau). We order the set of critical points $A = \{y_i \pm \epsilon_i, i = 1, \ldots, n\}$ such that $A = \{a_1 < a_2 < \ldots < a_{2n}\}$. Here we assume that all the values in $A$ are distinct, which can be achieved in practice by slightly perturbing the observations $\mathbf{y}$. When $\delta_1 < a_1$, the slope of $L(\delta_1; \mathbf{y}, \boldsymbol{\varepsilon})$ is $-n$. When $\delta_1$ increases, the slope is increased by 1 when each time $\delta_1$ hits a value in $A$. Therefore the slope is 0 when $a_n \leq \delta_1 < a_{n+1}$. When $\delta_1 > a_{2n}$, the slope becomes $n$. We can simply set $\delta_1 = a_n$ as the initial solution of FMR for $s = 0$. Suppose $\delta_1 = a_n = y_i \pm \epsilon_i$, then $y_i = \delta_1 \mp \epsilon_i$ becomes the first anchor of the initial solution. Since $\hat{\eta}_i = \delta_1$, $\forall i$, we can determine the Lagrange multipliers for those observations above ($\gamma_i = 1$), inside ($\gamma_i = 0$), and below ($\gamma_i = -1$) the margins. Since $\sum_{i=1}^{n} \gamma_i = 0$, we can also determine the Lagrange multiplier for the only anchor. We select the first break such that by adjusting which the error residual reduces the fastest. Denote the sum of residuals from locus $i$ to locus $n$ by $r_i$, that is,

$$r_i = \sum_{\substack{j=i \\ j \in \mathcal{R}_U}}^{n} \left( y_j - \hat{\eta}_j - \epsilon_j \right) + \sum_{\substack{j=i \\ j \in \mathcal{R}_L}}^{n} \left( \hat{\eta}_j - \epsilon_j - y_j \right), \; i = 2, \ldots, n.$$

Since $\hat{\eta}_j = \sum_{k=1}^{j} \delta_k$, we have

$$\frac{\partial r_i}{\partial \delta_i} = \sum_{\substack{j=i \\ j \in \mathcal{R}_U}}^{n} (-1) + \sum_{\substack{j=i \\ j \in \mathcal{R}_L}}^{n} (1) = \left| \mathcal{R}_L \cap \{i, \ldots, n\} \right| - \left| \mathcal{R}_U \cap \{i, \ldots, n\} \right|, \; i = 2, \ldots, n. \tag{2.17}$$

To reduce the error residual, $\delta_i$ should be positive if $\partial r_i / \partial \delta_i < 0$ and negative if $\partial r_i / \partial \delta_i > 0$. Therefore we select $\delta_{k^*}$ as the first non-zero break such that

$$k^* = \arg \max_{i \in \{2, \ldots, n\}} \left\{ \left| \frac{\partial r_i}{\partial \delta_i} \right| \right\} \text{ and } \operatorname{sign}(\delta_{k^*}) = -\frac{\partial r_{k^*}}{\partial \delta_{k^*}}. \tag{2.18}$$

The result (2.18) is quite intuitive: we insert a break and shift the segment after it to reduce as fast as possible the difference between the numbers of observations on the two sides of the margins. The initial value of the Lagrange multiplier $\lambda$ is set to $\left| \sum_{i=k^*}^{n} \gamma_i \right|$ since the KKT conditions require that $\lambda \cdot \operatorname{sign}(\delta_k) = \sum_{i=k}^{n} \gamma_i$ for all $\delta_k \neq 0$ and $\delta_{k^*}$ is the only non-zero break.

### 2.5.2 Selection of Candidate Breaks

Given the optimal solution for the current value of $s$, to find the next possible solution that reduces the error residual, we can shift either a segment delimited by existing non-zero breaks or a new segment created by introducing a new non-zero break to $\mathcal{P}$. This can be determined by analyzing the stationarity conditions (2.16). Suppose at a particular step of the path algorithm we have $|\mathcal{B}|=|\mathcal{P}|+1$ (which apparently holds for the initialization step where $|\mathcal{P}|=0$ and $|\mathcal{B}|=1$). Denote the sign of the $i$ th non-zero break in $\mathcal{P}$ by $s_{(i)}$ . Due to the condition $\sum_{k=i}^{n}\gamma_k=\lambda\cdot\text{sign}(\delta_i)$ , $\forall\delta_i\neq 0$, the relation between the Lagrange multipliers $\gamma_i$ and $\lambda$ can be expressed as

$$\sum_{k=i}^{j}\gamma_k=\begin{cases}-s_{(1)}\lambda, & i=1, \quad j=\mathcal{P}(1)-1\\ 2s_{(t)}\lambda\cdot\mathrm{I}(s_{(t)}\neq s_{(t+1)}), & i=\mathcal{P}(t), \quad j=\mathcal{P}(t+1)-1, \quad 1\leq t\leq m-1\\ s_{(m)}\lambda, & i=\mathcal{P}(m), \quad j=n\end{cases}, \tag{2.19}$$

where $\mathrm{I}(A)$ is an indicator function such that $\mathrm{I}(A)=1$ if condition $A$ is true and $\mathrm{I}(A)=0$ if $A$ is false. (2.19) implies that $\lambda$ is an indicator of the rate of change of the error residuals.

We first consider the case where an existing segment should be shifted, i.e. the amplitudes of its bounding breaks should be changed. In order to reduce the error residual, the signs of the breaks keep the same as the old ones. The anchor within the shifted segments will depart from the margin boundary and the corresponding $\gamma_i$, and consequently $\lambda$, will change. Denote the previous value of $\lambda$ as $\lambda^{\text{old}}$ and $\Delta\lambda=\lambda-\lambda^{\text{old}}$; denote the previous values of $\gamma_i$ as $\gamma_i^{\text{old}}$ and $\Delta\gamma_i=\gamma_i-\gamma_i^{\text{old}}$ , $i=1,\ldots,n$ . Note that for $i\notin\mathcal{B}$ , $\Delta\gamma_i=0$ . Since conditions (2.16) must be satisfied for both old and new optimal solutions, we have

$$\sum_{i=1}^{m+1}\Delta\gamma_{[i]}=0, \tag{2.20}$$

$$\Delta\lambda\cdot\text{sign}(\delta_{(i)})=\sum_{\substack{k\geq\mathcal{P}(i)\\ k\in\mathcal{B}}}\Delta\gamma_k\ ,\ i=1,\ldots,m. \tag{2.21}$$

In order to have a unique solution of $\Delta\gamma_{[i]}$, (2.20) and (2.21) must yield the following linear equations

$$\Delta\lambda\cdot\begin{bmatrix}0\\ s_{(1)}\\ s_{(2)}\\ \vdots\\ s_{(m)}\end{bmatrix}=\begin{bmatrix}1&1&1&\cdots&1\\ 0&1&1&\cdots&1\\ 0&0&1&\cdots&1\\ &\cdots&&\cdots&\\ 0&0&0&\cdots&1\end{bmatrix}\begin{bmatrix}\Delta\gamma_{[1]}\\ \Delta\gamma_{[2]}\\ \Delta\gamma_{[3]}\\ \cdots\\ \Delta\gamma_{[m+1]}\end{bmatrix}. \tag{2.22}$$

The solution to (2.22) is

$$\Delta\gamma'_{[i]}=\frac{\Delta\gamma_{[i]}}{\Delta\lambda}=\begin{cases}-s_{(i)}, & i=1\\ s_{(i-1)}-s_{(i)}, & 2\le i\le m\\ s_{(i-1)}, & i=m+1\end{cases}. \tag{2.23}$$

(2.23) is interpreted as the rate of change of $\gamma_{[i]}$ with respect to the change of $\lambda$. If there exists $i$ such that $\Delta\gamma_{[i]}=0$, we simply move the anchor $\mathcal{B}[i]$ to $\mathcal{R}_{\mathrm{U}}$, $\mathcal{M}$, or $\mathcal{R}_{\mathrm{L}}$ according to the sign of $\Delta\gamma'_{[i]}$ and set $\boldsymbol{\gamma}=\boldsymbol{\gamma}^{\mathrm{old}}$ and $\lambda=\lambda^{\mathrm{old}}$. In the following we only discuss the case where $\Delta\gamma_{[i]}\ne 0$, $\forall i$.

There are several ways that an anchor can move away from the margin boundary. When $\Delta\gamma'_{[i]}<0$ (which implies that $\Delta\gamma_{[i]}>0$ since the error residual should decrease, i.e. $\Delta\lambda<0$), an anchor moves from $\mathcal{B}_{\mathrm{U}}$ to $\mathcal{R}_{\mathrm{U}}$ ($0\le\gamma_{[i]}^{\mathrm{old}}<1$, $\gamma_{[i]}=1$) or from $\mathcal{B}_{\mathrm{L}}$ to $\mathcal{M}$ ($-1\le\gamma_{[i]}^{\mathrm{old}}<0$, $\gamma_{[i]}=0$). When $\Delta\gamma'_{[i]}>0$ (or equivalently $\Delta\gamma_{[i]}<0$), an anchor moves from $\mathcal{B}_{\mathrm{U}}$ to $\mathcal{M}$ ($0<\gamma_{[i]}^{\mathrm{old}}\le 1$, $\gamma_{[i]}=0$) or from $\mathcal{B}_{\mathrm{L}}$ to $\mathcal{R}_{\mathrm{L}}$ ($-1<\gamma_{[i]}^{\mathrm{old}}\le 0$, $\gamma_{[i]}=-1$). For all these cases, a possible value of $\Delta\lambda$ for each anchor can be calculated as

$$\Delta\lambda_i=\frac{\Delta\gamma_{[i]}}{\Delta\gamma'_{[i]}},\quad i=1,\ldots,m+1.$$

Note that those anchors in the segments whose delimiting non-zero breaks have the same sign ($\Delta\gamma'_{[i]}=0$) are not considered since according to (2.19) the error residuals will not change by moving those segments. Since the decreasing of $\lambda$ should take the smallest possible step in order to make each $\Delta\gamma_{[i]}$ valid ($\Delta\gamma_{[i]}\in[-1,0]$ or $\Delta\gamma_{[i]}\in[0,1]$), we select

$$\Delta\lambda_a = \max(\{\Delta\lambda_i, i = 1, \ldots, m+1\}) .$$

We should also consider the case where a new non-zero break should be introduced (consequently a new segment is created) in order to reduce the error residuals. For a break $\delta_i = 0$ to become non-zero in the new optimal solution, it must satisfy the condition $\lambda \cdot \text{sign}(\delta_i) - \sum_{j=i}^{n} \gamma_j = 0$. Therefore we have

$$\begin{aligned}
\lambda^{\text{old}} \cdot \text{sign}(\delta_i) - \sum\nolimits_{j \ge i} \gamma_j^{\text{old}} &= \lambda^{\text{old}} \cdot \text{sign}(\delta_i) - \sum\nolimits_{j \ge i} \gamma_j^{\text{old}} - \lambda \cdot \text{sign}(\delta_i) + \sum\nolimits_{j \ge i} \gamma_j \\
&= -\Delta\lambda \cdot \text{sign}(\delta_i) + \sum\nolimits_{j \ge i, j \in \mathcal{B}} \Delta\gamma_j \\
&= \Delta\lambda \left( \sum\nolimits_{j \ge i, j \in \mathcal{B}} \frac{\Delta\gamma_j}{\Delta\lambda} - \text{sign}(\delta_i) \right)
\end{aligned}$$

and consequently

$$\Delta\lambda = \frac{\sum_{j \ge i} \gamma_j^{\text{old}} - \lambda^{\text{old}} \cdot \text{sign}(\delta_i)}{\text{sign}(\delta_i) - \sum\limits_{\substack{j \ge i \\ j \in \mathcal{B}}} \frac{\Delta\gamma_j}{\Delta\lambda}} . \tag{2.24}$$

For each existing non-zero break $\delta_{(k)}^{\text{old}}$ we have $\Delta\lambda \cdot \text{sign}(\delta_{(k)}^{\text{old}}) = \sum_{j \ge \mathcal{P}(k), j \in \mathcal{B}} \Delta\gamma_j$, which implies

$$\sum_{\substack{j \ge i \\ j \in \mathcal{B}}} \frac{\Delta\gamma_j}{\Delta\lambda} = \begin{cases} 0, & i > \mathcal{B}[m+1] \\ \text{sign}(\delta_{(k-1)}^{\text{old}}), & \mathcal{B}[k-1] < i \le \mathcal{B}[k],\ 2 \le k \le m+1 \\ 0, & i \le \mathcal{B}[1] \end{cases} .$$

Note that when $[k-1] < i \le [k]$ and $2 \le k \le m+1$, $\text{sign}(\delta_i)$ must take value $-\text{sign}(\delta_{(k-1)}^{\text{old}})$ to make (2.24) valid. For $i > [m+1]$ and $i \le [1]$, we choose whichever sign for $s_i$ that yields a larger negative value of $\Delta\lambda_i$. A possible value of $\Delta\lambda$ for each candidate break can be determined by

$$\Delta\lambda_i = \begin{cases} s_i \sum\limits_{j \ge i} \gamma_j^{\text{old}} - \lambda^{\text{old}}, & 2 \le i \le \mathcal{B}[1] \text{ or } i > \mathcal{B}[m+1] \\ \frac{1}{2} s_{(k-1)}^{\text{old}} \sum\limits_{j=\mathcal{P}(k-1)}^{i-1} \gamma_j^{\text{old}} - \lambda^{\text{old}}, & \mathcal{B}[k-1] < i \le \mathcal{B}[k], 2 \le k \le m+1 \end{cases} \tag{2.25}$$

When deriving the second case of (2.25), we use the property $\lambda^{\text{old}}\cdot s_{(k-1)}^{\text{old}}=\sum_{j\geq\mathcal{P}(k-1)}\gamma_j^{\text{old}}$. In order to achieve the smallest step of reduction of $\lambda$, we select $\Delta\lambda$ by

$$\Delta\lambda_b=\max(\{\Delta\lambda_i, i=1,\ldots,m+1\})$$

We choose the final value of $\Delta\lambda$ as

$$\Delta\lambda=\max(\Delta\lambda_a,\Delta\lambda_b) \tag{2.26}$$

If we have $\Delta\lambda=\Delta\lambda_a$, we remove the corresponding anchor from set $\mathcal{B}$. If we have $\Delta\lambda=\Delta\lambda_b$, by inserting to $\mathcal{P}$ a new non-zero break $\delta_i$ with a chosen sign, an existing segment is split to two new segments: one of the new segments containing an existing anchor should have minimal error residual and cannot be further optimized; the other new segment should have error residual $\lambda^{\text{old}}+\Delta\lambda$ if it is the first/last segment or $\lambda^{\text{old}}+2\Delta\lambda$ otherwise, which addresses the majority of the error residual of the segment before splitting. Given the chosen value of $\Delta\lambda$, the Lagrange multipliers $\boldsymbol{\gamma}^{\text{old}}$ are updated by

$$\gamma_{[i]}=\gamma_{[i]}^{\text{old}}+\Delta\lambda\cdot\Delta\gamma'_{[i]},\quad i=1,\ldots,|\mathcal{B}|.$$

By either removing an existing anchor from $\mathcal{B}$ or adding a new non-zero break to $\mathcal{P}$, we have $|\mathcal{B}|=|\mathcal{P}|$.

### 2.5.3 Determining the Change of *s*

After the candidate segment to be moved is chosen in the previous step, we need to determine the next possible value of $s$ and how far the candidate segment can be moved. Suppose the optimal solution corresponding to $s^{\text{old}}$ is $\boldsymbol{\delta}^{\text{old}}=\{\delta_i^{\text{old}}\}_{i=1}^n$, $\sum_{i=2}^n\left|\delta_i^{\text{old}}\right|=s^{\text{old}}$, we want to find the maximum amount of change in $s$, denoted by $\Delta s=s-s^{\text{old}}$, such that the sets $\mathcal{B}$, $\mathcal{P}$, and $\mathcal{M}$ determined by the KKT conditions in the previous step do not change. The optimal solution $\boldsymbol{\delta}$ corresponding to $s=s^{\text{old}}+\Delta s$ must satisfy

$$y_{[i]} = \delta_1 + \sum_{k:\mathcal{P}(k)\le\mathcal{B}[i]} \delta_{(k)} \pm \epsilon_i = \delta_1^{\text{old}} + \sum_{k:\mathcal{P}(k)\le\mathcal{B}[i]} \delta_{(k)}^{\text{old}} \pm \epsilon_i\,,\ \ i = 1,\ldots,m\,, \tag{2.27}$$

$$\sum_{i=1}^{m} \left|\delta_{(i)}\right| - \sum_{i=1}^{m} \left|\delta_{(i)}^{\text{old}}\right| = \Delta s\,. \tag{2.28}$$

(2.27) and (2.28) lead to

$$\Delta\delta_1 + \sum_{k:\mathcal{P}(k)\le\mathcal{B}[i]} \Delta\delta_{(k)} = 0\,,\ \ i = 1,\ldots,m\,, \tag{2.29}$$

$$\sum_{i=1}^{m} \Delta\delta_{(i)} \cdot \text{sign}(\delta_{(i)}^{\text{old}}) = \Delta s\,, \tag{2.30}$$

where $\Delta\delta_i = \delta_i - \delta_i^{\text{old}}$, $i = 1,\ldots,n$. We introduce vectors $\mathbf{d} = [d_0, d_1, \ldots, d_m]^{\text{T}}$, $\mathbf{s} = [s_0, s_1, \ldots, s_m]^{\text{T}}$, and $\mathbf{c} = [0,\ldots,0,1]^{\text{T}}$ where

$$d_0 = \frac{\Delta\delta_1}{\Delta s}\,,\ \ d_i = \frac{\Delta\delta_{(i)}}{\Delta s}\,,\ i = 1,\ldots,m$$

$$s_0 = 0\,,\ \ s_i = \text{sign}(\delta_{(i)}^{\text{old}})\,,\ i = 1,\ldots,m$$

(2.29) and (2.30) become

$$d_0 + \sum_{k:\mathcal{P}(k)\le\mathcal{B}[i]} d_k = 0\,,\ \ i = 1,\ldots,m$$

$$\sum_{i=1}^{m} d_i s_i = 1$$

which can be written in a matrix form

$$\begin{bmatrix} \mathbf{D} \\ \mathbf{s}^{\text{T}} \end{bmatrix} \cdot \mathbf{d} = \mathbf{c}\,, \tag{2.31}$$

where $\mathbf{D}$ is a $m$ by $m+1$ matrix. To have a unique solution of $\mathbf{d}$, the matrix $\begin{bmatrix} \mathbf{D}^{\text{T}} & \mathbf{s} \end{bmatrix}^{\text{T}}$ must have full rank $m+1$. Therefore each row of $\mathbf{D}$ must have at least one more 1 than its previous row. Consequently, $\mathbf{D}$ can only take one of the three forms as follows:

(1) $\mathbf{D}=\begin{bmatrix}\mathbf{1} & \mathbf{L}'\end{bmatrix}$, where $\mathbf{1}$ is a $m$-vector of ones and $\mathbf{L}'$ is a $m$ by $m$ matrix where $[\mathbf{L}']_{ij}=1$ for $i>j$ and $[\mathbf{L}']_{ij}=0$ for $i\leq j$;

(2) $\mathbf{D}=\begin{bmatrix}\mathbf{1} & \mathbf{L}\end{bmatrix}$, where $\mathbf{L}$ is a lower triangular matrix whose non-zero elements are all ones;

(3) $\mathbf{D}=\begin{bmatrix}\mathbf{1} & \mathbf{L}^*\end{bmatrix}$, where $\mathbf{L}^*$ is composed by the first $i$ rows of $\mathbf{L}'$ and the last $m-i$ rows of $\mathbf{L}$.

The solutions to (2.31) have three forms corresponding to the three cases of $\mathbf{D}$:

(1) $d_m = s_m$ and $d_i = 0$, $i = 0,\ldots,m-1$;

(2) $d_0 = -s_1$, $d_1 = s_1$, and $d_i = 0$, $i = 2,\ldots,m$;

(3) if $s_i \neq s_{i+1}$, $d_i = 0.5s_i$, $d_{i+1} = -0.5s_i$, and $d_k = 0$, $\forall k \neq i$. There is no solution when $s_i = s_{i+1}$, which complies with the analysis in the previous section that a segment delimited by breaks with the same sign must have zero error residual and hence should not be further adjusted.

The solutions have intuitive interpretations. The first case corresponds to moving a tail segment of the profile of $\hat{\boldsymbol{\eta}}$ starting from locus $\mathcal{P}(m)$ in the direction of $\text{sign}(\delta_{(m)}^{\text{old}})$. The second case corresponds to moving the first segment of $\hat{\boldsymbol{\eta}}$ ended at locus $\mathcal{P}(1)-1$ in the opposite direction of $\text{sign}(\delta_{(1)}^{\text{old}})$. The last case corresponds to moving a gate-shaped segment from loci $\mathcal{P}(i)$ to $\mathcal{P}(i+1)-1$ in the direction of $\text{sign}(\delta_{(i)}^{\text{old}})$; $\hat{\eta}_i$ outside the segment is not affected since $\delta_{(i)}^{\text{old}}$ and $\delta_{(i+1)}^{\text{old}}$ change in opposite directions but with the same amount, i.e. $\Delta\delta_{(i)} = -\Delta\delta_{(i+1)}$. It shall be noted that, compared with the path algorithm for fused quantile regression in (Li and Zhu, 2007), solving the change of breaks in FMR does not involve matrix inversion and is purely analytic, which significantly improves the computational efficiency of FMR.

At each locus, the change of error residual with respect to $\Delta s$ is

$$\Delta r_i = \frac{(y_i - \hat{\eta}_i) - (y_i - \hat{\eta}_i^{\text{old}})}{\Delta s} = -\frac{\Delta \delta_1}{\Delta s} - \sum_{k:\mathcal{P}(k)\le i} \frac{\Delta \delta_{(k)}}{\Delta s} = -d_0 - \sum_{k:\mathcal{P}(k)\le i} d_k\,, \quad i = 1,\ldots,n\,.$$

Corresponding to the three cases discussed earlier, the solution of $\Delta r_i$ are

(1) $\Delta r_i = -s_m$, $\forall i \ge \mathcal{P}(m)$; $\Delta r_i = 0$, $\forall i < \mathcal{P}(m)$;

(2) $\Delta r_i = s_1$, $\forall i < \mathcal{P}(1)$; $\Delta r_i = 0$, $\forall i \ge \mathcal{P}(1)$;

(3) $\Delta r_i = -0.5 s_k$, $\mathcal{P}(k) \le i < \mathcal{P}(k+1)$; $\Delta r_i = 0$, $\forall i$, $i < \mathcal{P}(k)$ or $i \ge \mathcal{P}(k+1)$.

If $\Delta r_i < 0$, the segment moves upward and an observation in $\mathcal{R}_{\text{U}}$ or $\mathcal{M}$ attaches to $\mathcal{B}_{\text{U}}$ or $\mathcal{B}_{\text{L}}$, respectively. For each locus, a possible value of $\Delta s$ is solved by

$$\Delta s_i = \begin{cases} [\epsilon_i - (y_i - \hat{\eta}_i^{\text{old}})] / \Delta r_i, & i \in \mathcal{R}_{\text{U}} \\ [-\epsilon_i - (y_i - \hat{\eta}_i^{\text{old}})] / \Delta r_i, & i \in \mathcal{M} \end{cases}$$

If $\Delta r_i > 0$, the segment moves downward and an observation in $\mathcal{M}$ or $\mathcal{R}_{\text{L}}$ attaches to $\mathcal{B}_{\text{U}}$ and $\mathcal{B}_{\text{L}}$, respectively. For each locus, a possible value of $\Delta s$ is solved by

$$\Delta s_i = \begin{cases} [\epsilon_i - (y_i - \hat{\eta}_i^{\text{old}})] / \Delta r_i, & i \in \mathcal{M} \\ [-\epsilon_i - (y_i - \hat{\eta}_i^{\text{old}})] / \Delta r_i, & i \in \mathcal{R}_{\text{L}} \end{cases}$$

To take the smallest increment of $s$, we choose

$$\Delta s^* = \min\{\Delta s_i\}\,.$$

The breaks are then updated by

$$\delta_{(i)} = \delta_{(i)}^{\text{old}} + \Delta s^* \cdot d_i\,, \quad i = 0,\ldots,m\,.$$

The error residuals at the loci within the moved segment are updated according to the new values of $\boldsymbol{\delta}$.

We have shown that $\lambda$ either reduces or stays unchanged at each iteration of the algorithm. Since $\lambda$ is a non-negative Lagrange multiplier as specified in (2.13), the path algorithm should stop when $\lambda$ becomes 0. On the other hand, $s$, which is the sum of absolute magnitudes of the non-zero breaks, either increases or stays unchanged during the iterations. Therefore, we can either wait for the algorithm to stop (in order to obtain the full solution-path), or alternatively we can set a non-negative threshold $s^*$ such that when $s \geq s^*$ we terminate the algorithm.

## 2.6 Estimation of the Margin Size

The parameters $\boldsymbol{\epsilon} = \{\epsilon_i\}_{i=1}^n$ of the loss functions control the solution of FMR. Here we propose a heuristic method to determine the parameters for the case $\epsilon_i = \epsilon$ , $i = 1, \ldots, n$ . Intuitively, the widths of margins should be proportional to the scales of noises in the signals. Therefore, we can decide $\epsilon$ based on the estimation of noise variance in the signal profile.

In real-world copy number data, the noises usually follow normal distributions after appropriately transforming the original signals. In Figure 2.3, we show several Q-Q plots of the empirical distributions of noises in SNP array copy number profiles versus the theoretical standard normal distribution. We randomly select four samples of normal, serous borderline tumor (SBT), low-grade (LG), and high-grade (HG) serous carcinomas from a SNP array copy number dataset of ovarian cancers. For each sample, we randomly select 1000 observations from one or more continuous segments in the signal profile. The observations are standardized to have zero mean and unit standard deviation. The empirical distribution is then compared to the standard normal distribution using Q-Q plot. We can see that in general the empirical distributions resemble the standard normal distribution.

In order to estimate the noise distribution in the signals, we assume that the independent noises $\{N_i\}_{i=1}^n$ in (2.1) conform to normal distribution $N(0, \sigma^2)$ . Accordingly, a copy number signal $Y_i$ conforms to normal distribution $N(\eta_i, \sigma^2)$ . Since there are only a limited number of states in the true copy numbers $\{\eta_i\}_{i=1}^n$ , the observations $\{y_i\}_{i=1}^n$ thus can be considered as being drawn independently from a mixture of normal distributions with the true copy number states as the means. Therefore we can estimate the variance $\sigma^2$ by fitting a standard finite normal mixture (SFNM) (McLachlan and Peel, 2000) to the pooled observations $\{y_i\}_{i=1}^n$ .

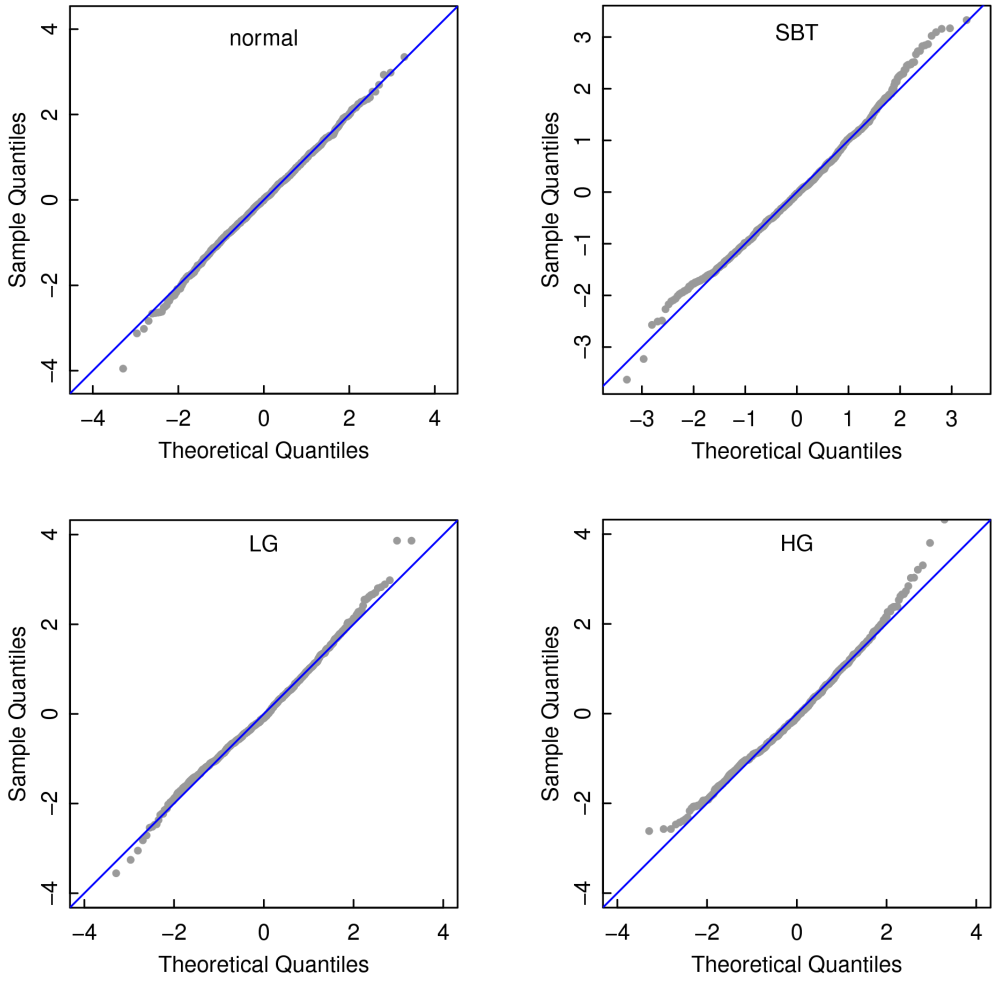


Figure 2.3. The Q-Q plots of the empirical distributions of standardized SNP array copy number signals versus the theoretical standard normal distribution.

The probability density function (pdf) of an SFNM with $K$ components (McLachlan and Peel, 2000) can be expressed as

$$p(y) = \sum_{k=1}^{K} \alpha_k p(y \mid \mu_k, \sigma_k^2),$$

where $p(y \mid \mu_k, \sigma_k^2)$ is the pdf of normal distribution $\mathcal{N}(\mu_k, \sigma_k^2)$; $\alpha_k$, $\mu_k$, and $\sigma_k^2$ are the mixing proportion, mean, and variance of the $k$ th component in the mixture. We use Expectation Maximization (EM) algorithm to compute the estimates $\hat{\alpha}_k$, $\hat{\mu}_k$, and $\hat{\sigma}_k^2$ of the corresponding parameters (Bilmes, 1998) via the following iterative steps:

E step: $$\hat{p}_{i,k} = p(k \mid y_i, \Theta) = \frac{\hat{\alpha}_k p(y_i \mid \hat{\mu}_k, \hat{\sigma}_k^2)}{\sum_{k'=1}^{K} \hat{\alpha}_{k'} p(y_i \mid \hat{\mu}_{k'}, \hat{\sigma}_{k'}^2)}$$

M step: $$\hat{\alpha}'_k = \frac{1}{n}\sum_{i=1}^{n} \hat{p}_{i,k}$$

$$\hat{\mu}'_k = \frac{\sum_{i=1}^{n} \hat{p}_{i,k} y_i}{\sum_{i=1}^{n} \hat{p}_{i,k}}$$

$$\hat{\sigma}'^2_k = \frac{\sum_{i=1}^{n} \hat{p}_{i,k} \cdot (y_i - \hat{\mu}'_k)^2}{\sum_{i=1}^{n} \hat{p}_{i,k}}$$

where $\Theta' = \{\hat{\alpha}'_k, \hat{\mu}'_k, \hat{\sigma}'^2_k\}_{k=1}^{K}$ are the estimates of the parameters of the components at the current step of the EM algorithm, and $\Theta = \{\hat{\alpha}_k, \hat{\mu}_k, \hat{\sigma}_k^2\}_{k=1}^{K}$ are the estimates at the previous step. The EM algorithm terminates when a pre-determined maximum number of steps is reached or the absolute difference between the conditional expectations of log-likelihood $Q(\Theta', \Theta)$ (Bilmes, 1998) of two consecutive steps is smaller than a pre-defined threshold, where

$$Q(\Theta', \Theta) = \sum_{k=1}^{K}\sum_{i=1}^{n} \log(\hat{\alpha}'_k p(y_i \mid \hat{\mu}'_k, \hat{\sigma}'^2_k)) p(k \mid y_i, \Theta). \tag{2.32}$$

In order to select an optimal number of components for the mixture, we test a range of values of $K$ and choose the one yielding the smallest minimum description length (MDL) (Wang, et al., 2000)

$$\mathrm{MDL}(K) = -Q + \frac{3K-1}{2}\log n,$$

where $Q$ is the value of (2.32) in the final step of the EM algorithm. Denote the optimal number of components by $K^*$ and the corresponding estimates of parameters by $\Theta^* = \{\hat{\alpha}_k^*, \hat{\mu}_k^*, \hat{\sigma}_k^{*2}\}_{k=1}^{K^*}$, we set the noise variance $\tilde{\sigma}^2$ to be a weighted sum of the variances of the mixture components

$$\tilde{\sigma}^2 = \sum_{k=1}^{K^*} \hat{\alpha}_k^* \hat{\sigma}_k^{*2}.$$

Alternatively, we can also use the variance of the major component of the mixture, i.e.

$$\tilde{\sigma}^2 = \hat{\sigma}_k^{*2}, \quad k = \arg\max_{i=1,\ldots,K} \hat{\alpha}_i^* .$$

Given the estimate $\tilde{\sigma}^2$, the parameter $\epsilon$ of the loss function $L_\epsilon$ is defined as

$$\epsilon = a\tilde{\sigma} ,$$

where $a$ is a non-negative constant to be determined by users. We can select $a$ by testing a set of values using two-fold cross-validation (Eilers and de Menezes, 2005). For the sake of simplicity, we assume that there is an even number of observations. We split $\mathbf{y} = \{y_i\}_{i=1}^n$ into two subsets, $\mathbf{y}_{\text{odd}} = \{y_{2i-1}\}_{i=1}^{n/2}$ and $\mathbf{y}_{\text{even}} = \{y_{2i}\}_{i=1}^{n/2}$. For a given value of $a$, we first estimate copy numbers $\hat{\boldsymbol{\eta}} = \{\hat{\eta}_i\}_{i=1}^{n/2}$ using $\mathbf{y}_{\text{odd}}$, and measure the fitting or testing error on $\mathbf{y}_{\text{even}}$ as $e_1 = \sum_{i=1}^{n/2} |y_{2i} - \hat{\eta}_i|$. We then estimate copy numbers again using $\mathbf{y}_{\text{even}}$ , and measure the fitting error on $\mathbf{y}_{\text{odd}}$ as $e_2 = \sum_{i=1}^{n/2} |y_{2i-1} - \hat{\eta}_i|$. Among a set of values of $a$, we select the one that yields the smallest overall error $e_1 + e_2$.

In practice, we found that the detection results are not highly sensitive to small changes in $a$ and $1.5 \le a \le 2$ generally yields satisfactory results. Usually we can simply use empirical values of $a$ for different types of copy number data without using the two-fold CV to select $a$.

## 2.7 Two-tier FMR

We briefly introduce a method that estimates $\epsilon_i$ for different loci, which is straightforward to implement based on the path algorithm that we have derived. In CGH and SNP array copy number data, the noise variance may be inhomogeneous in different genomic regions and/or at different levels of copy number signals. Instead of using the same $\epsilon$ for the entire signal profile, we estimate $\epsilon_i$ for each locus according to the local noise levels. The values of $\boldsymbol{\varepsilon}$ can be determined using a two-step procedure. In the first step, we apply FMR using the same value for all $\epsilon_i$ (as in 2.6) to detect copy number segments. In the second step, we first subtract the estimated copy numbers from the observations, i.e. $\mathbf{y}' = \mathbf{y} - \hat{\boldsymbol{\eta}}$. Ideally, the profile of $\mathbf{y}'$ does not contain breaks and the fluctuation in the profile is attributed to noises. For each locus $i$, we compute the sample variance of the observations in its neighborhood, i.e. $\hat{\sigma}_i^2 = \mathrm{var}\left(\{y'_k\}_{k=i-l_1}^{i+l_2}\right)$, where $l_1, l_2 \geq 0$ define the neighborhood. Finally, we apply FMR to the original signals $\mathbf{y}$ again with parameters $\epsilon_i = a\hat{\sigma}_i$, $i = 1, \ldots, n$, to detect copy number changes.

## 2.8 Detection of Consensus Copy Number Changes using FMR

FMR can be extended to detect consensus copy number changes in multiple CGH or SNP arrays. We call this extension Multi-Profile FMR (MPFMR). A consensus copy number change is a chromosomal region in which a significant number of the genomes of a population (e.g., patients with the same disease) have abnormal copy numbers, either amplifications or deletions. In cancer research, since a consensus deletion or amplification occurs frequently in the patients' tumor genomes, those genes harbored in the region may be among the driving causes of tumor initiation and/or progression and hence become important candidates for screening tumor suppressor genes or oncogenes.

Denote the matrix of observed copy number signals by $\mathbf{Y}_{m \times n}$, where each row of $\mathbf{Y}$ is a signal profile and each column corresponds to a probe locus. To find consensus copy number changes, we fit a single piecewise constant profile to the pooled signal profiles. The corresponding optimization problem is

$$\min \sum_{i=1}^{m} \sum_{j=1}^{n} \left( \xi_{ij} + \zeta_{ij} \right) \tag{2.33}$$

$$\text{s.t.} \begin{cases} \sum_{k=2}^{n} |\delta_k| \le s \\ y_{ij} - \sum_{k=1}^{j} \delta_k \le \epsilon_i + \xi_{ij}, & i = 1,\ldots,m,\ j = 1,\ldots,n \\ \sum_{k=1}^{i} \delta_k - y_{ij} \le \epsilon_i + \zeta_{ij}, & i = 1,\ldots,m,\ j = 1,\ldots,n \\ \xi_{ij} \ge 0,\ \zeta_{ij} \ge 0, & i = 1,\ldots,m,\ j = 1,\ldots,n \end{cases} \tag{2.34}$$

(2.33) and (2.34) can be analyzed in the same way as the LP problem of FMR and solved using a modified path algorithm for FMR.

We select a solution of breakpoints from the entire solution path of MPFMR in the following steps. First, for each set of solutions with the same set of breakpoint loci, we only keep the one with the largest value of $s$, which has the smallest fitting error. Second, in each of the retained solutions, we merge a segment to its adjacent segment if (1) its delimiting breaks have the same sign, i.e., $\delta_i \cdot \delta_j > 0$, and (2) its length is shorter than a user defined threshold. We also merge two adjacent segments if (1) the delimiting breaks of either segment have the same sign, and (2) the difference between the estimated amplitudes of the two segments is smaller than $d \cdot \epsilon$, where $d$ is a user defined small positive number (e.g., $d$=0.1). By these two operations, we remove those segments that are either too short or have no significant difference from the adjacent segments in their estimated copy numbers. Finally, for each of the solutions, we adjust the amplitude of each segment to minimize the $\epsilon$-insensitive loss of the observations in the segment. Given the set of modified solutions, we use the Schwarz Information Criterion (Li and Zhu, 2007) to select an optimal solution,

$$\text{SIC}(\hat{\boldsymbol{\eta}} \mid \mathbf{Y}) = \log\left(\frac{L(\mathbf{Y}, \hat{\boldsymbol{\eta}})}{mn}\right) + K \cdot \frac{\log(mn)}{2mn},$$

where $m$ is the number of profiles, $n$ is the number of loci, $K$ is the number of segments in the solution, and $L(\mathbf{Y}, \hat{\boldsymbol{\eta}})$ is the total loss

$$L(\mathbf{Y}, \hat{\boldsymbol{\eta}}) = \sum_{i=1}^{m} \sum_{j=1}^{n} \left| y_{ij} - \hat{\eta}_j \right|.$$

Note that here we use the absolute difference instead of the $\epsilon$-insensitive loss as the loss function.

## 2.9 Chromosome Instability Index

We propose a biologically-plausible definition of Chromosome INstability (CIN) index to summarize the frequency and amplitudes of copy number variations in the copy number signals measured on a chromosome. Given the raw signal profile $\mathbf{y}$ of a chromosome $k$, we first use FMR to detect copy number alterations in $\mathbf{y}$. We make gain/loss calls on the detected segments delimited by non-zero breakpoints using empirical thresholds $t_{\text{gain}}$ and $t_{\text{loss}}$ such that a segment is considered a gain or a loss if its amplitude is greater than $t_{\text{gain}}$ or smaller than $t_{\text{loss}}$, respectively. Denote the $p$ gain/loss segments and their amplitudes by $S_1, \ldots, S_p$ and $a_1, \ldots, a_p$, respectively. We map the amplitude $a_i$, $0 \le a_i < t_{\text{loss}}$, of a loss segment $S_i$ to the scale of gain amplitudes by

$$a_i \leftarrow \frac{t_{\text{loss}} - a_i}{a_i}\left(A - t_{\text{gain}}\right) + t_{\text{gain}}, \tag{2.35}$$

where $A > t_{\text{gain}}$ is a pre-selected constant representing the maximum amplitude of gains. Finally, we calculate the CIN index of the signal profile as $\text{CIN}_k = (\sum_i a_i)/n_k$, where $n_k$ is the number of observations in $\mathbf{y}$. A genome-wide CIN index can be calculated as $\text{CIN} = \sum_k \log(\text{CIN}_k + 1)$. In 2.10.4, we demonstrate the application of the CIN index on an ovarian cancer copy number dataset.

## 2.10 Results

### 2.10.1 Analysis of Running Time

Copy number data assayed by CGH and SNP arrays usually consists of a large number of observations. In order to be applicable routinely to detect copy number changes in high-density CGH and SNP arrays, a detection algorithm should be computationally efficient. We designed

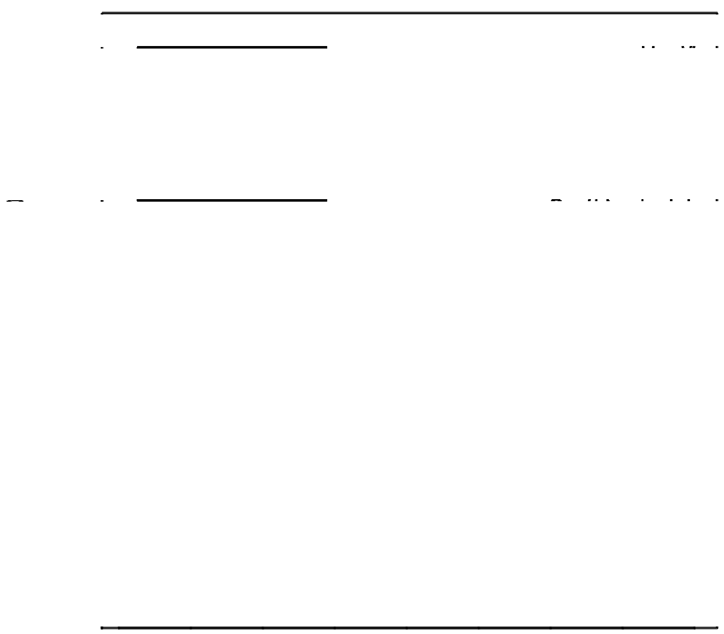


Figure 2.4. Running time of FMR on noise profiles of different lengths.

three experiments to empirically demonstrate the efficiency of FMR in terms of the running time using both synthetic data and real SNP array data.

In the first experiment, we show the running time of FMR on noise profiles of different lengths. For each profile, the observations are generated independently from a standard normal distribution $\mathcal{N}(\mu = 0, \sigma^2 = 1)$. We create 100 profiles of lengths $l \in \{2000k, k = 1, \ldots, 100\}$ and apply FMR with $a$ equal to 0.5, 1.0, 1.5 and 2.0 on each of them. The running time of FMR with respect to the length of profile is plotted in Figure 2.4. For each value of $a$, we fit the observed running time with a second order polynomial, which is plotted as a curve in the figure.

In the second experiment, we perform the same test on real SNP array data. We randomly select 10 SNP arrays of high-grade tumor samples from an ovarian cancer dataset. We concatenate the profiles into a single profile of 2383040 loci and uniformly sample 100 profiles of length $l \in \{2000k, k = 1, \ldots, 100\}$ from it. We apply FMR with $a$ equal to 0.5, 1.0, 1.5, and 2.0

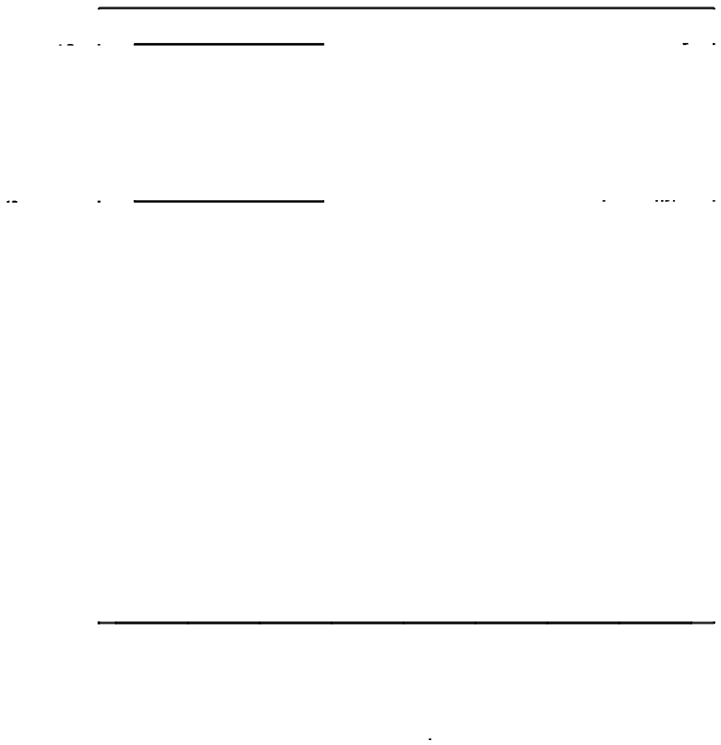


Figure 2.5. Running time of FMR on signal profiles of different lengths sampled from SNP array data of high-grade tumor samples in an ovarian cancer dataset.

on these profiles. The relationship between the running time and the length of profile is plotted in Figure 2.5. Again, the running time of FMR can be fitted using second order polynomials. Figure 2.4 and Figure 2.5 empirically show that the path algorithm of FMR has time complexity $O(n^2)$, where $n$ is the number of loci in a signal profile.

In the third experiment, we compare the running time of FMR, fused Lasso, and CBS on the SNP array data used in the previous experiment. We create 15 profiles of lengths $l \in \{\lceil 2383040 \cdot 2^{-k} \rceil, k = 0, \ldots, 14\}$ by uniformly sampling observations from the concatenated SNP profile of 2383040 loci. We use $a = 1.5$ for FMR and the default parameters for fused Lasso and CBS. The relationships between the running time and the length of profile (both in log-scale) for the three methods are plotted in Figure 2.6. FMR is faster than fused Lasso by 4 to 12 times. FMR also has significant advantage compared with CBS in most of the cases that

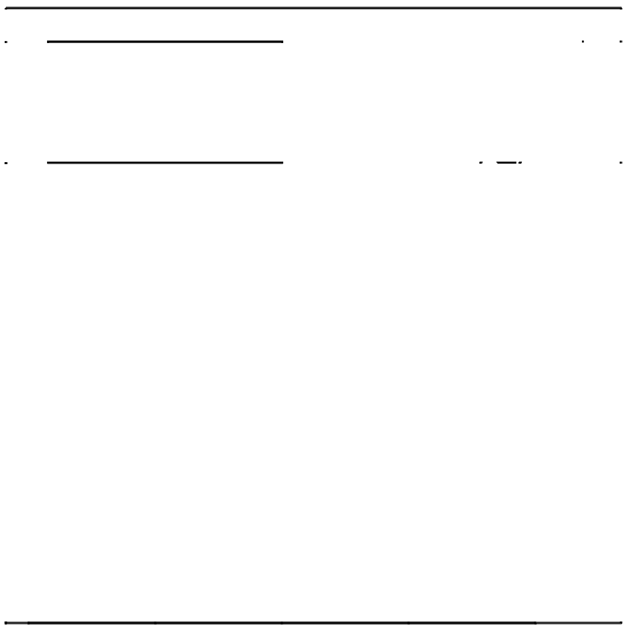


Figure 2.6. Comparison of the running time between FMR, fused Lasso, and CBS on SNP signal profiles of different lengths.

happen in real world SNP array data. The running time of these methods are also presented in Table 2.1. We can see that FMR is an efficient detection algorithm that can be used in routine analysis of high-density CGH and SNP arrays.

### 2.10.2 Detecting DNA Copy Number Changes in A Single Signal Profile

In this section, we test the performance of FMR for detecting copy number changes in a single signal profile. We first compare FMR with several widely adopted copy number change detection methods using two simulation schemes. We then visually demonstrate the detection results of FMR on real CGH and SNP array copy number data.

In the first simulation study, we test the performance of FMR for detecting DNA copy number gains using signal profiles with a single level of amplifications. The results and conclusions can be generalized to copy number losses by detecting gains in the negative signals

Table 2.1. Comparison of the running time (in seconds) between FMR, fused Lasso, and CBS on SNP signal profiles of different lengths.

| Profile Length | FMR | Fused Lasso | CBS |
|---:|---:|---:|---:|
| 146 | 0.001 | 0.005 | 0.004 |
| 291 | 0 | 0.007 | 0.03 |
| 582 | 0.001 | 0.012 | 0.119 |
| 1,164 | 0.004 | 0.024 | 1.671 |
| 2,328 | 0.007 | 0.068 | 2.059 |
| 4,655 | 0.019 | 0.15 | 2.913 |
| 9,309 | 0.058 | 0.467 | 3.596 |
| 18,618 | 0.187 | 1.436 | 4.378 |
| 37,235 | 0.676 | 4.979 | 9.384 |
| 74,470 | 2.668 | 18.574 | 22.21 |
| 148,940 | 13.885 | 72.742 | 46.462 |
| 297,880 | 64.142 | 286.075 | 138.609 |
| 595,760 | 273.409 | 1138.321 | 532.702 |
| 1,191,520 | 1073.81 | 4528.064 | 1115.484 |
| 2,383,040 | 4227.453 | 19910.67 | 2331.777 |

of a profile containing losses. In the second simulation study, we test the performance of FMR for detecting multi-level copy number changes using signal profiles generated by Hidden Markov Models. In both simulations, we create synthetic data with different noise distributions and varying signal-to-noise ratios in order to systematically compare different detection methods.

In the simulation studies, the existing copy number change detection methods for comparison include fused Lasso (Tibshirani and Wang, 2008), Circular Binary Segmentation (CBS) (Venkatraman and Olshen, 2007), Gain and Loss Analysis of DNA (GLAD) (Hupé, et al., 2004), Hidden Markov Models (HMM), and Haar wavelets (HaarSeg) (Ben-Yaacov and Eldar, 2008). The corresponding software packages used in the experiments are listed in Table 2.2. For HMM, we use the BioHMM function in the BioConductor package snapCGH (Marioni, et al., 2006). BioHMM is originally proposed to incorporate clone distances into HMM to detect copy

Table 2.2. The existing copy number change detection methods and their software packages used for comparison in the simulation studies.

| **Method** | **Software** |
|---|---|
| Fused Lasso (Tibshirani and Wang, 2008) | CRAN, cghFLasso 0.2-1 |
| Circular Binary Segmentation (CBS) (Venkatraman and Olshen, 2007) | BioConductor, DNAcopy 1.20.0 |
| Gain and Loss Analysis of DNA (GLAD) (Hupé, et al., 2004) | BioConductor, GLAD 2.6.0 |
| BioHMM (Marioni, et al., 2006) | BioConductor, snapCGH 1.16.0 |
| HaarSeg (Ben-Yaacov and Eldar, 2008) | http://www.ee.technion.ac.il/Sites/People/YoninaEldar/ HaarSeg R implementation, version 1.0 |

number changes in array CGH data. In our simulated data the loci are equally distanced and we use BioHMM as a generic implementation of HMM.

### *2.10.2.1 Simulated data with a single level of amplifications*

We first compare FMR with five existing copy number change detection methods on simulated signal profiles containing a single level of amplifications. The performance is measured by area under the receiver operating characteristic curve (AUC).

The simulated data is constructed in a similar way as in (Lai, et al., 2005). To generate a simulated signal profile, we first create a profile of true copy numbers $\boldsymbol{\eta} = \{\eta_i\}_{i=1}^n$, $n = 2000$, with 10 evenly distanced amplified segments, each of length $l$, such that $\eta_i = a > 0$ within those segments and $\eta_i = 0$ otherwise. We then generate i.i.d. noises $\varepsilon_i$, $i = 1, \ldots, n$, from distribution $p_{\text{noise}}$ and transform them by $\varepsilon_i' = (\varepsilon_i - \upsilon) / \sigma$, where $\upsilon$ and $\sigma$ are the mode and standard deviation of distribution $p_{\text{noise}}$. The transformed noises are added to the true copy numbers to create a simulated signal profile $\mathbf{y} = \{y_i = \eta_i + \varepsilon_i'\}_{i=1}^n$. We use four different noise distributions in the simulations, including the standard normal distribution, t-distribution with degree of freedom

$\mathrm{df}=4$, log-normal distribution with parameters $\mu=0$ and $\sigma=0.5$, and an empirical distribution of noises in real SNP microarray signal profiles. Besides the normal distribution that is frequently used as the noise distribution in literatures, we use t and log-normal distributions to generate noises with long tail and skewed distributions. The empirical distribution of SNP array noises is used to mimic noises in real microarray data. The pdf of log-normal distribution with parameters $\mu$ and $\sigma$ is

$$p(x;\mu,\sigma)=\frac{1}{x\sigma\sqrt{2\pi}}e^{-\frac{(\log(x)-\mu)^2}{2\sigma^2}}\;,\;\; x>0\,,$$

where $\log(\cdot)$ is the natural logarithm. For SNP microarray noises, $\upsilon$ and $\sigma$ are defined to be the sample mean and sample standard deviation of the empirical distribution.

For each combination of noise distribution $p_{\text{noise}}$, length of amplified segments $l$, $l\in\{5,10,20,50,100\}$, and amplitude $a$, $a\in\{0.1,0.5,1.0,1.5,2.0\}$, we create a dataset consisting of 100 signal profiles in order to measure the average performance of each detection method. Since the noises are transformed to have $\mathrm{SD}=1$, the values of $a$ are effectively SNRs. For a simulated signal profile $\mathbf{y}=\{y_i\}_{i=1}^n$, we denote the estimated copy numbers generated by a detection method as $\hat{\boldsymbol{\eta}}=\{\hat{\eta}_i\}_{i=1}^n$. In order to make aberrant calls, we set a threshold $\theta$ such that any locus $i$ with $\hat{\eta}_i>\theta$ is considered an amplification. For a particular value of $\theta$, the true positive rate (TPR) and false positive rate (FPR) can be defined as (Lai, et al., 2005)

$$\mathrm{TPR}(\theta)=\frac{\#\left\{\{i\,|\,\hat{\eta}_i>\theta\}\cap\{i\,|\,i\in I,\eta_i>\theta\}\right\}}{\#\{i\,|\,i\in I,\eta_i>\theta\}}$$

$$\mathrm{FPR}(\theta)=\frac{\#\left\{\{i\,|\,\hat{\eta}_i>\theta\}\cap\{i\,|\,\eta_i<\theta\}\right\}}{\#\{i\,|\,\eta_i<\theta\}}$$

where $\#\{A\}$ is the cardinality of set $A$ and $I$ is the set of loci within the true amplified segments, i.e. $I=\{i\,|\,\eta_i=a\}$. By varying the threshold $\theta$, we get pairs of TPR and FPR and consequently the empirical ROC curve. The area under the ROC curve (AUC) can be calculated by the trapezoidal rule. For each dataset, we calculate the AUC for each of the 100 signal profiles and use the mean and SD of AUC as the performance measure: larger means and smaller SDs of AUC indicate better performances.

We apply FMR and other competing methods on the simulated datasets. For FMR, we set the parameter $\epsilon = 0.5\hat{\sigma}$ in the $\epsilon$-insensitive loss function, where $\hat{\sigma}$ is the estimated noise SD of a signal profile as described earlier. We choose the solution in the whole solution path such that the corresponding value of $s$ is equal to the sum of the largest 0.5% of the absolution differences $\{| y_i - y_{i-1} |\}_{i=2}^{n}$. For the other methods, we use the default parameters in their software packages. The performances of all the methods under different noise distributions are listed in Table 2.3 to Table 2.6. For each dataset, the best performance is highlighted in bold face. We can see that FMR has superior performance in terms of the mean AUC in majority of the cases. FMR also tends to have smaller SDs of AUC compared with the others when the performances are close. In Figure 2.7 to Figure 2.10, we plot the ROC curves of the mean FPRs versus mean TPRs for those datasets where $l \in \{5,10,20,50\}$ and $a \in \{0.5,1.0,1.5,2.0\}$, which are more informative for comparing the competing methods. On each ROC curve, each data point (cross) corresponds to the mean FPR and mean TPR at a particular threshold across all 100 signal profiles in a dataset. We plot the ROC curves in the region of $0 \le \text{FPR} \le 0.2$ and $0 \le \text{TPR} \le 1$ in order to highlight the sensitivities of the competing methods under low FPRs. When the noises conform to normal, t, and log-normal distributions, FMR has better sensitivities compared with the other methods in most of the cases. For the datasets with SNP array noises, HaarSeg has better sensitivities as compared to the others when $l = 5$ and $l = 10$, while FMR performs slightly better than HaarSeg when $l = 20$ and $l = 50$. In general, the experimental results show that FMR is a competitive method for detecting gain/loss segments in copy number signal profiles.

Table 2.3. The means and standard deviations of AUC of the copy number change detection methods on datasets where the noises conform to the standard normal distribution.

| *a* | *l* | **FMR** | **FLasso** | **CBS** | **GLAD** | **BioHMM** | **HaarSeg** |
|---|---|---|---|---|---|---|---|
| 0.1 | 5 | **0.5479** (0.0953) | 0.5221 (0.0809) | 0.5000 (0.0000) | 0.5000 (0.0000) | 0.4994 (0.0067) | 0.5058 (0.0290) |
|  | 10 | **0.5604** (0.0801) | 0.5353 (0.0658) | 0.5000 (0.0000) | 0.5000 (0.0000) | 0.5000 (0.0007) | 0.5004 (0.0283) |
|  | 20 | **0.5776** (0.0619) | 0.5647 (0.0697) | 0.5002 (0.0029) | 0.5000 (0.0000) | 0.5000 (0.0000) | 0.5073 (0.0262) |
|  | 50 | 0.5942 (0.0538) | **0.6136** (0.0677) | 0.5004 (0.0032) | 0.5000 (0.0000) | 0.5003 (0.0026) | 0.5122 (0.0249) |
|  | 100 | 0.5947 (0.0512) | **0.6262** (0.0687) | 0.5008 (0.0056) | 0.5000 (0.0000) | 0.5009 (0.0057) | 0.5126 (0.0282) |
| 0.5 | 5 | **0.6708** (0.0757) | 0.5888 (0.0782) | 0.5000 (0.0000) | 0.5000 (0.0000) | 0.5000 (0.0000) | 0.5074 (0.0265) |
|  | 10 | **0.7666** (0.0625) | 0.6929 (0.0703) | 0.5000 (0.0003) | 0.5000 (0.0000) | 0.5001 (0.0009) | 0.5191 (0.0327) |
|  | 20 | **0.8311** (0.0447) | 0.7683 (0.0652) | 0.5043 (0.0131) | 0.5000 (0.0000) | 0.5193 (0.0480) | 0.5349 (0.0301) |
|  | 50 | 0.8674 (0.0326) | **0.8678** (0.0589) | 0.5221 (0.0439) | 0.5000 (0.0000) | 0.7239 (0.0975) | 0.5408 (0.0307) |
|  | 100 | 0.8741 (0.0266) | **0.8861** (0.0540) | 0.5241 (0.0502) | 0.5000 (0.0000) | 0.8473 (0.0628) | 0.5512 (0.0307) |
| 1.0 | 5 | **0.8152** (0.0627) | 0.6913 (0.0681) | 0.5023 (0.0116) | 0.5000 (0.0000) | 0.5046 (0.0250) | 0.5227 (0.0329) |
|  | 10 | **0.9275** (0.0357) | 0.8263 (0.0699) | 0.5273 (0.0454) | 0.5003 (0.0035) | 0.6357 (0.1133) | 0.5487 (0.0328) |
|  | 20 | **0.9638** (0.0176) | 0.8992 (0.0627) | 0.7020 (0.1127) | 0.5000 (0.0000) | 0.8426 (0.0763) | 0.5958 (0.0671) |
|  | 50 | **0.9810** (0.0072) | 0.9621 (0.0237) | 0.9577 (0.0180) | 0.5019 (0.0128) | 0.9581 (0.0150) | 0.8461 (0.0846) |
|  | 100 | 0.9834 (0.0064) | **0.9840** (0.0077) | 0.9719 (0.0095) | 0.5005 (0.0047) | 0.9685 (0.0108) | 0.8609 (0.0818) |
| 1.5 | 5 | **0.9143** (0.0450) | 0.8264 (0.0701) | 0.5255 (0.0394) | 0.5007 (0.0048) | 0.6402 (0.1339) | 0.5416 (0.0322) |
|  | 10 | **0.9775** (0.0146) | 0.9349 (0.0453) | 0.7485 (0.1152) | 0.5035 (0.0127) | 0.8885 (0.0515) | 0.5966 (0.0558) |
|  | 20 | **0.9921** (0.0056) | 0.9715 (0.0311) | 0.9636 (0.0196) | 0.5247 (0.0399) | 0.9633 (0.0177) | 0.9294 (0.0538) |
|  | 50 | **0.9954** (0.0024) | 0.9946 (0.0026) | 0.9828 (0.0076) | 0.7844 (0.1471) | 0.9827 (0.0068) | 0.9909 (0.0049) |
|  | 100 | 0.9965 (0.0014) | **0.9971** (0.0014) | 0.9868 (0.0044) | 0.9159 (0.1287) | 0.9874 (0.0041) | 0.9939 (0.0028) |
| 2.0 | 5 | **0.9561** (0.0292) | 0.9310 (0.0453) | 0.6570 (0.1084) | 0.5100 (0.0191) | 0.8615 (0.0807) | 0.5632 (0.0475) |
|  | 10 | **0.9939** (0.0040) | 0.9869 (0.0157) | 0.9599 (0.0307) | 0.5893 (0.0711) | 0.9608 (0.0252) | 0.8337 (0.0835) |
|  | 20 | **0.9972** (0.0023) | 0.9946 (0.0063) | 0.9841 (0.0088) | 0.8523 (0.0938) | 0.9824 (0.0091) | 0.9919 (0.0069) |
|  | 50 | 0.9985 (0.0011) | **0.9987** (0.0009) | 0.9912 (0.0046) | 0.9886 (0.0046) | 0.9912 (0.0040) | 0.9966 (0.0019) |
|  | 100 | 0.9989 (0.0007) | **0.9991** (0.0006) | 0.9929 (0.0027) | 0.9917 (0.0025) | 0.9943 (0.0020) | 0.9975 (0.0013) |

Table 2.4. The means and standard deviations of AUC of the copy number change detection methods on datasets where the noises conform to t-distribution with df= 4.

| *a* | *l* | **FMR** | **FLasso** | **CBS** | **GLAD** | **BioHMM** | **HaarSeg** |
|---|---|---|---|---|---|---|---|
| 0.1 | 5 | **0.5286** (0.0672) | 0.5055 (0.0298) | 0.4998 (0.0020) | 0.4998 (0.0020) | 0.5010 (0.0142) | 0.5014 (0.0118) |
| | 10 | **0.5228** (0.0390) | 0.4995 (0.0065) | 0.5001 (0.0010) | 0.5001 (0.0010) | 0.5002 (0.0027) | 0.4999 (0.0059) |
| | 20 | **0.5675** (0.0550) | 0.5474 (0.0617) | 0.5002 (0.0033) | 0.5002 (0.0036) | 0.5006 (0.0137) | 0.5042 (0.0244) |
| | 50 | **0.5855** (0.0457) | 0.5658 (0.0536) | 0.5001 (0.0005) | 0.5002 (0.0022) | 0.4996 (0.0093) | 0.5095 (0.0207) |
| | 100 | **0.5780** (0.0352) | 0.5761 (0.0499) | 0.5000 (0.0005) | 0.5000 (0.0004) | 0.5007 (0.0042) | 0.5060 (0.0163) |
| 0.5 | 5 | **0.6657** (0.0689) | 0.5331 (0.0377) | 0.5002 (0.0020) | 0.5002 (0.0020) | 0.4966 (0.0230) | 0.5077 (0.0192) |
| | 10 | **0.7784** (0.0525) | 0.5848 (0.0495) | 0.5004 (0.0032) | 0.5000 (0.0003) | 0.5030 (0.0269) | 0.5091 (0.0221) |
| | 20 | **0.8230** (0.0420) | 0.7241 (0.0663) | 0.5006 (0.0045) | 0.5001 (0.0021) | 0.5353 (0.0758) | 0.5316 (0.0349) |
| | 50 | **0.8409** (0.0281) | 0.8175 (0.0625) | 0.5015 (0.0075) | 0.5002 (0.0020) | 0.7102 (0.1383) | 0.5593 (0.0501) |
| | 100 | **0.8208** (0.0256) | 0.6663 (0.0797) | 0.5028 (0.0127) | 0.5002 (0.0022) | 0.7450 (0.1328) | 0.5390 (0.0398) |
| 1.0 | 5 | **0.8526** (0.0543) | 0.6457 (0.0618) | 0.5007 (0.0042) | 0.5007 (0.0069) | 0.5538 (0.0963) | 0.5178 (0.0342) |
| | 10 | **0.9334** (0.0238) | 0.7785 (0.0728) | 0.5035 (0.0171) | 0.5028 (0.0102) | 0.6447 (0.1425) | 0.5448 (0.0400) |
| | 20 | **0.9601** (0.0152) | 0.8401 (0.0648) | 0.5332 (0.0629) | 0.5021 (0.0089) | 0.9122 (0.0468) | 0.7296 (0.0986) |
| | 50 | **0.9618** (0.0126) | 0.8007 (0.0711) | 0.9151 (0.1095) | 0.5101 (0.0278) | 0.9542 (0.0173) | 0.9262 (0.0496) |
| | 100 | 0.9740 (0.0094) | **0.9756** (0.0091) | 0.9541 (0.0819) | 0.5247 (0.0601) | 0.9582 (0.0270) | 0.9522 (0.0252) |
| 1.5 | 5 | **0.9515** (0.0290) | 0.7217 (0.0870) | 0.5050 (0.0205) | 0.5057 (0.0136) | 0.6607 (0.1596) | 0.5356 (0.0422) |
| | 10 | **0.9829** (0.0108) | 0.9089 (0.0674) | 0.6091 (0.1074) | 0.5342 (0.0399) | 0.9200 (0.0521) | 0.7345 (0.1020) |
| | 20 | **0.9914** (0.0050) | 0.9639 (0.0326) | 0.9526 (0.0653) | 0.6557 (0.0998) | 0.9708 (0.0169) | 0.9729 (0.0197) |
| | 50 | **0.9930** (0.0031) | 0.9898 (0.0050) | 0.9851 (0.0080) | 0.9709 (0.0324) | 0.9830 (0.0095) | 0.9891 (0.0045) |
| | 100 | **0.9946** (0.0025) | 0.9944 (0.0030) | 0.9864 (0.0055) | 0.9855 (0.0046) | 0.9871 (0.0078) | 0.9912 (0.0037) |
| 2.0 | 5 | **0.9763** (0.0195) | 0.8413 (0.0830) | 0.5421 (0.0722) | 0.5611 (0.0588) | 0.8909 (0.1274) | 0.6531 (0.0855) |
| | 10 | **0.9941** (0.0052) | 0.9870 (0.0203) | 0.9206 (0.0657) | 0.7461 (0.0823) | 0.9696 (0.0214) | 0.9260 (0.0522) |
| | 20 | **0.9969** (0.0034) | 0.9961 (0.0037) | 0.9864 (0.0104) | 0.9609 (0.0287) | 0.9853 (0.0122) | 0.9903 (0.0059) |
| | 50 | 0.9981 (0.0016) | **0.9984** (0.0014) | 0.9923 (0.0046) | 0.9900 (0.0048) | 0.9899 (0.0076) | 0.9944 (0.0037) |
| | 100 | 0.9983 (0.0010) | **0.9988** (0.0010) | 0.9923 (0.0033) | 0.9923 (0.0024) | 0.9929 (0.0043) | 0.9952 (0.0024) |

Table 2.5. The means and standard deviations of AUC of the copy number change detection methods on datasets where the noises conform to log-normal distribution with $\mu = 0$ and $\sigma = 0.5$.

| *a* | *l* | **FMR** | **FLasso** | **CBS** | **GLAD** | **BioHMM** | **HaarSeg** |
|---|---|---|---|---|---|---|---|
| 0.1 | 5 | **0.5499** (0.0865) | 0.5172 (0.0711) | 0.5000 (0.0011) | 0.5007 (0.0110) | 0.5272 (0.0367) | 0.5031 (0.0303) |
| | 10 | **0.5687** (0.0768) | 0.5351 (0.0787) | 0.4999 (0.0026) | 0.5004 (0.0032) | 0.5230 (0.0276) | 0.5019 (0.0320) |
| | 20 | **0.5851** (0.0687) | 0.5569 (0.0656) | 0.5002 (0.0012) | 0.5012 (0.0051) | 0.5171 (0.0215) | 0.5085 (0.0310) |
| | 50 | 0.5951 (0.0430) | **0.6044** (0.0616) | 0.5000 (0.0002) | 0.4995 (0.0068) | 0.5189 (0.0145) | 0.5114 (0.0278) |
| | 100 | 0.5975 (0.0383) | **0.6042** (0.0529) | 0.5004 (0.0026) | 0.5005 (0.0059) | 0.5211 (0.0138) | 0.5057 (0.0269) |
| 0.5 | 5 | **0.7310** (0.0656) | 0.6222 (0.0694) | 0.5001 (0.0028) | 0.5006 (0.0068) | 0.6186 (0.0408) | 0.5105 (0.0386) |
| | 10 | **0.8128** (0.0503) | 0.6887 (0.0725) | 0.5003 (0.0020) | 0.5018 (0.0072) | 0.6095 (0.0342) | 0.5258 (0.0288) |
| | 20 | **0.8587** (0.0321) | 0.7937 (0.0491) | 0.5005 (0.0028) | 0.5032 (0.0116) | 0.6177 (0.0318) | 0.5564 (0.0495) |
| | 50 | **0.8858** (0.0243) | 0.8838 (0.0493) | 0.5020 (0.0134) | 0.5021 (0.0075) | 0.6343 (0.0399) | 0.5930 (0.0760) |
| | 100 | 0.8870 (0.0226) | **0.9020** (0.0416) | 0.5007 (0.0053) | 0.5023 (0.0069) | 0.6685 (0.0608) | 0.5988 (0.0625) |
| 1.0 | 5 | **0.8906** (0.0381) | 0.7239 (0.0702) | 0.5008 (0.0058) | 0.5031 (0.0110) | 0.7472 (0.0362) | 0.5349 (0.0381) |
| | 10 | **0.9560** (0.0166) | 0.8474 (0.0581) | 0.5015 (0.0068) | 0.5085 (0.0208) | 0.7569 (0.0412) | 0.5895 (0.0637) |
| | 20 | **0.9725** (0.0085) | 0.9296 (0.0442) | 0.5185 (0.0519) | 0.5152 (0.0326) | 0.7819 (0.0633) | 0.8007 (0.0999) |
| | 50 | **0.9819** (0.0058) | 0.9684 (0.0306) | 0.8007 (0.1771) | 0.5336 (0.0601) | 0.8221 (0.0706) | 0.9635 (0.0211) |
| | 100 | **0.9817** (0.0064) | 0.9596 (0.0394) | 0.8776 (0.1535) | 0.5845 (0.1021) | 0.8420 (0.0764) | 0.9682 (0.0189) |
| 1.5 | 5 | **0.9704** (0.0167) | 0.8488 (0.0519) | 0.5044 (0.0141) | 0.5221 (0.0303) | 0.8617 (0.0371) | 0.5791 (0.0549) |
| | 10 | **0.9911** (0.0042) | 0.9674 (0.0260) | 0.5368 (0.0556) | 0.5600 (0.0631) | 0.8916 (0.0454) | 0.8043 (0.0927) |
| | 20 | **0.9951** (0.0022) | 0.9890 (0.0123) | 0.8899 (0.1049) | 0.6855 (0.1060) | 0.9175 (0.0548) | 0.9815 (0.0146) |
| | 50 | **0.9962** (0.0017) | 0.9951 (0.0036) | 0.9867 (0.0057) | 0.9795 (0.0105) | 0.9134 (0.0683) | 0.9912 (0.0033) |
| | 100 | **0.9963** (0.0022) | 0.9959 (0.0028) | 0.9873 (0.0050) | 0.9854 (0.0059) | 0.8994 (0.0684) | 0.9921 (0.0038) |
| 2.0 | 5 | **0.9948** (0.0027) | 0.9445 (0.0451) | 0.5187 (0.0363) | 0.5883 (0.0625) | 0.9275 (0.0236) | 0.7069 (0.0978) |
| | 10 | **0.9981** (0.0010) | 0.9963 (0.0026) | 0.8509 (0.1296) | 0.7860 (0.1034) | 0.9574 (0.0217) | 0.9725 (0.0300) |
| | 20 | **0.9988** (0.0007) | 0.9983 (0.0011) | 0.9920 (0.0048) | 0.9711 (0.0232) | 0.9688 (0.0271) | 0.9920 (0.0042) |
| | 50 | **0.9989** (0.0007) | 0.9986 (0.0011) | 0.9944 (0.0027) | 0.9912 (0.0030) | 0.9741 (0.0293) | 0.9946 (0.0028) |
| | 100 | **0.9988** (0.0009) | **0.9988** (0.0012) | 0.9933 (0.0033) | 0.9929 (0.0029) | 0.9748 (0.0313) | 0.9953 (0.0032) |

Table 2.6. The means and standard deviations of AUC of the copy number change detection methods on datasets where the noises are sampled from real SNP microarray profiles.

| *a* | *l* | **FMR** | **FLasso** | **CBS** | **GLAD** | **BioHMM** | **HaarSeg** |
|---|---|---|---|---|---|---|---|
| 0.1 | 5 | 0.5355 (0.0717) | 0.5237 (0.0703) | 0.5054 (0.0486) | 0.5061 (0.0408) | **0.5385** (0.0586) | 0.5330 (0.0792) |
| | 10 | 0.5400 (0.0698) | 0.5308 (0.0679) | 0.5103 (0.0478) | 0.5080 (0.0343) | 0.5294 (0.0502) | **0.5434** (0.0737) |
| | 20 | **0.5633** (0.0757) | 0.5526 (0.0729) | 0.5172 (0.0392) | 0.4980 (0.0378) | 0.5355 (0.0386) | 0.5572 (0.0621) |
| | 50 | **0.5677** (0.0679) | 0.5659 (0.0711) | 0.5250 (0.0406) | 0.5046 (0.0332) | 0.5318 (0.0276) | 0.5567 (0.0544) |
| | 100 | 0.5593 (0.0598) | **0.5627** (0.0750) | 0.5192 (0.0326) | 0.5054 (0.0376) | 0.5286 (0.0253) | 0.5479 (0.0491) |
| 0.5 | 5 | 0.6103 (0.0836) | 0.5677 (0.0771) | 0.5327 (0.0566) | 0.5203 (0.0483) | 0.6256 (0.0630) | **0.6287** (0.0845) |
| | 10 | **0.6823** (0.0848) | 0.6288 (0.0889) | 0.5541 (0.0549) | 0.5209 (0.0413) | 0.6424 (0.0543) | 0.6729 (0.0783) |
| | 20 | **0.7359** (0.0658) | 0.6854 (0.0677) | 0.5905 (0.0550) | 0.5327 (0.0371) | 0.6430 (0.0342) | 0.7050 (0.0544) |
| | 50 | **0.7985** (0.0557) | 0.7740 (0.0615) | 0.6743 (0.0670) | 0.5497 (0.0413) | 0.6570 (0.0276) | 0.7456 (0.0453) |
| | 100 | **0.8075** (0.0466) | 0.8062 (0.0537) | 0.7092 (0.0748) | 0.5729 (0.0506) | 0.6557 (0.0288) | 0.7420 (0.0406) |
| 1.0 | 5 | 0.7091 (0.0732) | 0.6387 (0.0703) | 0.5718 (0.0601) | 0.5698 (0.0514) | **0.7490** (0.0465) | 0.7446 (0.0712) |
| | 10 | **0.8192** (0.0634) | 0.7156 (0.0700) | 0.6459 (0.0826) | 0.5748 (0.0602) | 0.7657 (0.0435) | 0.8162 (0.0507) |
| | 20 | **0.9060** (0.0339) | 0.8246 (0.0673) | 0.7938 (0.0720) | 0.6108 (0.0686) | 0.7930 (0.0327) | 0.8735 (0.0351) |
| | 50 | **0.9460** (0.0256) | 0.9125 (0.0475) | 0.9009 (0.0346) | 0.6834 (0.0723) | 0.8188 (0.0346) | 0.8896 (0.0305) |
| | 100 | **0.9546** (0.0203) | 0.9368 (0.0386) | 0.9284 (0.0231) | 0.7049 (0.0841) | 0.8166 (0.0454) | 0.8941 (0.0234) |
| 1.5 | 5 | 0.7862 (0.0648) | 0.7069 (0.0662) | 0.6470 (0.0823) | 0.6178 (0.0797) | 0.8381 (0.0445) | **0.8480** (0.0588) |
| | 10 | **0.9261** (0.0396) | 0.8479 (0.0651) | 0.8140 (0.0776) | 0.6804 (0.0904) | 0.8711 (0.0380) | 0.9254 (0.0292) |
| | 20 | **0.9746** (0.0154) | 0.9124 (0.0533) | 0.9379 (0.0318) | 0.7781 (0.0705) | 0.9024 (0.0297) | 0.9462 (0.0211) |
| | 50 | **0.9875** (0.0066) | 0.9808 (0.0160) | 0.9592 (0.0167) | 0.9069 (0.0464) | 0.9325 (0.0276) | 0.9544 (0.0142) |
| | 100 | **0.9903** (0.0047) | 0.9885 (0.0074) | 0.9689 (0.0102) | 0.9297 (0.0321) | 0.9333 (0.0328) | 0.9565 (0.0129) |
| 2.0 | 5 | 0.8240 (0.0603) | 0.7980 (0.0765) | 0.7642 (0.0962) | 0.7560 (0.0785) | 0.8996 (0.0339) | **0.9287** (0.0399) |
| | 10 | **0.9668** (0.0227) | 0.9304 (0.0424) | 0.9366 (0.0430) | 0.8516 (0.0706) | 0.9391 (0.0239) | 0.9635 (0.0200) |
| | 20 | **0.9918** (0.0066) | 0.9724 (0.0259) | 0.9693 (0.0168) | 0.9247 (0.0404) | 0.9598 (0.0174) | 0.9761 (0.0103) |
| | 50 | **0.9963** (0.0023) | 0.9955 (0.0031) | 0.9808 (0.0074) | 0.9684 (0.0122) | 0.9725 (0.0118) | 0.9798 (0.0071) |
| | 100 | 0.9965 (0.0026) | **0.9970** (0.0025) | 0.9858 (0.0058) | 0.9732 (0.0096) | 0.9758 (0.0138) | 0.9822 (0.0056) |

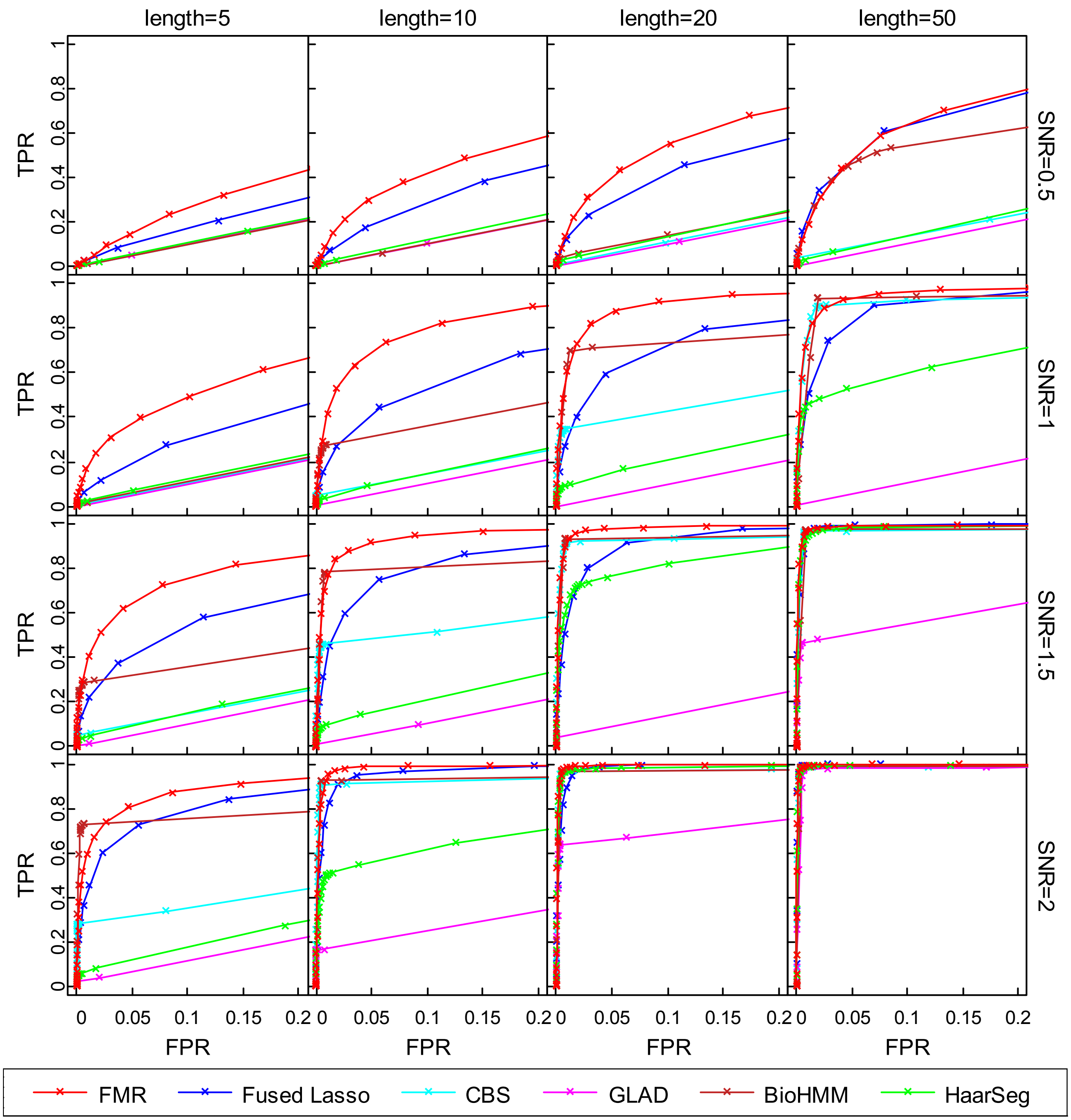


Figure 2.7. ROC curves of the detection methods on datasets where $l \in \{5, 10, 20, 50\}$ and $a \in \{0.5, 1.0, 1.5, 2.0\}$, and the noises conform to the standard normal distribution.

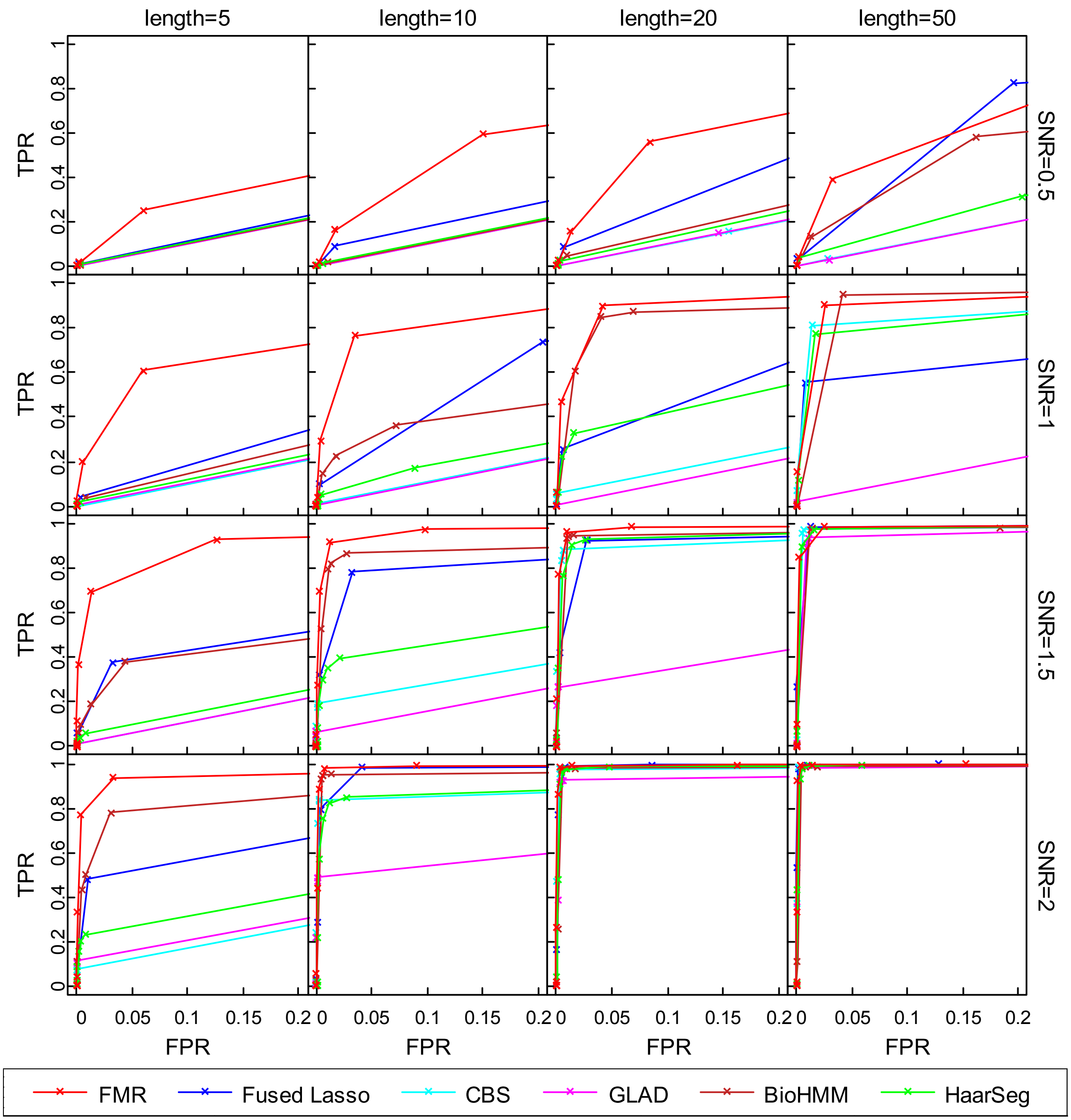


Figure 2.8. ROC curves of the detection methods on datasets where $l \in \{5, 10, 20, 50\}$ and $a \in \{0.5, 1.0, 1.5, 2.0\}$, and the noises conform to t-distribution with df=4.

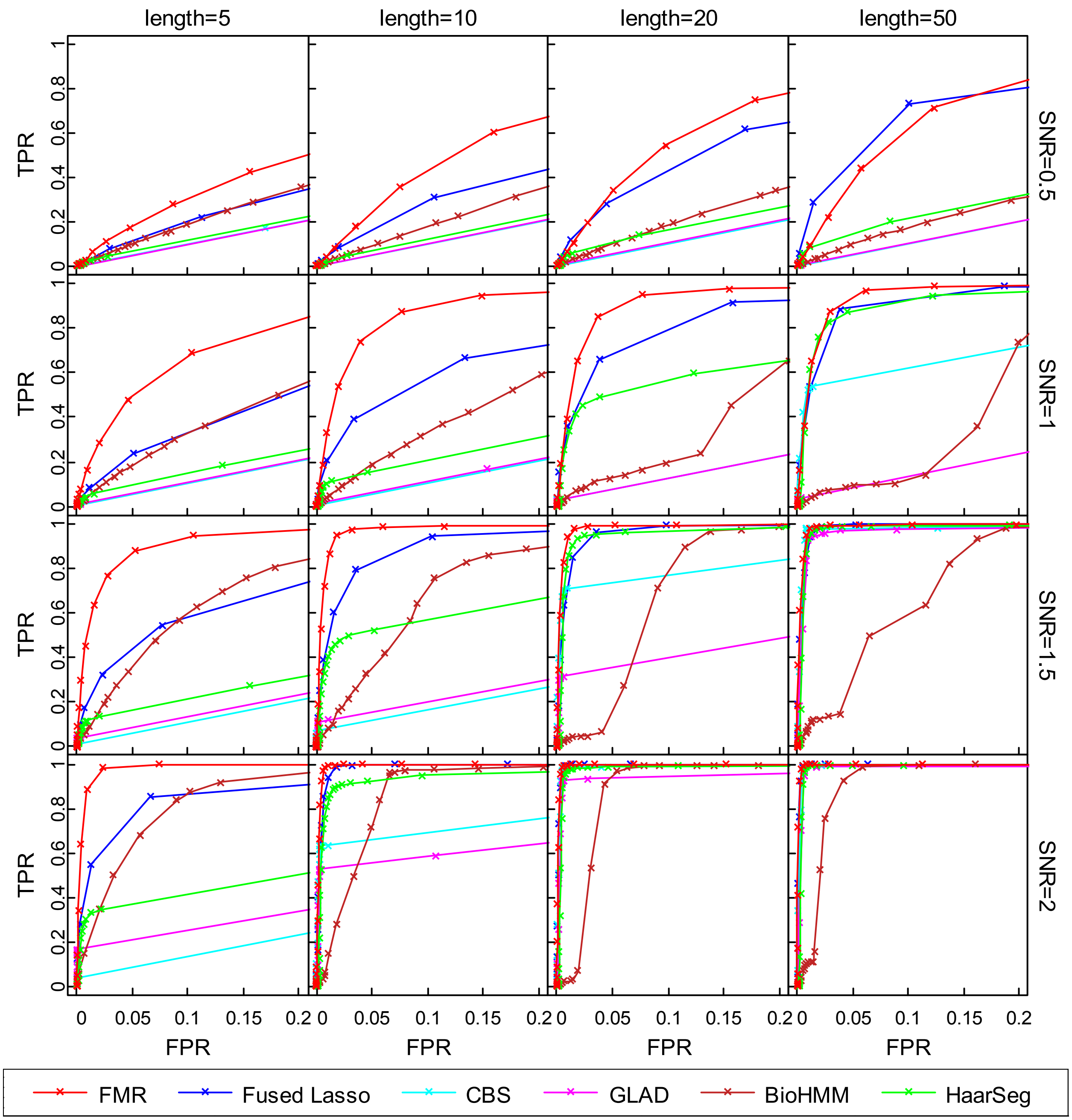


Figure 2.9. ROC curves of the detection methods on datasets where $l \in \{5, 10, 20, 50\}$ and $a \in \{0.5, 1.0, 1.5, 2.0\}$, and the noises conform to log-normal distribution with $\mu = 0$ and $\sigma = 0.5$.

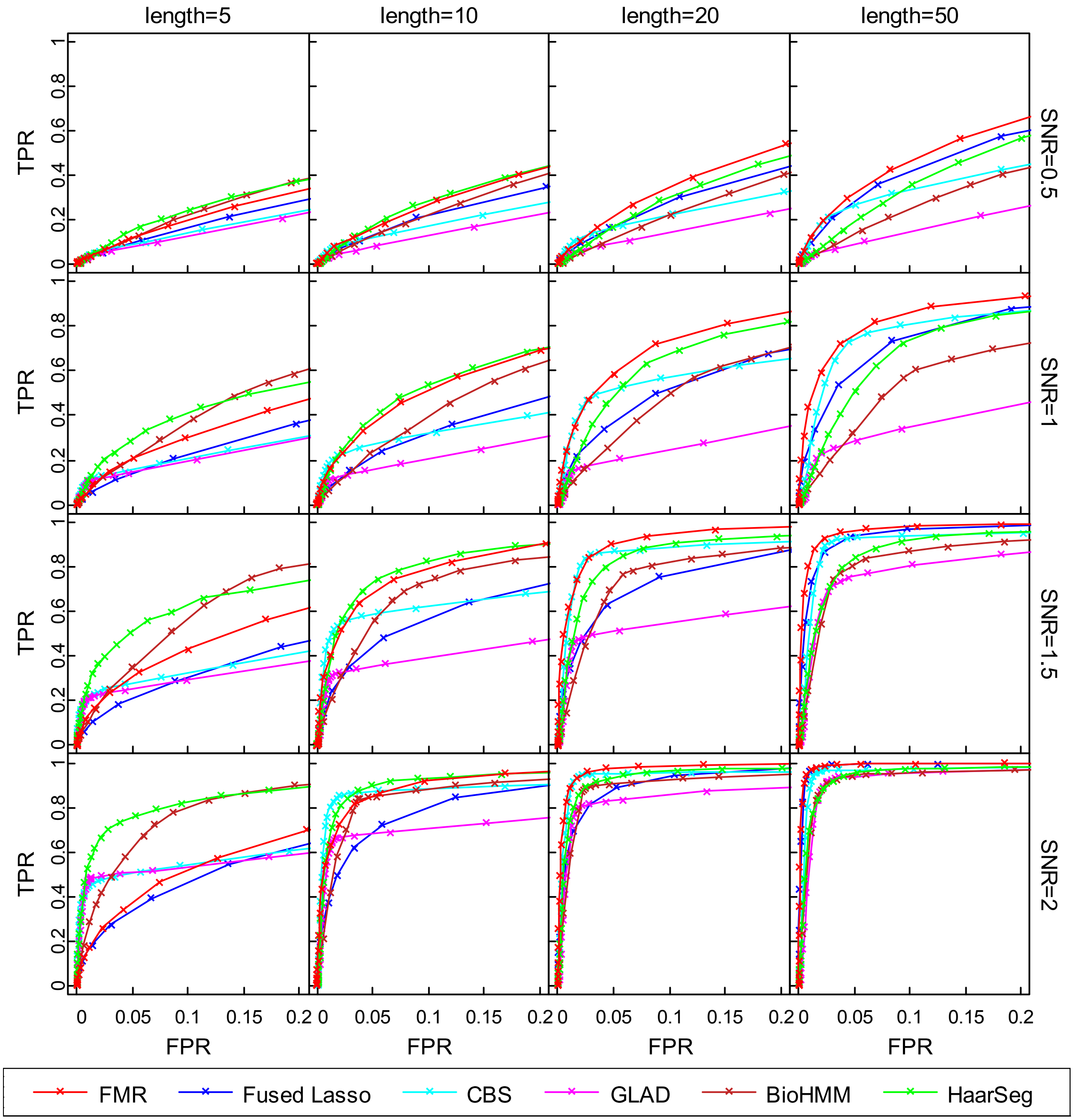


Figure 2.10. ROC curves of the detection methods on datasets where $l \in \{5,10,20,50\}$ and $a \in \{0.5,1.0,1.5,2.0\}$, and the noises are sampled from real SNP microarray profiles.

*2.10.2.2 Simulated data with multiple copy number states*

In the second simulation study, we compare FMR with the competing methods for detecting copy number changes in signal profiles with multiple copy number states. We create a simulated signal profile using Hidden Markov Model where the hidden states are generated by a stationary Markov chain and the emission probability is a selected noise distribution. Denote the true copy numbers by $\boldsymbol{\eta} = \{\eta_i\}_{i=1}^n$, $n = 2000$, where $\eta_i \in \{0.5, 1, 2, 3, 4, 5\}$. Here we use 0.5 to represent the state of homozygous deletions in order to have valid log2-ratio values. The state transition probabilities are governed by a designated conditional distribution

$$P(S_i = \eta_i \mid S_{i-1} = \eta_{i-1}) = \begin{cases} 0.975, & \eta_{i-1} = \eta_i = 2 \\ 0.95, & \eta_{i-1} = \eta_i \neq 2 \\ 0.03, & \eta_{i-1} \neq \eta_i = 2 \\ 0.005, & \text{otherwise} \end{cases}, \quad i = 2, \ldots, n,$$

where $S_i$ is the random variable of copy number state at the $i$th locus. The probabilities of the initial states are defined as $P(S_1 = 2) = 0.975$ and $P(S_1 = \eta) = 0.005$, $\forall \eta \neq 2$.

After the true copy numbers are determined, we generate i.i.d. noises $\boldsymbol{\varepsilon} = \{\varepsilon_i\}_{i=1}^n$ using noise distribution $p_{\text{noise}}$, which can be a standard normal distribution, t-distribution with $\text{df} = 4$, log-normal distribution with $\mu = 0$ and $\sigma = 0.5$, or an empirical distribution of noises in real SNP microarray data. The noises are then transformed by $\varepsilon_i' = [(\varepsilon_i - \upsilon)/\sigma] \cdot \sigma'$, where $\upsilon$ and $\sigma$ are the mode and SD of the noise distribution, and $\sigma' \in \{0.1, 0.2, 0.5, 1\}$ is the designated noise SD. The log2-ratio copy number signals of a profile are then generated by $y_i = \log_2(\eta_i / 2) + \varepsilon_i'$, $i = 1, \ldots, n$. The combinations of noise distributions and different values of $\sigma'$ yield 16 datasets. For each dataset, we create 100 signal profiles according to the described scheme.

We apply a detection method on each of the signal profiles in a dataset to generate estimated copy numbers $\hat{\boldsymbol{\eta}} = \{\hat{\eta}_i\}_{i=1}^n$. For FMR, we use $\epsilon = 1.5\hat{\sigma}$ in the $\epsilon$-insensitive loss function, where $\hat{\sigma}$ is the estimated noise SD. We select the solution with the smallest SIC (Li and Zhu, 2007) in the solution path as the optimal solution

$$\mathrm{SIC}(\hat{\boldsymbol{\eta}} \mid \mathbf{y}) = \log\left(\frac{1}{n}\sum_{i=1}^{n} L_\epsilon(y_i, \hat{\eta}_i)\right) + K\frac{\log n}{2n},$$

where $K$ is the number of segments delimited by non-zero breakpoints in the estimated copy numbers $\hat{\boldsymbol{\eta}}$. For all the other methods, we use the default parameters provided in the corresponding software packages.

For each detection method, we re-estimate the copy number of each detected segment using its mean signal intensity. Specifically, given a detected segment in $\hat{\boldsymbol{\eta}}$ delimited by loci $i_1$ and $i_2$, the estimated copy number $\hat{\eta}_i$, $i_1 \le i \le i_2$, is replaced by

$$\hat{\eta}_i' = \frac{1}{i_2 - i_1 + 1}\sum_{k=i_1}^{i_2}\hat{\eta}_k, \quad \forall i,\ i_1 \le i \le i_2.$$

We use the sum of squared errors SSE=$\sum_{i=1}^{n}(\hat{\eta}_i' - \eta_i)^2$ as the performance measure and inspect the empirical distributions of SSE of the 100 signal profiles in each dataset (Figure 2.11 to Figure 2.14). The four sub-figures in each figure show the distributions of SSE for $\sigma'$ equal to 0.1, 0.2, 0.5, and 1, respectively. In Figure 2.11, BioHMM performs the best for all different values of $\sigma'$ due to the fact that the signal profiles in these datasets are generated using HMM with normal distributions as the emission probabilities. For all the other noise distributions, BioHMM only performs well for the cases of large SNRs ($\sigma' = 0.1$). Although it has been shown in (Lai, et al., 2005) and (Willenbrock and Fridlyand, 2005) that CBS has superior performance compared with many existing copy number change detection methods, the experimental results here show that CBS only has good performance when the SNRs are small. FMR, fused Lasso, and HaarSeg have similar performances across different datasets; but for the cases of real SNP array noises, FMR performs better than the other two methods, which suggests that FMR is more suitable for real applications. Furthermore, although the performance of FMR is not always the best, it is fairly consistent across different datasets with the sole user-determined parameter, i.e. the coefficient $a$ in $\epsilon = a\hat{\sigma}$ . This suggests that FMR is a robust copy number change detection method and its performance has less dependency on noise distributions and signal-to-noise ratios.

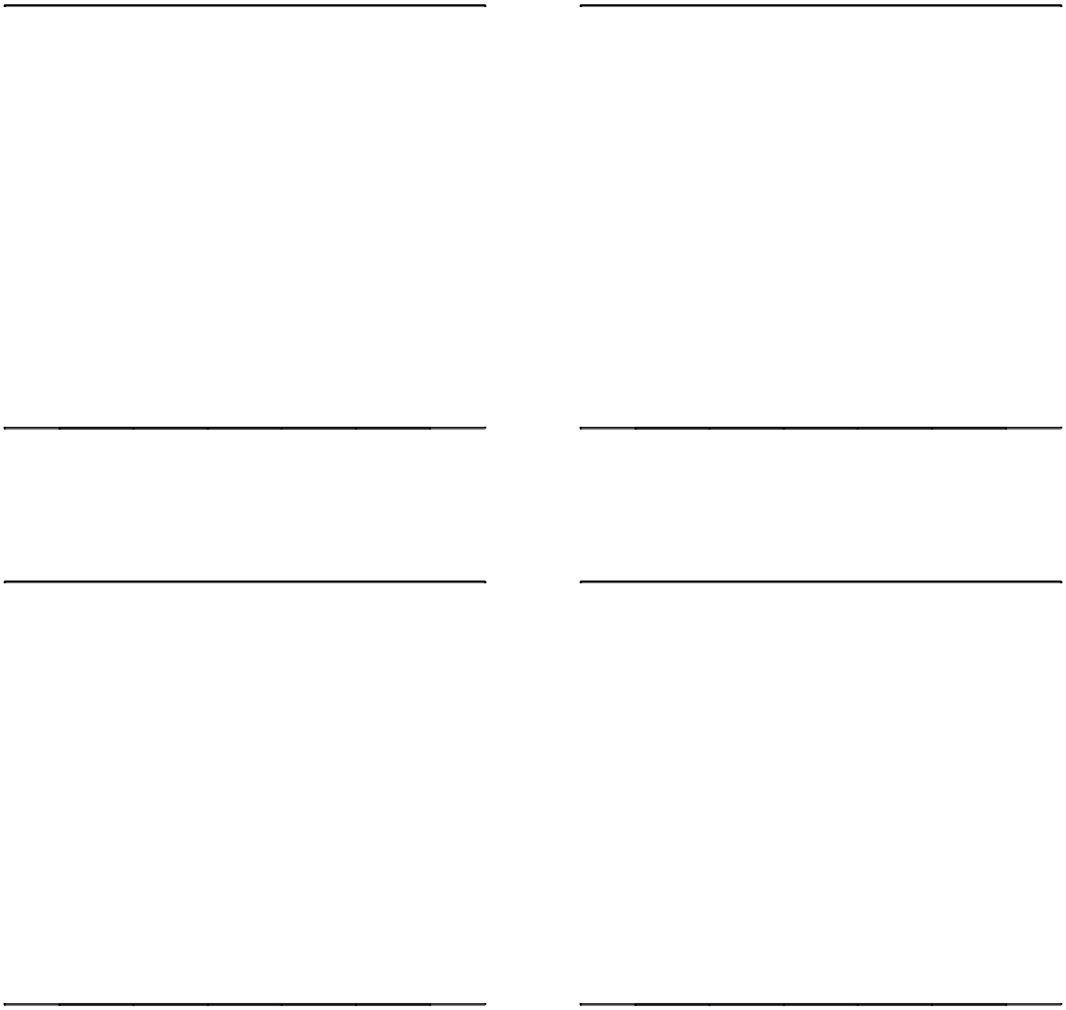


Figure 2.11. Distributions of SSE of copy number change detection methods on the simulated datasets where noises conform to normal distributions.

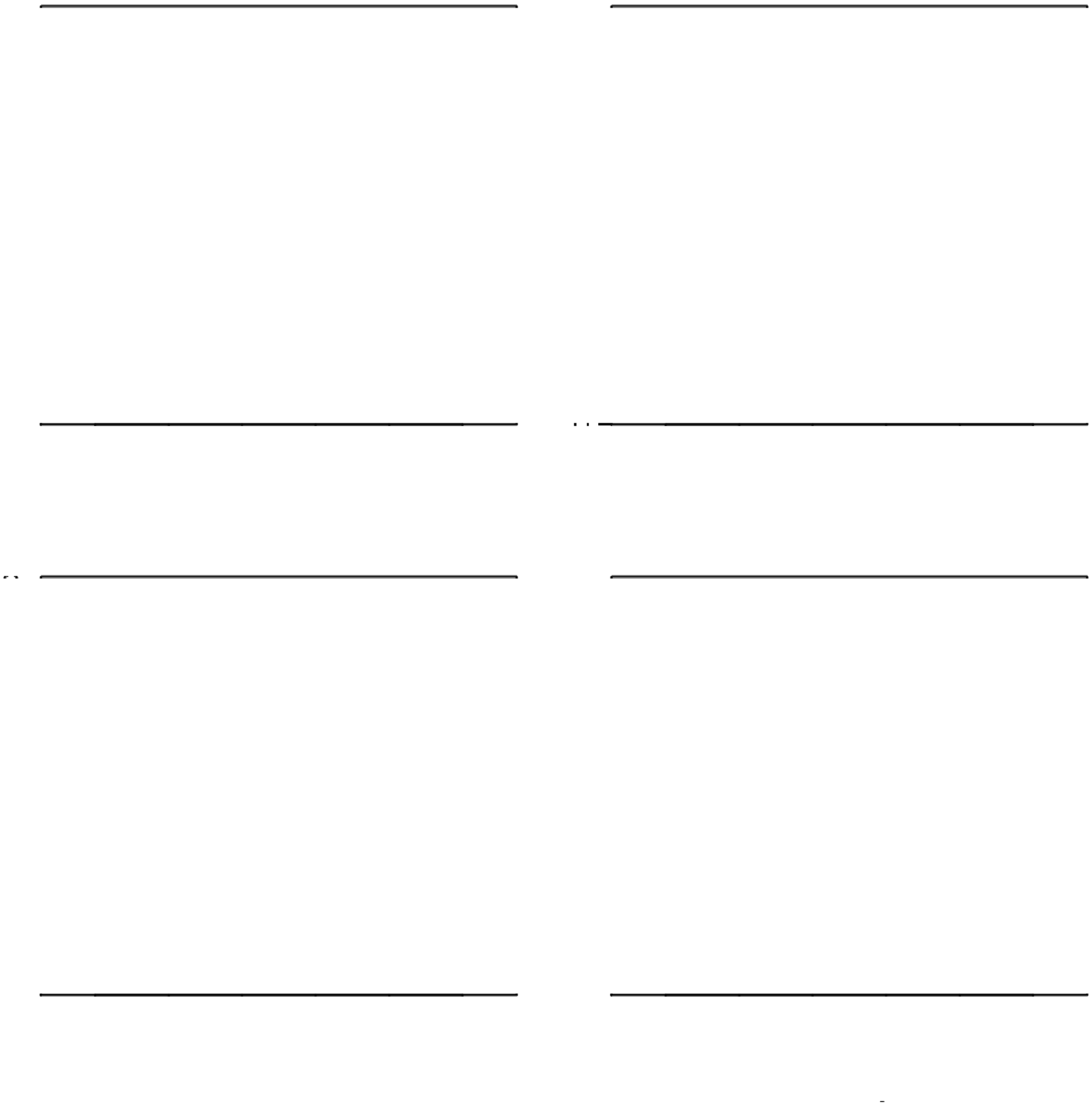


Figure 2.12. Distributions of SSE of copy number change detection methods on the simulated datasets where noises conform to t-distributions with df=4.

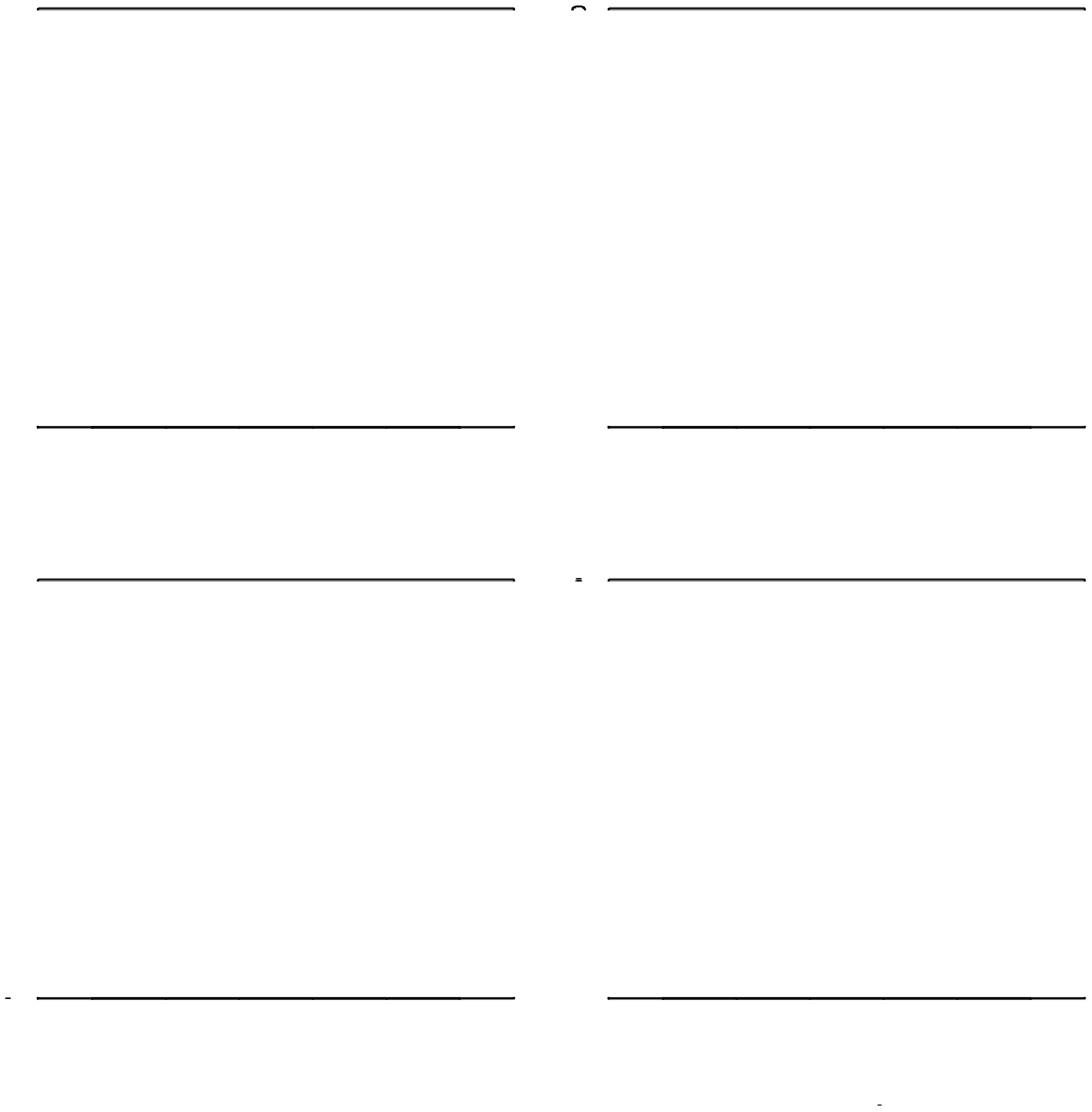


Figure 2.13. Distributions of SSE of copy number change detection methods on the simulated datasets where noises conform to log-normal distribution with $\mu = 0$ and $\sigma = 0.5$.

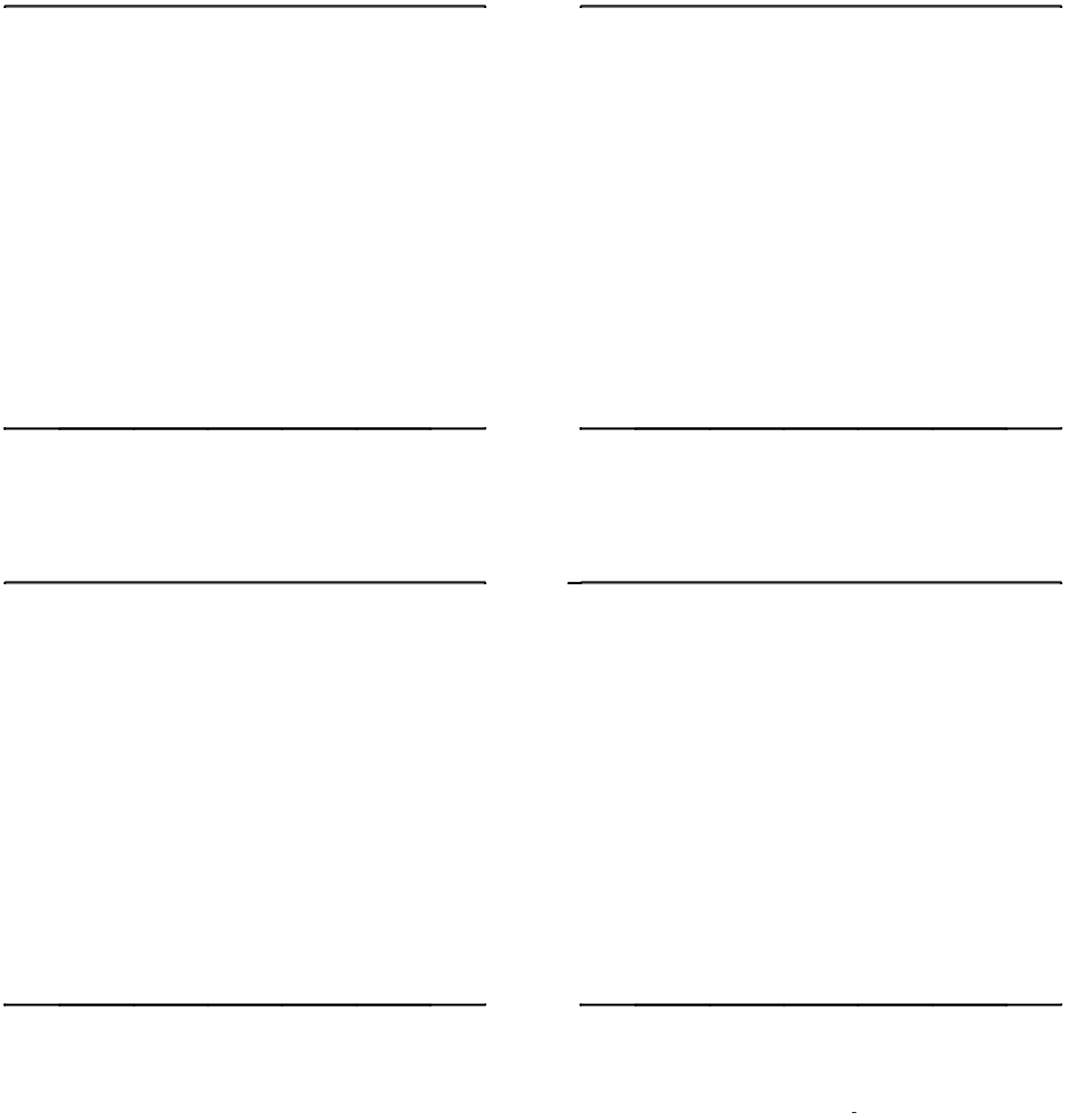


Figure 2.14. Distributions of SSE of copy number change detection methods on the simulated datasets where noises are sampled from real SNP microarray data.

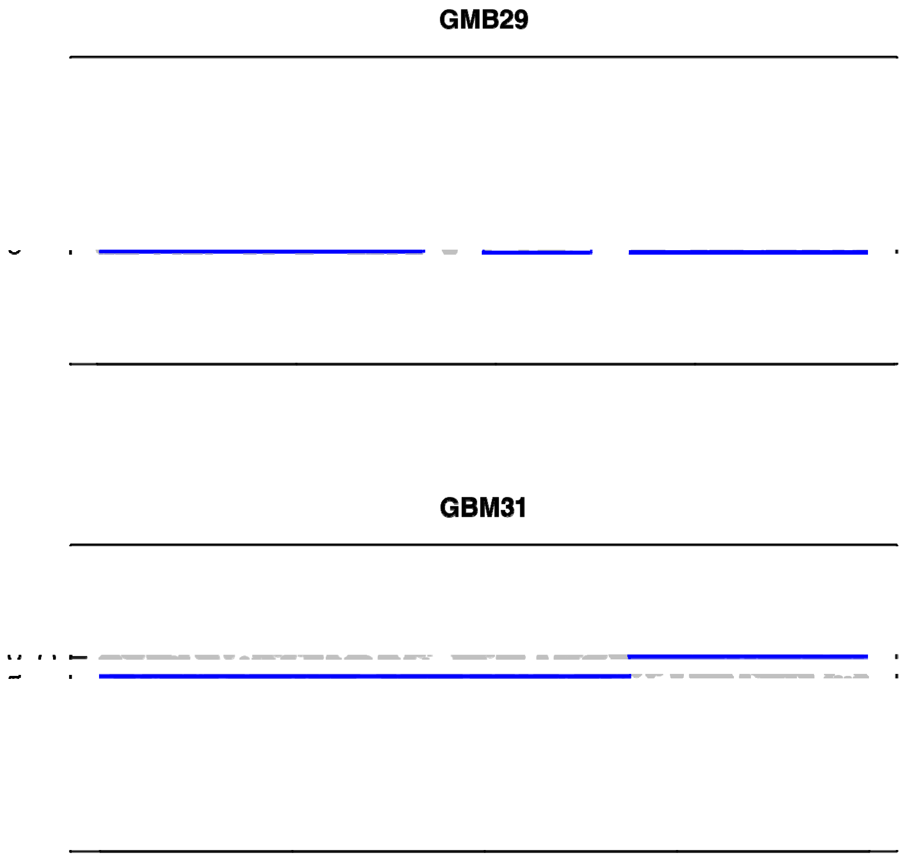


Figure 2.15. Detection results of FMR on chromosome 7 of sample GBM29 and chromosome 13 of sample GBM31 in the CGH copy number dataset of glioblastoma multiforme tumors.

### *2.10.2.3 Examples of CGH data*

We visually demonstrate FMR on two CGH datasets that are frequently used in literatures for testing copy number change detection methods. For both datasets, we use $\epsilon = 1.5\hat{\sigma}$ in the loss function and select the optimal solution in the solution path using SIC. The estimated copy numbers are further smoothed by joining adjacent segments whose estimated copy numbers are close (the difference is smaller than $0.1\epsilon$).

The first CGH dataset consists of 26 samples of primary glioblastoma multiforme (GBM) tumors (Bredel, et al., 2005). We test FMR on the two profiles used for comparison in (Lai, et al.,

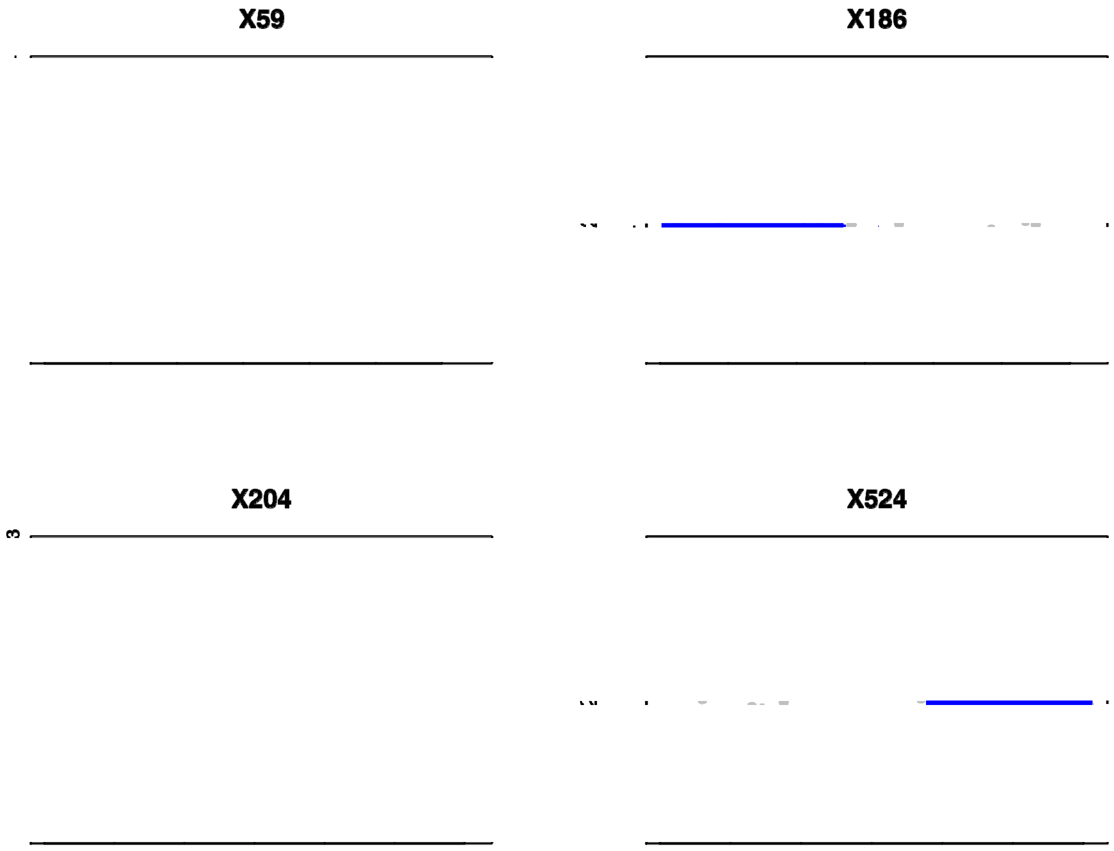


Figure 2.16. Detection results of FMR on the first chromosomes of samples X59, X186, X204, and X524 in the CGH copy number dataset of colorectal cancers.

2005), which are chromosome 13 of sample GBM31 and chromosome 7 of sample GBM29 (Figure 2.15). For sample GBM29, FMR has the same result as CGHSeg (Fig. 4 in (Lai, et al., 2005)), which is the only method tested in (Lai, et al., 2005) that identifies the three true amplifications without generating false detections. In comparison, quantile regression generates an excessive number of false positive breakpoints and CBS misses two true breakpoints between the first and second amplifications. For sample GBM31, FMR successfully detects the breakpoint between the deleted and the unaltered regions as CBS does, which is the only method in (Lai, et al., 2005) that unambiguously detects the true breakpoint. Once again, quantile regression generates an excessive number of false positives.

The other CGH dataset consists of 125 primary colorectal cancers (Nakao, et al., 2004). We test FMR on the first chromosomes of samples X59, X186, X204, and X524 (Figure 2.16), which are used in (Eilers and de Menezes, 2005; Li and Zhu, 2007) to demonstrate quantile regression. We removed loci with missing values in the original data. In general, FMR produces fewer breakpoints compared with quantile regression while still captures the trend of copy number changes as quantile regression does. We should note that both FMR and the clone distance penalized quantile regression in (Li and Zhu, 2007) pick up the amplification around 75Mb in the first chromosome of X524 (between loci 35 and 40 of sample X524 in Figure 2.16), which is missed by the original quantile regression method in (Eilers and de Menezes, 2005).

### *2.10.2.4 Examples of SNP microarray data*

We test FMR on a SNP microarray copy number dataset of ovarian cancers provided by Johns Hopkins Medical Institutions (Kuo, et al., 2009). The dataset was obtained from the SNP array analysis on a total of 13 normal samples and 37 affinity-purified serous carcinomas. The serous carcinoma samples include 12 serous borderline tumors (SBT), 12 low-grade (LG) and 13 high-grade (HG) serous carcinomas. The SNPs were genotyped using the 250K StyI arrays (Affymetrix, Santa Clara, CA) in the Microarray Core Facility at the Dana-Farber Cancer Institute, Boston, MA (http://chip.dfci.harvard.edu). Data was normalized and the raw copy number signals were extracted using dChip software (Zhao, et al., 2004). Mapping information of SNP locations and cytogenetic bands were based on curation of Affymetrix and University of California Santa Cruz hg17.

We select three samples in the dataset and test FMR on the down-sampled signal profiles of the first chromosome consisting of 3000 loci each. We use $\epsilon = 1.5\hat{\sigma}$ in the $\epsilon$-insensitive loss function and select optimal solutions using SIC. We further smooth the detection results of FMR by joining adjacent segments with similar estimated copy numbers (difference smaller than $0.1\epsilon$) and removing staircase segments shorter than 5 SNP loci. In Figure 2.17, the first example (top)

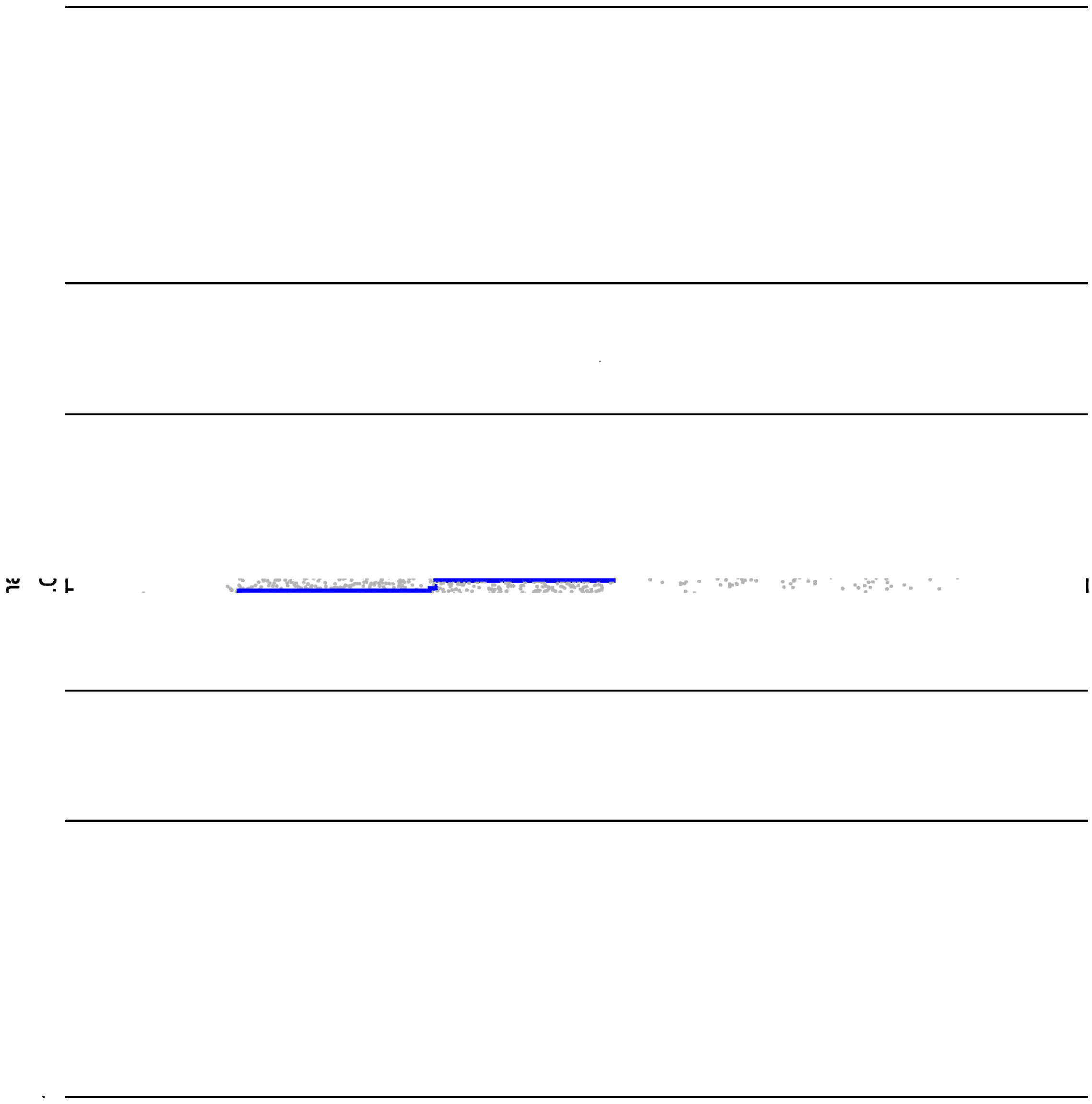


Figure 2.17. Detection results of FMR on three signal profiles in the SNP microarray copy number dataset of ovarian cancers.

contains both local and chromosomal scale alterations; the second example (middle) contains large scale deletions and two levels of large scale amplifications; the last example (bottom) contains a large number of short amplifications and deletions. These examples show that FMR is suitable for detecting copy number changes in various configurations of local and global scale alterations.

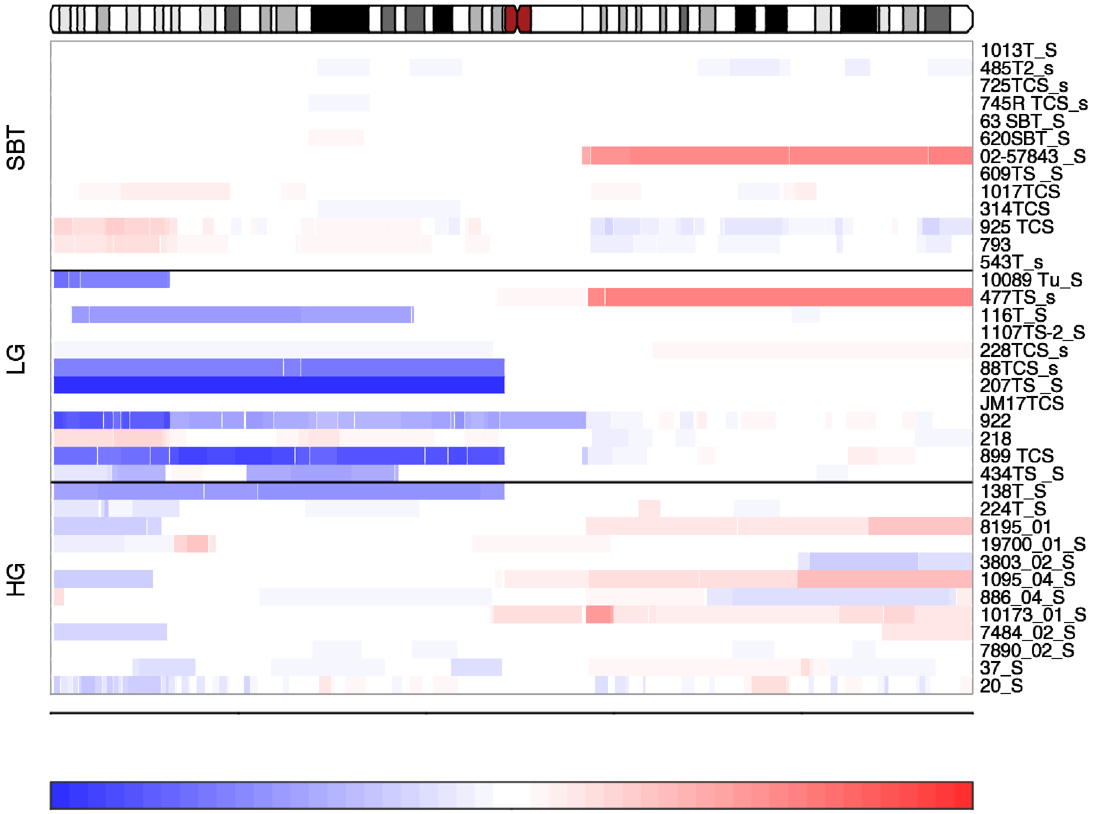


Figure 2.18. Detection results of FMR on the first chromosomes of SBT, LG, and HG samples in the SNP microarray copy number dataset of ovarian cancers.

We also visualize the detection results of FMR on chromosome 1 and 7 of these serous carcinoma samples in Figure 2.18 and Figure 2.19, respectively. The copy number signals are first converted to log2-ratios and then the signal profiles are linearly transformed such that the major components estimated by SFNM have zero means and unit SDs. Figure 2.18 shows that LG samples usually have whole-arm deletions on the first chromosome. Both Figure 2.18 and Figure 2.19 show that HG samples usually have more frequent alterations compared with HG and SBT samples on the two chromosomes. By visualizing the detection results, we can have a global view of the patterns of copy number alterations in the three subtypes of ovarian cancers.

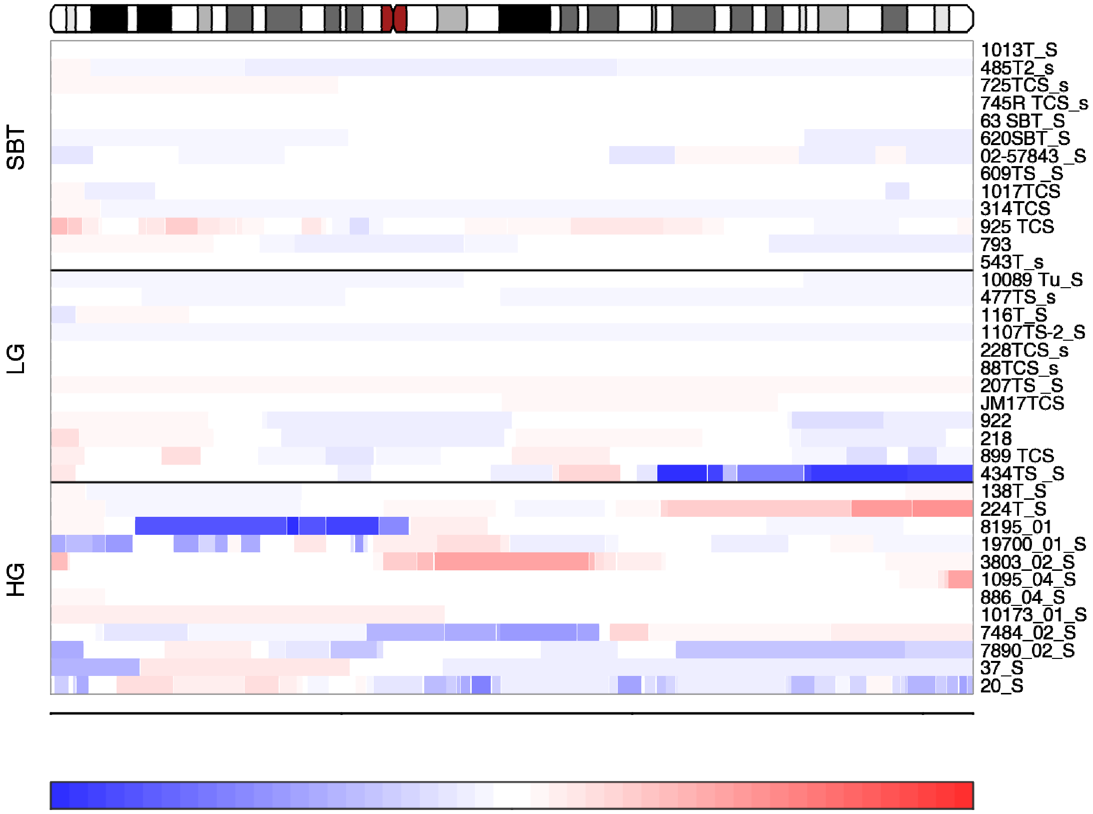


Figure 2.19. Detection results of FMR on chromosome 7 of SBT, LG, and HG samples in the SNP microarray copy number dataset of ovarian cancers.

### 2.10.3 Detecting Consensus Copy Number Changes in Population Data

DNA copy number variations (CNVs) in normal human genomes are associated with phenotypic diversities and disease susceptibilities. DNA copy number alterations (CNAs) in tumor genomes are considered hallmarks of tumorigenesis. CNAs that frequently occur in tumor genomes may harbor important tumor suppressors or amplicons that affect the progression of tumors. Detecting consensus copy number changes in CGH and SNP arrays can help to reveal these important genomic structural variations. In this section, we test MPFMR on simulated data and a real CGH dataset of 270 samples in the international HapMap DNA collection (Redon, et al., 2006).

Figure 2.20. A simulation dataset ( $p = 0.4$ , $I = [0.4, 0.6]$ ) and the consensus copy number altered regions.

*2.10.3.1 Experiments on simulation data*

We compare MPFMR with the Bayesian Segmentation Approach (BSA) (Wu, et al., 2009) and Correlation Matrix Diagonal Segmentation (CMDS) (Zhang, et al., 2009) using a simulation scheme proposed in (Wu, et al., 2009). Each simulated dataset consists of 100 profiles, each with 200 loci. There are three consensus copy number altered regions $S_1$ , $S_2$ , and $S_3$ with copy numbers 5, 4, and 3 located between loci 36 to 40, 91 to 100, and 141 to 160, respectively. All of the loci outside these regions have copy number 2. Different from the simulation in (Wu, et al., 2009) where all the 100 profiles contain the three regions, we insert $S_1$ , $S_2$ , and $S_3$ only in a portion $p$ , $0 \le p \le 1$ , of the 100 profiles in order to simulate consensus copy number changes with varying frequencies in the population. Each copy number profile $\boldsymbol{\eta}$ is then mixed with a diploid normal profile by $a \cdot \boldsymbol{\eta} + (1 - a) \cdot 2$ to simulate the effect of normal tissue contamination, where $a$ is generated from a uniform distribution with support $[0.3, 0.7]$. Finally, we transform the copy numbers by $\log_2(\eta / 2)$ and add i.i.d. noises generated from a normal distribution

Table 2.7. The numbers of missed true breakpoints and false positives of BSA, CMDS, and MPFMR on the simulated datasets.

| $p$ | $I$ | **BSA** | **CMDS** ($w$=5) | **CMDS** ($w$=10) | **CMDS** ($w$=20) | **CMDS** ($w$=30) | **MPFMR** |
|---|---|---|---|---|---|---|---|
| 1 | [0.1,0.2] | 0/0 | 2/0 | 5/1 | 6/0 | 6/0 | 0/0 |
|  | [0.2,0.4] | 0/0 | 3/1 | 5/1 | 6/0 | 6/0 | 0/0 |
|  | [0.4,0.6] | 0/6 | 6/0 | 6/0 | 6/0 | 6/0 | 0/0 |
| 0.8 | [0.1,0.2] | 0/0 | 2/0 | 4/0 | 6/2 | 6/0 | 0/0 |
|  | [0.2,0.4] | 0/0 | 2/0 | 4/0 | 6/0 | 6/0 | 0/0 |
|  | [0.4,0.6] | 0/2 | 2/0 | 4/0 | 4/2 | 6/0 | 0/0 |
| 0.6 | [0.1,0.2] | 0/0 | 2/0 | 4/0 | 5/1 | 6/0 | 0/0 |
|  | [0.2,0.4] | 0/2 | 2/0 | 5/1 | 6/0 | 6/0 | 0/0 |
|  | [0.4,0.6] | 0/6 | 2/0 | 4/0 | 6/0 | 6/0 | 0/0 |
| 0.4 | [0.1,0.2] | 0/10 | 4/0 | 4/0 | 4/0 | 6/0 | 0/0 |
|  | [0.2,0.4] | 1/7 | 2/0 | 5/1 | 6/0 | 6/0 | 0/0 |
|  | [0.4,0.6] | 0/2 | 2/0 | 4/0 | 6/0 | 6/0 | 0/0 |
| 0.2 | [0.1,0.2] | 3/4 | 2/0 | 4/0 | 5/1 | 6/0 | 2/0 |
|  | [0.2,0.4] | 2/2 | 2/0 | 4/0 | 5/1 | 6/0 | 0/0 |
|  | [0.4,0.6] | 3/5 | 2/0 | 4/0 | 5/1 | 6/0 | 1/0 |

$\mathcal{N}(\mu=0,\sigma^2)$ to the transformed copy numbers. The standard deviation $\sigma$ of noises is generated from a uniform distribution with support $I$. We create 15 datasets corresponding to all the combinations of proportions $p \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$ and supports $I = [0.1, 0.2]$, $I = [0.2, 0.4]$, and $I = [0.4, 0.6]$. As an example, the simulated dataset of $p = 0.4$ and $I = [0.4, 0.6]$ is plotted in Figure 2.20. The blue bands indicate the positions of the consensus copy number altered regions.

To measure the performance of a detection method on the simulated data, we count the number of missed true breakpoints and the number of falsely detected breakpoints (false positives). For a true breakpoint at locus $i$, we consider a detected breakpoint in the window $[i-2, i+2]$ as a true positive. If multiple breakpoints are detected in the window, only one is counted as true positive and all the others are counted as false positives. The performances of MPFMR, BSA, and CMDS on the 15 datasets are listed in Table 2.7. In each cell of the table, the numbers of misses and false positives are separated by a slash. For MPFMR, we use $\epsilon = 1.5\hat{\sigma}$ for each dataset, where $\hat{\sigma}$ is the average of the noise standard deviations estimated on individual profiles. For CMDS, we test the performance for window size $w$ equal to 5, 10, 20, and 30. The results show that except for the two cases where $p = 0.2$, MPFMR can detect all the breakpoints of the consensus regions, and it performs better than BSA and CMDS on all the datasets. BSA has reasonable performance when the consensus copy number changes occur with high frequencies, but the performance drops quickly when the frequencies become smaller, for which BSA tends to generate more false positives compared with MPFMR. The performance of CMDS highly depends on the window size. When the window size is larger than the length of copy number segments, the method cannot detect any consensus regions. Unfortunately, the authors (Zhang, et al., 2009) did not provide any method to automatically determine the window size. The experimental results show that MPFMR is a competitive consensus copy number change detection method compared with the recently published BSA and CMDS methods.

### *2.10.3.2 HapMap WGTP CGH dataset*

Redon et al (2006) proposed the first generation copy number variation map of human genomes on the 270 HapMap samples, which consist of four populations: 45 Han Chinese from Beijing, China (CHB), 45 Japanese from Tokyo, Japan (JPT), 90 Yoruba from Nigeria (YRI), and 90 European descents from Utah, USA (CEU). The copy number data is acquired using Affymetrix GeneChip Human Mapping 500K early access arrays and CGH Whole Genome TilePath (WGTP) arrays. The WGTP data consists of 26574 large-insert clones covering 93.7% of the euchromatic portion of  human genomes (Redon, et al., 2006). Here we apply MPFMR to detect consensus copy number altered regions on the WGTP dataset and measure the overlap between the detected regions with the 913 CNV regions (CNVR) discovered using the WGTP platform in (Redon, et al., 2006).

Table 2.8. The numbers of CNVRs detected by MPFMR and Redon et al's method and their overlaps on each chromosome.

| | **# CNVRs (MPFMR)** | **Overlap with Redon** | **# CNVRs (Redon)** | **Overlap with MPFMR** |
|---|---|---|---|---|
| Chr 1 | 96 | 39.58% | 67 | 44.78% |
| Chr 2 | 85 | 37.65% | 71 | 33.80% |
| Chr 3 | 75 | 34.67% | 63 | 44.44% |
| Chr 4 | 81 | 39.51% | 61 | 42.62% |
| Chr 5 | 85 | 35.29% | 57 | 49.12% |
| Chr 6 | 77 | 42.86% | 66 | 43.94% |
| Chr 7 | 59 | 49.15% | 57 | 40.35% |
| Chr 8 | 44 | 38.64% | 37 | 40.54% |
| Chr 9 | 38 | 52.63% | 35 | 25.71% |
| Chr 10 | 55 | 45.45% | 43 | 48.84% |
| Chr 11 | 55 | 32.73% | 40 | 45.00% |
| Chr 12 | 64 | 31.25% | 33 | 45.45% |
| Chr 13 | 32 | 37.50% | 25 | 48.00% |
| Chr 14 | 22 | 36.36% | 22 | 31.82% |
| Chr 15 | 30 | 73.33% | 35 | 45.71% |
| Chr 16 | 32 | 56.25% | 34 | 35.29% |
| Chr 17 | 44 | 61.36% | 29 | 68.97% |
| Chr 18 | 27 | 25.93% | 16 | 37.50% |
| Chr 19 | 16 | 81.25% | 24 | 45.83% |
| Chr 20 | 27 | 44.44% | 20 | 55.00% |
| Chr 21 | 6 | 33.33% | 6 | 33.33% |
| Chr 22 | 22 | 86.36% | 26 | 65.38% |
| Chr X | 60 | 30.00% | 38 | 42.11% |
| Chr Y | 11 | 81.82% | 8 | 75.00% |

We first remove missing values in the WGTP dataset by the following steps. For each of the 24 chromosomes, we remove a signal profile if more than 5% of the clones have no measurements. For each of the clones with missing values in a retained profile, we assign the average intensity of the clones in its neighborhood as its copy number, where the size of the neighborhood is 10. We then add a noise value generated from a normal distribution $\mathcal{N}(0, \hat{\sigma}^2)$ to the copy number. The variance $\hat{\sigma}^2$ is estimated from the clones without missing values in the profile.

We apply MPFMR on the filtered WGTP dataset to detect consensus CNV regions. We select thresholds to make gain/loss calls in the following steps. We first apply FMR on each individual signal profile $\mathbf{y}_i = \{y_{ij}\}_{j=1}^{n}$, $i = 1, \ldots, m$, to get estimated copy numbers $\hat{\boldsymbol{\eta}}_i$, and then extract $\hat{\boldsymbol{\eta}}_i$ from $\mathbf{y}_i$. Ideally, $\mathbf{y}_i - \hat{\boldsymbol{\eta}}_i$ will only contain noises. We then apply MPFMR on the noise profiles $\{\mathbf{y}_i - \hat{\boldsymbol{\eta}}_i\}_{i=1}^{m}$ to detect segments of consensus copy number changes. We select threshold $t$ such that 5% of the $n$ loci are within the segments whose estimated amplitudes are greater than $t$ or smaller than $-t$. Finally, we apply MPFMR on the original profiles $\{\mathbf{y}_i\}_{i=1}^{m}$ to detect consensus segments and make gain/loss calls with threshold $t$ and $-t$. In this way, we can control the false discovery rate in the selected consensus segments to be less than 0.05.

We found 1472 consensus CNV segments, which are further merged into 1143 CNVRs by concatenating consecutive segments of the same gain/loss types. The numbers of CNVRs found on each chromosome and their overlaps with Redon et al's CNVRs are listed in Table 2.8: the second and fourth columns show the numbers CNVRs detected by MPFMR and Redon et al's method on each chromosome, respectively; the third and fifth columns show the percentage of the CNVRs detected by MPFMR or Redon et al's method that have at least one overlapping CNVR detected by the other method, respectively. We can see that there are considerable overlaps between the CNVRs detected by MPFMR and Redon et al's results. We should note that a CNVR detected by MPFMR contains CNVs of either duplications or deletions; while in Redon et al's results, a CNVR can contain both types.

We use chromosome 17 as an example to illustrate and compare the CNVRs detected by MPFMR and those in (Redon, et al., 2006). The first and second halves of chromosome 17 are plotted in Figure 2.21 (a) and (b). We also apply BSA to detect CNVRs on the chromosome, which only generates 5 regions of duplications and one region of deletions. We should note that

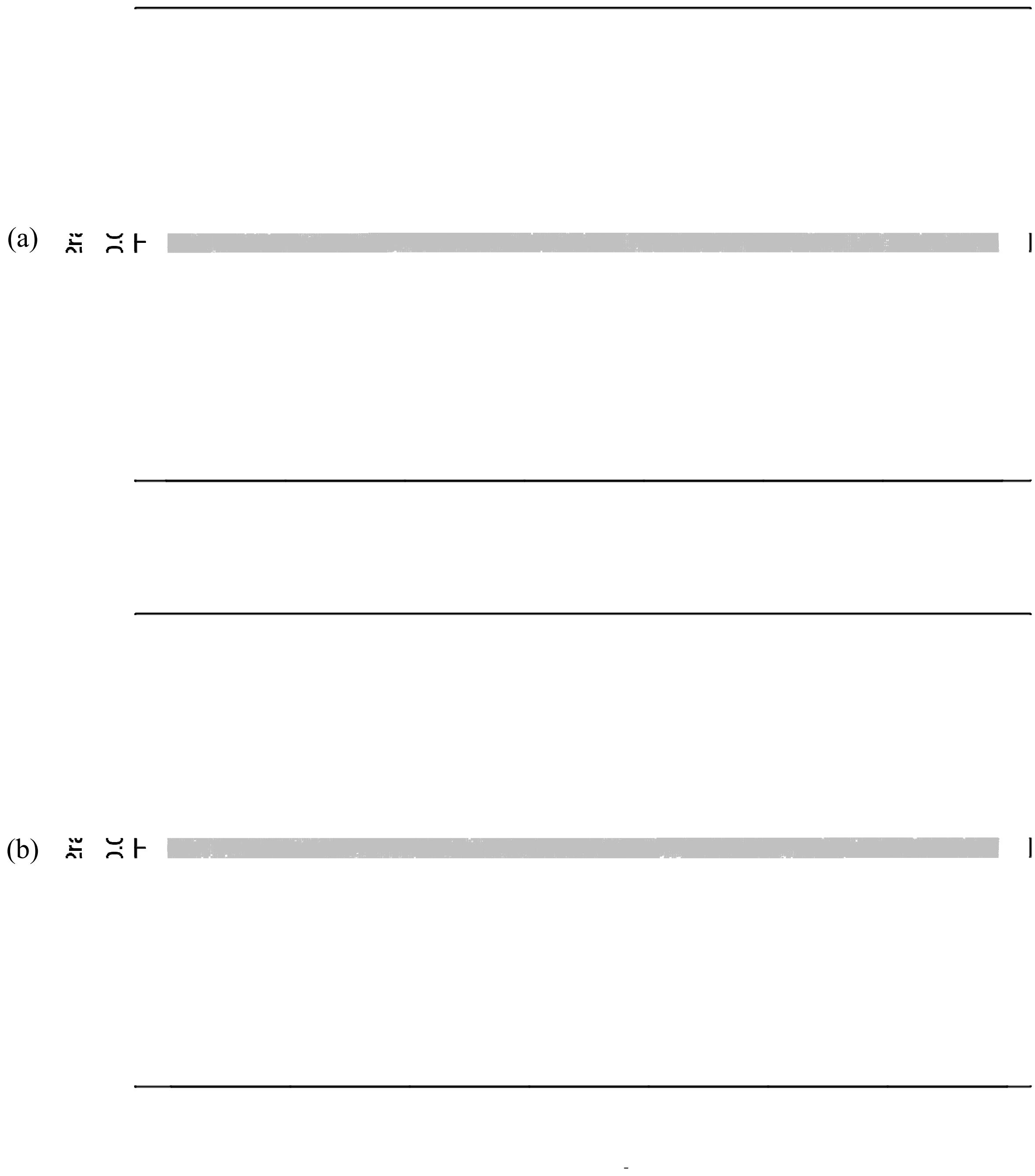


Figure 2.21. Log2ratio signals on chromosome 17 and consensus regions of copy number variations detected by MPFMR, Redon et al.'s method, and BSA.

in (Wu, et al., 2009), the authors applied BSA separately on the data of the four populations. In

Figure 2.22. Distributions of signals and CNVRs detected by MPFMR and Redon et al's method between clones 321 to 348 in chromosome 17.

order to make the results comparable, here we apply BSA on the same filtered data as used to test MPFMR. We plot the CNVRs detected by MPFMR, Redon et al.'s method, and BSA as three rows of bands below the log2ratio signals in each sub-figure. For MPFMR (top row) and BSA (bottom row), each CNVR is plotted as a red or blue band if it contains duplications or deletions, respectively; for Redon et al.'s method (middle row) where the CNVRs do not have directions, each CNVR is plotted as a green band. We can see that although the CNVRs detected by BSA are all included in Redon et al.'s results and overlap perfectly with the CNVRs detected MPFMR, BSA misses too many regions that are detected by the other methods. Compared with BSA, MPFMR not only generates more CNVRs overlapping with Redon et al.'s results, it also

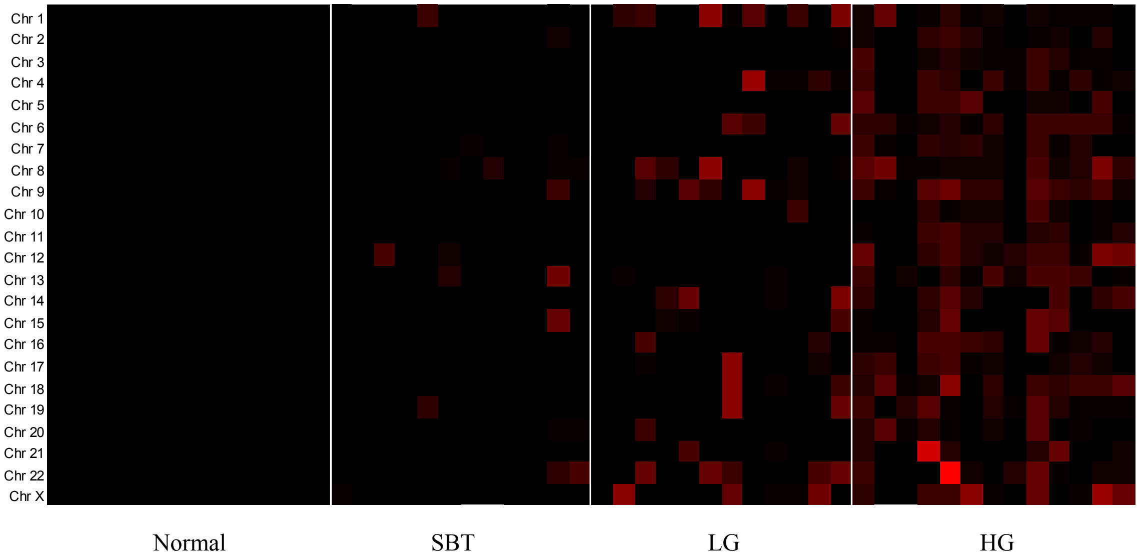


Figure 2.23. Chromosome instability (CIN) indices of normal and ovarian serous carcinomas calculated based on the copy number alterations detected by FMR. Each column of the heatmap corresponds to a sample and each row corresponds to a chromosome.

detects potential candidate CNVRs that are complement to Redon et al.' CNVRs. As an example, we plot in Figure 2.22 the distributions of the log2ratio signals between loci 321 to 348 in chromosome 17. The CNVRs detected by MPFMR and Redon et al.'s method are plotted as two rows of colored bands as explained earlier. MPFMR is able to detect two potential regions of duplications and deletions at locus 340 and 348 respectively, which are not detected by Redon et al.'s method. Furthermore, in the CNVR between loci 324 to 327 in Redon et al.'s results, MPFMR detects two shorter regions of deletions, which enhance the resolution of Redon et al.'s CNVR. These experimental results show that MPFMR is a competitive method for detecting consensus copy number changes for genetic structural variation studies.

### 2.10.4 Chromosome Instability Index

We apply and visualize the CIN index defined in 2.9 on the ovarian cancer copy number data described in 2.10.2.4. We set the $\epsilon$ coefficient $\alpha = 1.645$ for FMR, which is approximately the 95% quantile of the standard normal distribution. Figure 2.23 shows different patterns of chromosome-specific CIN indices of the four classes of samples and Figure 2.24 shows the

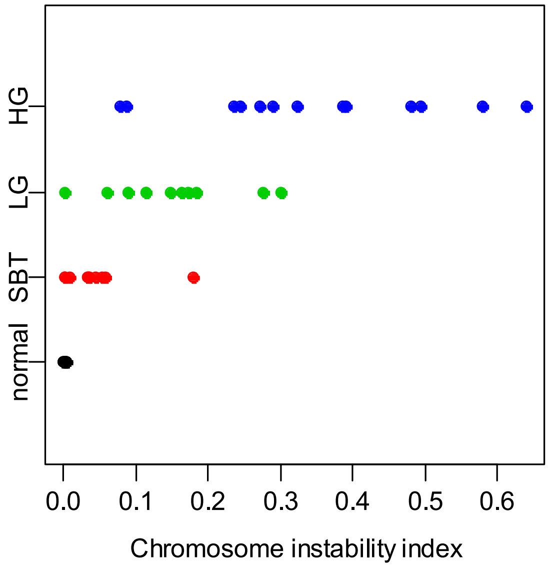


Figure 2.24. Genome-wide CIN indices of normal and ovarian serous carcinomas calculated based on the copy number alterations detected by FMR. Each dot represents a sample and each row corresponds to a phenotype.

global trend of genome-wide CIN indices. HG tumors have high instabilities across all chromosomes while LG tumors are unstable only in some chromosomes, such as chromosome 8, 9, and 22. SBT tumors have lower instabilities compared with LG and HG tumors. The stabilities of the normal samples are reflected by the lowest intensities in the CIN heatmap. The transitions of stabilities from SBT to LG and then to HG are consistent with existing knowledge of ovarian serous carcinomas. SBT and LG are usually considered indolent tumors and it is now believed that LG carcinomas are developed from SBT. Compared with SBT and LG, HG carcinomas are more aggressive and develop fast.

## 2.11 Conclusions and Discussions

DNA copy number change is an important form of structural variation in genomes. CNVs are known to be associated with phenotypic diversities and disease susceptibilities and CNAs may harbor important functional sequences, such as oncogenes and tumor suppressor genes. Recent advances in microarray technologies, specifically high-density CGH and SNP microarrays, allow researchers to investigate copy number changes in sub-microscopic level. Detecting copy number changes in microarray data is a challenging task due to the large numbers of observations and low SNRs in the signal profiles of oligonucleotide based microarrays. Detecting CNAs in tumor genomes further complicates the problem due to the complex signal patterns caused by normal tissue contamination and heterogeneous cell populations. These challenges require a detection method to be both sensitive and robust. In this dissertation, we try to tackle some of the challenges with the proposed fused margin regression method. FMR belongs to a family of regression based copy number change detection methods that use the first order variable fusion constraint to enforce piecewise constant solutions in the estimated copy numbers. The fusion constraint penalizes the $l_1$-norm of the differences between the estimated copy numbers of adjacent loci, which yields a sparse solution of breakpoints in the profile of estimated copy numbers. The difference between FMR and the other regression based methods is that FMR uses the $\epsilon$-insensitive loss function to penalize the deviations between the estimated and observed copy numbers.

Compared with other existing copy number change detection methods, FMR has several advantages:

1. FMR is an intuitive and simple regression model. It casts the copy number change detection problem to a constrained error minimization problem. FMR only requires a single user-supplied parameter, which controls the width of the margin in the $\epsilon$-insensitive loss function; the parameter can be intuitively determined based on the noise variance estimated from the signal profiles;
2. FMR is less sensitive to noises and outliers compared with fused quantile regression and (fused) Lasso due to its $\epsilon$-insensitive loss function, which assigns zero errors to the observations in the margins around the estimated copy numbers and linearly scaled errors to those outside the margins;

3. FMR is a distribution-free model. Unlike methods such as HMM and CBS, FMR does not enforce any assumptions on the number of copy number states and noise distributions in the signal profiles. The simulation studies show that FMR is a robust detection method in the sense that its performance does not significantly fluctuate with the changes of noise distributions and SNRs. FMR performs particularly well compared with the other competing methods when the noises are sampled from real SNP array data. We use the same parameter for FMR across all simulation cases and achieve consistent performances, which also suggests the robustness of FMR;
4. The linear programming problem associated with FMR can be efficiently solved by the solution-path algorithm. The computational efficiency allows FMR to be routinely applied in copy number change analysis on data generated by all existing and potentially future high-throughput platforms using affordable desktop computers;
5. FMR and its path algorithm can model and solve both problems of detecting copy number changes in a single profile and detecting consensus copy number changes in population data.

Based on the detection results of FMR on individual copy number signal profiles, we propose an intuitive definition of chromosome instability index that summarizes the global trend of copy number alterations in genomes. In the experiments of 2.10.4, we can observe distinct CIN patterns of the normal samples and SBT, LG, and HG serous carcinoma samples.

For a given value of $\epsilon$ or equivalently the coefficient $a$ in $\epsilon = a\hat{\sigma}$, the path algorithm generates all possible solutions of breakpoints corresponding to discernable values of $s$, from which an optimal solution can be chosen using a pre-defined threshold of $s$ or a model selection criterion such as SIC. Here we discuss and compare three alternative approaches for choosing appropriate values of $a$. The most straightforward approach, as used in the experiments, is to select a fixed coefficient $a$ that yields generally plausible detection results for all signal profiles.

The results in 2.10.2.2 show that $a = 1.5$ generates satisfactory detection accuracies across all the simulated datasets with different SNRs and noise distributions. The second approach, as described in 2.6, is to use a two-fold cross-validation (CV) to select an optimal value for $a$ from a grid of values (Eilers and de Menezes, 2005). Since the problem of estimating copy numbers is unsupervised by its nature, the two-fold CV can yield an over-fitting choice of $a$. The third approach is to use training data with known copy number states to select $a$. Presumably, the training data should have a similar distribution of CNVs/CNAs as that of the testing data. We compare the three approaches on the simulated data examined in 2.10.2.2. For each dataset, we retain 50 signal profiles as the training set and use the rest 50 profiles as the testing set. In the first approach ("Fixed"), we use $a = 1.5$; in the second approach ("CV"), we apply FMR on each testing profile where $a$ is selected from $\{1 + 0.1k, k = 0, \ldots, 10\}$ using two-fold CV; in the third approach ("Train"), we select $a$ from $\{1 + 0.1k, k = 0, \ldots, 10\}$ such that the average SSE over the 50 training profiles is minimized. We then use the selected $a$ for FMR to detect copy number changes in the 50 testing profiles. The distributions of SSE of the three approaches on the testing data are plotted in Figure 2.25. In more than half of the 16 datasets, the two-fold CV procedure does not help improving the performance at all, which is largely due to the over-fitting problem. On the other hand, selecting $a$ using training data yields better performances as compared to using fixed $a$ in some of the datasets. Unfortunately, training data is usually not available due to the unsupervised nature of the detection problem. Furthermore, we have observed different values of $a$ selected on different training datasets, which implies that the value of $a$ selected on one training dataset with a particular noise distribution may not generalize to data with different noise distributions. The comparison between the three approaches shows that using a fixed empirical value of $a$ is a reasonable way to determine $\epsilon$ considering the unsupervised nature of the copy number change detection problem.

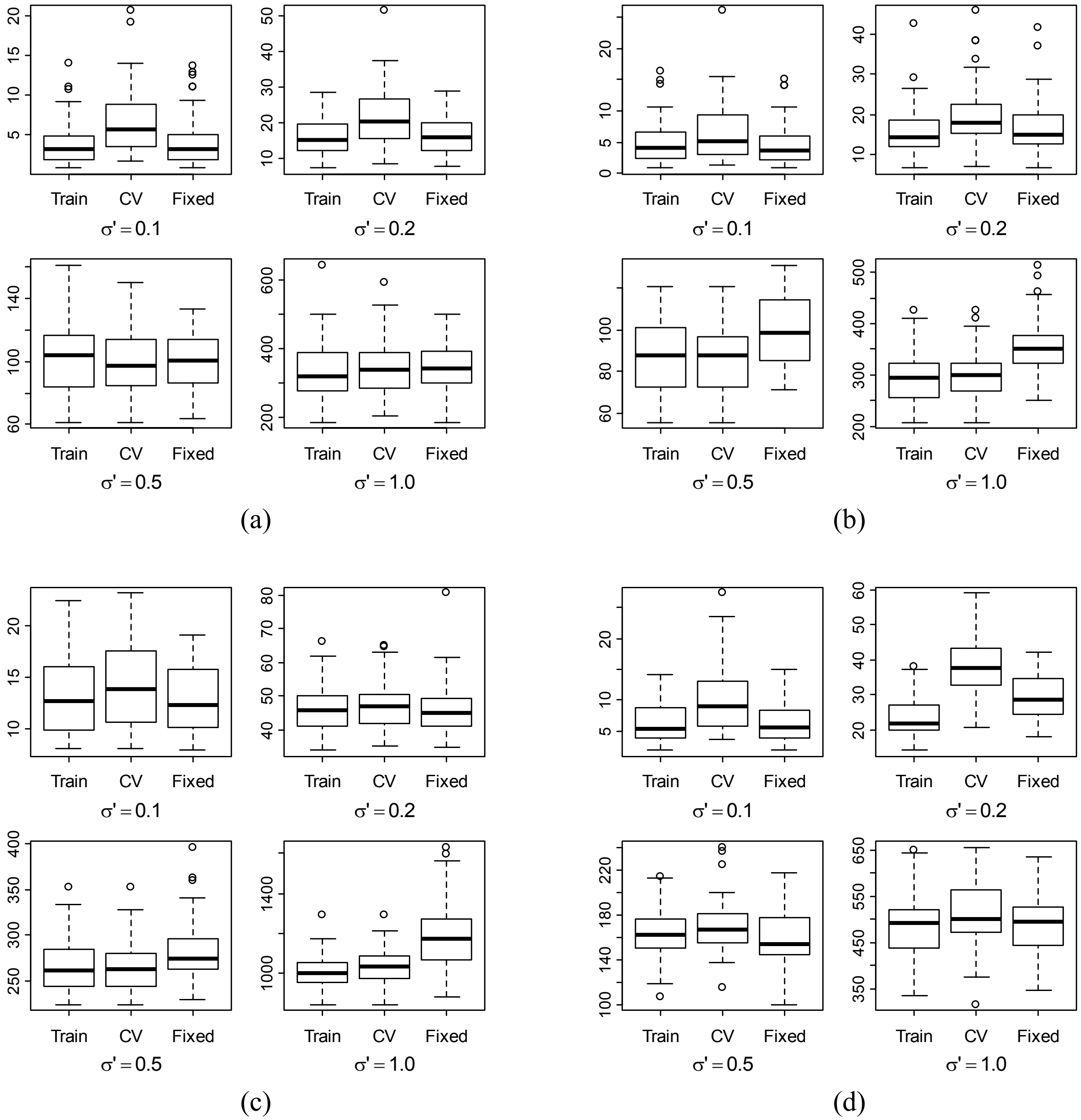


Figure 2.25. Distributions of SSE of different approaches for determining the coefficient of $\epsilon$ on simulated datasets where the noises conform to (a) the standard normal distribution, (b) t-distribution with df=4, (c) log-normal distribution with $\mu = 0$ and $\sigma = 0.5$, and (d) an empirical distribution of noises in real SNP microarray data.

# 3 Learning Tree of Phenotypes Using Gene Expression Data

## 3.1 Introduction

The emergence and development of high throughput genomic data acquisition technologies have been posing new challenges in analyzing high dimensional, small sample size, and multiclass data to the statistics, machine learning, and pattern recognition communities. In supervised learning where the information of disease phenotypes is available, major efforts have been made to discover discriminative biomarkers and design accurate and generalizable classifiers (McLachlan, et al., 2004), while few studies have been conducted to explore the structural information embedded in the molecular level data. The structural information can provide novel insights for understanding the relationships between multiple diseases and/or disease subtypes. Learning the structural information from data remains a highly challenging task in machine learning research, mainly due to the very large search space and the lack of prior knowledge. In this chapter, we try to tackle the problem of learning Tree of Phenotypes (TOP) using multiclass, high dimensional, and small sample size gene expression data. The concept of TOP resembles phylogenetic trees in evolutionary biology (Ciccarelli, et al., 2006; Felsenstein, 2004; Holmes, 1999). In a phylogenetic tree, each species is represented by a single DNA or protein sequence as its unique signature; while in a TOP, the equivalence to a species is a disease phenotype, which is represented by the gene expression values of multiple samples of the disease.

The task of learning TOP is tightly related to the problem of multi-category classification. Conventionally, the main focus of supervised learning on gene expression data is to learn the dependencies between gene expression values and phenotypic labels for constructing prognostic/diagnostic classifiers. When the number of phenotypes exceeds two, parallelstructured multiclass prediction models are usually used, such as one-versus-rest multiclass Support Vector Machines (Ramaswamy, et al., 2001), neural networks (Khan, et al., 2001), Nearest Shrunken Centroids (Tibshirani, et al., 2002), etc. Although widely adopted, parallel classification schemes do not explicitly encode the relationships between multiple classes. On the other hand, hierarchical or tree-structured classifiers, or simply tree classifiers, not only generate multiclass predictions but also directly represent the relationships between multiple phenotypes in their hierarchical structures. In tree classifiers, the multi-category classification task is tackled by a series of simpler tasks, each of which may only require a simple classification model to yield an overall high prediction accuracy. The hierarchical structure of tree classifiers can either be constructed based on prior knowledge or learned directly from data (Simon, 2003). For example, Shedden et al (2003) proposed a pathology and tissue ontogeny based tree structure for building a hierarchical classifier of 14 human cancers. Such a fixed tree is independent of data and therefore will not introduce variances in the prediction outcomes of the tree classifier. On the other hand, the information content of the tree is largely confined by the often incomplete prior knowledge. For example, in Shedden et al's tree structure, seven tumor types, such as lung cancer and prostate cancer, are grouped together to form a monolithic terminal node corresponding to non-Mullerian tumors, which does not display the relationships between these seven cancers. Furthermore, the knowledge based tree may not be consistent with the latent data structure in the gene expression data. For instance, two cancers that are similar in their gene expression patterns can be far apart in the knowledge based tree. Various efforts have been made to computationally learn the structures of tree classifiers from data. The Classification and Regression Tree (CART) is one of the most widely used hierarchical classifiers (Breiman, et al., 1984). Park an Hastie (2005) use several tree building methods to construct hierarchical structures based on data and train a Nearest Shrunken Centroid (NSC) classifier (Tibshirani, et al., 2002) at each node. Tibshirani and Hastie (2007) treated each class of samples as an entirety and use the margin size of a Support Vector Machine (SVM) (Vapnik, 1998) trained on a pair of

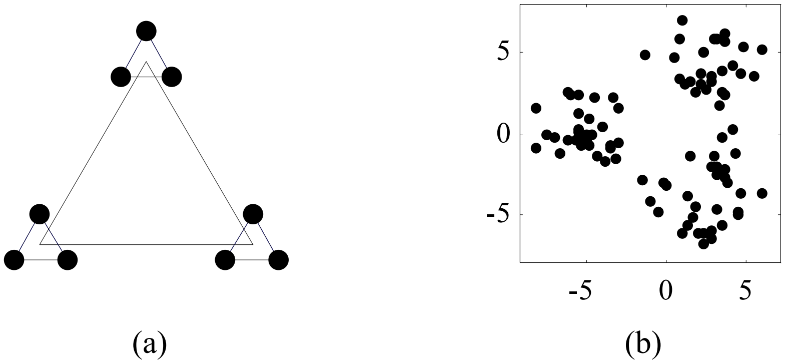
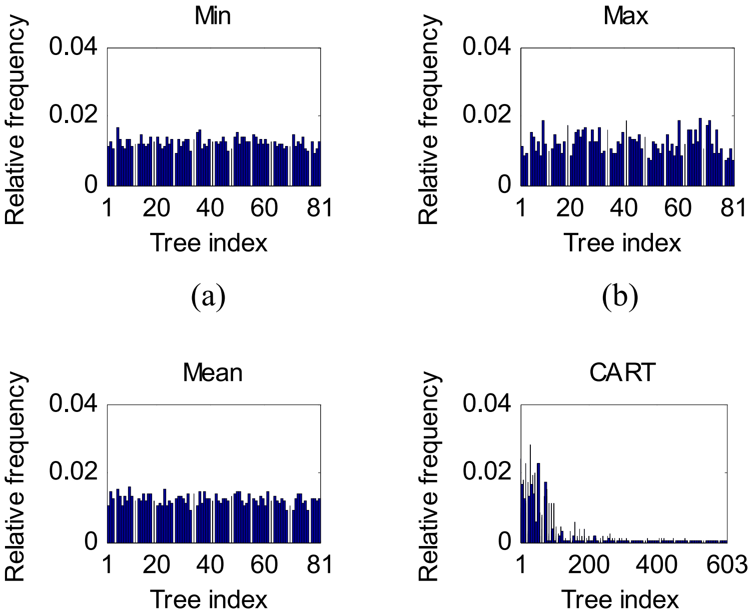


Figure 3.1. (a) The structure of a two-layer generative model. Each black dot is the center of a class; (b) The projection of a dataset drawn from the generative model in the 2-D space of the first two principal components of the sample covariance matrix.

classes as their distance. Based on these pairwise distances, they use the conventional single linkage and complete linkage hierarchical clustering (HC) methods to grow trees in a bottom-up fashion. They also proposed a top-down splitting algorithm that efficiently determines the maximum-margin binary partition of the classes at each node. There are two major problems associated with these automatic tree learning methods. First, these methods can only generate binary trees, which may not be the most informative structures to describe the data. Second, tree structures learned from small sample size, high dimensional gene expression data may be highly unstable, which compromises the interpretability and application of the learned structures.

We demonstrate how the conventional binary tree learning methods would fail using data drawn from a two-layer generative toy model as illustrated in Figure 3.1 (a). The first layer of the model is an equilateral triangle $\Delta_0$ and the second layer consists of three equilateral triangles, each with its geometric center located at a vertex of $\Delta_0$. All the second layer triangles have the same edge length. The 2-dimensional structure is randomly rotated in a 10-dimensional space. Each vertex of the second layer triangles is the mean of a multivariate normal distribution with identity covariance matrix. We create a dataset by drawing 10 observations from each of the nine normal distributions and appending noises drawn from a 190-dimensional spherical normal distribution where each dimension has unit variance and randomly assigned mean. Finally, we



(a) (b)

(c) (d)

Figure 3.2. The empirical distributions of binary trees learned by single linkage HC (a), complete linkage HC (b), centroid linkage HC (c), and CART (d).

randomly shuffle the 200 dimensions of the dataset. The projection of such a dataset in its first two principal component subspace is plotted in Figure 3.1 (b). We create 5000 datasets from the generative model, and for each dataset we apply a tree learning algorithm to generate a binary tree. We tested single linkage HC, complete linkage HC, centroid linkage HC, and CART on the 5000 datasets. In the HC methods, we use the sample mean to represent a class as an entirety. In CART, we truncated each tree to have at least nine terminal nodes such that each terminal node contains samples from the same class. The histograms of the unique binary trees generated by these methods are plotted in Figure 3.2. Each HC method generated 81 structures, which consist of all the possible structures if the two layers are correctly identified. CART generated much more variants in the tree structures compared with the HC methods.

From our preliminary evaluation on the existing methods, we can see that these conventional binary tree learning methods failed to reveal the true parallel structure in the data. The empirical distributions of the trees learned by the HC methods imply that the parallel structure is broken down into multiple sequential binary splits with approximately equal probability. By exploring such instability of trees, we may be able to derive a tree learning method that can recover the latent parallel structure.

## 3.2 Motivations and Contributions

We are motivated by the following reasons to propose novel TOP learning methods:

1. We hypothesize that heterogeneous disease/disease subtypes have diverse relationships between themselves (e.g., non-binary and multi-scale). Given genomic data of the phenotypes and an appropriate distance measure, some phenotypes have more similar patterns than the others. Such inhomogeneous relationships can be naturally represented by tree structures;
2. Although trees of disease phenotypes can be built based on histological and pathological knowledge, those structures may not be consistent with the latent or true structures of the phenotypes in the molecular level data. Learning TOP from genomic data shall reveal those hidden yet informative relationships between phenotypes;
3. In a small sample size context often seen in microarray gene expression data, conventional tree learning algorithms usually generate highly variable tree structures given different realizations of the same data distribution, which leads to non-reproducible outcomes. In order to learn a tree structure with high confidence, the tree learning algorithm must take into account the relationship between the small sample size and stability of learned tree structures;
4. When learning a tree of phenotypes in a tree space from a given dataset, the conventional tree learning methods only search a subspace consisting of all-binary trees. Therefore, for data generated from non-binary structures as shown in the previous simulation, the conventional methods are incapable of learning the true tree structures and consequently

demonstrate high instabilities. We hypothesize that, by reducing the instability of the outcomes of a tree learning method, we have better chance to identify the true latent structure that cannot be reliably learned otherwise.

To tackle the problem of learning TOP using genomic data, we explore two learning methods in this chapter. In section 3.3, we introduce a visualization based human interactive data modeling method for learning coarse-to-fine hierarchical relationships between multiple disease phenotypes. Benefit from its interactive nature, the users can incorporate domain knowledge into the tree structures. To generate a highly reproducible tree structure, the method incorporates a leave-one-out cross validation based stability analysis. In section 3.4, we propose a fully automatic, stability analysis based framework for learning TOP from gene expression data. The method uses a node bandwidth constraint to reach a balance between the reproducibility and descriptive power of tree structures learned by hierarchical group-clustering methods. A bootstrap based stability analysis approach is developed to generate a full spectrum of stable tree structures. We test the proposed methods using two microarray gene expression datasets of human diseases. The results show that our methods can effectively generate highly reproducible trees of phenotypes.

## 3.3 Learning Tree of Phenotypes Using SA-ccsmVISDA

A tree of phenotypes is a natural way to describe the relationships between diseases or disease subtypes. We devise a method that learns the latent relationships between phenotypes in gene expression data with the guidance of human interaction, namely Color-Coded Supervised Mode VIsual Statistical Data Analyzer (ccsmVISDA) (Feng, et al., 2006), an extension of the original VISDA algorithm (Bakay, et al., 2006; Wang, et al., 2000; Wang, et al., 2003; Zhu, et al., 2008). To assure good generalizability of the learned tree structure for small sample size data, ccsmVISDA is incorporated within a leave-one-out stability analysis (SA-ccsmVISDA). The final learned tree is the one receiving the most votes among all the leave-one-out trees.

### 3.3.1 ccsmVISDA

The ccsmVISDA algorithm hierarchically displays the classes and constructs a tree. We call a tree node with two or more classes a composite or an internal node, and a tree node with only one class a terminal node. Starting from the root node, the samples are partitioned into clusters to grow the tree. A cluster is considered as a composite node if it contains more than one class; otherwise, it is a terminal node. At each composite node, the local data is first projected into a visualization subspace that allows the user to interactively initialize the clustering. The cluster partition is iteratively updated until a stable state is reached. During the updating, samples from the same class are forced to be assigned to the same cluster, i.e. clusters learn to fully own either one or multiple classes.

Before constructing a tree, we first filter the genes by their signal-to-noise ratios (SNR) in order to remove those non-discriminatory genes and ease the computational demand. Suppose that the dataset consists of $K$ classes with $p$ genes, and the $k$ th class has $n_k$ samples, $k = 1,\ldots,K$. Denote the sample mean and standard deviation of data in class $k$ on gene $i$ by $\mu_{ik}$ and $\sigma_{ik}$, respectively, $k = 1,\ldots,K$, $i = 1,\ldots,p$. We define the overall SNR as

$$SNR(i) = \frac{\sum_{u=1}^{K-1}\sum_{v=u+1}^{K} \pi_u \pi_v (\mu_{iu} - \mu_{iv})^2}{\sum_{k=1}^{K} \pi_k \sigma_{ik}^2},$$

$$\pi_k = \frac{n_k}{\sum_{j=1}^{K} n_j}, \quad k = 1,\ldots,K.$$

The top $m$ genes with the highest SNRs are used to represent the data. Here $m$ is proportional to $K$ and determined empirically. We apply ccsmVISDA on the filtered data.

Suppose at a composite node there are $n_0$ samples with $m$ genes coming from $K_0$ classes. Denote the mean vector and the covariance matrix of class $c$ by $\boldsymbol{\mu}_c$ and $\mathbf{C}_c$, respectively. All the samples are first projected into a two-dimensional space selected by multiclass Fisher's discriminant analysis (Fukunaga, 1990), which utilizes the class information to find the most discriminatory subspace for the $K_0$ classes. The projection space is spanned by the two vectors that maximize Fisher's criterion, i.e.

$$\mathbf{W}_0 = \arg\max_{\mathbf{W}} \left\{ \mathrm{trace}(\mathbf{W}^{\mathrm{T}}\mathbf{S}_{\mathrm{w}}^{-1}\mathbf{S}_{\mathrm{b}}\mathbf{W}) \right\},$$

where $\mathbf{W}$ and $\mathbf{W}_0$ are $m$ by 2 matrices. The within class scatter matrix $\mathbf{S}_{\mathrm{w}}$ and the between class scatter matrix $\mathbf{S}_{\mathrm{b}}$ are defined as (Fukunaga, 1990)

$$\mathbf{S}_{\mathrm{w}} = \sum_{c=1}^{K_0} \pi_c \mathbf{C}_c,$$

$$\mathbf{S}_{\mathrm{b}} = \sum_{i=1}^{K_0-1} \sum_{j=i+1}^{K_0} \pi_i \pi_j (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^{\mathrm{T}}.$$

Here $\pi_c$ is the mixing proportion of class $c$, i.e. $\pi_c = |I_c|/n_0$, with $I_c$ the index set of the samples from class $c$, and $|I_c|$ the cardinality of set $I_c$. Each sample $\mathbf{y}$ in the $m$-dimensional space is projected into the 2-D space by

$$\mathbf{x} = \mathbf{W}_0^{\mathrm{T}}\mathbf{y}.$$

Given the 2-D projection of the samples, the user is invited to determine both the number of clusters $M$, $2 \le M \le K_0$, and the initial center of each cluster in the projection plot. A cluster is modeled by a normal distribution $p(\mathbf{x}\,|\,\boldsymbol{\mu},\mathbf{C})$ with mean $\boldsymbol{\mu}$ and covariance matrix $\mathbf{C}$. In order to get a robust partition of the samples, the user is further required to select the centers of two more partitioning scenarios that have $M-1$ and $M+1$ clusters. All three scenarios will undergo the same clustering process and the best one will be selected. Given the user-defined initial centers $\boldsymbol{\mu}_{\mathbf{x}1}^{(1)},\ldots,\boldsymbol{\mu}_{\mathbf{x}M_0}^{(1)}$ of $M_0$ clusters in the 2-D projection, $M_0 \in \{M-1, M, M+1\}$, each sample $\mathbf{x}_i$ is assigned to a cluster $g_i$ such that

$$g_i = \arg\min_{k\in\{1,\ldots,M_0\}} \left\{ \left\| \mathbf{x}_i - \boldsymbol{\mu}_{\mathbf{x}k}^{(1)} \right\| \right\}, \quad i = 1,\ldots,n_0.$$

The initial covariance matrix $\mathbf{C}_{\mathbf{x}k}^{(1)}$ of each cluster is defined to be an identity matrix and the initial prior probability of each cluster is calculated by

$$\pi_k^{(1)} = \sum_{i=1}^{n_0} \delta(g_i - k) / n_0 , \quad k = 1, \ldots, M_0 ,$$

where $\delta(\cdot)$ is the Kronecker function

$$\delta(x) = \begin{cases} 1, & x = 0 \\ 0, & x \neq 0 \end{cases} .$$

The initial partitioning of samples is iteratively updated by an EM-like, two step procedure. In the first step of an iteration, each sample is assigned to a cluster. Denote the posterior probability of sample $\mathbf{x}_i$ belonging to cluster $k$ at the $n$th iteration by $z_{ik}^{(n)}$,

$$z_{ik}^{(n)} = \frac{\pi_k^{(n)} p(\mathbf{x}_i \mid \boldsymbol{\mu}_{\mathbf{x}k}^{(n)}, \mathbf{C}_{\mathbf{x}k}^{(n)})}{\sum_{j=1}^{M_0} \pi_j^{(n)} p(\mathbf{x}_i \mid \boldsymbol{\mu}_{\mathbf{x}j}^{(n)}, \mathbf{C}_{\mathbf{x}j}^{(n)})} , \quad n \geq 1 ,$$

where $\pi_k^{(n)}$, $\boldsymbol{\mu}_{\mathbf{x}k}^{(n)}$, and $\mathbf{C}_{\mathbf{x}k}^{(n)}$ are the estimated prior probability, mean, and covariance matrix of the $k$th cluster at the $n$th iteration, respectively. Denote the index set of samples in class $c$ by $I_c$, $c = 1, \ldots, K_0$; each class $c$ is exclusively assigned to a unique cluster $k_c$ such that

$$k_c = \arg \max_{k \in \{1, \ldots, M_0\}} \left\{ \sum_{i \in I_c} z_{ik}^{(n)} \right\} , \quad c = 1, \ldots, K_0 . \tag{3.1}$$

Note that after operation (3.1), multiple classes can be assigned to the same cluster. The partition is updated accordingly:

$$S_k^{(n)} = \bigcup_{\substack{c: k_c = k \\ c \in \{1, \ldots, K_0\}}} I_c , \quad k = 1, \ldots, M_0 .$$

Apparently the partition $\mathcal{S}^{(n)} = \left\{ S_1^{(n)}, \ldots, S_{M_0}^{(n)} \right\}$ satisfies

$$\bigcup_{k=1}^{M_0} S_k^{(n)} = \{1, \ldots, n_0\} \quad \text{and} \quad S_i^{(n)} \bigcap \mathrm{S}_j^{(n)} = \varnothing , \ \forall i \neq j .$$

In the second step of the same iteration, the prior probability, mean, and covariance matrix of each cluster are updated (for the next iteration) by

$$\boldsymbol{\mu}_{\mathbf{x}k}^{(n+1)} = \frac{\sum_{i\in S_k^{(n)}} \mathbf{x}_i}{\left|S_k^{(n)}\right|},$$

$$\pi_k^{(n+1)} = \frac{\left|S_k^{(n)}\right|}{\sum_{i=1}^{M_0} \left|S_i^{(n)}\right|},$$

$$\mathbf{C}_{\mathbf{x}k}^{(n+1)} = \frac{\sum_{i\in S_k^{(n)}} \left(\mathbf{x}_i - \boldsymbol{\mu}_{\mathbf{x}k}^{(n+1)}\right)\left(\mathbf{x}_i - \boldsymbol{\mu}_{\mathbf{x}k}^{(n+1)}\right)^{\mathrm{T}}}{\left|S_k^{(n)}\right|}.$$

The two steps are repeated until the partition no longer changes, i.e. $\exists n'$, $\mathcal{S}^{(n'-1)} = \mathcal{S}^{(n')}$. Denote the distribution parameters estimated at the final iteration by $\pi_k$, $\boldsymbol{\mu}_{\mathbf{x}k}$, and $\mathbf{C}_{\mathbf{x}k}$, $k = 1,\ldots,M_0$, the distribution of the partition is characterized by the SFNM

$$p(\mathbf{x}) = \sum_{k=1}^{M_0} \pi_k p(\mathbf{x} \mid \boldsymbol{\mu}_{\mathbf{x}k}, \mathbf{C}_{\mathbf{x}k}). \qquad (3.2)$$

Finally, the best partition among the three candidate scenarios is selected by the user under the guidance of Minimum Description Length (MDL) scores (Wang, et al., 2000). The MDL score for the partition with $M_0$ clusters is

$$\mathrm{MDL}(M_0) = -\sum_{i=1}^{n_0} \log p(\mathbf{x}_i) + \frac{6M_0 - 1}{2} \cdot \log n_0, \;\; M_0 \in \{M-1, M, M+1\},$$

where $p(\mathbf{x})$ is the pdf of SFNM in (3.2). To grow the tree of phenotypes, each cluster in the selected partition becomes a child node of the current composite node, and the described partitioning procedure is repeated recursively until each newly generated node only contains samples from the same class.

We pursue such a human-interactive visualization approach within the mixture modeling for two complementary reasons. First, users of ccsmVISDA can incorporate their domain knowledge about the relationships between phenotypes during construction of the trees by designating the number and centers of clusters of labeled data. Second, the automatic clustering procedure can reveal new latent structures in the data and help the users to discover new relationships between phenotypes. The merits of this approach have been empirically evidenced by biological studies (Bakay, et al., 2006; Zhao, et al., 2003).

### 3.3.2 Stable Solution of the Tree

Learning a tree from all the samples may cause over-fitting due to small sample size and potentially poor data quality. Thus, we embed the ccsmVISDA learning procedure within a leave-one-out stability analysis to generate leave-one-out trees, and select the one occurs with the highest frequency as the final solution. The winning tree reflects the underlying stable structural information in the data since it is learned from the dataset and amongst all learned trees best survives small disturbances of the data (Poggio, et al., 2004). The stability analysis based ccsmVISDA (SA-ccsmVISDA) is more robust compared with the original ccsmVISDA in the sense that, given different realizations of the data distribution, SA-ccsmVISDA will generate similar solutions. The robustness of the tree learning algorithm is critical for making scientific discoveries, since a learned tree must be highly reproducible in the face of small data variations in order to be used as a hypothesis for the underlying relationships between phenotypes.

Denote by $\Omega_K$ the sample space of all possible tree structures with $K$ terminal nodes. Since $\Omega_K$ contains a finite number of elements, we can designate a bijection $T : \Omega_K \mapsto I_K$ that maps structures in $\Omega_K$ to indices in set $I_K = \{1, \ldots, |\Omega_K|\}$. Therefore, for a TOP learning problem with $K$ phenotypes consisting of $n_k$ observations each, the outcome of a tree learning method $L$ is simply an index in $I_K$. We can use the entropy of the distribution $p(l)$ of random variable $L$ to measure the stability of the tree learning method,

$$\mathrm{H}(L) = -\sum_{l \in I_K} p(l) \log p(l) .$$

In SA-ccsmVISDA, we estimate $p(l)$ by applying $L$ on the leave-one-out (LOO) datasets $\mathcal{Y}_{\text{LOO}} = \left\{ \mathbf{Y}^{(-1)}, \ldots, \mathbf{Y}^{(-n)} \right\}$ of a training dataset $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$, where $\mathbf{Y}^{(-i)}$ is the subset derived by removing $\mathbf{y}_i$ from $\mathbf{Y}$. Denote by $\hat{p}(l)$ the empirical distribution of trees learned by $L$ on $\mathcal{Y}_{\text{LOO}}$, we measure the stability of the tree learning method $L$ by

$$S(L \mid \mathcal{Y}_{\text{LOO}}) = -\sum_{l \in I_K} \hat{p}(l) \log \hat{p}(l). \tag{3.3}$$

The output of SA-ccsmVISDA is the tree structure corresponding to the mode of $\hat{p}(l)$, i.e.

$$T^{-1}(\arg\max_{l \in I_K} \hat{p}(l)).$$

In the experiments, we also show on the winning tree the separability between the phenotypes at a composite node as the common length of edges connecting the composite node to its child nodes. The separability is defined as the average of the Fisher's criteria between all pairs of phenotypes at a composite node in the 2-D projection space, calculated using all the samples.

The leave-one-out based SA-ccsmVISDA approach for tree learning is practically feasible given the small sample sizes in most of the existing microarray datasets. The amount of human interaction required for each of the LOO trials is not always the same. In the experiments we have observed that leaving one sample out usually will not change the top level structures of the trees. The projections will mostly change at the deeper levels under the composite nodes that include the left-out sample. For such instances, we only need to apply ccsmVISDA once on those branches that are not subject to change in a specific subset of LOO trials. Thus, in practice, the human interaction in SA-ccsmVISDA is less intensive than that required for repetition of the ccsmVISDA procedure over all LOO training datasets. If the sample sizes of microarray datasets increase in the future, we can modify our approach to be semi or fully automated by exhaustively searching the optimal number of clusters at each composite node; but in such a way the users will have less or no chance to incorporate their domain knowledge into the tree structures. For existing microarray data with small sample sizes, our SA-ccsmVISDA method has a reasonable balance between robustness and practical feasibility.

### 3.3.3 Hierarchical Classifiers

The tree structure learned by SA-ccsmVISDA can be used to build hierarchical classifiers. Classification tree is a general framework for solving multiclass classification problems (Simon, 2003). We can use potentially any classifier on the internal nodes to make intermediate predictions; we can also use different subspaces (gene subsets) on different internal nodes. Subspace feature selection will help not only improving classification performance, but also finding the genes that account for the similarities or differences between subsets of phenotypes (Shedden, et al., 2003).

To select genes for classification, we used the feature filtering and selection approach proposed by Shedden et al (2003). First we remove the control genes from data. Each expression value $x$ is transformed by $\log[\max(x,0)+50]$. All those genes on which the sample standard deviations are smaller than 0.7 are removed. For each of the $K$ clusters (child nodes) of a composite node, the genes are ranked according to the difference between the mean of the samples in the cluster and that of the samples in the rest $K-1$ clusters pooled together. The top $\alpha$ genes with the largest differences are selected for each cluster and the union of the $K$ gene sets is used as the feature space for classification. Here $\alpha$ is determined by the users.

Given the selected feature space, we use one-versus-rest multiclass Support Vector Machines (OVR-MSVM) (Ramaswamy, et al., 2001) as the node classifiers. For a given dataset, we evaluate both hard classifiers where the outputs are class labels, and soft classifiers where the outputs are estimated post probabilities (Platt, 1999). In the hard classification scheme, for each cluster of a composite node, a binary SVM is trained in the selected gene subspace to separate the samples of the cluster from all the other samples. A testing sample is assigned to the cluster whose associated binary SVM has the largest real-valued output, and it is passed on to the child node corresponding to the cluster. When the testing sample reaches a terminal node, it is assigned to the phenotype associated with the node. In the soft classification scheme, an OVR-MSVM is trained in the same way as in the hard classification scheme except that for each binary SVM the real-valued output is transformed to a posterior probability using the method in (Platt, 1999). The output $g_k(\mathbf{x})$, $k = 1,\ldots,M_0$, of the $M_0$ binary SVMs are normalized by

$$\hat{p}(k \mid \mathbf{x}) = \frac{g_k(\mathbf{x})}{\sum_{i=1}^{M_0} g_i(\mathbf{x})}, \quad k = 1,\ldots,M_0. \tag{3.4}$$

A testing sample is then passed on to each of the child nodes for further testing. According to the chain rule of hierarchical classification (Wang, et al., 2000), the posterior probability of a testing sample belonging to class $k$ is the product of all the estimated posterior probabilities (3.4) on the path from the root node of the tree to the terminal node corresponding to class $k$. A testing sample is assigned to a class according to the Max A Posteriori (MAP) rule. We will see in the experiments that the soft classification scheme improves the performance of the tree classifier.

## 3.4 Automatic Learning Tree of Phenotypes Using Bandwidth Constrained Stability Analysis

In the previous section, we propose a visualization based tree learning approach where the users can incorporate domain knowledge into the tree learning. We now propose a fully automatic stability analysis based tree learning method for situations where human interaction is not feasible or domain knowledge is absent. We consider the tree learning problem as an estimation problem where the purpose is to determine a complex parameter that describes the structural information embedded in the population data. The sample space of the phenotypes is described as a random vector $X \in \mathbb{R}^m$ and its associated class label $Y \in \{1,\ldots,K\}$ with joint distribution $p(x,y)$, $(x,y) \in \mathbb{Z}$, $\mathbb{Z} = \mathbb{R}^m \times \{1,\ldots,K\}$. Depending on the tree learning method used, there are different tree structures associated with $p(x,y)$. For example, the full linkage and single linkage hierarchical clustering methods may generate different trees on the same data. We shall introduce the tree learning methods investigated in section 3.4.1. Given an i.i.d. sample $\mathcal{Z} = \{(X_i, Y_i)\}_{i=1}^{n}$ drawn from $p(x, y)$, we want to extract the hierarchical structure associated with a particular definition of tree by a tree learning algorithm $T : \mathbb{Z}^n \mapsto \mathbb{T}$, which maps a sample of size $n$ to a rooted tree. Billera et al (2001) propose a metric space of rooted, semi-labeled trees. A tree in the space is an acyclic graph with a root node and $K$ terminal nodes labeled by $K+1$ symbols, conventionally 0 for the root and $k \in \{1, \ldots, K\}$ for each terminal. A composite node (internal node of a tree) is fully determined by the sub-tree rooted at it and hence is not labeled. An edge in a tree corresponds to a binary partition of the $K+1$ symbols. An edge connecting internal nodes is called internal edge. The tree learning algorithm $T$ will generally create different tree structures given different realizations of a sample. Such variability hampers the interpretability of the trees and also introduces variances in the follow-up applications of the trees. For example, the classification error of a tree classifier may inflate due to the variance introduced by an unstable tree learner (Breiman, 1996). We try to tackle this problem by introducing a stability analysis based framework to control the variance of a tree learning algorithm, which is introduced in section 3.4.2.

### 3.4.1 Hierarchical Group Clustering

In this section, we introduce four phenotypic tree learning methods investigated in this chapter, which are based on the agglomerative hierarchical clustering strategy in unsupervised learning mode. In these methods, each class is treated as an entirety, i.e. each node in a learned tree contains either all or none of the observations of a particular class, hence we call these methods Hierarchical Group Clustering (HGC). The bottom-up tree building procedures start from individual classes where each class is treated as a single class point in the unsupervised HC methods. A tree is then built based on these points in the conventional way.

Denote the matrix of $n$ samples measured on $m$ features (genes) by an $n$ by $m$ matrix $\mathbf{X}$, where each row of $\mathbf{X}$ corresponds to a sample and each column of $\mathbf{X}$ corresponds to a feature. Denote the $i$th row vector, the $j$th column vector, and the observation of the $i$th sample on the $j$th feature in $\mathbf{X}$ by $\mathbf{x}^{(i)} \in \mathbb{R}^m$, $\mathbf{x}_j \in \mathbb{R}^n$, and $x_j^{(i)}$, respectively. Denote the labels of the $n$ samples by $\mathbf{y} = [y_1, \ldots, y_n]^{\mathrm{T}}$, $y_i \in \{1, \ldots, K\}$. The $k$th class, denoted by $C_k$, is defined as the index set of the samples with label $k$, i.e. $C_k = \{i \mid y_i = k, \forall i \in \{1, \ldots, n\}\}$, $k = 1, \ldots, K$. The classes constitute a partition of the indices of all the samples, i.e. $\bigcup_{k=1}^{K} C_k = \{1, \ldots, n\}$ and $C_i \bigcap C_j = \varnothing$, $\forall i \neq j$. A group $G_S$ is defined as the union of one or more classes indexed by label set $S \subseteq \{1, \ldots, K\}$, i.e. $G_S = \bigcup_{k \in S} C_k$.

In the HGC methods, a binary tree is built from a set of classes. At the beginning, each class is a candidate group (terminal node) in a pool. Given a dataset consisting of $K$ classes, a binary tree is built with $K-1$ steps of merging. For each step, the two groups in the pool with the minimum dissimilarity (maximum similarity) are picked from the pool and combined to form a new group (internal node), which is put back into the pool as a candidate for subsequent merging steps. The merging procedure continues until there are only two groups left in the pool, which are merged to create the root node of the binary HGC tree. We use four dissimilarity measures defined based on groups.

*Centroid linkage based on group means* This dissimilarity measure for HGC is the same as the centroid linkage measure in HC (Duda, et al., 2001). It is defined as the Euclidean distance between the mean vectors of two candidate groups,

$$d_1(G_{S_1}, G_{S_2}) = \left\| \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2 \right\|,$$

$$\boldsymbol{\mu}_i = \frac{1}{|G_{S_i}|} \sum_{k \in G_{S_i}} \mathbf{x}^{(k)}, \ \ i = 1, 2.$$

*Centroid linkage based on group medians* The dissimilarity is defined as the Euclidean distance between the median vectors of two groups

$$d_2(G_{S_1}, G_{S_2}) = \|\mathbf{v}_1 - \mathbf{v}_2\|.$$

The $j$ th element of the median vector $\mathbf{v}_i$ is the median of the observations on the $j$ th feature in group $G_{S_i}$.

*Centroid linkage based on Fisher separability* The dissimilarity is defined as (Fukunaga, 1990)

$$d_3(G_{S_1}, G_{S_2}) = \mathrm{trace}\left(\mathbf{S}_{\mathrm{w}}^{-1} \cdot \mathbf{S}_{\mathrm{b}}\right).$$

The between class scatter matrix $\mathbf{S}_{\mathrm{b}}$ and the within class scatter matrix $\mathbf{S}_{\mathrm{w}}$ are defined as (Fukunaga, 1990)

$$\mathbf{S}_{\mathrm{b}} = (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^{\mathrm{T}},$$

$$\mathbf{S}_{\mathrm{w}} = \frac{n_1}{n_1 + n_2}\boldsymbol{\Sigma}_1 + \frac{n_2}{n_1 + n_2}\boldsymbol{\Sigma}_2,$$

where $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ are the sample mean and covariance matrix of the samples in group $G_{S_i}$; $n_i$ is the size of $G_{S_i}$.

*Complete linkage based on maximized margin* Tibshirani and Hastie (2007) proposed to use single linkage and complete linkage HC to construct classification trees, where the dissimilarity measure between two classes is the size of the margin of a linear SVM trained on the two classes

$$M(C_i, C_j) = \frac{2}{\|\mathbf{w}_{ij}\|}. \tag{3.5}$$

The vector $\mathbf{w}_{ij} \in \mathbb{R}^m$ is the solution to the following quadratic programming problem

$$\min \frac{1}{2} \| \mathbf{w}_{ij} \|^2 ,$$

$$\text{s.t.} \begin{cases} \left\langle \mathbf{w}_{ij} \cdot \mathbf{x}^{(k)} \right\rangle + b \geq 1, & \forall k \in C_i \\ \left\langle \mathbf{w}_{ij} \cdot \mathbf{x}^{(k)} \right\rangle + b \leq -1, & \forall k \in C_j \end{cases} .$$

Note that the measure (3.5) only applicable to two classes that are linearly separable, therefore it requires a relatively large number of features in order to achieve the linear separability. It is found that the complete linkage HC combined with the measure (3.5) usually generates more balanced tree structures compared with single linkage HC (Tibshirani and Hastie, 2007), which implies better biological interpretability; therefore we adopt the complete linkage strategy in our study. The dissimilarity between two groups in complete linkage HGC is defined as (Fukunaga, 1990)

$$d_4(G_{S_1}, G_{S_2}) = \max_{u \in S_1, v \in S_2} M(C_u, C_v) .$$

### 3.4.2 Node Bandwidth Constrained Stability Analysis

As shown in Figure 3.1 and Figure 3.2, a tree learning algorithm may generate different hierarchical structures given different realizations of a data distribution. In order to analyze the variability of the trees, we can consider the tree learning problem in the perspective of model selection (Hastie, et al., 2001). We characterize the complexity of a tree by the maximum number of child nodes (out-degree) an internal node (split) can have. Given a set of $K$ classes, a binary tree has the maximum descriptive power among all possible tree structures, accompanying with more parameters to be estimated. To be more specific, we have to estimate the lengths of $K-2$ internal edges in a binary tree with $K$ terminal nodes, while less than or equal to $K-2$ edge lengths if we allow the tree to have non-binary splits. Therefore a binary tree would always have higher complexity compared with non-binary trees. As in other model selection problems, given a limited number of observations, we prefer a mechanism to trade the descriptive power (model complexity) with the generalization of the solutions. Different from classification problems where the generalization is measured by the expected classification error, which can be estimated by various cross validation methods, we do not have such measures for the tree learning problem due to its unsupervised nature (i.e., true tree structure is unknown but needs to be learned). In order to measure and control the variability of the trees, we introduce the concept of "node bandwidth" – the maximum number of child nodes a node split can have. By constraining the node bandwidth, we can control the split size and the number of levels a tree can have, and indirectly control the variability (and complexity) of the learned trees. Given a finite number of observations, we use bootstrap (Efron and Tibshirani, 1993) to generate a simulated resampling distribution of trees, and use this distribution to study the variability of a tree learning algorithm.

In order to incorporate node bandwidth into tree learning, we can either design a tree learning algorithm that honors this parameter in its learning mechanism, or we can apply an arbitrary tree learning algorithm and then modify the output trees to have a desired node bandwidth (e.g., selective merging). For the first approach, there are several potential problems: first, at each split the tree learning algorithm has to search the optimal number of child nodes from 2 to the given node bandwidth, which is time consuming for a large number of classes; second, it is a nontrivial task to define a distance measure between multiple groups; and lastly, the distances between different numbers of groups may not be directly comparable.

We therefore have chosen to pursue the second approach in our study. We first apply one of the selected four tree learning algorithm as described in 3.4.1 on bootstrap samples of the training data to generate a set of trees. We then use the node bandwidth as a parameter to restructure the learned trees. Finally we estimate the distribution of the modified trees. In order to achieve a specific node bandwidth, we merge the hierarchies of the trees in the direction from the roots to the terminal nodes (top-down). Let the desired node bandwidth be $F$. Obviously we have $2 \le F \le K$, where $K$ is the number of classes. Define the length of path $l(u,v)$ as the number of edges between node $u$ and its descendant $v$. For a non-terminal node $u$, its child nodes belong to the set $D(u) = \{v \mid l(u,v) = 1\}$. If $u$ is a terminal node, we define $D(u) = \varnothing$. At a particular node $u$, a merging procedure can be briefly described as follows:

(a) If $D(u) = \varnothing$, terminate the merging process;

(b) If $|\bigcup_{v \in D(u)} D(v)| + |\{v \in D(u) \mid D(v) = \varnothing\}| > F$, keep $u$ intact and continue to process each child node of $u$ from step (a);

(c) Otherwise, remove the nodes in $\{v \in D(u) \mid D(v) \neq \varnothing\}$ and connect the nodes in $\bigcup_{v \in D(u)} D(v)$ as new child nodes of $u$ if the stability of the splits at $u$ is improved. We continue the process on each child node of $u$ from step (a).

After the top-down merging on the bootstrapped trees, we have a set of new tree structures with less variation compared with the original tree structures. By varying the node bandwidth, we can get a full spectrum of tree structures with different complexities and their corresponding bootstrap resampling distributions. We then characterize a bootstrap distribution of trees by its entropy and the maximum relative frequency of all distinct tree structures. Specifically, denote the unique bootstrap tree structures and their relative frequencies by $t_1, \ldots, t_n$ and $f_1, \ldots, f_n$, the entropy of the bootstrap distribution is then $H = -\sum_{i=1}^{n} f_i \log(f_i)$ and the maximum frequency is $\max_{i \in \{1,\ldots,n\}} f_i$.

## 3.5 Results

We test SA-ccsmVISDA and the node bandwidth constrained stability analysis based tree learning method on two gene expression datasets. The MIT human cancer dataset consists of 144 training samples and 54 testing samples in 16063 genes from 14 human cancers (Ramaswamy, et al., 2001). We test our methods on the combined set of training and testing samples, where the 8 metastatic samples are removed. The muscular dystrophy dataset, provided by Children National Medical Center (CNMC), Center for Genetic Medicine, consists of 108 samples with 11252 genes from 9 diagnostic groups of muscular dystrophies. The name and the number of samples of

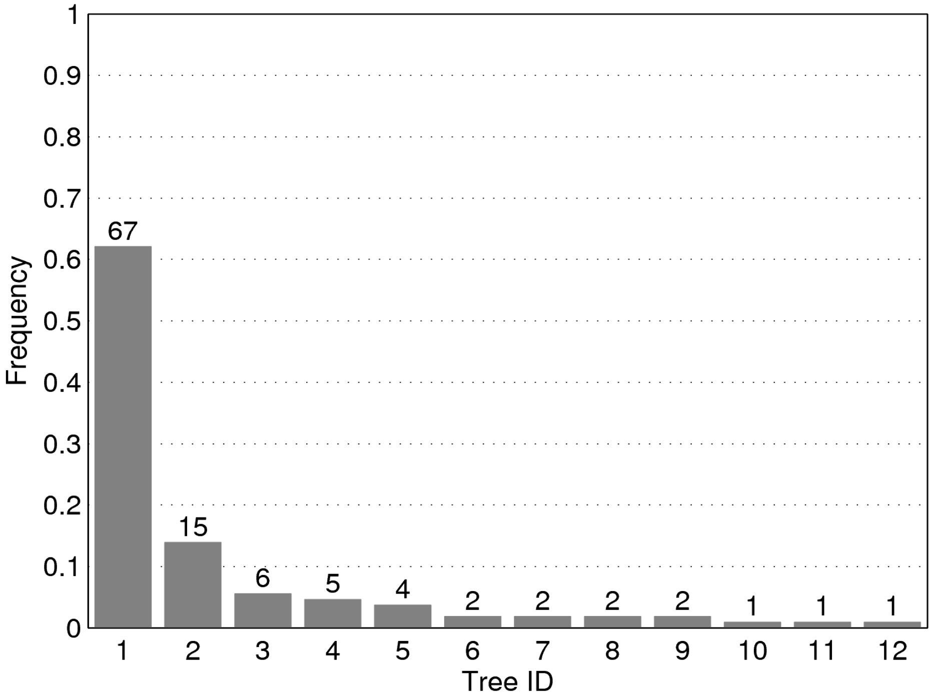


Figure 3.3. The empirical distribution of the tree structures learned by SA-ccsmVISDA for the muscular dystrophy dataset. The abscissa is the index of the tree structures in descending order of frequencies, and the ordinate is the frequency.

each group are: amyotrophic lateral sclerosis (*ALS*, *n*=9); acute quadriplegic myopathy (*AQM*, *n*=5); calpain III deficiency (*Calpain3*, *n*=10); Duchenne muscular dystrophy (*DMD*, *n*=10); dysferlin deficiency (*Dysferlin*, *n*=10); fukutin related protein deficiency (*FKRP*, *n*=7); fascioscapulohumeral dystrophy (*FSH*, *n*=14); normal human muscle (*NHM*, *n*=18); and juvenile dermatomyositis (*JDM*, *n*=25).

### 3.5.1 Learning Tree of Phenotypes Using SA-ccsmVISDA

In the experiments, we applied SA-ccsmVISDA on the muscular dystrophy dataset to generate the tree of 9 subtypes of muscular dystrophies and on the MIT cancer dataset to generate the tree of 14 human cancers. For each dataset, we also evaluated the performance of the tree classifiers built on the tree of phenotypes.

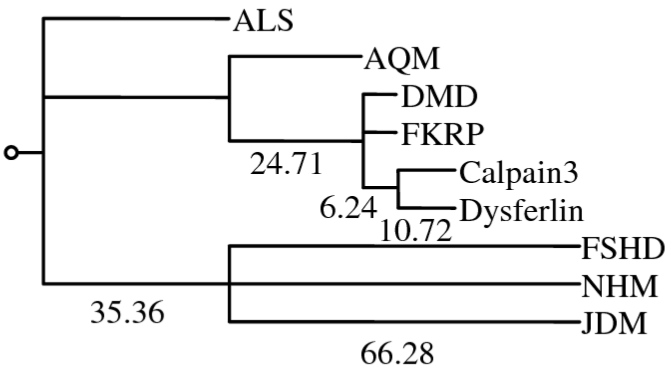


Figure 3.4. The structure of the winning tree for the muscular dystrophy dataset. The subtypes of muscular dystrophies are shown on the terminal nodes. The separability measures are shown along the edges.

### *3.5.1.1 Muscular Dystrophy Dataset*

We applied SA-ccsmVISDA on the muscular dystrophy data and derived 108 leave-one-out trees showing 12 different structures. The empirical distribution of the trees is shown in Figure 3.3. The entropy calculated using (3.3) is 1.4208. The frequency of the winning tree is $67/108 \approx 0.62$.

The structure of the winning tree is shown in Figure 3.4 with subtype names on the terminal nodes. The winning tree is supported by many known clinical, genetic, and histological features of these disorders. ALS is the only denervating disorder, due to die-back of motor neurons. A number of the muscular dystrophies are caused by abnormalities in the plasma membrane of the muscle fiber: Calpain3, DMD, Dysferlin and FKRP are all such membrane dystrophies, and all group together. FSHD is a unique disorder due to a heterozygous deletion in chromosome 4q, and by our SA-ccsmVISDA approach this maps distinctly, as does normal human muscle (NHM) and an autoimmune disease, JDM.

Based on the winning tree structure, we built hard and soft tree classifiers using the gene subspace selection method (Shedden, et al., 2003) described in 3.3.3 and OVR-MSVM as the node classifiers. The gene expression values were transformed by $\log(x)$ before evaluating the

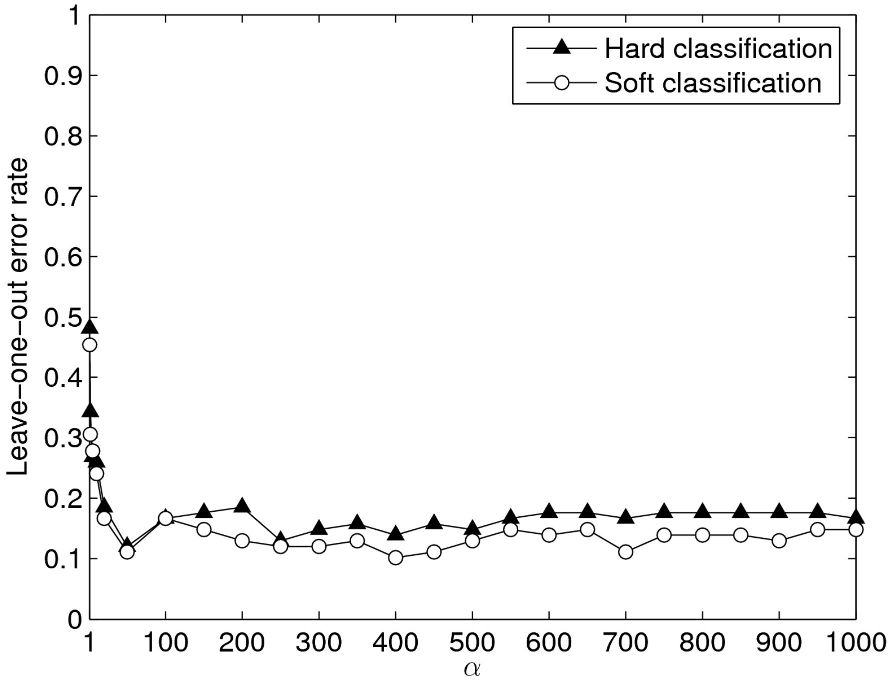


Figure 3.5. Leave-one-out error rates of hard and soft classifiers on the muscular dystrophy dataset for $C = 1.0$.

Table 3.1. Leave-one-out error rates (in percentage) for hard and soft classification schemes on the muscular dystrophy dataset. The numbers in parentheses are the values of $\alpha$. The lowest error is in bold face.

| $C$ | 0.001 | 0.01 | 0.1 | 1.0 | 10.0 |
|---|---|---|---|---|---|
| Hard | 23.15 (800) | 12.96 (250) | 12.04 (50) | 12.04 (50) | 12.04 (50) |
| Soft | 12.04 (300) | 11.11 (400) | 11.11 (300) | **10.19** (**400**) | 11.11 (50) |

classifiers. In OVR-MSVM we use linear SVMs as the binary components. The complexity of a linear SVM can be controlled by a penalty value, $C$. We use the same value of $C$ for each linear SVM in each of the OVR-MSVMs. We tested 5 values for $C$: 0.001, 0.01, 0.1, 1.0 and 10.0. In order to determine an optimal subspace size, for each $C$ value we tested 25 different values for $\alpha$: 1, 2, 5, 10, 20 and the sequence $\{50k, k = 1, \ldots, 20\}$. All these tree classifiers are evaluated by leave-one-out cross validation. In Table 3.1, we list the lowest error rates (in percentage) for both soft and hard classification schemes for each value of $C$. The values in parentheses are the values of $\alpha$ at which the performances are achieved. In Figure 3.5, we plot the leave-one-out

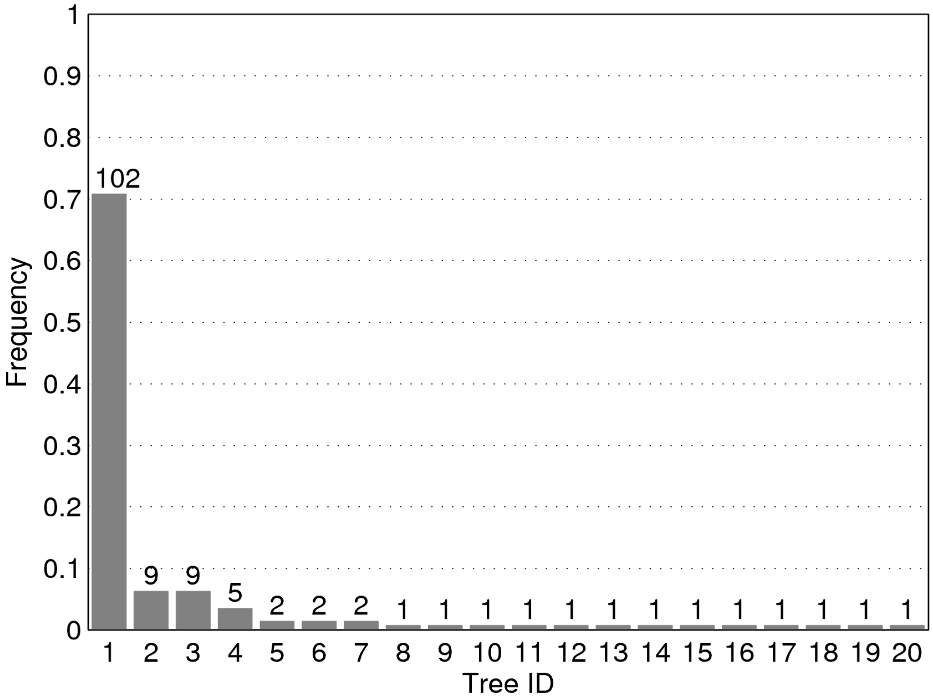


Figure 3.6. The empirical distribution of the tree structures learned by SA-ccsmVISDA for the MIT cancer dataset.

error rates of soft and hard classification as functions of $\alpha$ for $C = 1.0$. It can be seen that soft classification improves the performance for most cases. The lowest error rate is 10.19% when $C = 1.0$ and $\alpha = 400$.

### *3.5.1.2 MIT Cancer Dataset*

We applied SA-ccsmVISDA on the training set of the MIT cancer dataset and generated 144 trees in the leave-one-out loop. The 144 trees demonstrate 20 different structures, and their empirical distribution is shown in Figure 3.6. The entropy of the empirical distribution is 1.3344. The frequency of the winning tree is $102/144 \approx 0.71$. The structure of the winning tree with the cancer types shown on the terminal nodes is illustrated in Figure 3.7.

Using the same scheme as for the muscular dystrophy dataset, we evaluated the hard and soft tree classifiers for the MIT cancer data. Before the evaluation, the gene expression values were transformed and the genes were filtered as in (Shedden, et al., 2003). The lowest leave-one-

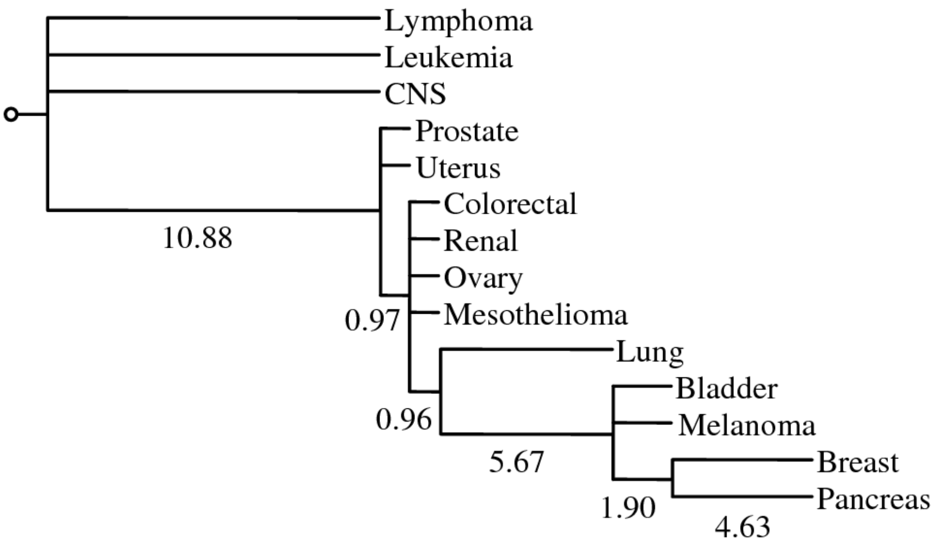


Figure 3.7. The structure of the winning tree for the MIT cancer dataset. The cancer types are shown on the terminal nodes. The separability measures are shown along the edges.

out error rates for different values of $C$ are listed in Table 3.2. The curves of leave-one-out errors of soft and hard classifications as functions of $\alpha$ are illustrated in Figure 3.8. The highest classification accuracy of our tree classifiers is 87.5% when $C = 1$ and $\alpha = 200$. Our results compare favorably with those of Ramaswamy et al (2001), who used a parallel OVR-MSVM to classify the 14 cancers and achieved a 78% accuracy on the training set of 144 samples using leave-one-out cross validation.

Table 3.2. Leave-one-out error rates (in percentage) for soft and hard classification schemes on the MIT cancer data. In the parentheses are the values of $\alpha$. The lowest error is in bold face.

| $C$ | 0.001 | 0.01 | 0.1 | 1.0 | 10.0 |
|---|---|---|---|---|---|
| Hard | 21.53 (450) | 16.67 (200) | 15.97 (200) | 15.97 (200) | 15.97 (200) |
| Soft | 16.67 (250) | 13.19 (250) | 13.19 (200) | **12.50** (**200**) | 13.19 (200) |

As a means to explore the biological implications of our solution, we compared our tree to Shedden et al's tree based on pathologic and ontologic knowledge (Shedden, et al., 2003). Our solution has notable similarities to this tree, consistently classifying lymphoma, leukemia, CNS and epithelial cancers into groups in which lymphoma and leukemia are closely related and CNS

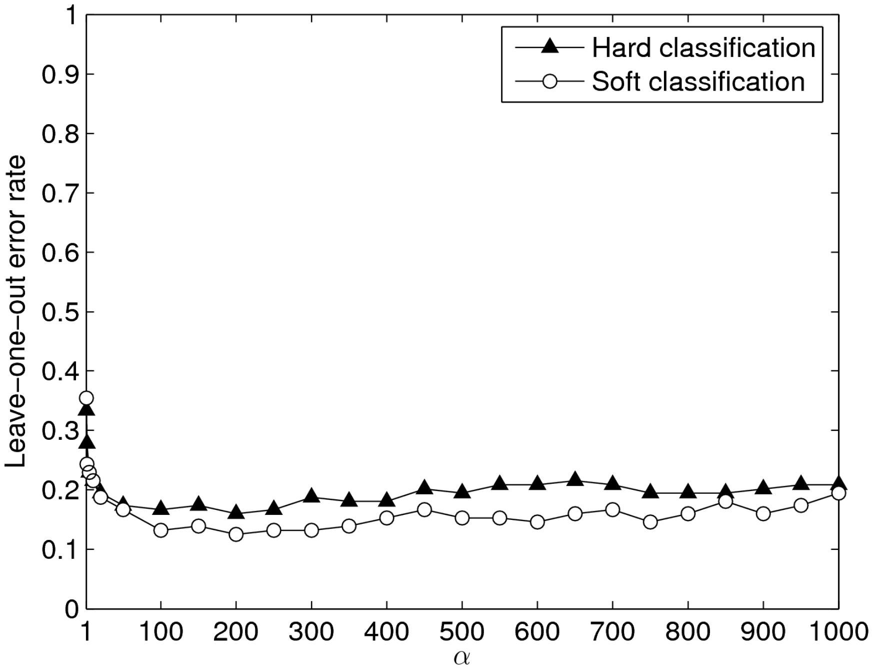


Figure 3.8. Leave-one-out error rates of the hard and soft classifiers on MIT cancer data for $C = 1.0$.

and epithelial cancers are closely positioned. Cancers of the uterus, breast, lung, colon, bladder, kidney and pancreas are consistently and appropriately classified into the same group. A major difference between the two trees is the location of the mesotheliomas (a rare cancer) and melanomas. Applying an independent data-driven approach to the same dataset, Tibshirani and Hastie (2007) generated a tree broadly similar to ours, in which the melanomas and mesotheliomas also clustered differently than predicted by the Shedden et al's construct. Further analysis of the similarities among the melanomas, mesotheliomas and their most closely related cancers may generate new insights into common molecular functions among these cancers.

### 3.5.2 Learning Tree of Phenotypes Using Node-Bandwidth Constrained Stability Analysis

We demonstrate our node bandwidth constrained stability analysis based tree learning method using the MIT cancer dataset and the muscular dystrophy dataset. For each dataset we create 500 bootstrap samples and apply the four hierarchical group clustering methods to generate bootstrap distributions of trees. These trees are then merged with different node bandwidths to generate new distributions of non-binary trees. For each particular combination of HGC method and node bandwidth, we characterize the tree distribution by the number of unique tree structures, the maximum relative frequency among all the trees, and the entropy of the distribution.

*3.5.2.1 MIT Cancer Dataset*

We test our method using the combined training and testing set consisting of 190 samples for the MIT cancer dataset. The data is preprocessed as in (Shedden, et al., 2003). There are totally 7095 genes left after the preprocessing. We apply each HGC method on the bootstrap samples to generate a set of trees and merge them under different node bandwidth $F$ . The properties of the tree distributions are plotted in Figure 3.9. Due to the small sample size, almost all the bootstrap samples yield different tree structures before controlling the node bandwidth. With the increase of the node bandwidth, the variability of the merged tree structures decreases.

We can see that the SVM margin based complete linkage method outperforms other HGC methods in terms of the stability. We plot the tree structures with the maximum relative frequencies under different node bandwidths in Figure 3.10. As claimed in (Tibshirani and Hastie, 2007), the complete linkage margin trees usually have balanced structures. We can see that the sub-structure composed by phenotypes *Leukemia*, *Lymphoma*, and *CNS*, and the global structure of these three phenotypes versus all other solid tumors (the top level binary split) are maintained throughout all different node bandwidths. Our method preserves these highly resolved, highly stable local structures.

Note that the trees of node bandwidth 2, 3, and 4 are not compatible with each other in their deeper level sub-structures. This is due to the fact that there are many different tree structures under these node bandwidths and none of them receives significantly larger counts compared with the other structures; therefore a tree could become the mode of the bootstrap distribution purely by chance. This reveals the stability issue in those HGC methods without properly controlled complexities: a tree learning algorithm creating trees with high structural complexity (small node bandwidth) will highly probably generate a non-reproducible tree structure given data with small sample sizes. When we increase the node bandwidth, more stable structure will emerge. For example, the local structure of the group of *Pancreas*, *Breast*, *Bladder*, and *Melanoma*, versus the group of *Colorectal*, *Lung*, *Uterus*, *Ovary*, *Renal*, *Mesothelioma*, and *Prostate* occurs in the winning tree structures at node bandwidths 6, 7, 8, and 9. The reason that multiple node bandwidths can share the same tree structure is that the stability of the tree distribution will not be increased by allowing splits with more child nodes, and we prefer to preserve structures with higher complexity/resolution with the same stability. From the figures we can see that at the node bandwidth $F = 11$, the winning tree has a significantly larger frequency compared with the other structures, and the entropy of the tree distribution also drops drastically at $F = 11$. Hence using the distribution characteristics as the criteria for model selection, we will choose node bandwidth 11 to generate the best tree structure for the margin based complete linkage HGC embedded in stability analysis.

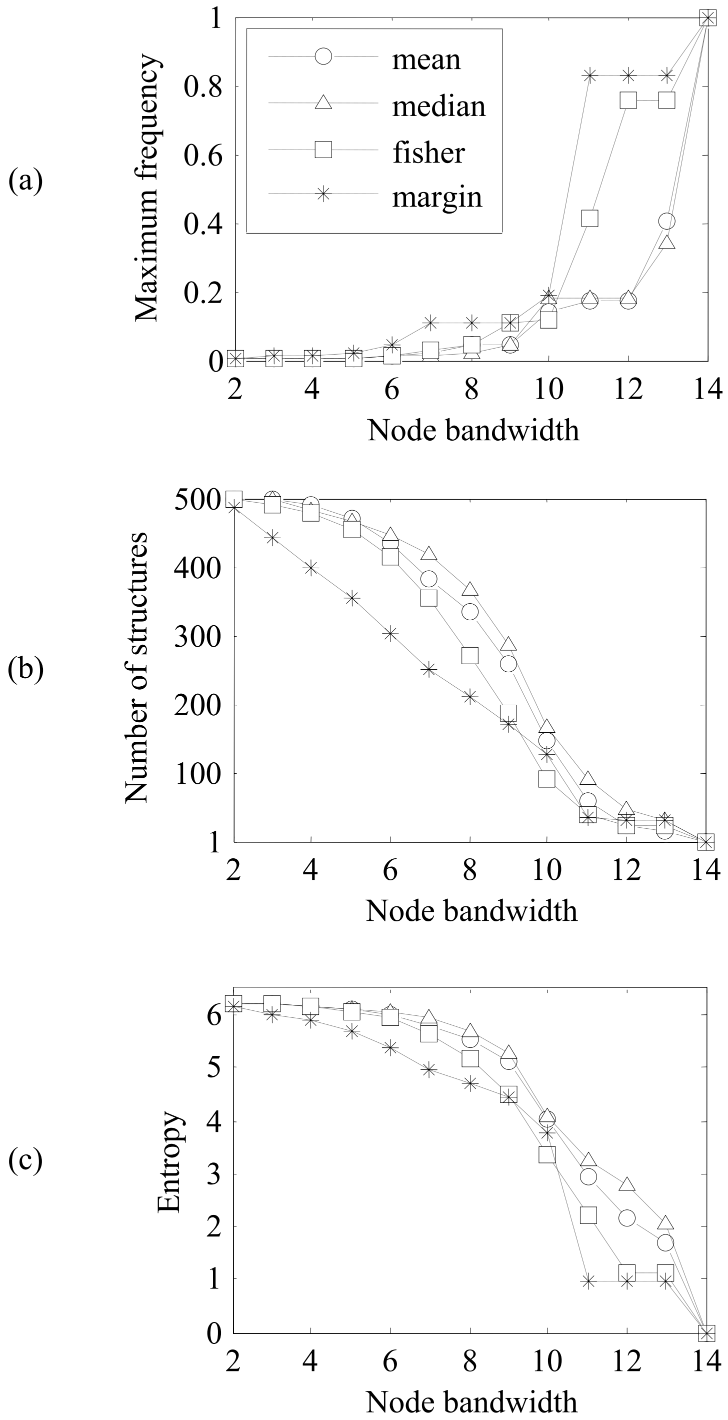


Figure 3.9. Characteristics of the bootstrap distributions of the HGC methods under different node bandwidths (from 2 to 14) for MIT cancer dataset. (a) the maximum relative frequency of all tree structures; (b) the number of unique tree structures; (c) the entropy of the bootstrap distribution.

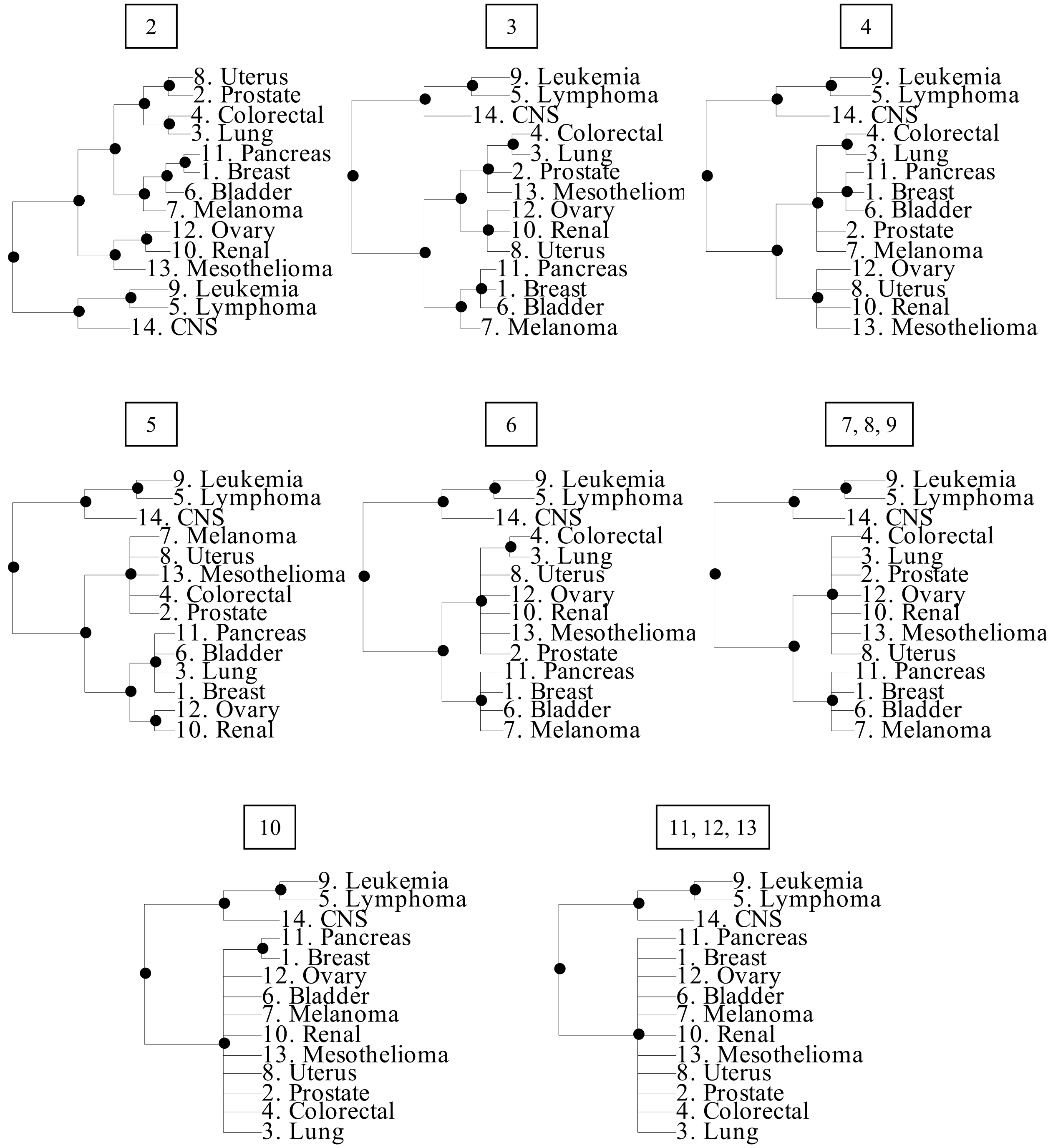


Figure 3.10. The tree structures with the highest relative frequencies generated by the SVM margin based HGC method at different node bandwidths for MIT cancer dataset. The numbers in the box on the top of each structure are the node bandwidths. Different node bandwidths may yield the same tree structure.

### *3.5.2.2 Muscular Dystrophy Dataset*

Following the same procedure as used for MIT cancer dataset, we first generate 500 bootstrap samples and apply the HGC methods to generate bootstrap distributions of trees. These trees are then subjected to stability analysis based tree merging, controlled by the node bandwidth $F \in \{2,\ldots,9\}$. In Figure 3.11, we plot the number of unique tree structures, the maximum relative frequencies, and the entropy of the bootstrap distribution as functions of the node bandwidth.

Again, the margin based complete linkage method stands out with good stability properties. But when the node bandwidth is greater than or equal to 7, the mean-distance based HGC method yields higher maximum relative frequencies. We plot the winning tree structures generated by this approach in Figure 3.12. Note that in the tree structure corresponding to node bandwidth 5, *FSH* is peeled off in the first level, which is different from the structures corresponding to node bandwidths 4 and 6. By examining the frequencies of trees, we found that the winning tree structure at node bandwidth 5 has the second largest frequency at node bandwidth 4 and 6. These two groups of structures dominate the tree frequencies.

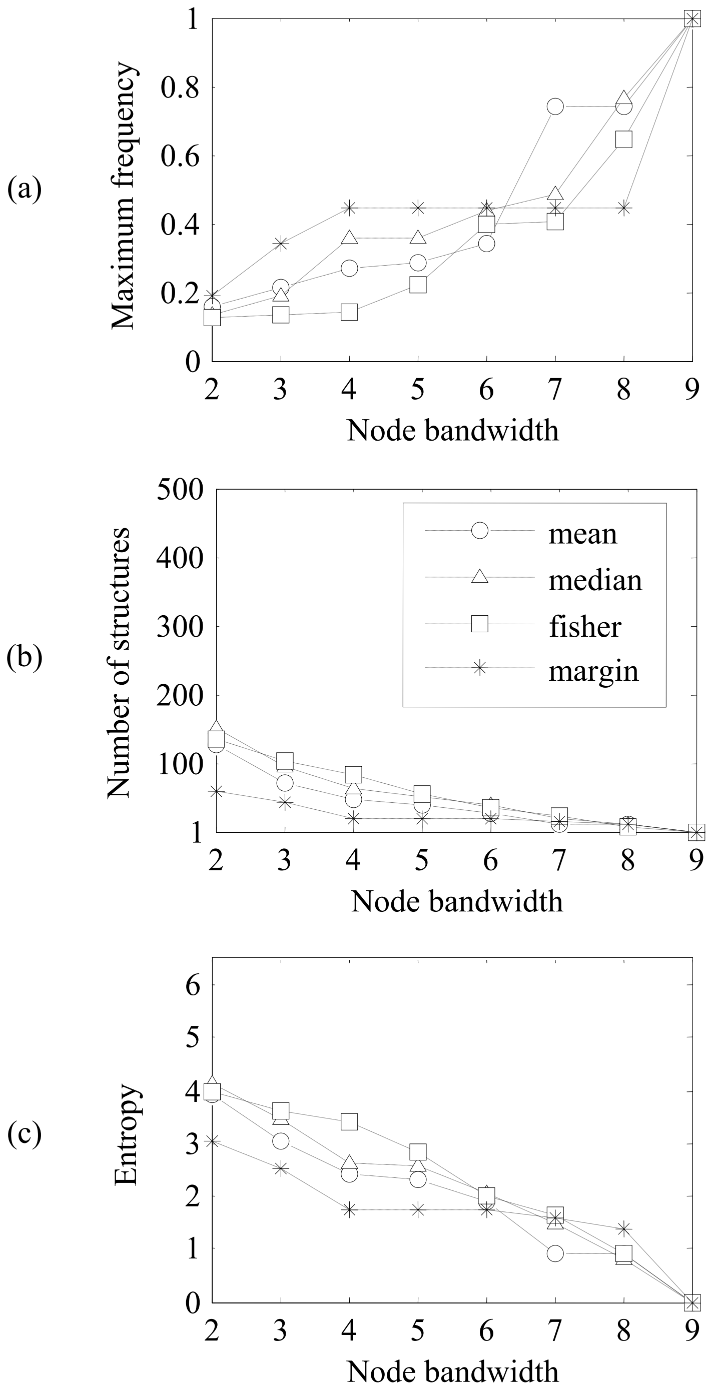


Figure 3.11. Characteristics of the bootstrap distributions of the tree structures under different node bandwidths (from 2 to 9) for the muscular dystrophy dataset. (a) the maximum relative frequency; (b) the number of unique tree structures; (c) the entropy of the tree distribution.

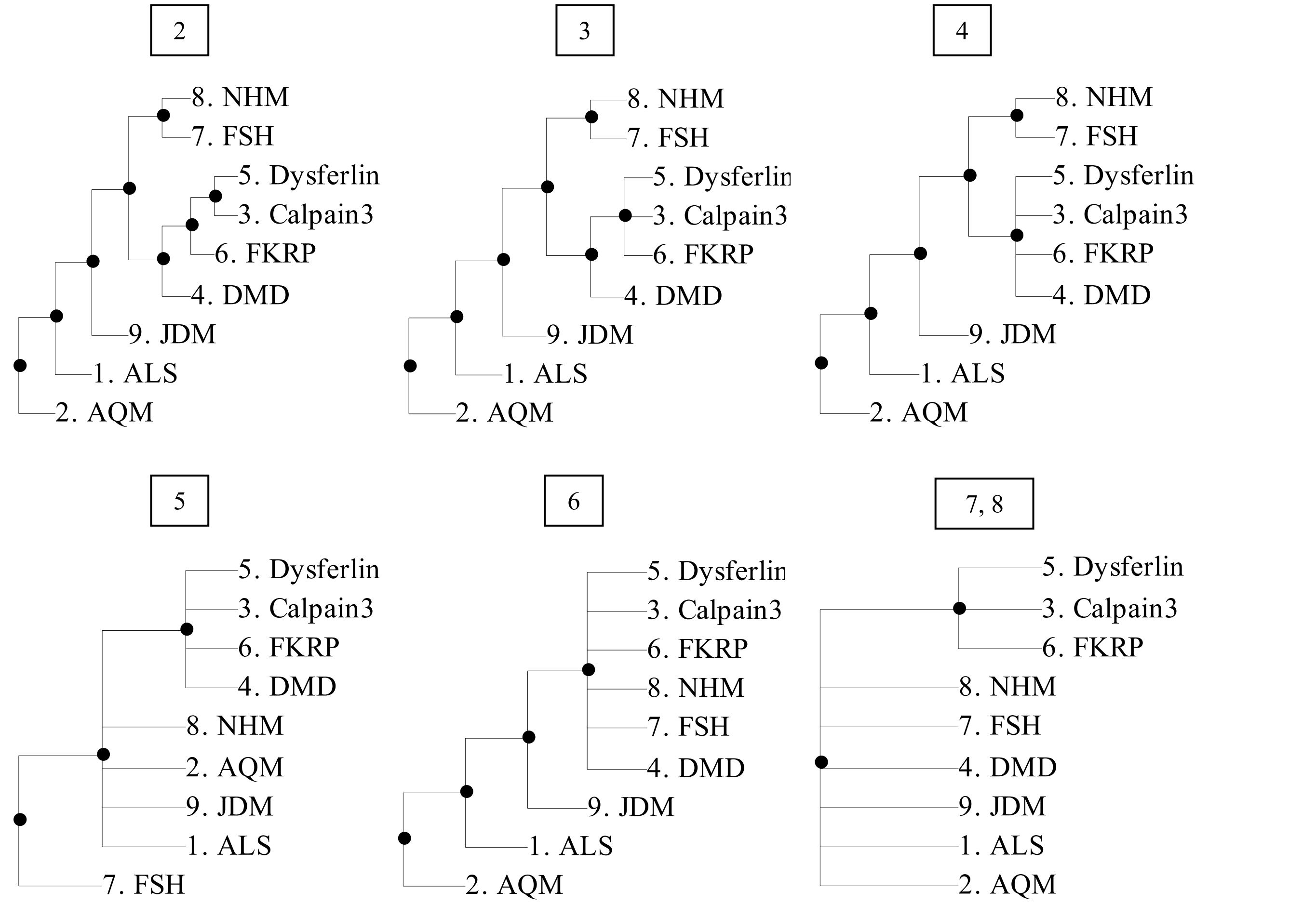


Figure 3.12. The tree structures with the highest frequencies generated by the mean-distance based HGC method at different node bandwidths for muscular dystrophy dataset. The numbers in the box on the top of each structure are the node bandwidths. Different node bandwidths may yield the same tree structure.

## 3.6 Conclusions and Discussions

In this chapter, we try to address the problem of learning the tree of phenotypes using multiclass, small sample size, high-dimensional microarray gene expression data. The purpose of learning TOP is to discover the latent relationships between multiple diseases/disease subtypes in molecular level data and represent them in a highly reproducible, coarse-to-fine tree structure of phenotypes. Though existing knowledge can be used to construct TOP, the resolutions of such trees are largely confined by the completeness of the prior knowledge. Furthermore, the prior knowledge may not be consistent with the latent structures embedded in molecular level data.

Learning TOP from gene expression data can help to improve our understanding about the relationships between heterogeneous diseases.

In order to use the learned TOP as hypotheses for analyzing the relationships between diseases, it is important for a TOP learning method to generate reproducible tree structures given different realizations of a data distribution. Given small sample size, high dimensional gene expression data, a tree learning method can easily create highly variable tree structures. In order to suppress such variability, we can purposely perturb a given dataset to generate artificial replicates of the original data distribution, and observe and control the variation of the outcomes of the tree learning method.

We propose two methods to learn TOP from gene expression data. The first method, SA-ccsmVISDA, is a stability analysis based human interactive visualization approach. SA-ccsmVISDA creates a hierarchy of phenotypes in a top-down fashion. At each node during tree construction, the data of involved phenotypes is projected into a 2-D space selected by Fisher's discriminant analysis. The user initializes the partitioning of phenotypes into groups by providing the number of groups and the center of each group. The partition is iteratively updated by an EM-like procedure, which guarantees that samples from the same phenotype is exclusively assigned to a single group, until a stable state of the partition is reached. Based on this partition, child nodes corresponding to phenotypic groups are grown under the current node. The same tree growing procedure is recursively applied on each new child node, until each phenotypic class becomes a single group, which become terminal nodes of the phenotypic tree. The tree building procedure is embedded in leave-one-out resampling of the original dataset: for each leave-one-out dataset, a TOP is built. The final outcome of SA-ccsmVISDA is the tree structure occurs with the highest frequency in all the leave-one-out trees. There are several unique features associated with SA-ccsmVISDA: (a) the method provides a coarse-to-fine decomposition of the mixture of heterogeneous phenotypes; (b) it allows domain experts to incorporate their knowledge in the tree structures; (c) by incorporating a leave-one-out stability analysis, the method generates a highly reproducible TOP that can be reliably used in hypothesis-driven scientific discoveries. We applied SA-ccsmVISDA on two gene expression datasets of human diseases. The TOP learned on the muscular dystrophy dataset is supported by various clinical, genetic, and histological knowledge of the muscular dystrophy disorders. The TOP learned on the MIT human cancer datasets has local structures consistent with Shedden et al's pathological tree (Shedden, et al., 2003); it also contains structures that are not shown in Shedden et al's tree, which can potentially provide new insights about the related cancer phenotypes. Based on the learned tree structure, we can perform various down-stream analyses. For example, we built hierarchical classifiers using the learned tree structures and achieved good prediction accuracies on the two datasets of human diseases.

SA-ccsmVISDA is an effective tool to incorporate domain knowledge into tree structures given a limited size of samples. When the prior knowledge is not available, or the sample size makes it impractical to apply ccsmVISDA in the leave-one-out stability analysis, we need a fully automatic learning method to generate highly reproducible phenotypic trees. Conventional tree learning algorithms, such as CART and HC, generate highly variable tree structures on different replicates of the same data distribution. Considering the tree structures as the outcomes, we can control the tree learning procedure in the perspective of model selection in order to balance the generalizability and complexity of the tree structures. We propose a node bandwidth constrained stability analysis based TOP learning method. The method consists of three components: (a) the hierarchical group clustering method for generating binary phenotypic trees; (b) a merging procedure for modifying the binary tree structures learned by HGC in order to achieve certain node bandwidths; (c) a bootstrap stability analysis wrapper for estimating the distribution of trees and selecting appropriate node bandwidths. By controlling the node bandwidth, we try to balance the generalizability and complexity of the tree structures. The generalizability of a tree structure is reflected by its frequency of occurrence, or reproducibility, in the stability analysis. The final outcome of the method is the tree structure with the highest frequency at an appropriately selected node bandwidth based on the characteristic curves of the bootstrap tree distributions. We should note that there can be numerous alternative methods for each of the three components of our method. For example, the components of HGC learning and node bandwidth controlling can be replaced by more sophisticated methods that search the optimal local/global tree structure at a given node bandwidth. The value of our proposed work is to consider the tree structure as an important outcome of learning (rather than a by-product of training tree classifiers), parameterize the tree complexity, and learn the tree structures under the framework of model selection using stability analysis.

# 4 Summary and Future Works

In this dissertation, we have developed computational methods for analyzing DNA copy number changes and learning the tree of phenotypes using DNA microarray data. In this chapter, we summarize our contributions and discuss possible future developments.

## 4.1 Summary of Contributions

### 4.1.1 Computational Analysis of DNA Copy Number Changes

We propose the Fused Margin Regression method for detecting (1) copy number changes in a single signal profile and (2) consensus copy number changes in population data. In contrast to most existing methods that can handle only one of the two detection problems, FMR uses a unified optimization model for both problems. FMR applies the variable fusion constraint (Land and Friedman, 1996) to enforce sparse solutions of the breakpoints and the $\epsilon$-insensitive loss function to generate robust estimations of copy numbers for signal profiles with low SNRs and complex copy number patterns. We compare FMR with several widely used existing methods on systematically constructed simulations and observed robust and competitive performance under different noise distributions and SNRs. The applications of FMR on real CGH and SNP array copy number datasets further demonstrate the applicability of FMR to help scientific discoveries.

High density CGH and SNP microarrays can generate a significant amount of copy number measurements. To make FMR applicable to high-throughput data, we adapt the solution-path algorithm to solve the computation-intensive optimization problem associated with FMR. The experimental results show that, equipped with the path algorithm, FMR can efficiently detect copy number changes in potentially all existing high-density microarrays.

In addition, we also propose a chromosome instability measure base on the copy number segments detected by FMR. We apply the CIN measure on an ovarian cancer copy number dataset and observed its capability in distinguishing CIN patterns between different subtypes of ovarian serous carcinomas and the normal samples.

### 4.1.2 Learning Tree of Phenotypes

We propose the SA-ccsmVISDA method to learn the tree of phenotypes using microarray gene expression data. The inner part of the method is a supervised-mode extension of the VISDA algorithm, which allows the users to interactively initialize the clusters during hierarchical decomposition of phenotypes. The outer loop of the method is a leave-one-out stability analysis, which is used to select a stable tree among all the tree structures learned from leave-one-out resampling of the original dataset.

We also propose a fully automatic method to learn the tree of phenotypes. We propose the concept of tree-node bandwidth and use it as a parameter to control the trade-off between the descriptive power and stability of the learned trees. We use a bootstrap based stability analysis to generate stable tree structures with different complexities: for each bootstrap sample, we first use a hierarchical group clustering method to generate a binary tree, and then apply a top-down merging procedure to modify the binary tree structures in order to achieve a preferred node bandwidth. For a particular node bandwidth, we select the most stable tree structure as the output among all trees learned from the bootstrap samples.

We test both tree learning methods on a muscular dystrophy gene expression dataset and a multiclass human cancer gene expression dataset. The results show that using the proposed methods, we can discover stable yet highly resolved tree structures from small sample size, high-dimensional microarray gene expression data.

## 4.2 Future Works

### 4.2.1 Future Works on DNA Copy Number Data Analysis

The experimental results in Chapter 2 have shown that FMR is an efficient and robust copy number change detection methods for high density CGH and SNP microarray data. Here we discuss several possible future works that further improve the FMR method or use the outcome of FMR to perform down-stream analyses.

In most CGH and SNP array platforms, the probes are not evenly distributed across each chromosome, i.e. in some chromosomal regions the probes are denser than those in other regions. Some copy number change detection methods incorporate the genomic distance between probes into their models by very intuitive approaches (Marioni, et al., 2006; Stjernqvist, et al., 2007), and there are no concrete experimental evidence showing that those approaches indeed improve the detection results. We believe that using the distance information should be based on biological insights rather than simple intuitions. We can further explore this issue to improve the performance of FMR.

A tumor tissue sample is usually a mix of normal stromal cells and tumor cells at different stages of development. It is reasonable to assume that (1) there are several cell populations in the sample; (2) within each population the copy number genotypes of the cells are largely the same; (3) each copy number alteration detected in the tissue sample is caused by the copy number alteration(s) in one or more cell populations; (4) the copy numbers in different cell populations are in the same discrete state space. Based on these assumptions, it is possible to define a computational model to decompose the copy number measurements on a tumor sample to the copy number genotypes of multiple cell populations.

In the experiments of Chapter 2, we have shown that the CIN index exhibits distinct patterns on different subtypes of ovarian serous carcinomas. Considerable amount of research has been conducted on gene expression based diagnosis/prognosis for various human diseases. It is intriguing to see whether copy number data can be used for the same purpose. Copy number data usually contains much more features ($10^5$ to $2\times10^6$) than gene expression data ($10^4$ to $3\times10^4$), and those features are spatially correlated. The supervised feature selection methods designed for gene expression data may not be directly applicable to copy number data for dimensionality reduction. Using summary statistics such as the CIN index can circumvent this problem. Given copy number and gene expression data measured on the same subjects as in the TCGA project (The Cancer Genome Atlas Research Network, 2008), we can compare the performances between the prediction models built on CIN indices and those built on gene expressions. As an even further step, we can explore using gene expression and copy number data jointly for diagnosis/prognosis.

### 4.2.2 Future Works on Learning Tree of Phenotypes

Learning tree of phenotypes using genomic data is a novel concept that has not been explicitly addressed before in bioinformatics, pattern recognition, and machine learning research. The methods proposed in the dissertation can be improved in many aspects. For example, we use a single feature (gene) space in all stages of the tree learning. In the future work, we can explore the impact of feature selection on the stability of learned trees.

In SA-ccsmVISDA, we build a non-binary tree structure using hierarchical and divisive visualization and clustering. In the node bandwidth constrained tree learning method, we derive non-binary trees by modifying binary tree structures. Both methods use stability analysis to select an established tree structure as the output of tree learning. We have been exploring a novel method that incorporates stability analysis and non-binary split selection in a top-down tree growing procedure. The method generates a unique, potentially non-binary tree with stable local structures (splits) on a given dataset. The basic idea can be described as follows. During the process of top-down tree growing, at each terminal node with $K > 1$ classes (phenotypes), we first generate $N_{\mathrm{B}}$ bootstrap samples. For each bootstrap sample, we use a partitioning algorithm, for instance hierarchical group clustering (3.4.1), to generate partitions $P_1^{(k)},\ldots,P_{N_\mathrm{B}}^{(k)}$ for each split size $k\in\{2,\ldots,K\}$. Suppose there are $M\le N_\mathrm{B}$ unique partition structures in $\{P_i^{(k)}\}_{i=1}^{N_\mathrm{B}}$ and denote those structures and their relative frequencies by $\hat{\mathcal{P}}^{(k)}=\{\hat{P}_1^{(k)},\ldots,\hat{P}_M^{(k)}\}$ and $\hat{\mathbf{f}}^{(k)}=\{\hat{f}_1^{(k)},\ldots,\hat{f}_M^{(k)}\}$, respectively. The stability of partition size $k$ is measured by

$$s_K(k)=\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\left|\hat{p}(i,j)-p_0(K,k)\right|,$$

where $\hat{p}(i,j)$ is the relative frequency that classes $i$ and $j$ are grouped together in all the $N_\mathrm{B}$ partitions; $p_0(K,k)$ is the expected probability that any two classes are grouped together if the $K$ classes are randomly (uniformly) assigned to $k$ groups, i.e., there is no structure in the data at all. It can be shown that

$$p_0(K,k)=S(K-1,k)/S(K,k),$$

where $S(a,b)$ is the *Sterling number of the second kind.* The purpose of subtracting $p_0(K,k)$ from $\hat{p}(i,j)$ is to remove the intrinsic bias associated with $k$ in using $\hat{p}(i,j)$ as a stability measure. The stable split size $k^*$ is selected by

$$k^*=\arg\max_k\{s_K(k)\},$$

and the optimal partition $\hat{P}_m^{(k^*)}$ is the partition of split size $k^*$ with the highest relative frequency,

$$m=\arg\max_{i\in\{1,\ldots,M\}}\{\hat{f}_i^{(k^*)}\}\ .$$

According to $\hat{P}_m^{(k^*)}$, we create child nodes for the current node, where each child node corresponds to a group in the partition. We perform stability analysis on each child node to generate deeper levels of the tree. The tree growing procedure is recursively executed until each terminal node contains only one class.

# Bibliography

D.G. Albertson, C. Collins, F. McCormick and J.W. Gray (2003) Chromosome aberrations in solid tumors, *Nat Genet*, **34**(4):369-376.

R. Andersson, C.E. Bruder, A. Piotrowski, U. Menzel, H. Nord, J. Sandgren, T.R. Hvidsten, T. Diaz de Stahl, J.P. Dumanski and J. Komorowski (2008) A segmental maximum a posteriori approach to genome-wide copy number profiling, *Bioinformatics*, **24**(6):751-758.

M. Bakay, Z. Wang, G. Melcon, L. Schiltz, J. Xuan, P. Zhao, V. Sartorelli, J. Seo, E. Pegoraro, C. Angelini, B. Shneiderman, D. Escolar, Y.W. Chen, S.T. Winokur, L.M. Pachman, C. Fan, R. Mandler, Y. Nevo, E. Gordon, Y. Zhu, Y. Dong, Y. Wang and E.P. Hoffman (2006) Nuclear envelope dystrophies show a transcriptional fingerprint suggesting disruption of Rb-MyoD pathways in muscle regeneration, *Brain*, **129**(Pt 4):996-1013.

E. Ben-Yaacov and Y.C. Eldar (2008) A fast and flexible method for the segmentation of aCGH data, *Bioinformatics*, **24**(16):i139-145.

R. Beroukhim, G. Getz, L. Nghiemphu, J. Barretina, T. Hsueh, D. Linhart, I. Vivanco, J.C. Lee, J.H. Huang, S. Alexander, J. Du, T. Kau, R.K. Thomas, K. Shah, H. Soto, S. Perner, J. Prensner, R.M. Debiasi, F. Demichelis, C. Hatton, M.A. Rubin, L.A. Garraway, S.F. Nelson, L. Liau, P.S. Mischel, T.F. Cloughesy, M. Meyerson, T.A. Golub, E.S. Lander, I.K. Mellinghoff and W.R. Sellers (2007) Assessing the significance of chromosomal aberrations in cancer: methodology and application to glioma, *Proc Natl Acad Sci U S A*, **104**(50):20007-20012.

L.J. Billera, S.P. Holmes and K. Vogtmann (2001) Geometry of the space of phylogenetic trees, *Adv Appl Math*, **27**(4):733-767.

J.A. Bilmes (1998) A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. *ICSI Technical Report*. TR-97-021.

S. Boyd and L. Vandenberghe (2004) *Convex Optimization*. Cambridge University Press, Cambridge.

M. Bredel, C. Bredel, D. Juric, G.R. Harsh, H. Vogel, L.D. Recht and B.I. Sikic (2005) High-resolution genome-wide mapping of genetic alterations in human glial brain tumors, *Cancer Res*, **65**(10):4088-4096.

L. Breiman (1996) Bagging Predictors, *Machine Learning*, **24**(2):123-140.

L. Breiman, J. Friedman, C.J. Stone and R.A. Olshen (1984) *Classification and Regression Trees*. Chapman & Hall.

T.A. Brown (2007) *Genomes*. Garland Science, New York.

N.P. Carter (2007) Methods and strategies for analyzing copy number variation using DNA microarrays, *Nat Genet*, **39**(7 Suppl):S16-21.

E. Check (2005) Human genome: patchwork people, *Nature*, **437**(7062):1084-1086.

F.D. Ciccarelli, T. Doerks, C.v. Mering, C.J. Creevey, B. Snel and P. Bork (2006) Toward automatic reconstruction of a highly resolved tree of life, *Science*, **311**(5765):1283-1287.

S.J. Diskin, T. Eck, J. Greshock, Y.P. Mosse, T. Naylor, C.J. Stoeckert, Jr., B.L. Weber, J.M. Maris and G.R. Grant (2006) STAC: A method for testing the significance of DNA copy number aberrations across multiple array-CGH experiments, *Genome Res*, **16**(9):1149-1158.

R.O. Duda, P.E. Hart and D.G. Stork (2001) *Pattern Classification*. Wiley, New York.

B. Efron and R.J. Tibshirani (1993) *An Introduction to the Bootstrap*. Chapman & Hall, New York.

P.H. Eilers and R.X. de Menezes (2005) Quantile smoothing of array CGH data, *Bioinformatics*, **21**(7):1146-1153.

J. Felsenstein (2004) *Inferring Phylogenies*. Sinauer.

Y. Feng, Z. Wang, Y. Zhu, J. Xuan, D.J. Miller, R. Clarke, E.P. Hoffman and Y. Wang (2006) Learning the Tree of Phenotypes Using Genomic Data and VISDA. *Sixth IEEE Symposium on Bioinformatics and BioEngineering*.

L. Feuk, A.R. Carson and S.W. Scherer (2006) Structural variation in the human genome, *Nature Reviews Genetics*, **7**(2):85-97.

R. Fletcher (2000) *Practical Methods of Optimization*. Wiley.

K. Fukunaga (1990) *Introduction to Statistical Pattern Recognition*. Academic Press, Boston.

L. Gunter and J. Zhu (2007) Efficient Computation and Model Selection for the Support Vector Regression, *Neural Computation*, **19**(6):1633-1655.

T. Hastie, R. Tibshirani and J. Friedman (2001) *The elements of statistical learning : data mining, inference, and prediction*. Springer, New York.

S. Holmes (1999) Phylogenies: an overview. In Halloran, M.E. and Geisser, S. (eds), *IMA Series volume 112: Statistics and Genetics*. 81-119.

T. Huang, B. Wu, P. Lizardi and H. Zhao (2005) Detection of DNA copy number alterations using penalized least squares regression, *Bioinformatics*, **21**(20):3811-3817.

P. Hupé, N. Stransky, J.P. Thiery, F. Radvanyi and E. Barillot (2004) Analysis of array CGH data: from signal ratio to gain and loss of DNA regions, *Bioinformatics*, **20**(18):3413-3422.

A.J. Iafrate, L. Feuk, M.N. Rivera, M.L. Listewnik, P.K. Donahoe, Y. Qi, S.W. Scherer and C. Lee (2004) Detection of large-scale variation in the human genome, *Nat Genet*, **36**(9):949-951.

A.K. Jain, R.P.W. Duin and J. Mao (2000) Statistical pattern recognition: a review, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **22**(1):4-37.

J. Khan, J.S. Wei, M. Ringner, L.H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C.R. Antonescu, C. Peterson and P.S. Meltzer (2001) Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks, *Nat Med*, **7**(6):673-679.

S. Kim and A. Misra (2007) SNP genotyping: technologies and biomedical applications, *Annu Rev Biomed Eng*, **9**(289-320.

R. Koenker and G. Bassett (1978) Regression Quantiles, *Econometrica*, **46**(1):33-50.

K.T. Kuo, B. Guan, Y. Feng, T.L. Mao, X. Chen, N. Jinawath, Y. Wang, R.J. Kurman, M. Shih Ie and T.L. Wang (2009) Analysis of DNA copy number alterations in ovarian serous tumors identifies new molecular genetic changes in low-grade and high-grade carcinomas, *Cancer Res*, **69**(9):4036-4042.

K.T. Kuo, T.L. Mao, X. Chen, Y. Feng, K. Nakayama, Y. Wang, R. Glas, M.J. Ma, R.J. Kurman, M. Shih Ie and T.L. Wang (2010) DNA copy numbers profiles in affinity-purified ovarian clear cell carcinoma, *Clin Cancer Res*, **16**(7):1997-2008.

W.R. Lai, M.D. Johnson, R. Kucherlapati and P.J. Park (2005) Comparative analysis of algorithms for identifying amplifications and deletions in array CGH data, *Bioinformatics*, **21**(19):3763-3770.

S. Land and J. Friedman (1996) Variable fusion: a new method of adaptive signal regression. *Technical Report*. Department of Statistics, Stanford University.

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

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

Y. Li and J. Zhu (2007) Analysis of array CGH data for cancer studies using fused quantile regression, *Bioinformatics*, **23**(18):2470-2476.

J.C. Marioni, N.P. Thorne and S. Tavare (2006) BioHMM: a heterogeneous hidden Markov model for segmenting array CGH data, *Bioinformatics*, **22**(9):1144-1146.

S.A. McCarroll, F.G. Kuruvilla, J.M. Korn, S. Cawley, J. Nemesh, A. Wysoker, M.H. Shapero, P.I. de Bakker, J.B. Maller, A. Kirby, A.L. Elliott, M. Parkin, E. Hubbell, T. Webster, R. Mei, J. Veitch, P.J. Collins, R. Handsaker, S. Lincoln, M. Nizzari, J. Blume, K.W. Jones, R. Rava, M.J. Daly, S.B. Gabriel and D. Altshuler (2008) Integrated detection and population-genetic analysis of SNPs and copy number variation, *Nat Genet*, **40**(10):1166-1174.

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

G.J. McLachlan, K.-A. Do and C. Ambroise (2004) *Analyzing Microarray Gene Expression Data*. Wiley-Interscience, Hoboken, N.J.

K. Nakao, K.R. Mehta, J. Fridlyand, D.H. Moore, A.N. Jain, A. Lafuente, J.W. Wiencke, J.P. Terdiman and F.M. Waldman (2004) High-resolution analysis of DNA copy number alterations in colorectal cancer by array-based comparative genomic hybridization, *Carcinogenesis*, **25**(8):1345-1357.

A.B. Olshen, E.S. Venkatraman, R. Lucito and M. Wigler (2004) Circular binary segmentation for the analysis of array-based DNA copy number data, *Biostatistics*, **5**(4):557-572.

M.-Y. Park and T. Hastie (2005) *Hierarchical Classification using Shrunken Centroids*,Technical Report, Stanford University.

D. Pinkel and D.G. Albertson (2005) Array comparative genomic hybridization and its applications in cancer, *Nat Genet*, **37 Suppl**(S11-17.

J.C. Platt (1999) Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In Smola, A., Bartlett, P., Schölkopf, B. and Schuurmans, D. (eds), *Advances in Large Margin Classifiers*. MIT Press, Cambridge.

T. Poggio, R. Rifkin, S. Mukherjee and P. Niyogi (2004) General conditions for predictivity in learning theory, *Nature*, **428**(6981):419-422.

J.R. Pollack, T. Sorlie, C.M. Perou, C.A. Rees, S.S. Jeffrey, P.E. Lonning, R. Tibshirani, D. Botstein, A.L. Borresen-Dale and P.O. Brown (2002) Microarray analysis reveals a major direct role of DNA copy number alteration in the transcriptional program of human breast tumors, *Proc Natl Acad Sci U S A*, **99**(20):12963-12968.

S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C.H. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J.P. Mesirov, T. Poggio, W. Gerald, M. Loda, E.S. Lander and T.R. Golub (2001) Multiclass cancer diagnosis using tumor gene expression signatures, *Proc Natl Acad Sci U S A*, **98**(26):15149-15154.

R. Redon, S. Ishikawa, K.R. Fitch, L. Feuk, G.H. Perry, T.D. Andrews, H. Fiegler, M.H. Shapero, A.R. Carson, W. Chen, E.K. Cho, S. Dallaire, J.L. Freeman, J.R. Gonzalez, M. Gratacos, J. Huang, D. Kalaitzopoulos, D. Komura, J.R. MacDonald, C.R. Marshall, R. Mei, L. Montgomery, K. Nishimura, K. Okamura, F. Shen, M.J. Somerville, J. Tchinda, A. Valsesia, C. Woodwark, F. Yang, J. Zhang, T. Zerjal, L. Armengol, D.F. Conrad, X. Estivill, C. Tyler-Smith, N.P. Carter, H. Aburatani, C. Lee, K.W. Jones, S.W. Scherer and M.E. Hurles (2006) Global variation in copy number in the human genome, *Nature*, **444**(7118):444-454.

C. Rouveirol, N. Stransky, P. Hupe, P.L. Rosa, E. Viara, E. Barillot and F. Radvanyi (2006) Computation of recurrent minimal genomic alterations from array-CGH data, *Bioinformatics*, **22**(7):849-856.

K.A. Shedden, J.M. Taylor, T.J. Giordano, R. Kuick, D.E. Misek, G. Rennert, D.R. Schwartz, S.B. Gruber, C. Logsdon, D. Simeone, S.L. Kardia, J.K. Greenson, K.R. Cho, D.G. Beer, E.R. Fearon and S. Hanash (2003) Accurate molecular classification of human cancers based on gene expression using a simple classifier with a pathological tree-based framework, *Am J Pathol*, **163**(5):1985-1995.

R. Simon (2003) Diagnostic and prognostic prediction using gene expression profiles in high-dimensional microarray data, *British Journal of Cancer*, **89**(9):1599-1604.

A.J. Smola and B. Schölkopf (2004) A tutorial on support vector regression, *Statistics and Computing*, **14**(199-222.

A. Statnikov, C.F. Aliferis, I. Tsamardinos, D. Hardin and S. Levy (2005) A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis, *Bioinformatics*, **21**(5):631-643.

S. Stjernqvist, T. Ryden, M. Skold and J. Staaf (2007) Continuous-index hidden Markov modelling of array CGH copy number data, *Bioinformatics*, **23**(8):1006-1014.

The Cancer Genome Atlas Research Network (2008) Comprehensive genomic characterization defines human glioblastoma genes and core pathways, *Nature*, **455**(7216):1061-1068.

The Wellcome Trust Case Control Consortium (2010) Genome-wide association study of CNVs in 16,000 cases of eight common diseases and 3,000 shared controls, *Nature*, **464**(7289):713-720.

R. Tibshirani and T. Hastie (2007) Margin trees for high-dimensional classification, *J Mach Learn Res*, **8**(637-652.

R. Tibshirani, T. Hastie, B. Narasimhan and G. Chu (2002) Diagnosis of multiple cancer types by shrunken centroids of gene expression, *Proc Natl Acad Sci U S A*, **99**(10):6567-6572.

R. Tibshirani, M. Saunders, S. Rosset, J. Zhu and K. Knight (2005) Sparsity and smoothness via the fused lasso, *Journal of the Royal Statistical Society Series B*, **67**(1):91-108.

R. Tibshirani and P. Wang (2008) Spatial smoothing and hot spot detection for CGH data using the fused lasso, *Biostatistics*, **9**(1):18-29.

M.A. van de Wiel, F. Picard, W.N. van Wieringen and B. Ylstra (2010) Preprocessing and downstream analysis of microarray DNA copy number profiles, *Brief Bioinform*:bbq004.

V.N. Vapnik (1998) *Statistical Learning Theory*. John Wiley & Sons, New York.

E.S. Venkatraman and A.B. Olshen (2007) A faster circular binary segmentation algorithm for the analysis of array CGH data, *Bioinformatics*, **23**(6):657-663.

Y. Wang, L. Luo, M.T. Freedman and S.-Y. Kung (2000) Probabilistic Principal Component Subspaces: A Hierarchical Finite Mixture Model for Data Visualization, *IEEE Transactions on Neural Networks*, **11**(3):625-636.

Y. Wang, D.J. Miller and R. Clarke (2008) Approaches to working in high-dimensional data spaces: gene expression microarrays, *Br J Cancer*, **98**(6):1023-1028.

Z. Wang, Y. Wang, J. Lu, S.-Y. Kung, J. Zhang, R. Lee, J. Xuan, J. Khan and R. Clarke (2003) Discriminatory Mining of Gene Expression Microarray Data, *Journal of VLSI Signal Processing*, **35**(3):255-272.

S.J. White, L.E. Vissers, A. Geurts van Kessel, R.X. de Menezes, E. Kalay, A.E. Lehesjoki, P.C. Giordano, E. van de Vosse, M.H. Breuning, H.G. Brunner, J.T. den Dunnen and J.A. Veltman (2007) Variation of CNV distribution in five different ethnic populations, *Cytogenet Genome Res*, **118**(1):19-30.

H. Willenbrock and J. Fridlyand (2005) A comparison study: applying segmentation to array CGH data for downstream analyses, *Bioinformatics*, **21**(22):4084-4091.

L.Y. Wu, H.A. Chipman, S.B. Bull, L. Briollais and K. Wang (2009) A Bayesian segmentation approach to ascertain copy number variations at the population level, *Bioinformatics*, **25**(13):1669-1679.

B. Ylstra, P. van den Ijssel, B. Carvalho, R.H. Brakenhoff and G.A. Meijer (2006) BAC to the future! or oligonucleotides: a perspective for micro array comparative genomic hybridization (array CGH), *Nucleic Acids Res*, **34**(2):445-450.

J. Zhang, L. Feuk, G.E. Duggan, R. Khaja and S.W. Scherer (2006) Development of bioinformatics resources for display and analysis of copy number and other structural variants in the human genome, *Cytogenet Genome Res*, **115**(3-4):205-214.

Q. Zhang, L. Ding, D.E. Larson, D.C. Koboldt, M.D. McLellan, K. Chen, X. Shi, A. Kraja, E.R. Mardis, R.K. Wilson, I.B. Borecki and M.A. Province (2009) CMDS: a population-based method for identifying recurrent DNA copy number aberrations in cancer from high-resolution data, *Bioinformatics*, **26**(4):464-469.

P. Zhao, J. Seo, Z. Wang, Y. Wang, B. Shneiderman and E.P. Hoffman (2003) In vivo filtering of in vitro expression data reveals MyoD targets, *Comptes Rendus Biologies*, **326**(10):1049-1065.

X. Zhao, C. Li, J.G. Paez, K. Chin, P.A. Janne, T.H. Chen, L. Girard, J. Minna, D. Christiani, C. Leo, J.W. Gray, W.R. Sellers and M. Meyerson (2004) An integrated view of copy number and allelic alterations in the cancer genome using single nucleotide polymorphism arrays, *Cancer Res*, **64**(9):3060-3071.

J. Zhu, S. Rosset, T. Hastie and R. Tibshirani (2004) 1-norm Support Vector Machines. In Thrun, S., Saul, L. and Schölkopf, B. (eds), *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA.

Y. Zhu, H. Li, D.J. Miller, Z. Wang, J. Xuan, R. Clarke, E.P. Hoffman and Y. Wang (2008) caBIG VISDA: modeling, visualization, and discovery for cluster analysis of genomic data, *BMC Bioinformatics*, **9**(1):383.