

RICE UNIVERSITY

**Statistical Methods for Bioinformatics: Estimation of Copy  
Number and Detection of Gene Interactions**

by


**Beibei Guo**


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

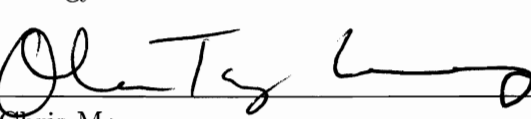
**Doctor of Philosophy**

APPROVED, THESIS COMMITTEE:

  
\_\_\_\_\_  
Rudy Guerra, Chair  
Professor of Statistics

  
\_\_\_\_\_  
Marina Vannucci, Co-advisor  
Professor of Statistics

  
\_\_\_\_\_  
Michael Gustin  
Professor of Biochemistry and Cell  
Biology

  
\_\_\_\_\_  
Chris Man  
Assistant Professor of Pediatrics, Baylor  
College of Medicine

HOUSTON, TEXAS

AUGUST, 2010

## Abstract

# Statistical Methods for Bioinformatics: Estimation of Copy Number and Detection of Gene Interactions

by

Beibei Guo

Identification of copy number aberrations in the human genome has been an important area in cancer research. In the first part of my thesis, I propose a new model for determining genomic copy numbers using high-density single nucleotide polymorphism genotyping microarrays. The method is based on a Bayesian spatial normal mixture model with an unknown number of components corresponding to true copy numbers. A reversible jump Markov chain Monte Carlo algorithm is used to implement the model and perform posterior inference. The second part of the thesis describes a new method for the detection of gene-gene interactions using gene expression data extracted from microarray experiments. The method is based on a two-step Genetic Algorithm, with the first step detecting main effects and the second step looking for interacting gene pairs. The performances of both algorithms are examined on both simulated data and real cancer data and are compared with popular existing algorithms. Conclusions are given and possible extensions are discussed.

## Acknowledgements

I would like to thank my committee for all of their creative and intelligent guidance for the projects. I want to thank my Committee Chair, Dr. Rudy Guerra, for his guidance and encouragement for completing the projects. I want to thank Dr. Marina Vannucci for her additional mentorship, support, and guidance into the field of Bayesian Statistics. I appreciate Dr. Michael Gustin for sharing his own experiences and expertise to help with the projects. I am very grateful to Dr. Chris Man for his time and insightful comments on the projects.

I appreciate Dr. Ching Lau for lending me his expertise in biology and bioinformatics, and for providing me the data on leukemia or ependymoma.

I want to express my gratitude to Dr. Jian Wang, for his time and efforts in helping me get familiar with softwares and datasets.

I am also very indebted to my family for their unwavering support through my graduate studies.

Last, I want to thank the funding source. This research was funded by the Rice-TCH Bioinformatics Collaboration leaded by Drs. David Poplack and Kathy Ensor..

# Contents

<b>Abstract</b>	<b>ii</b>
<b>Acknowledgements</b>	<b>iii</b>
<b>List of Figures</b>	<b>vi</b>
<b>List of Tables</b>	<b>x</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Overview . . . . .	1
1.2 Outline . . . . .	3
<b>2 Estimation of Genomic Copy Number with Single Nucleotide Poly-</b>	
<b>    morphism Genotyping Arrays</b>	<b>4</b>
2.1 Introduction . . . . .	4
2.2 Biology Background . . . . .	6
2.2.1 Basic Genetics . . . . .	6
2.2.2 Studies for copy number estimation . . . . .	7
2.3 Existing Methods . . . . .	14
2.3.1 Normal Mixtures for CGH Data . . . . .	17
2.3.2 Probe based Models for SNP Data . . . . .	19
2.3.3 Robust CN Algorithms for SNP Data . . . . .	21
2.3.4 Comparison of the three algorithms . . . . .	22

2.4	Comparison of copy number estimates from CGH and SNP arrays . .	26
2.5	Finite mixture models for SNP arrays . . . . .	32
2.6	Gaussian Markov random field . . . . .	34
2.7	Proposed Method - Bayesian model for copy number estimation . . .	35
2.7.1	Prior distributions . . . . .	37
2.7.2	Posterior inference . . . . .	40
2.8	Simulation Study . . . . .	42
2.9	Real data application . . . . .	57
2.10	Conclusion . . . . .	62
<b>3</b>	<b>Detection of Gene Interactions for Classification using Gene Expression Data</b>	<b>67</b>
3.1	Introduction . . . . .	67
3.2	Biological background . . . . .	68
3.3	Existing methods . . . . .	71
3.3.1	Multigene Expression Profile Model . . . . .	72
3.3.2	A Introduction to Genetic Algorithm . . . . .	77
3.3.3	k-Nearest-Neighbor Classifiers . . . . .	80
3.3.4	Genetic Algorithm/k-nearest-Neighbor Method . . . . .	81
3.3.5	Comparison of the three algorithms . . . . .	84
3.4	Proposed GA Method . . . . .	85
3.5	Simulation Study . . . . .	89
3.6	Real data application . . . . .	110
3.7	Conclusion . . . . .	130
<b>A</b>	<b>Details of the derivations of the MCMC algorithm</b>	<b>139</b>

# List of Figures

2.1	A Karyotyping image. . . . .	8
2.2	Technology of CGH and aCGH. . . . .	10
2.3	Technology of SNP microarray. . . . .	12
2.4	Technology of SNP microarray. . . . .	13
2.5	Comparison of dChip and CNAG. . . . .	24
2.6	Comparison of dChip and CNAG. This shows more noise than previous one. . . . .	25
2.7	A concordant case between SNP and CGH copy number inference. The data are ENP samples from Texas Children's Hospital. In the top panel, each dot represents a SNP. Each green dot represents log-ratio and red dot represents inferred integer copy numbers. In the bottom panel, each dot represents a BAC. The black boxes on top show cytobands. Yellow, red and green represent normal, gain ( $CN \geq 3$ ), and loss ( $CN = 0$ , or $1$ ) respectively. However, red and green dots with crosses on them mean that they are outliers, so should be considered normal instead. . . . .	28

2.8	A discordant case between SNP and CGH copy number inference. The data are ENP samples from Texas Children's Hospital. In the top panel, each dot represents a SNP. Each green dot represents log-ratio and red dot represents inferred integer copy numbers. In the bottom panel, each dot represents a BAC. The black boxes on top show cytobands. Yellow, red and green represent normal, gain ( $CN \geq 3$ ), and loss ( $CN = 0$ , or 1) respectively. However, red and green dots with crosses on them mean that they are outliers, so should be considered normal instead. . . . .	29
2.9	Heatmap of concordance. We have 16 samples (16 columns) and 22 autosome (22 rows). The more red the cell is, the larger the p-value is; the more white the cell is, the smaller the p-value is. . . . .	31
2.10	SNP copy number inference based on normal mixtures implemented with EM and Bayesian algorithms. Both algorithms assume (spatially) independent log-ratios. . . . .	33
2.11	Scenario 1 . . . . .	50
2.12	Scenario 2 . . . . .	51
2.13	Scenario 3 . . . . .	52
2.14	Scenario 8 . . . . .	53
2.15	Case 1 chromosome 6. The red line is the loess curve with window size .3. There is no evidence of a genome wave. . . . .	59
2.16	Case 2 chromosome 12. The red line is the loess curve with window size .3. There is no evidence of a genome wave. . . . .	60
2.17	A segment of chromosome 6 from case 1. Cytogenetics data show a loss at 6q1206q21. . . . .	64

2.18	Chromosome 12 from case 2. We validated two regions using qPCR, around positions 50Mb and 110Mb. The average copy numbers based on qPCR for these two regions resulted in the values 1.43, 1.55, respectively, with approximate 95% confidence intervals of (1.2, 1.71), and (1.33, 1.81), respectively. The validation results support a loss, in concordance with the Bayes method and in contrast to the inference based on CNAG indicating a normal copy number throughout the stretch.	65
2.19	Case 2 CGH result analyzed using GLAD software in Bioconductor. The aCGH platform we used was a BAC array platform, which contains 2,621 BAC clones and has a 3Mb resolution. This array is based on a 2-color competitive hybridization platform (Cy3/Cy5). The experiments were conducted by hybridizing the fluorescent-labeled tumor DNA with reference DNA on the array. As the plot indicates, the region from around 45Mb to around 120Mb shows copy number loss, a finding that agrees with the result provided by our Bayesian model. Furthermore, the tail region of the chromosome indicates a copy number gain, which again confirms our findings and contradicts the CNAG result. . . . .	66
3.1	A hypothetical example of gene interactions . . . . .	70
3.2	Overview of microarray technology. . . . .	71
3.3	Single point Crossover . . . . .	80
3.4	A single mutation . . . . .	81
3.5	Flowchart of the GA procedure . . . . .	82
3.6	Selection frequencies of each gene for scenario 1 . . . . .	93
3.7	Selection frequencies of each gene for scenario 2 . . . . .	95
3.8	Selection frequencies of each gene for scenario 3 . . . . .	96
3.9	Selection frequencies of each gene for scenario4 . . . . .	98
3.10	Selection frequencies of each gene for scenario 5 . . . . .	99
3.11	Selection frequencies of each gene for scenario 6 . . . . .	101



3.12	Selection frequencies of each gene for scenario 7 . . . . .	102
3.13	Selection frequencies of each gene for scenario 8 . . . . .	103
3.14	Selection frequencies of each gene for breast cancer data . . . . .	111
3.15	Joint pattern of gene pair 1 identified by GA . . . . .	114
3.16	Joint pattern of gene pair 2 identified by GA . . . . .	115
3.17	Joint pattern of gene pair 3 identified by GA . . . . .	116
3.18	Joint pattern of gene pair 4 identified by GA . . . . .	117
3.19	Joint pattern of gene pair 5 identified by GA . . . . .	118
3.20	Joint pattern of gene pair 6 identified by GA . . . . .	119
3.21	Joint pattern of gene pair 7 identified by GA . . . . .	120
3.22	Joint pattern of gene pair 8 identified by GA . . . . .	121
3.23	Joint pattern of gene pair 1 with selection frequency of 0 . . . . .	122
3.24	Joint pattern of gene pair 2 with selection frequency of 0 . . . . .	123
3.25	Joint pattern of gene pair 3 with selection frequency of 0 . . . . .	124
3.26	Joint pattern of gene pair 4 with selection frequency of 0 . . . . .	125
3.27	Joint pattern of gene pair 5 with selection frequency of 0 . . . . .	126
3.28	Joint pattern of gene pair 6 with selection frequency of 0 . . . . .	127
3.29	Joint pattern of gene pair 7 with selection frequency of 0 . . . . .	128
3.30	Joint pattern of gene pair 8 with selection frequency of 0 . . . . .	129

# List of Tables

2.1	Concordance between SNP and BAC copy number. The entry represents the percentage. The 16 columns represent the 16 EPN samples, and the 22 rows represent the 22 autosome. . . . .	30
2.2	Misclassification rates from simulation study. . . . .	54
2.3	False negative rates from simulation study. . . . .	55
2.4	False positive rates from simulation study. . . . .	56
3.1	Misclassification on breast cancer data (van't Veer <i>et al.</i> [2002]). Source: Yan <i>et al.</i> [2008]. . . . .	77
3.2	Misclassification including GA on breast cancer data. . . . .	86
3.3	ranks of true interacting pairs for Scenario 1 . . . . .	94
3.4	ranks of true interacting pairs for Scenario 2 . . . . .	94
3.5	ranks of true interacting pairs for Scenario 3 . . . . .	97
3.6	ranks of true interacting pairs for Scenario 4 . . . . .	97
3.7	ranks of true interacting pairs for Scenario 5 . . . . .	97
3.8	ranks of true interacting pairs for Scenario 6 . . . . .	100
3.9	ranks of true interacting pairs for Scenario 7 . . . . .	100
3.10	ranks of true interacting pairs for Scenario 8 . . . . .	100
3.11	Ranks of true interacting pairs for 10 replicates for Scenario 1 under the proposed GA algorithm . . . . .	105
3.12	Ranks of true interacting pairs for 10 replicates for Scenario 2 under the proposed GA algorithm . . . . .	106

3.13 Ranks of true interacting pairs for 10 replicates for Scenario 3 under the proposed GA algorithm . . . . .	107
3.14 ranks of true interacting pairs for 10 replicates for Scenario 4 under the proposed GA algorithm . . . . .	108
3.15 ranks of true interacting pairs for 10 replicates for Scenario 5 under the proposed GA algorithm . . . . .	109
3.16 Correlations of the two groups . . . . .	113
3.17 Correlations of the two groups . . . . .	123

# Chapter 1

## Introduction

### 1.1 Overview

Statistical genomics is an application area of probability and statistics. It involves the development of models and methods for the analysis and interpretation of genomic data. It has recently received renewed interest because of significant advancements in biotechnology and breakthroughs in genetics and molecular biology. Diverse genomic data generated by high-throughput biotechnologies requires new computational and statistical methods for proper analysis and interpretation.

Bioinformatics is the application of statistics and computer science to biological fields including molecular biology and genomics. It now entails the creation and advancement of databases, algorithms, computational and statistical techniques and theories to solve formal and practical problems arising from the management and analysis of biological data. Identification of genetic differences between two sample classes is essential to disease diagnosis, treatment and prevention. In recent years, microarrays have become powerful tools to address this problem.

The present dissertation investigates two problems in bioinformatics: copy num-

ber variation and classification of disease status based on gene-gene interactions.

Copy number is the number of copies of a particular segment of DNA sequence. It was generally thought that DNA sequences were almost always present in two copies in a genome. However, discoveries have revealed that segments of DNA, ranging in size from single-nucleotide to millions of DNA bases, can vary in copy-number. We define copy number gain if copy number is greater than two, and loss if copy number is fewer than two. In cancer, copy number losses and gains are known to contribute to alterations in the expression of tumour-suppressor genes and oncogenes, respectively; see, for example Knuutila [1998] and Knuutila [1999]. Developmental abnormalities, such as Down(Izraeli S. [2005]), Prader Willi(Donlon [T.A. *et al*]), Angelman and Cri du Chat syndromes(wikipedia), result from gain or loss of one copy of a chromosome or chromosomal region. Thus, detection and mapping of copy number abnormalities provide an approach for associating aberrations with disease phenotype and for identifying critical genes. A factor that seems to affect detecting of copy number aberrations is the fact that many cancer samples (with copy number aberrations) are "contaminated" by normal cells. The estimation of copy number would be much more accurate if we can account for normal cell contamination.

In the study of human genetics, mapping of complex traits is a major challenge. In contrast to mendelian traits, which can be attributed to mutations of single genes, complex traits involve more than one gene. Complex traits are much more common in the population and include asthma, hypertension, heart disease, Alzheimer's disease, and diabetes, among many others. Understanding how interactions among genes contribute to the trait is having a large impact on biomedical research, agriculture and evolutionary biology. Most current strategies of selecting informative genes for classification are based on predictor variables representing individual genes. The simplest example is a t-test. These tests are very easy to perform, but they ignore

information contained in gene-gene interactions. A review of current techniques for detection of gene-gene interactions is given in Musani *et al* [2007]. We believe that taking into account gene-gene interactions can help improve classification accuracy.

This thesis discusses some of the current methods and their limitations for the problems of copy number inference and classification based on gene-gene interactions. A new approach is proposed for each problem. The new methods are assessed by simulation studies, as well as applications involving real cancer data.

## 1.2 Outline

The dissertation is organized as follows. In Chapter 2 we discuss copy number estimation, including biology background, existing statistical methods to estimate copy number, the proposed Bayesian mixture model, and the application to simulated and real data. Chapter 3 covers the detection of gene-gene interactions, with its own parts of biology, existing methods and the proposed method, as well as the applications. The thesis concludes with discussions of possible extensions.

## Chapter 2

# Estimation of Genomic Copy Number with Single Nucleotide Polymorphism Genotyping Arrays

### 2.1 Introduction

Gene dosage variations occur in many diseases, as well as in normal populations (e.g., Pinkel *et al.* [1998], Wang *et al* [2009]). In cancer, copy number losses and gains are known to contribute to alterations in the expression of tumour-suppressor genes and oncogenes, respectively; see, for example Knuutila [1998, 1999]. Developmental abnormalities, such as Down(Izraeli S. [2005]), Prader Willi(Donlon [T.A. *et al*]), Angelman and Cri du Chat syndromes(wikipedia), result from gain or loss of one copy of a chromosome or chromosomal region. Thus, detection and mapping of copy number abnormalities provide an approach for associating aberrations with disease phenotype and for identifying critical disease-causing genes. As an example, Redon *et al.* [2006] constructed a first-generation copy number variation (CNV) map of the human genome through the study of 270 HAPMAP normal individuals from four populations (30 parent-offspring trios of the Yoruba from Nigeria (YRI), 30 parent-

offspring trios of European descent from Utah, USA (CEU), 45 unrelated Japanese from Tokyo, Japan (JPT) and 45 unrelated Han Chinese from Beijing, China (CHB)) with ancestry in Europe, Africa or Asia, Consortium IH [2003]. A copy number variant (CNV) is a segment of DNA in which differences of copy-number (number of copies of a molecule or portions of it) have been found by comparison of two or more genomes. A total of 1,447 copy number variable regions, covering 360 megabases (i.e., 12% of the genome), were identified in this study. These CNVRs contained genes, disease loci, functional elements and segmental duplications. Their map of copy number variation in the human genome demonstrates the ubiquity and complexity of this form of genomic variation. The abundance of functional sequences of all types both within and flanking areas of copy number variation suggests that the contribution of CNVs to phenotypic variation is likely to be appreciable.

DNA from the individuals in the study of Redon *et al.* [2006] was analyzed for CNV using two technologies: single-nucleotide polymorphism (SNP) genotyping arrays and comparative genomic hybridization (CGH). Comparative genomic hybridization (CGH) is a molecular-cytogenetic method for the analysis of copy number changes (gains/losses) in the DNA content of a given subject's DNA and often in tumor cells. Array-based Comparative Genomic Hybridization (aCGH) is a technique to detect genomic copy number variations at a higher resolution level than chromosome-based CGH (Pinkel *et al.* [1998]). The method is based on hybridization of fluorescently labeled tumor DNA and reference DNA on a microarray platform containing Bacterial Artificial Chromosome (BAC) clones. As a gold standard, it is robust in identifying long (say, greater than 1MB) segments of chromosomal alterations. However, although the resolution of aCGH has been improved, it is still not high enough to detect amplifications or deletions of relatively short segments (less than 10 kb), Toruner [2007] and Redon *et al.* [2006]. The high-density SNP array, which can accommodate hundreds of thousands of SNP probe sets simultaneously, is an alternative approach to detect genome wide copy number aberrations with much higher resolution than CGH,



see Bignell [2004]. Compared to CGH, SNP array based experiments are newer and are becoming more popular for copy number analysis. Below I provide more detail on these studies.

## 2.2 Biology Background

### 2.2.1 Basic Genetics

The human body is built from 100 trillion cells. The human genome inside the nucleus of each cell is comprised of 3 billion nucleotides packaged into two sets of 23 chromosomes, one set inherited from each parent. Each chromosome is a DNA double helix. The nucleotides that make up DNA include A (adenine), T (thymine), G (guanine), C (cytosine), with A pairing with T and G pairing with C. A gene is a specific contiguous subsequence of DNA whose A-T-G-C sequence is the code required for constructing a protein. Roughly 1.2 percent of the genome is made of genes. The central dogma of molecular biology is the process of DNA  $\rightarrow$  mRNA  $\rightarrow$  protein, with the two intervening steps called transcription and translation, respectively.

Copy number is the number of copies of a particular segment of DNA sequence. It was generally thought that DNA sequences were almost always present in two copies in a genome. However, discoveries have revealed that segments of DNA, ranging in size from single-nucleotide to millions of DNA bases, can vary in copy-number. There are two primary mechanisms of genomic rearrangements that lead to copy number changes; non-allelic homologous recombination (NAHR) is the most common form and non-homologous end-joining is also known but seems to occur less often (Shaw [2004]). The NAHR mechanism is characterized by the presence of low copy repeats (LCRs) that serve as substrates for the recombination. Briefly, two LCRs, A and B, that are directly oriented misalign so that one is atop the other (imagine an S-shaped curve), and subsequent homologous recombination results in a deletion with a sin-

gle recombinant LCR. Similar mechanisms lead to gains. DNA replication errors is another source of copy number change. As found in Redon *et al.* [2006], 12% of the genome of normal population has copy number gains or losses.

Differences in the DNA sequence of our genomes contribute to our uniqueness. These changes influence most traits including susceptibility to disease. Copy number changes in DNA can cause abnormal mRNA transcript amounts and consequently affect the functioning of proteins. In particular, amplification of an oncogene or deletion of a tumor suppressor gene are important steps in elucidating mechanisms for tumorigenesis. There are many examples of copy number change associated with disease. Down Syndrome is the most common numerical abnormality found in newborns, which is characterized by trisomy chromosome 21. Cancer is also associated with copy number changes. In an analysis of breast cancer (Shadeo [A *et al.*]), seven cell lines showed 75 gains and 48 losses in the genome. A prostate cancer study (Wolf [M *et al.*]) across 4 cell lines showed association with 28 gains and 18 losses, while a colorectal cancer study (Douglas [E *et al.*]) across 48 cell lines and 37 primary CRCs showed gain of chromosome 20, 13, and 8q and smaller regions of amplification such as chromosome 17q11.2-q12. Thus, studying DNA copy number is important in biological and medical research.

### 2.2.2 Studies for copy number estimation

There are many different ways to detect copy number variation. Among them, karyotyping, comparative genomic hybridization (CGH), array CGH, and SNP microarrays are standard well-known methods.

**Karyotyping** is the study of chromosomes and the related disease states caused by numerical and structural chromosome abnormalities. A variety of cell or tissue

types can be used to perform these studies. Normally chromosomes can't be seen with a light microscope but during cell division they become condensed enough to be easily analyzed at 1000X. Under the microscope chromosomes appear as thin, thread-like structures. Karyotyping images can only show large ( $\geq 20\text{kb}$ ) segments of copy number change; for example, chromosome arms. Figure 2.1 shows a karyotype of Down syndrome defined as having three copies of chromosome 21.

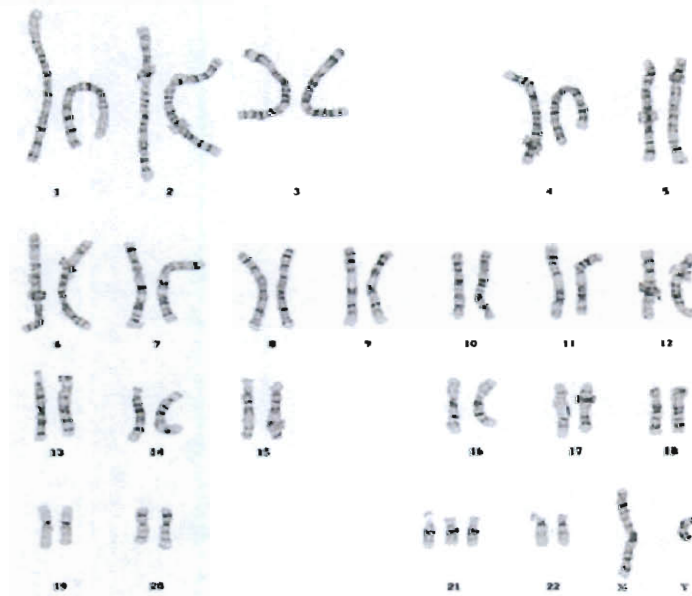


Figure 2.1: A Karyotyping image.

Source: Cytogenetics information site (<http://home.comcast.net/dmgt350/cytogenetics>).

**Comparative genomic hybridization** Comparative genomic hybridization (CGH) is a molecular-cytogenetic method for the analysis of copy number changes (gains/losses) in the DNA content of a given subject's DNA and often in tumor cells. Array-based Comparative Genomic Hybridization (aCGH) is a technique to detect genomic copy number variations at a higher resolution level than chromosome-based CGH (Pinkel *et al.* [1998]). The method is based on the hybridization of fluorescently labeled tumor DNA and reference DNA on a microarray platform containing Bacterial Artificial Chromosome (BAC) clones or spotted DNA. Using image analysis, regional differences in the fluorescence ratio of tumor to reference DNA can be detected and used for quantifying copy number changes. In CGH experiments (Figure 2.2), DNA from test (tumor) tissue and from normal tissue is labeled with different dyes. After mixing test and reference DNA, the mixture is hybridized to normal metaphase chromosomes or, for array-CGH (aCGH) to a slide containing thousands of defined BAC probes. The (fluorescence) color ratio along the chromosomes is used to evaluate regions of DNA gain or loss in the test sample. As shown in Figure 2.2, the more green we get, the smaller the copy number is, and the more red we get, the larger the copy number is. Array-CGH (aCGH) gives a higher resolution ( $\leq 20\text{Kb}$ ) than conventional CGH (the BAC clones are smaller). As a gold standard, it is robust to identify long segments of chromosomal alterations. However, although the resolution of aCGH has been improved, it is still not high enough to detect amplifications or deletions of relatively short segments (Toruner [2007] and Redon *et al.* [2006]).

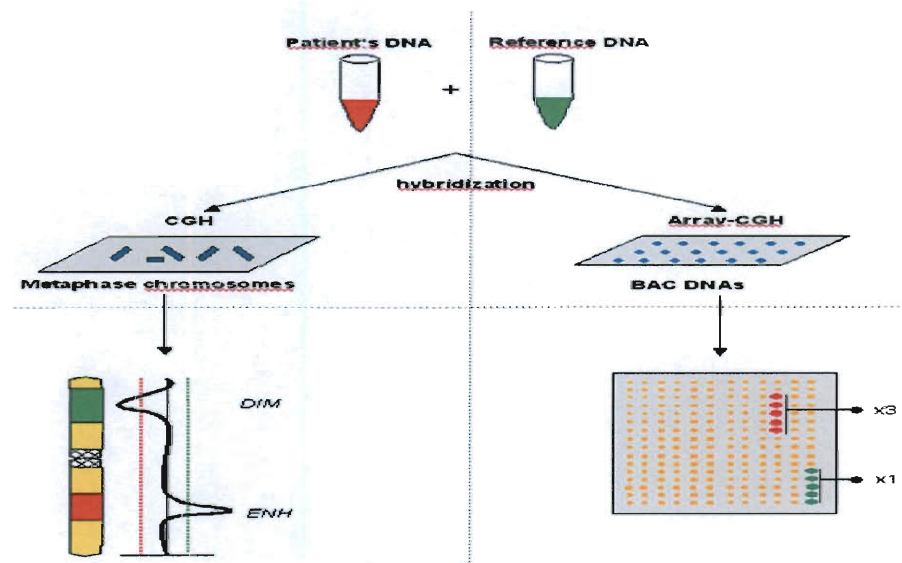


Figure 2.2: Technology of CGH and aCGH.

Source: [www.array-cgh.de](http://www.array-cgh.de)

**SNP Microarrays** are also used for copy number studies and are quickly becoming the standard approach to these studies. CGH, as a gold standard, is very robust in identifying relatively longer segments of chromosomal alterations. SNP arrays provide higher resolution, and thus can detect amplifications or deletions of relatively short segments. Compared to CGH, SNP array based experiments are newer, cheaper and easier to run (Winchester [2009]).

A SNP (single nucleotide polymorphism) is a DNA sequence variation occurring when a single nucleotide in the genome differs between members of a population. Almost all common SNPs are operationally defined by two alleles. They occur approximately every 1-2kb in the human genome. SNPs are used to help researchers pinpoint genes that are associated with disease. Since DNA copy number may occur on a very short region, platforms that provide higher resolutions are desirable. The technology for production of high-density oligonucleotide arrays was pioneered by Affymetrix, a biotechnology company. These arrays were originally designed as genotyping arrays and can accommodate hundreds of thousands of SNPs, simultaneously. They provide much higher resolution (about 10kb)(Bignell [2004]) than aCGH and thus can hopefully measure much smaller segments of copy number variation. Currently, there is no specific platform for evaluating SNP copy number. However, SNP genotyping arrays can and are being used for copy number studies. Both Affymetrix and another biotechnology company, Illumina, manufacture genotyping arrays that are popular platforms for copy number estimation.

Figure 2.3 shows the Affymetrix platform used to generate the data I have analyzed. The surface of the Affymetrix chip is like a giant checkerboard with hundreds of thousands of squares that has been shrunk down to the size of a thumbnail. Each square on the checkerboard holds millions of copies of one unique type of probe. In this format a probe is 25 base-pairs long and 20 different probe pairs (PM (perfect match)

and MM (mis-match)) are used to interrogate each SNP. The observable **measurements** are the hybridization intensities, which represent the amount of target bound to the probes on the microarray. The simple average of PM-MM differences for all probe pairs in a probe set is used as the copy number index for the target SNP. The 40 probes for the same SNP are designed according to a probe quartet unit. A probe quartet includes four probes: a probe pair for allele A and a probe pair for allele B. A probe pair includes a perfect match and a mismatch. To make the SNP copy number estimates more reliable multiple probe quartets are used per SNP, each distinguished by a location shift of the SNP from the center (location 13) of the 25-mer probe. Probe quartets defined by sense/antisense strands and five shifts ( $-2, -1, 0, 1, 2$ ) lead to 40 probes per SNP. In Figure 2.4, the upper left panel shows definition of a probe and PM and MM definitions. For each SNP, the perfect match (PM) is a 25 base-pair long probe that is completely complementary, and the mis-match (MM) probes are shifted, as shown in the upper right panel. So at each SNP location, there are 'replications'. The lower panel shows cells identifying the probes by genotypes.

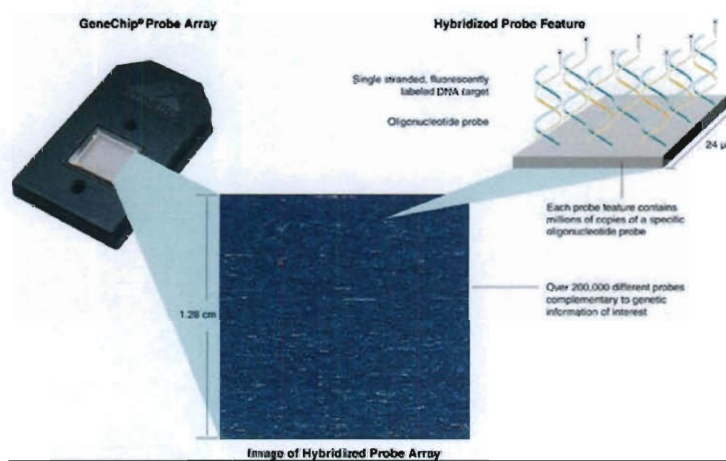


Figure 2.3: Technology of SNP microarray.

Image: Affymetrix

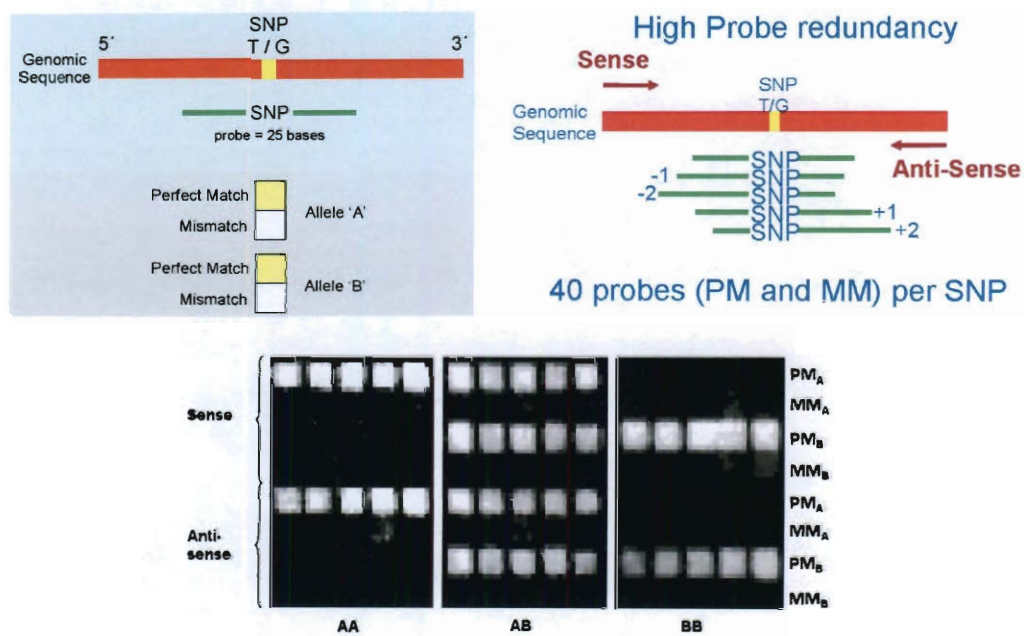


Figure 2.4: Technology of SNP microarray.

Image: Affymetrix



A simple description of the method is the following: For a given subject, DNA is obtained and fragmented at known locations so that the SNPs are far away from the ends of these fragments; the fragmented DNA is amplified with a polymerase chain reaction (PCR) reaction; the sample is labeled and hybridized to an array containing probes designed to interrogate the resulting fragments; the array is rinsed so that any DNA that didn't pair is washed away; a laser scanner detects and measures the intensity of the fluorescent signal being emitted for each probe. This is done separately for tumor DNA and normal DNA. The ratios of tumor to normal samples are then used to estimate copy number at each SNP location.

The raw data from SNP arrays are fluorescent intensities that must be normalized and then ultimately transformed to log-ratios.

## 2.3 Existing Methods

In order to estimate copy number from the various data types described we need statistical methods to optimize the signal-to-noise ratio. The first analytical methods were simple but often effective, involving smoothing of the log-ratios and applying a threshold to determine if the ratio over a potential region signified an amplification or a deletion. For example, a moving average was used to process the ratios, and a 'normal versus normal' hybridization was used to compute a threshold level (Pollack *et al* [2002]).

A number of statistical methods have been proposed to estimate copy numbers from various platforms. Two of the most popular methods for SNP arrays are dChip and Copy Number Analyser for GeneChip (CNAG). Zhao *et al.* [2004] proposed dChip, an algorithm that derives model-based estimates of SNP copy numbers that incorporate probe effects and a hidden Markov model (HMM) to infer integer-valued copy numbers. Although the current version of the dChip software can accommodate the

newer SNP arrays, such as the Affymetrix 250K array, it is not optimized for it. Nannya *et al.* [2005] developed the CNAG algorithm, which accounts for the length and GC content of the PCR products. Accounting for the length and content of GC elements in the probes seems to improve copy number inference (Nannya *et al.* [2005]). Another source of variation that can affect a copy number analysis is the so-called “genome wave” (Marioni *et al.* [2007]; Diskin *et al.* [2007]) a genome-wide spatial autocorrelation pattern in signal intensity. Since the genome wave may be confounded with the copy number profile across a chromosome, investigators should examine their intensity data for its presence and adjust the data accordingly. Since the genomic wave is thought to be in large part due to GC content (Marioni *et al.* [2007]), the CNAG algorithm can also be thought of as an adjustment for wave effects possibly present in SNP array data. Again, an HMM is used to infer integer copy numbers.

The HMM approach can also be found in the algorithms underlying QuantiSNP (Colella *et al.* [2007]) and PennCNV (Wang *et al.* [2007]), both of which use the log-R-ratio and *B*-allele frequency to infer the copy number state of each SNP. The *B*-allele frequency is the frequency of one allele. These two methods consider a six-state Markov model which distinguishes copy-neutral loss-of-heterozygosity from the normal state. Most HMM based algorithms use the Viterbi algorithm (Rabiner [1989]) to infer integer copy numbers.

To date, there are a handful of Bayesian methods for copy number inference. Most are for CGH data, but a few exist for SNP data. Rueda *et al.* [2007] proposed RJACGH, a nonhomogeneous HMM in a Bayesian context for CGH data. Instead of prespecifying the number of states as a conventional HMM, a reversible jump Markov Chain Monte Carlo (MCMC) method is used to allow for varying numbers of hidden states. Bayesian model averaging is used to obtain final estimates. Pique-Regi *et al.* [2008] developed a method called Genome Alteration Detection Algorithm (GADA) that is based on sparse Bayesian learning (Tipping [2001]). The approach takes advantage of

the a priori assumption that the number of copy number alterations (break points) is sparse with respect to the number of probes. As with several other methods, advantage is also taken of the fact that the copy number pattern across a chromosome can be modeled as a piecewise constant function or vector. The GADA output gives copy number results in the form of a segmentation, viz., a collection of ordered segments defined by their breakpoints and amplitudes. To obtain integer-valued copy numbers or alteration status (loss, normal, gain), the identified segments must be analyzed by a thresholding procedure, such as that proposed by Pique-Regi *et al* [2008]. GADA can be applied to both CGH and SNP based data. Rancoita *et al* [2009] also make use of piecewise constant modeling in their algorithm, mBPCR, which is a modification of the original Bayesian Piecewise Constant Regression (BPCR) method developed by Hutter [2007]. This method is general for data that take the form of a piecewise constant function with unknown segment numbers, boundaries, and levels. Rancoita *et al.* illustrate the mBPCR method using SNP data, but it appears that log-ratios based on CGH data can also be analyzed.

In addition to those described above, several other statistical methods have been developed for copy number analysis. They vary in their assumptions, inference (segmentation, alteration status, integer copy number), platform (CGH, SNP), input data (e.g., CEL files or generic normalized log-ratio), and software implementation (e.g., commercial, web-based, customized academic program). Winchester [2009] describe and compare a number of methods. No method stands out as uniformly best and Winchester *et al.* suggest analyzing copy number data with at least two different methods to assess consistency and robustness of results. Several of the methods cited above are included in the comparison.

Most of the copy number methods assume normalized log-ratios as input. Relatively few include adjustments for known factors affecting inference. GC content and fragment length have been mentioned as factors affecting copy number inference. Another factor from tumor samples is normal cell contamination. Indeed, most

tumor samples are heterogeneous and include both cancer cells (with copy number aberrations) and normal cells (that can also include copy number aberrations). The larger percentage of normal cells present, the more difficult it is to infer copy number aberrations in the tumor cells; the log-ratios tend to shrink to the null value of zero. None of the above methods implement an adjustment for normal cell contamination. Below we show how our proposed method can account for this factor. Below I focus on the methods most closely related to the work I have done.

### 2.3.1 Normal Mixtures for CGH Data

Broet and Richardson [2006] proposed a three-state (gain/loss/normal) normal mixture model based framework for CGH data. Denote  $Z_{i,k}$  as a random variable corresponding to the normalized log-ratio measurement for the  $i$ th BAC ordered along chromosome  $k$ . Let  $L_{i,k}$  be an unobserved categorical variable taking the values  $c = 1, 2, 3$  with probabilities  $\{\omega_{c,i,k} : c = 1, 2, 3\}$  where  $0 \leq \omega_{c,i,k} \leq 1$  and  $\sum_{c=1}^3 \omega_{c,i,k} = 1$ .  $L_{i,k} = c$  indicates that BAC  $i$  of chromosome  $k$  is in state  $c$ . Here  $c = 1$  corresponds to the loss copy state,  $c = 2$  the normal copy state and  $c = 3$  the gain copy state. For each fixed  $c$ , we model the log-ratio measurement as arising from a normal distribution with mean  $\mu_c$  and variance  $\sigma_c^2$ . The marginal density of  $Z_{i,k}$  can thus be written in the form of a three-component normal mixture:

$$f(z_{i,k}) = \sum_{c=1}^3 \omega_{c,i,k} \times N(z_{i,k} | \mu_c, \sigma_c^2).$$

The quantities  $\omega_{c,i,k}$  are the weights for BAC  $i$  of chromosome  $k$ . We know that the log-ratios across the chromosome are not independent; they are correlated. Neighboring BACs tend to be in the same state, and in particular, they tend to have the same weights. Thus, Broet and Richardson [2006] introduced a spatial structure on the weights for each chromosome. They relate the weights  $\omega_{c,i,k}$  to three latent Markov random fields,  $\mathbf{x}_{c,k} = \{x_{c,i,k} : 1 \leq i \leq n_k\}$ , where  $n_k$  is the number of BACs on chromosome  $k$ . By a logistic transformation, the weights are a function of the MRF,

$$\omega_{c,i,k} = \frac{\exp(x_{c,i,k})}{\sum_{l=1}^3 \exp(x_{l,i,k})}$$

Here  $\mathbf{x}_{c,k}$  are three independent latent vectors, each distributed according to a Gaussian conditional autoregression model:

$$x_{c,i,k} \mid x_{c,(-i),k}, \tau \sim N \left( \frac{1}{m_{i,k}} \sum_{l \in \delta_{i,k}} x_{c,l,k}; \frac{\tau_{c,k}^2}{m_{i,k}} \right)$$

where  $x_{c,(-i),k}$  is the vector  $\mathbf{x}$  with the  $i^{th}$  element removed. This is the nearest neighbour Markov random field model where each BAC sequence has two adjacent neighbours, except for the sequences at the ends of the chromosome.  $\delta_{i,k}$  is the set of indices for the neighbors of BAC  $i$  for chromosome  $k$ , and  $m_{i,k}$  is the corresponding number of neighbours. In this model,  $\tau_{c,k}^2$  acts as a smoothing prior for the spatial fields and consequently controls the dependence among the weights. In particular, small values correspond to smoother realizations. In the paper, 2 neighbors for each BAC are used, and the first and last BAC simply have 1 neighbor.

Let's recall the Gaussian conditional autoregression model. Suppose  $\mathbf{X} = (X_1, \dots, X_n)^T$  has density

$$p(\mathbf{x}) \propto e^{-\frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x}}, \quad \mathbf{x} \in R^n$$

where  $\mathbf{Q}$  is positive definite symmetric matrix. Then

$$X_i \mid x_{-i} \sim N \left( \sum_{j=1}^n \beta_{ij} x_j, k_j \right)$$

where  $\beta_{ii} = 0$ ,  $\beta_{ij} = -Q_{ij}/Q_{ii}$  ( $i \neq j$ ) and  $k_i = 1/Q_{ii} > 0$ . When the specification of the density is based on the precision matrix  $\mathbf{Q}$  rather than on the dispersion matrix  $V = Q^{-1}$ , it is usually referred to as a conditional autoregressive formulation.

In a Bayesian framework, all unknown quantities are given prior distributions. For  $\tau_{c,k}^2$ , they specify a *Gamma*(.01, .01). The mean parameter  $\mu_2$  is fixed to zero. They choose flat priors for the other means, in the observed range of data  $[a, b]$ ,  $a < 0, b > 0$  so that  $\mu_1 \sim U(a, 0)$  and  $\mu_3 \sim U(0, b)$ . For the variance component,  $\sigma_c^2$ , they use an inverse gamma *IG*(.1, .1). Inference for parameters of interest is undertaken by sampling from their joint posterior distribution using an MCMC sampler. In particular, the posterior probabilities  $\hat{p}_{c,i,k}$  of belonging to each state for each BAC  $i$  of chromosome  $k$  can be directly estimated as averages of the weights from the output of the algorithm.

The authors demonstrated the performance of this model through comparison with other existing methods using both simulated and real cancer data. They found that the Bayesian method for aCGH platform perform better than their frequentist counterparts. There are two limitations of the current method. First it does not incorporate the distance between BACs, and second it does not consider overlapping BACs.

### 2.3.2 Probe based Models for SNP Data

Li and Wong [2001] describe model-based estimates of gene expression that incorporate probe effects. Their model was motivated by the fact that intensities from different probes associated with the same SNP are highly variable. The model has been extended to copy number estimation as follows. For any given SNP, let  $\theta_i$  be a copy number index for the SNP in the  $i^{th}$  sample. Assume that the intensity value of a probe will increase linearly as  $\theta_i$  increases, but that the rate of increase will be different for different probes. It is also assumed that within the same probe pair, the PM intensity will increase at a higher rate than the MM intensity. The MM and PM models are:

$$MM_{ij} = \nu_j + \theta_i \alpha_j + \epsilon_{ij}$$

$$PM_{ij} = \nu_j + \theta_i \alpha_j + \theta_i \phi_j + \epsilon_{ij}$$

where  $PM_{ij}$  and  $MM_{ij}$  denote the PM and MM intensity values for the  $i$ th array and  $j$ th probe pair for this SNP,  $\nu_j$  is the baseline response of the  $j$ th probe pair,  $\alpha_j$  is the rate of increase of MM of the  $j$ th probe pair,  $\phi_j$  is the additional rate of increase in the corresponding PM response, and  $\epsilon$  is random error. The model for individual probe responses implies a simple model for the PM-MM differences:

$$y_{ij} = PM_{ij} - MM_{ij} = \theta_i \phi_j + \epsilon_{ij}$$

where PM and MM are normalized intensities, and  $\epsilon_{ij}$  are *iid* normal distributed with mean 0 and unknown variance. The least square estimate for  $\theta_i$  is used as the copy number index for the SNP in the  $i$ th sample. For each SNP, the estimates copy number indices index of all the normal samples are averaged to obtain a mean signal associated with 2 copies. The "raw copy number" for test sample  $i$  is defined as  $2(\text{estimated index}/\text{mean signal of normal samples})$ . To infer integer copy numbers from the raw data, they use a Hidden Markov Model. The emission probabilities are assumed to be  $t$  distributed with 40 degree of freedom. Transition probabilities are given by Haldane's map function  $\theta = \frac{1}{2}(1 - e^{-2d})$  where  $d$  is the genetic distance between the SNP of interest and a neighbor SNP, and  $2\theta$  is the probability that the copy number of the SNP of interest will return to the background distribution, independent from the copy number of the previous SNP. Assume  $N$  is the number of copy number classes, so consecutive copy numbers from  $0, 1, \dots, N-1$  are assumed as classes. The background distribution is taken as 0.9 for the 2-copy state and  $0.1/(N-1)$  for other copy number states, where  $N$  is the largest copy number integer considered. These background distributions are used as the initial probabilities of the HMM. The Viterbi algorithm is used to obtain the most likely underlying copy number sequence.

The HMM is applied to all of the chromosomes and all the samples separately. By applying the algorithm to real cancer data, Li and Wong successfully identified many known regions of copy number variation as well as several novel homozygous deletions and high-level amplifications. dChip, which is the software to implement this method, has been used with many diseases, such as malignant melanoma, acute myeloid leukemias, and lung carcinoma.

### 2.3.3 Robust CN Algorithms for SNP Data

Nannya *et al.* [2005] developed a robust algorithm (CNAG) for copy number by accounting for the length and GC content of the PCR products (probes), based on the fact that log-ratios can be biased due to DNA fragment length (in bp) and CG content of the fragment that includes the SNP. Briefly, in the PCR process longer segments need more time to replicate than shorter segments, which results in different amounts of DNA. The GC content affects the efficiency of hybridization. Adjustments for these known biases would presumably lead to better estimates of copy number (Nannya *et al.* [2005]).

Let  $S_i$  denote the sum of signals from the 10 perfect match probes for the A allele and those for the B allele for SNP  $i$ . Denote the relative copy number at the  $i$ th SNP locus between two samples by  $\Lambda_i^{1,2} = \log_2(S_i^{sample1}/S_i^{sample2})$ . To adjust for GC content and fragment length the following model includes their effects via quadratic polynomial regression to generate adjusted log-ratios,

$$\Lambda_i^{1,2} = {}^c\Lambda_i^{1,2} + \sum_{j=1}^2 (a_j + b_j x_j + c_j x_j^2)$$

where  ${}^c\Lambda_i^{1,2}$  represents the adjusted or corrected copy number and  $x_1$  and  $x_2$  represent the length and GC content of the fragment that contains the SNP. This compensation provides a more accurate estimate  ${}^c\Lambda_i^{1,2}$  of copy number, showing a lower



SD than the unadjusted copy number ( $\Lambda_i^{1,2}$ ). The inferred integer copy number is derived through an HMM similar to the HMM used with dchip except that CNAG uses Kosambi's map function,  $\frac{1}{2}\tanh(2\theta)$ , as transition probabilities, where  $\theta$  is the recombination fraction between two neighboring SNPs. Nannya *et al.* [2005] applied the model to real data and the results showed a dramatic reduction in SD values and it works well for high density arrays. This methodology has been used to identify copy number variation in many DNA samples from diseases such as leukemia, rectal cancer, mental retardation and multiple myeloma.

### 2.3.4 Comparison of the three algorithms

CGH technology is considered the gold standard for copy number inference. SNP based experiments are newer, cheaper, and easier to run. However, their relative accuracy is not well understood yet. dChip is a very popular method to estimate SNP copy number and it is the first to give integer copy numbers based on SNP data. However, CNAG may be the most popular way to estimate high density SNP array copy numbers. CGHmix is a relative new method to estimate CGH copy number, which proposed to use spatially correlated mixture models. The three algorithms have their own merits and disadvantages. CGHmix incorporates spatial dependence into a three state mixture model, which has an intuitive interpretation and the Bayesian context makes the model very flexible. It also estimates parameters based on the whole genome instead of chromosome-wide. However, the MCMC algorithm it uses cannot deal with cases where there is only one cluster (one CN along a chromosome) because the model assumes there are exactly three clusters. dChip considers probe effects and gives integer copy numbers, but it does not take into account the PCR conditions. And, although the current version of the dChip software can accommodate the Affymetrix 250k arrays, it is not optimized for it. CNAG improves the log-ratios

dramatically by accounting for PCR products and also gives integer copy numbers. Drawbacks to both dchip and CNAG: they apply the algorithm to one chromosome at a time, and thus do not consider neighboring information along the entire genome. Also, the Viterbi algorithm for the HMM has some problems that can lead to overfitting.

For comparative purposes, I applied dChip and CNAG to real datasets. Figure 2.5 and Figure 2.6 consider two different subjects with dChip on the left and CNAG on the right. In the plots, each dot stands for a SNP. The green dots represent log-ratios and the red dots represent inferred integer copy numbers from either dchip or CNAG. For the case in Figure 2.5, both dchip and CNAG do well in the sense that the inferred copy numbers show little noise (very few scattered red dots). The two algorithms give very similar integer copy number estimates. We see that almost everywhere the inferred copy number is normal, except that there are losses at the beginning of chromosome 2, gain of q-arm of chromosome 11, loss at the beginning of chromosome 17, and gain of chromosome X in both. Generally, the CNAG results are better than dchip in the sense that the log-ratios are less noisy, but as with the case in Figure 2.6 we still see that CNAG can result in noisy integer inferred copy number as illustrated by sporadic red dots between chromosomes 4 through 8. Ideally, the red dots would appear as contiguous non-overlapping segments. dChip does even worse.

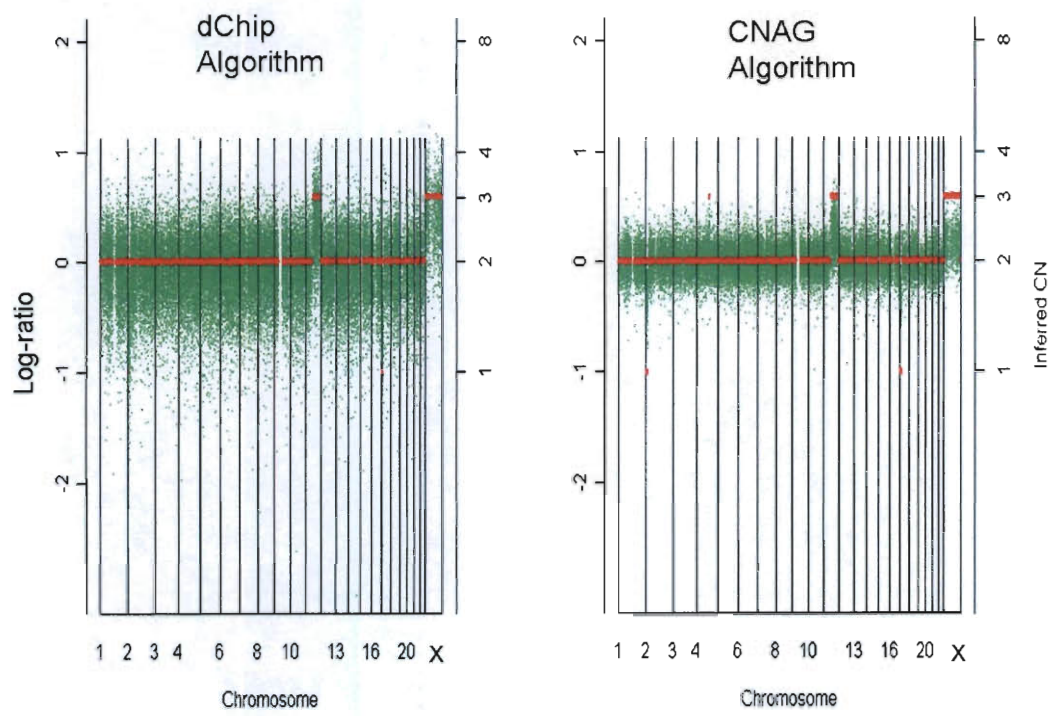


Figure 2.5: Comparison of dChip and CNAG.

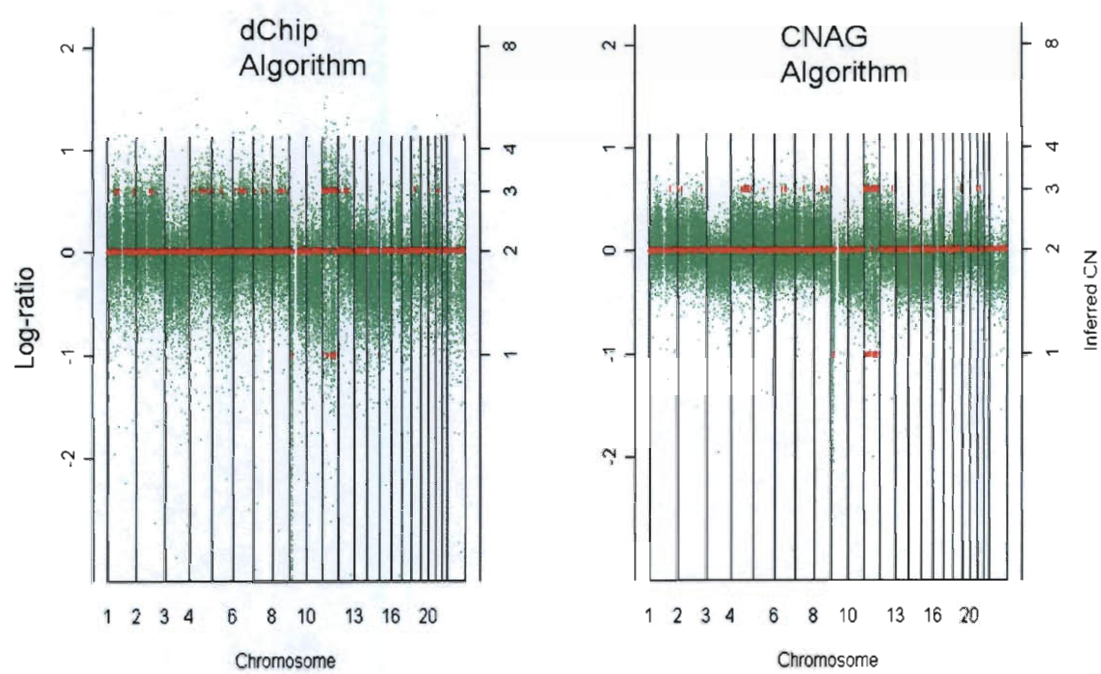


Figure 2.6: Comparison of dChip and CNAG. This shows more noise than previous one.

## 2.4 Comparison of copy number estimates from CGH and SNP arrays

CGH serves as the gold standard for copy number estimation. However, since its resolution is relatively low, investigators are interested in SNP arrays to obtain more reliable copy number information. SNP microarrays have higher resolution so hopefully they can detect smaller copy number variation segments. In order to see if we can depend on SNP arrays in place of aCGH, I compared copy number estimates from SNP arrays and CGH platforms. The main conclusion is that the two estimates are mostly concordant, but some cases do show discrepancies. There are two main reasons for the differences. First, the BACs used for CGH platform are big ( $\sim 100bp$ ) fragments, so their copy number estimates may reflect averages of different copy numbers within the BAC itself. Second, most CGH algorithms make use of the whole genome instead of one chromosome at a time for reference. Figure 2.7 and Figure 2.8 show a concordant case and a discordant case, respectively, with the SNP platform shown on the top panel and the CGH platform on the bottom panel. The data are ENP samples from Texas Children's Hospital. In the top panel, each dot represents a SNP. The green dots represent log-ratios and red dots represent inferred integer copy numbers. In the bottom panel, each dot represents a BAC. The black boxes on top show cytobands. Yellow, red and green represent normal, gain ( $CN \geq 3$ ), and loss ( $CN = 0$ , or 1) respectively. However, red and green dots with crosses on them mean that they are outliers, so should be considered normal instead. Figure 2.7 is a perfect concordant case where all the chromosomes are identified as normal status for both platforms. Figure 2.8 is a discordant case where, for example, chromosome 4 and chromosome 7 are identified as normal under the SNP platform, but as gains under the CGH platform.

I also did some numerical comparisons for a more objective measure of relative

accuracy. First, to make SNP copy numbers comparable with BAC copy numbers, we assign a copy number status to each SNP according to the inferred copy number: normal if it is 2; gain if it is greater than 2; loss if it is less than two. We then match the SNPs and BACs according to their positions along the chromosome. Considering only the matched SNPs, ignoring those that are not in any BAC, we compute the percentage (across each chromosome) of the matched SNPs that have the same copy number status as their corresponding BACs. This is obtained for each chromosome per subject ( $n = 16$ ). These percentages are shown in Table 2.1 and are measures of concordance between BAC copy numbers and SNP copy numbers. Using a 5% significance level, simple  $z$ -tests show concordance of approximately 89%, meaning that 89% of the null hypotheses are not rejected. A corresponding heatmap is also shown in Figure 2.9. Here we note that a main source of discrepancy is that due to subject-to-subject variation; that is, SNP and CGH copy number differences are largely confined to within subject, which may be a quality control issue.

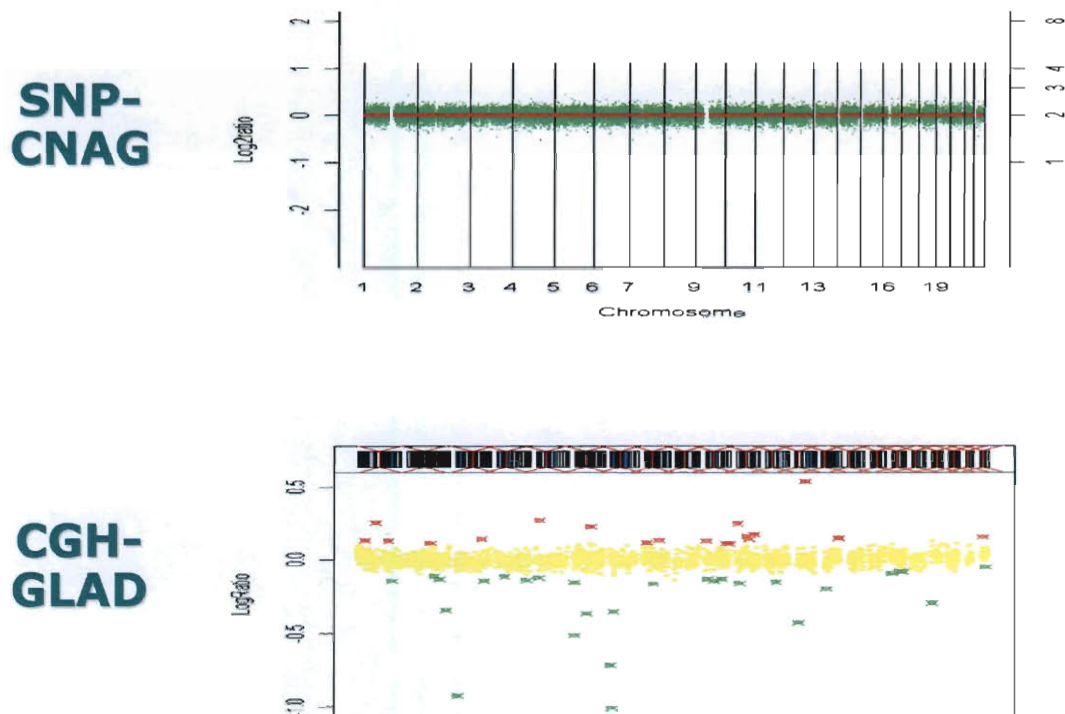
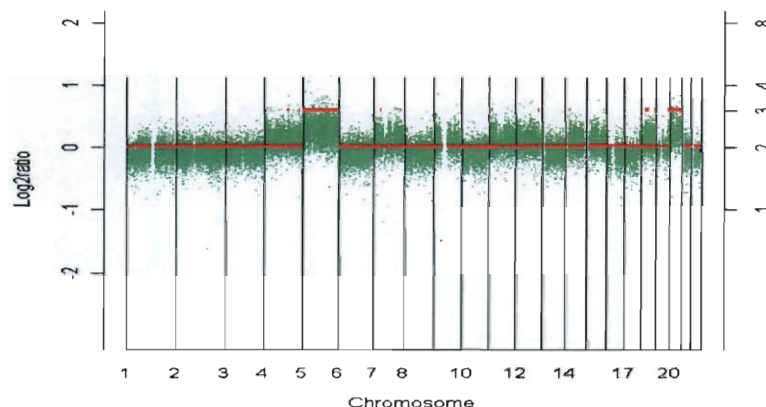


Figure 2.7: A concordant case between SNP and CGH copy number inference. The data are ENP samples from Texas Children's Hospital. In the top panel, each dot represents a SNP. Each green dot represents log-ratio and red dot represents inferred integer copy numbers. In the bottom panel, each dot represents a BAC. The black boxes on top show cytobands. Yellow, red and green represent normal, gain ( $CN \geq 3$ ), and loss ( $CN = 0$ , or  $1$ ) respectively. However, red and green dots with crosses on them mean that they are outliers, so should be considered normal instead.

**SNP-  
CNAG**



**CGH-  
GLAD**

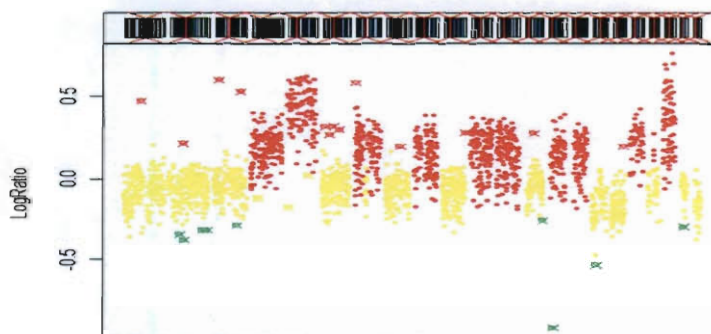


Figure 2.8: A discordant case between SNP and CGH copy number inference. The data are ENP samples from Texas Children's Hospital. In the top panel, each dot represents a SNP. Each green dot represents log-ratio and red dot represents inferred integer copy numbers. In the bottom panel, each dot represents a BAC. The black boxes on top show cytobands. Yellow, red and green represent normal, gain ( $CN \geq 3$ ), and loss ( $CN = 0$ , or  $1$ ) respectively. However, red and green dots with crosses on them mean that they are outliers, so should be considered normal instead.



Table 2.1: Concordance between SNP and BAC copy number. The entry represents the percentage. The 16 columns represent the 16 EPN samples, and the 22 rows represent the 22 autosome.

Case	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
chr1	1	1	0.994	0.984	1	1	1	1	1	1	0.543	0	0	1	1	1
chr2	1	1	1	0.885	1	1	1	1	0.875	1	1	1	1	1	1	1
chr3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
chr4	1	1	0	0.785	1	1	0.185	0.95	0.95	1	1	1	0.981	1	1	1
chr5	1	1	0.915	1	1	1	0.996	1	1	1	0.921	1	1	1	1	1
chr6	1	1	1	0.889	1	1	1	1	1	1	1	1	1	1	1	1
chr7	1	1	0.062	0	1	1	0	1	1	1	1	1	0	1	1	1
chr8	1	1	1	0.956	1	1	0.939	1	1	1	1	1	1	1	1	1
chr9	1	1	0	0.782	1	1	0.993	0.993	1	1	0.995	1	0	0.659	1	1
chr10	1	1	1	0.003	1	1	1	1	1	1	0.996	1	1	1	1	1
chr11	1	0.052	0.047	0.07	1	1	1	0.979	0.42	1	1	1	1	1	1	1
chr12	1	1	0	1	1	1	0.02	1	1	1	1	1	0.99	1	1	1
chr13	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
chr14	1	1	1	1	1	1	1	1	1	1	1	1	0.982	1	1	1
chr15	0.995	0.996	0	0.008	0.996	0.996	0.996	0.996	0.996	0.996	0.996	0.996	0.996	0.996	0.996	0.996
chr16	1	1	1	1	1	1	0	0	1	1	1	0	0	1	0.948	1
chr17	1	1	1	0.068	1	1	1	0	0.507	1	1	0	0.082	1	1	1
chr18	1	1	0.143	0	1	1	1	0.994	1	1	1	1	1	1	1	1
chr19	1	1	0.692	0	1	0.923	0	0	1	1	1	0.077	0.077	0	1	1
chr20	1	1	1	0.977	1	1	0	1	1	1	1	0	1	1	1	1
chr21	1	1	1	0.969	1	1	1	1	1	1	1	1	1	1	1	1
chr22	0.974	0.974	0.974	0.026	0.974	0.974	0.026	0.026	0.974	0.974	0.974	0.026	0.026	0.974	0.974	0.974



## 2.5 Finite mixture models for SNP arrays

Finite normal mixture models with  $k$  components have densities of the form

$$f(y_j) = \sum_{i=1}^k \pi_i \phi(y_j; \mu_i, \Sigma_i).$$

They are nowadays applied in such diverse areas as biometrics, genetics, medicine, and marketing finance. There exist various features of finite mixture distributions that render them useful in statistical modelling. First, finite mixture distributions arise in a natural way as a marginal distribution for statistical models involving discrete latent variables such as clustering. On the other hand, mixture models can capture many specific properties of real data such as multimodality and skewness. They are flexible in modeling and easy to implement. Applying mixtures to the SNP copy number context, we have

$$f(x) = p_L N(x; \mu_L, \sigma_L^2) + p_N N(x; \mu_N, \sigma_N^2) + p_G N(x; \mu_G, \sigma_G^2)$$

where  $p_L, p_N, p_G$  are proportions of loss, normal and gain and they sum to one.

There are several standard parameter estimation methods for mixture models assuming iid data, such as the EM algorithm and standard Bayesian approaches. In brief, for the EM algorithm we introduce unobserved allocation variables. In the E-step, we take conditional expectations of the allocation variables given the data. In the M-step, we calculate ML estimates of proportions and distributions parameters. In the Bayesian approach we assign all unknown variables (conjugate) priors and use a Gibbs sampler to simulate posterior distributions. I applied these two algorithms to real data. The results for one case are shown in Figure 2.10. The clusters are very noisy since these two algorithms assume independent data while here SNP copy numbers are correlated. As with CGH data, mixture models for correlated data are needed for SNP based copy number estimation.

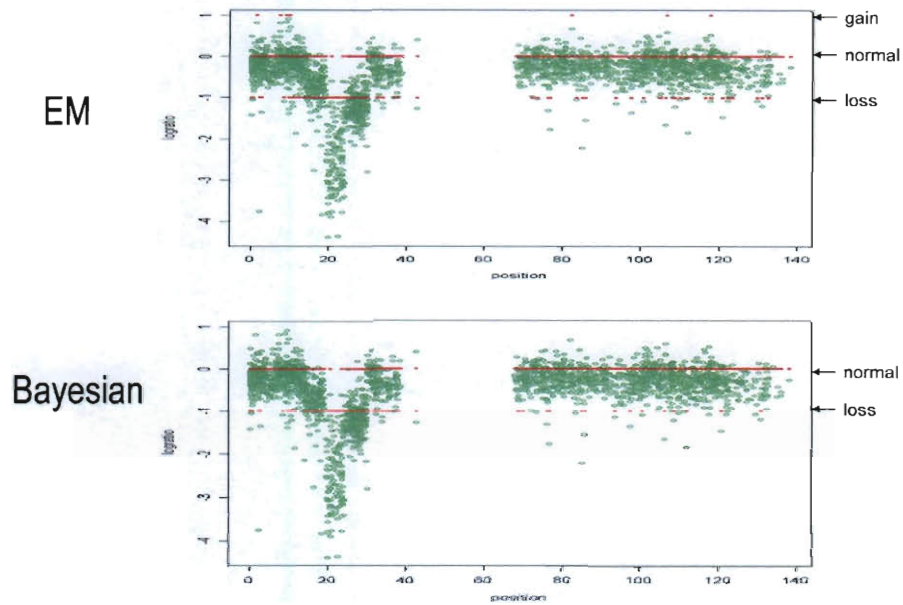


Figure 2.10: SNP copy number inference based on normal mixtures implemented with EM and Bayesian algorithms. Both algorithms assume (spatially) independent log-ratios.

## 2.6 Gaussian Markov random field

Gaussian Markov random fields have been widely used in spatial statistics and Bayesian image analysis, where they are intended to describe dependencies between random variables at fixed sites in Euclidean space. The main appeal of these distributions is in the Markovian interpretation of their full conditionals.

Suppose that the random vector  $X = (X_1, \dots, X_n)^T$  has density

$$p(x) \propto e^{-\frac{1}{2}x^T Q x} \quad (2.1)$$

where  $Q$  is an  $n \times n$  positive definite symmetric matrix. It follows that

$$X_i | x_{-i} \propto N\left(\sum_j \beta_{ij} x_j, k_i\right) \quad (2.2)$$

where  $\beta_{ii} = 0$ ,  $\beta_{ij} = -Q_{ij}/Q_{ii}$ ,  $i \neq j$ , and  $k_i = 1/Q_{ii} > 0$ . The symmetry of  $Q$  requires that

$$\beta_{ij} k_j = \beta_{ji} k_i \quad (2.3)$$

Note that  $i$  and  $j$  are "neighbors" if and only if  $\beta_{ij} \neq 0$ , in which case we write  $i \sim j$ . The variable  $x$  on the right-hand side of (2.1) can be replaced by  $x - \mu$ , where  $\mu$  is an arbitrary real  $n$ -vector, to allow for location shift.

Positive definiteness of  $Q$  may need to be checked on an individual basis but the identity

$$x^T Q x = \sum_i Q_{i+} x_i^2 - \sum_{i < j} Q_{ij} (x_i - x_j)^2 \quad (2.4)$$

where subscripts  $+$  denote summation over replaced indices, implies that a sufficient condition is that the  $\beta'_{ij}$ s are all non-negative and  $\beta_{i+} \leq 1$  for all  $i$ , with strict inequality for at least one  $i$ . When the specification of  $p(x)$  is based on a precision matrix  $Q$ , rather than on the dispersion matrix  $V = Q^{-1}$ , the model is usually referred to as a conditional autoregressive model.

Since the value of any  $X_i$  only depends on its neighbors, neighboring  $X_i$ 's have similar values. If we consider a line, representing a chromosome, then by using GMRF, we can borrow strength across SNP locations. In this case, neighbors are simply  $a_1$  SNPs on the left and  $a_2$  SNPs on the right.

## 2.7 Proposed Method - Bayesian model for copy number estimation

Generally, Bayesian estimates are more accurate than frequentist estimates. Another issue is the fact that the sample of cancer cells is always contaminated by normal cells. The larger percentage of normal cells present, the more difficult it is to infer copy number aberrations for the tumor cells; the log-ratios tend to shrink to zero. As of this writing, we are not aware of any method that accounts for normal cell contamination and gives integer copy numbers. For these reasons, there is a need for a novel copy number approach to address these issues.

Here we propose a Bayesian spatial normal mixture model for inferring SNP-based integer copy number. Bayesian mixture models were used by Broet and Richardson [2006] for CGH-based copy number estimation. There the authors considered a three-state (loss/normal/gain) mixture model and introduced a spatial structure to reflect correlated segments (e.g. BACs). Spatial correlation was induced through the weights of the mixture via Markov random fields. In our approach, instead of considering three states, we allow for an unknown number of mixture components and achieve inference using a reversible jump Markov chain Monte Carlo method, because the number of components is unknown. As in Broet and Richardson [2006] we use Markov random fields to account for correlated neighboring SNPs. In contrast to models that incorporate HMMs to infer integer copy numbers, our modeling approach uses information (neighboring SNPs) on both sides of a SNP. In addition, we account for

cell contamination by shrinking the theoretical copy number log-ratios towards zero. The implementation only requires ordered (normalized) log-ratios and, therefore, may be applied to data from any platform suitable for copy number estimation.

Let  $y_i$  be the preprocessed log-ratio of SNP  $i$  ordered along the chromosome. Following the notation of Fernandez and Green [2002], we consider a normal mixture model with  $k$  unknown components corresponding to  $k$  copy numbers,

$$p(y_i|k, \omega, \mu, \sigma^2) = \sum_{j=1}^k \omega_{ij} N(y_i|\mu_j, \sigma_j^2) \quad (2.5)$$

where  $\mu = (\mu_1, \dots, \mu_k)$  and  $\sigma^2 = (\sigma_1^2, \dots, \sigma_k^2)$  represent the vectors of means and variances of the  $k$  components. The matrix of weights  $\omega = (\omega_{ij})$  is such that  $0 \leq \omega_{ij} \leq 1$  and  $\sum_{j=1}^k \omega_{ij} = 1$ , for all  $i$ . In our application the components represent the true copy numbers (i.e., copy number equals to 0, 1, 2, 3, ...). Given a chromosome with  $n$  SNPs, let  $z_1, \dots, z_n$  be the allocation variables, indicating to which mixture component SNP  $i$  belongs. These are marginally distributed according to a multinomial distribution with

$$p(z_i = j|k, \omega, \mu, \sigma^2) = \omega_{ij}, \quad (2.6)$$

for  $j = 1, \dots, k$ . Since copy number aberrations tend to occur over contiguous segments, we impose that neighboring SNPs have similar multinomial probabilities of belonging to the copy number classes. To this end, for  $k$  components we introduce  $k$  independent Gaussian Markov random fields (GMRF),  $x_j = (x_{ij}, i = 1, \dots, n)$ , for  $j = 1, \dots, k$ , see Fernandez and Green [2002] and Broet and Richardson [2006]. Each GMRF,  $x$ , is assumed to have joint density

$$f(\mathbf{x}_j|h) = c(h) \exp\left[-\frac{1}{2}\left\{h \sum_{i \sim i'} (x_{ij} - x_{i'j})^2 + \sum_{i=1}^n x_{ij}^2\right\}\right] \quad (2.7)$$

where  $\sum_{i \sim i'}$  denotes the sum over all neighbors of  $i$  and where  $c(h) = (2\pi)^{-\frac{n}{2}} \prod_{i=1}^n (1 + h g_i)^{\frac{1}{2}}$ , with  $g_1, \dots, g_n$  the eigenvalues of a matrix  $A = (a_{ii'})$  coding the adjacencies,

with  $a_{ii} = \nu_i$  (number of neighbors of location  $i$ , and off-diagonal elements  $a_{ii'} = -1$  if locations  $i$  and  $i'$  are neighbors and zero otherwise). For each  $j$ ,  $\mathbf{x}_j$  is a realization of spatial correlation. For neighbors of a certain SNP, we use  $a_1$  SNPs on the left and  $a_2$  SNPs on the right. Since the conditional distribution of  $x_i$  only depends on its neighbors, neighboring  $x_i$ 's will tend to have similar values. The parameter  $h$  is non-negative and controls smoothing among neighboring SNPs: large values of  $h$  induce smoother realizations in the GMRF, and as  $h \rightarrow 0$  independent realizations take place. For the weights,  $\omega_{ij}$ , there are constraints that they are non-negative and sum to one, so we borrow spatial correlation from the GMRF's by defining logistic transformations of the type

$$\omega_{ij} = \frac{\exp(x_{ij}/\phi)}{\sum_{l=1}^k \exp(x_{il}/\phi)}, \quad j = 1, \dots, k, \quad i = 1, \dots, n, \quad (2.8)$$

where  $\phi$  is a positive scaling factor specified by the user. As  $h$  increases, all the  $x'_{ij}$ s shrink toward zero.  $\phi$  can compensate for this. In the simulation study in the next section, we will investigate robustness of the results to different values of  $\phi$  and varying number of neighbors.

### 2.7.1 Prior distributions

In this section we discuss the prior distributions for the model parameters, including the number  $k$  of mixture components, the normal mixture means and variances, and the smoothing parameter  $h$ .

#### 1. *Number of mixture components, $k$*

The number of components  $k$  is given a truncated Poisson distribution with a mean of 2,

$$k \sim TPoisson\{1, \dots, k_{max}\}, \quad (2.9)$$

with  $k_{max}$  a pre-specified integer representing the largest number of components.



The probability mass function is

$$g(k = t) = C \frac{2^t e^{-2}}{t!} \quad \text{if } 1 \leq k \leq k_{max} \quad (2.10)$$

where

$$C = \frac{1}{\sum_{i=1}^{i=k_{max}} \frac{2^i e^{-2}}{i!}} \quad (2.11)$$

We take  $k_{max} = 7$  for illustration purposes, corresponding to copy numbers 0, 1, 2, 3, 4, 5, and  $> 5$ . Here 7 is arbitrary, and we can use any positive value that makes sense for the data under consideration. If  $k_{max} = 7$ , it means we will have at most 7 classes. And the classes are not necessarily consecutive numbers.

## 2. Normal mixture means

We deviate from the approach of Fernandez and Green [2002] by constructing  $k_{max}$  uniform distributions,  $\{\nu_j = U(a_j, b_j), j = 1, \dots, k_{max}\}$  for the copy number class means, and assuming that each component mean  $\mu_i$  follows one of these uniform distributions, independently of all others. The uniform interval boundaries are very important. We choose the intervals to be non-overlapping and to contain the theoretical copy number values. According to Nannya *et al.* [2005], the observed mean values for the 7 components without contamination are approximately  $-1.24, -.49, 0.365, .657, .899$  and  $1.106$ , for copy numbers 0, 1, 2, 3, 4, 5 and  $> 5$ , respectively. However, due to normal cell contamination, the true log-ratios tend to shrink towards zero. Accordingly, we chose the intervals to contain the theoretical mean values and then set their extreme values to obtain disjoint intervals. In particular, results reported here were obtained using the following intervals:  $(-2, -.8), (-.6, -.25), (-.05, .05), (.15, .4), (.45, .65), (.75, .9), (.95, 1.3)$ , corresponding to copy numbers 0, 1, 2, 3, 4, 5,  $> 5$ , respectively. These intervals are the default values we use in the application and have worked well in most cases. Our results did not show sensitivity to the actual values we used for the extremes of the intervals; i.e., other disjoint sets of intervals worked well, too.

*Remark 1:* Due to normal cell contamination, the true log-ratios tend to shrink toward zero, and in practice some degree of normal cell contamination tends to be present. We thus decided to center the uniform distributions closer toward the null value of zero rather than at the theoretical means given above, except for  $CN=0$  and  $CN > 5$ . These exceptions are largely due to where we wanted to locate the respective uniform support; see Remark 3 below.

*Remark 2:* Moving the uniform intervals closer to zero resulted in some of the theoretical means being located close to a uniform boundary. For example, for  $CN=5$ , the theoretical mean of .899 is just inside the right boundary of .9. This does not cause a problem of misclassification since normal cell contamination brings the mean closer to the left boundary.

*Remark 3:* We also varied the length of the uniform intervals since the log scale makes the consecutive theoretical values become increasingly closer to each other; the consecutive pairwise distances between the theoretical means from -1.24 to 1.106 are .75, .49, .365, .292, .242, .207. If the uniform intervals were forced to be of equal length we would have either relatively short non-overlapping intervals or over-lapping long intervals. Since the uniform intervals are not of equal length, the gaps between the intervals are unequal, as well.

In cases where the percentage ( $p$ ) of normal cells is known or approximately known, then such intervals can be chosen to be centered at

$$\log_2 \left[ \frac{2p + j(1 - p) + b}{2 + b} \right],$$

for any copy number  $j$ , with background factor  $b$ , (Nannya *et al.* [2005]) and then choosing the length of the intervals so that the  $k_{max}$  intervals are non-overlapping.

### 3. Normal mixture variances

We assign an inverse gamma prior distribution to  $\sigma_j^2$ . In the application we center this distribution on 0.2 (from empirical data) and induce a vague specification by letting the variance be large.

#### 4. *Smoothing parameter*

We assign  $h$  a uniform distribution with a wide range,  $h \sim U(0, h_{max})$ , with  $h_{max} = 1,000,000$ , to induce smooth realizations.

I will provide further discussion of these prior selections below in the next section in the context of the simulations and real data applications.

## 2.7.2 Posterior inference

We employ MCMC with reversible jump to achieve posterior inference. Below is a brief step-by-step description of the method and additional details are given in Appendix A.

First, a brief introduction to Gibbs and Metropolis-Hastings sampling is presented. In Bayesian analysis, we give each unknown parameter a prior distribution. Combined with the likelihood, we get the posterior distribution for each parameter. If it's a known distribution, we can just sample from it, which is called Gibbs sampler. If the posterior distribution is not any known distribution, we apply Metropolis-Hastings algorithm. We draw samples from an arbitrary distribution  $q$ . Suppose the new draw is  $x'$  and the current value is  $x$ , define the acceptance ratio as  $\alpha = \min(1, \frac{p(x')q(x|x')}{p(x)q(x'|x)})$ . Then with probability  $\alpha$ , we accept the new draw, and with probability  $1 - \alpha$  we reject the new draw and keep the previous one.

- **Updating  $k$ :** This step causes creation or deletion of mixture model components, therefore requiring the sampler to jump between subspaces with different dimensions. To implement the sampler, we use reversible jump MCMC (RJMCMC), see Green [1995] and Richardson and Green [1997]. We update  $k' = k+1$

with probability  $b_k$ , and  $k' = k - 1$  with probability  $1 - b_k$  ( $b_1 = 1$ ,  $b_{k_{max}} = 0$ ,  $b_k = .5$  for  $k = 2, \dots, k_{max} - 1$ ). If  $k' = k + 1$ , we draw a new component  $*$  from the remaining  $k_{max} - k$  components with equal probability, and draw  $\mu_*$  from the corresponding uniform distribution. We also draw  $\sigma_*^2$  and  $x_*$  from the prior distributions. We use the fast sampling algorithm of Rue [2001] to generate a new GMRF,  $x_*$ . We then increase the dimensions of the vector parameters  $\mu' = (\mu, \mu_*)$ ,  $\sigma^{2'} = (\sigma^2, \sigma_*^2)$ , and  $x' = (x, x_*)$  and accept the new component with probability:

$$\min \left( 1, \frac{(1 - b_{k+1}) \frac{1}{k+1} g(k+1)}{b_k \frac{1}{k_{max}-k} g(k)} \prod_{i=1}^n \frac{\sum_{j=1}^{k+1} \omega'_{ij} N(y_i | \mu'_j, \sigma_j'^2)}{\sum_{j=1}^k \omega_{ij} N(y_i | \mu_j, \sigma_j^2)} \right). \quad (2.12)$$

If  $k' = k - 1$ , we instead randomly pick a component from the discrete uniform distribution on  $\{1, \dots, k\}$  and remove  $\mu_*, \sigma_*^2, x_*$  from  $\mu, \sigma^2, x$ . Similarly, the acceptance probability is

$$\min \left( 1, \frac{b_{k-1} \frac{1}{k_{max}-(k-1)} p(k-1)}{(1 - b_k) \frac{1}{k} p(k)} \prod_{i=1}^n \frac{\sum_{j=1}^{k-1} \omega'_{ij} N(y_i | \mu'_j, \sigma_j'^2)}{\sum_{j=1}^k \omega_{ij} N(y_i | \mu_j, \sigma_j^2)} \right). \quad (2.13)$$

- **Updating  $x$ :** We update each location using a Metropolis-Hastings step, see Metropolis *et al.* [1953] and Hastings [1970]. We perform these  $n$  updates sequentially.
- **Updating  $h$ :** We use a Metropolis-Hastings random walk with a proposal defined by a truncated normal distribution,  $h' \sim TN(h, \sigma_h^2) I(0 \leq h' \leq h_{max})$ . In applications we chose  $\sigma_h$  to have acceptance ratios between 40% and 70%.
- **Updating allocations:** Using a Gibbs step, we draw the  $n$  allocations independently from

$$p(z_i = j | \mathbf{y}, \mathbf{k}, \mu, \sigma^2, x, h, \phi) \propto \omega_{ij} N(y_i | \mu_j, \sigma_j^2) I[j \in \{1, \dots, k\}]. \quad (2.14)$$

- **Updating  $\mu, \sigma^2$ :** For each  $\mu_i$ , we find one of the  $k_{max}$  intervals which has the largest posterior probability, and sample  $\mu_i$  from a truncated normal distribution at this interval. In the iterations it may happen that two or more  $\mu_i$ 's are sampled to the same interval. In this case, we combine these components and update  $k$ . The new  $\mu_i, \sigma_i, x_i$  for the newly formed component are taken to be the weighted sum of the previous ones, where the weights are the sample sizes. We then redefine  $z$  and calculate  $\omega$ . We draw  $\sigma_i^2$  from its full conditional, which is again a inverse gamma distribution.

For posterior inference, the primary parameters of interest are the weights,  $\omega$ 's. We propose an allocation rule as follows: at each iteration we record the probability of each SNP belonging to each of the  $k_{max}$  components (we assign zero if a component is empty). After the MCMC is done, we average all the  $\omega$ 's and assign a SNP to the component that has the largest probability. We check reproducibility of the clustering with different starting values.

The run-times of the various copy number algorithms can range from less than a minute to days depending on the algorithm and the probe density of the array platforms. When applied to newer high-density arrays almost all methods have relatively high run-times (Pique-Regi *et al* [2008]). Reversible jump MCMC methods such as ours and RJACGH (Rueda *et al* [2007]) tend to be computationally expensive. Our current implementation may require several hours to more than 1 day per chip. However, our current version is implemented in MatLab and we have not attempted to optimize the code. Programming in some version of C and parallel computing by chromosome and/or chromosome arm will likely significantly reduce the time.

## 2.8 Simulation Study

We first investigate the performance of our model through simulation experiments. In the next Section we compare our method with an alternative method in the context

of actual tumor samples from leukemia and ependymoma cancers.

We conducted two sets of simulations studies. The first set was designed to examine the influence of hyperparameters in the prior specifications: the scaling parameter,  $\phi$ , of the logistic transformation for the GMRF and the number of smoothing neighbors,  $nb$ . Based on the results of the first set of experiments we then conducted a second set of experiments by setting these two parameters at fixed (default) values in order to assess performance of our algorithm.

In the first set of simulation studies we found that a small range of  $\phi$  was suitable over different configurations. In particular, we investigated sensitivity by choosing different values in the set  $\{.005, .01, .5, .1\}$ . For the number of neighboring SNPs (on either side) over which to smooth in the GMRF, we considered the two values, 1 and 4, for of total of 2 or 8 neighbors for each SNP. Boundary SNPs at the ends of the chromosomes simply used fewer SNPs. Based on the results of the first set of experiments we then conducted a second set of experiments by setting these two parameters at fixed values,  $\phi = 0.01$  and  $nb = 4$ , and varying the signal-to-noise ratio and location of the copy number breakpoints. We also varied the number of SNPs constituting the aberration regions. In all simulations, the standard deviation ( $\sigma_h$ ) of the proposal distribution (see section 2.7.2) to update the smoothing parameter,  $h$ , was chosen so that acceptance ratios (see section 2.7.2) would be between 40% and 70%. For all cases reported we used 50,000 sampling draws for inference after a 50,000 iteration burn-in period.

For the first set of simulations, we simulated 8 data sets representing 8 sets of (normalized) SNP log-ratios. Four data sets represent the case where there is a clear separation among three contiguous segments whose ordered copy numbers are 2, 4, and 2, corresponding to normal/gain/normal. We call this scenario the “non-overlap” case. The four versions differ by how many SNPs are in the gain segment: 100, 50, 25, 10. The second set of four data sets are analogous to the first set except for the fact that the log-ratios of the gain segment are not well separated from the normal

(CN=2) log-ratios on either side. We call this case the “overlap” case. In all cases examined we computed misclassification numbers and false negative fractions (i.e., numbers of false negatives divided by the numbers of non-normal SNPs). We comment on some of the results below.

#### *Non-overlap Case*

In the non-overlap scenario the log-ratios corresponding to CN=2 SNPs were independently drawn from a  $N(0, .1^2)$  distribution. The log-ratios corresponding to CN=4 SNPs were drawn from a  $N(.6, .1^2)$ , for all four SNP sample sizes of 10, 25, 50, and 100. We pick the mean to be .6 to make the components non-overlapping. The SNR here is 6. For this case we obtained excellent results and observed no sensitivity to the parameter  $\phi$  and to the number of neighbors.

#### *Overlap Case*

In the overlap case the log-ratios of the normal copy number SNPs were sampled from a  $N(0, .15^2)$ , and the gains from a  $N(.3, .15^2)$ . Here SNR is 2. For this case we obtained excellent results in all cases except one. When  $\phi = .1$  the algorithm can detect the three cases with 100/50/25 SNPs in the middle, but cannot detect the short segment of 10 SNPs.

From the first set of simulation experiments, we see that the results are not sensitive to the number of neighbors and the value of the parameter  $\phi$  as long as it's small. So in the second set of simulation studies, we fix the number of neighbors to be 4 on either side and  $\phi = .01$ .

For the second set of simulations we designed two patterns of copy number segments. For each pattern, we simulated four scenarios of SNP log-ratios. In practice, the log-ratios would be suitably normalized. The four scenarios are different configurations of true copy number, signal-to-noise ratio (SNR), normal cell contamination, and number of SNPs within the CNA region. For each scenario we report misclassification, false-negative and false-positive rates. All rates in Table 2.2-Table 2.4 are based on 50 sample replicates.

The *misclassification* rate reported is defined as  $P(CN \neq j \mid \text{true } CN = j)$ , for  $j \neq 2$ . For the special case  $j = 2$  we obtain the *false-positive rate*,  $FP = P(CN \neq 2 \mid \text{true } CN = 2)$ . The *false-negative rate* is defined as the chance of a true loss or gain classified as a normal copy number,  $FN = P(CN = 2 \mid \text{true } CN \neq 2)$ . We do not find it very useful to cite global rates since each depends on several factors, including the true CN, signal-to-noise ratio (SNR), normal cell contamination, and number of SNPs within the CNA region. We therefore report misclassification, false-negative and false positive rates given various combinations of these parameters. Other authors (e.g., Pique-Regi *et al* [2008]) define performance accuracy by breakpoint detection. This results in slightly different definitions of false-positive and false-negative rates than we do here. Since our model is based on mixture components corresponding to integer copy numbers it makes more sense for us to consider more specific false-negative and false-positive rates. As shown below, these rates also depend on factors other than true copy number.

A number of authors have used the simulation data of Willenbrock *et al* [2005] to assess their proposed copy number algorithms for aCGH data. However, we are specifically interested in how well SNP data performs. We, therefore, generated our own simulation data since the Willenbrock and Fridlyand simulated data was generated to emulate real tumor data from the aCGH copy number algorithm, DNACopy. As such, their simulated data represents levels, variance, and breakpoints (segmentation) specific to aCGH data analyzed with a specific algorithm. We also note that



our simulations generated simple text files of log-ratios. Therefore, we were unable to compare our method to those whose software implementation requires special data files, such as Affymetrix CEL files. The real data studies, however, did allow for such comparisons as we were able to obtain log-ratios from their analysis. In short, the simulations were for assessing our own method and the real data with validation were for performance assessment under real conditions and comparative purposes.

Table 2.2 shows misclassification rates (%) for eight different scenarios. Table 2.3 and Table 2.4 show false-negative and false-positive rates, respectively. We first discuss the misclassification (MC) rates in Table 2.2.

**Scenarios 1-4:** These scenarios assume the following ordered copy number segments with number of SNPs given in parentheses: 2(10), 3(5), 2(50), 1(10), 2(50), 3(20), 2(50), 3(40), 2(10). The widths of the copy number segments (5, 10, 20, 40, 50) correspond to those considered by Rancoita et al. Rancoita *et al* [2009]. The SD and SNR are given on the log2 ratio scale under a true CN of 3. Since in this table we report misclassification rates, we do not show the segments corresponding to a true copy number of 2, which would be the false-positive rate (Table 2.4). The rows are ordered by segment as given above, excluding segments with a normal copy number. Figure 2.11 shows a typical data set under Scenario 1 in which the SNR of 7.3 leads to clearly non-overlapping log-ratios across the segments. In this case, the MC rate is 0% independent of CN aberration and number of SNPs defining the respective segments. Scenarios 2, 3, and 4 have increasingly smaller SNRs and for a given true CN aberration the MC rate increases with decreasing SNR (left to right across columns). Figure 2.12 shows a data set under Scenario 2 with a SNR of 2.4. The overlap between CN classes is mild, but clear change points can still be observed when there are at least 10 SNPs. Here, a few of the CN=3 cases between SNPs 11-15 are classified as normals. Conversely, at about SNPs #190 and 250, normal CNs are classified as CN=3. The largest MC rate (16%) under Scenario 2 is that corresponding to a segment with true CN=3 and 5 SNPs. The other three cases under

Scenario 2 with at least 10 SNPs have a MC rate of no more than 5%. Figures 3A and 3B show two data sets under Scenario 3 with a SNR under 2, namely  $\text{SNR} = 1.8$ . Figure 2.13A shows correct classification of 4 of 5  $\text{CN}=3$  cases between SNPs 11-15, while Figure 2.13B shows all 5 of these  $\text{CN}=3$  cases misclassified as normals. However, Figure 3A shows more misclassifications of the  $\text{CN}=3$  cases between SNPs 230 and 240 than that in Figure 3B. With at least 10 SNPs in a segment, the MC rate is no more 11% under Scenario 3. Under Scenario 4 the SNR is 1.5 and as with Scenario 3 ( $\text{SNR} = 1.8$ ) the MC rate is about 50% when only 5 SNPs define the segment. With a SNR as small as 1.5, a relatively large ( $\geq 20$ ) number of SNPs are needed to accurately classify a copy number.

**Scenarios 5-8:** These represent the following ordered copy number segments with number of SNPs in parentheses: 2(10), 4(5), 2(50), 3(10), 2(50), 0(20), 2(50), 3(40), 2(10). As with Scenarios 1-4, for a given combination of CN and number of SNPs in the segment, the MC rate increases with decreasing SNR. Segments with a larger number of SNPs also lead to smaller MC rates than those with fewer SNPs. One interesting comparison is that between row 1 of Scenarios 1-4 ( $\text{CN}=3$  with 5 SNPs) with row 1 of Scenarios 5-8 ( $\text{CN}=4$  with 5 SNPs). Figure 2.14 shows a sample data set from Scenario 8 and there we observe that all five SNPs with  $\text{CN}=4$  at positions 11-15 are classified as  $\text{CN}=3$ . Examining the misclassifications across all 50 replicates for this configuration we found that the vast majority of SNPs with  $\text{CN}=4$  were labelled as a 3; hence, the misclassification rate of 98%. Note that the false-negative rate for this situation (Table 2.3, row 1, Scenario 8) was only 6%. On the other hand, the MC rate under Scenario 4 with  $\text{CN}=3$  with 5 SNPs was 50%, approximately half that for  $\text{CN}=4$  in Scenario 8. In general, larger copy number aberrations are more difficult to correctly identify than smaller ones. Indeed, the log scale shrinks the larger copy number ratios toward smaller ones, leading to misclassifications. Line 3 shows MC rates under a true copy number of 0. Figure 2.14 shows how distinct this aberration is from its neighbors regardless of the size of the SNR; the MC is constantly 0%.

Table 2.3 shows false-negative rates. Except for minor differences, the false-negative rates for Scenarios 1-4 are the same as the broader misclassification rates (Table 2.2). This shows that most of the misclassifications in Scenarios 1-4 were losses and gains that were called normal. Where there are differences between Table 2.2 and Table 2.3, we see that misclassification rates are at least as large as the false-negative rates as we would expect. It is worth noting that the aberrations studied in Scenarios 1-4 are "neighbors" of normal copy number, viz., CN=3 is one additional copy and CN=1 is one less copy. As such, it is not too surprising that the misclassification rates agreed with the false-negative rates. Especially in the presence of normal cell contamination we expect the log-ratios to regress toward the mean value of 0. This is contrast to Scenarios 5-8, which include more extreme aberrations of CN=0 and CN=4. Comparing the misclassification rates (Table 2.2) with the corresponding FN rates (Table 2.3), we see that the latter can be much smaller than the former. Large differences of MC vs FN rates are seen for CN=4 in Scenarios 6 (87% vs 0%), 7 (77% vs 4%), and 8 (98% vs 6%). Taken together this implies that almost all of the misclassifications for CN=4 were called as CN=3 and very few as CN=2. A manual calculation of the calls confirms this conclusion. Smaller differences between MC and FN rates occur in Scenario 6 with CN=3 and 10 SNPs (line 2, Table 2.2 and Table 2.3); the MC rate is 14% and the FN rate is 7%. Here, half of the 14% is due to normal calls and the other half to calls of CN=4. In Scenario 7 with true CN=3 and 10 SNPs the MC rate of 31% is 20% CN=2 (false-negative) and 11% CN=4. Similarly, the MC rates of 9% and 17% for Scenarios 6 and 7 with CN=3 and 40 SNPs (line 4, Table 2.2 and Table 2.3), respectively, are only due to false calls of CN= 2 and CN=4. It is, therefore, seen that when a true copy number of 3 is misclassified, it tends to be called a CN=4 with a smaller percentage of normal calls, CN=2. And, as discussed above, a true CN=4 tends to be called a 3 when misclassified. In this sense, if an investigator is only calling loss/normal/gain, even though misclassifications occur under true copy numbers of 3 and 4, they would both be correctly called as gains with

a small percentage of CN=2 (false-negative) calls. This is at least the behavior of the Bayes mixture model; other methods may apportion the misclassifications differently. In all scenarios (1-8) we observe a misclassification rate and a false-negative rate of 0% for CN=0 and 20 SNPs. No matter the signal-to-noise ratio, the distribution of log-ratios for CN=0 is well separated from the other copy number distributions and its call is constantly correct. For CN=1, the misclassification rates and corresponding false-negative rates are equal, showing that when misclassified this copy number is called a normal (false-negative).

Table 2.4 shows false-positive (FP) rates defined as a true normal copy number being classified as a gain or loss:  $P(CN \neq 2 \mid CN = 2)$ . Since the two patterns of copy number structure differed only in their gain and loss patterns we combined the data for the normal copy number segments. Thus the FP rates are based on 100 replicates instead of 50 as with the MC and FN rates in Tables 2.2 and 2.3. As with the FN rate, for a fixed number of SNPs defining the normal segment, the FP positive rate increases with decreasing SNR. And, for a given combination of SNR and normal cell contamination, the FP rate decreases with an increasing number of SNPs in the segment. Under the most difficult configuration considered, 10 SNPs with a SNR of 1.5 and 20% contamination, the false-positive rate was only 9%.

Rancoita *et al* [2009] compared their mBPCR method with six other methods and found that in general no method, including their own, was able to detect aberrations of width 5-10 probes. Lai *et al* [2005] reached similar conclusions. Use of alternative estimators for a certain covariance parameter led to the detection of these smaller segments, but this was accompanied by dividing larger segments into sub-segments. Our method, too, had trouble with regions defined by only 5 probes, although regions with at least 10 probes were fairly well identified unless the signal-to-noise ratio was on the order of 1.5 or higher.

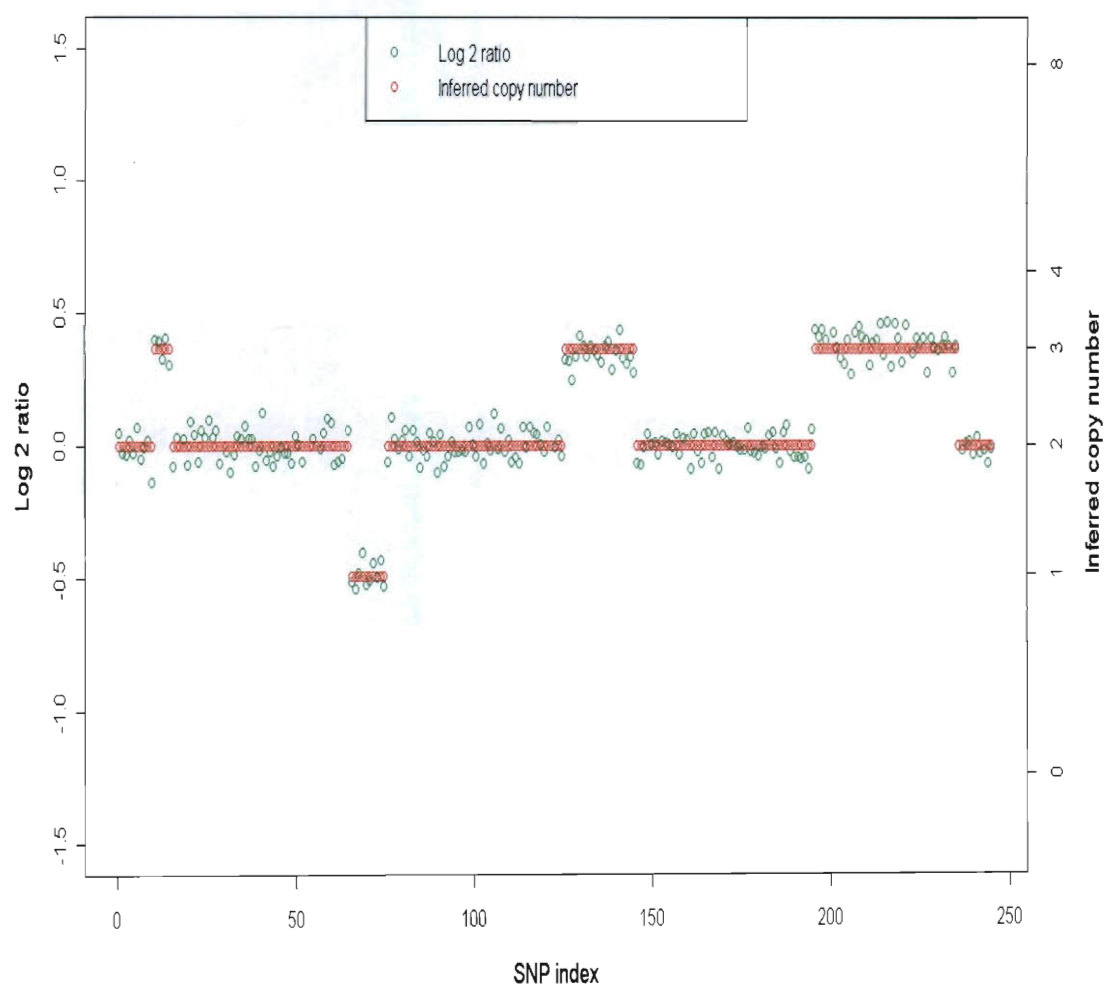


Figure 2.11: Scenario 1

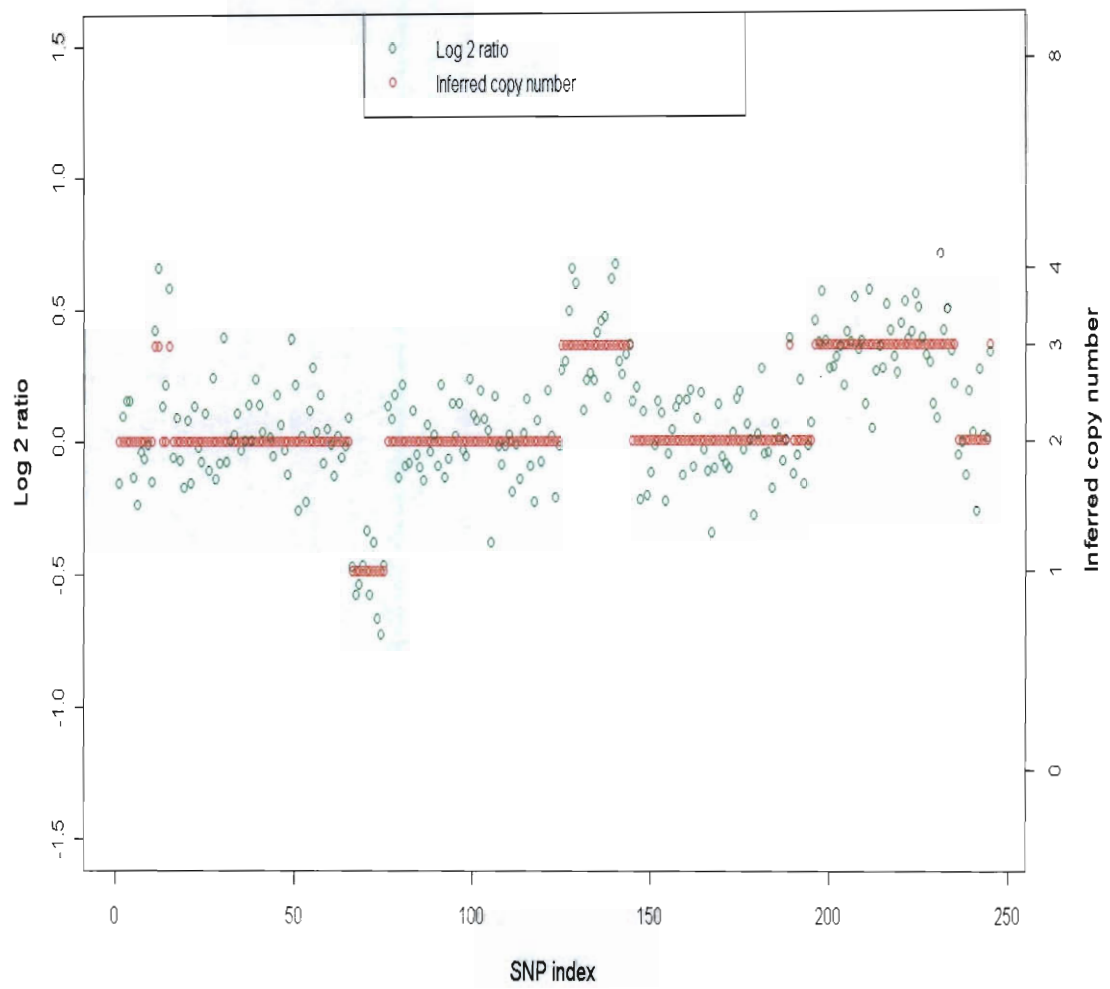


Figure 2.12: Scenario 2

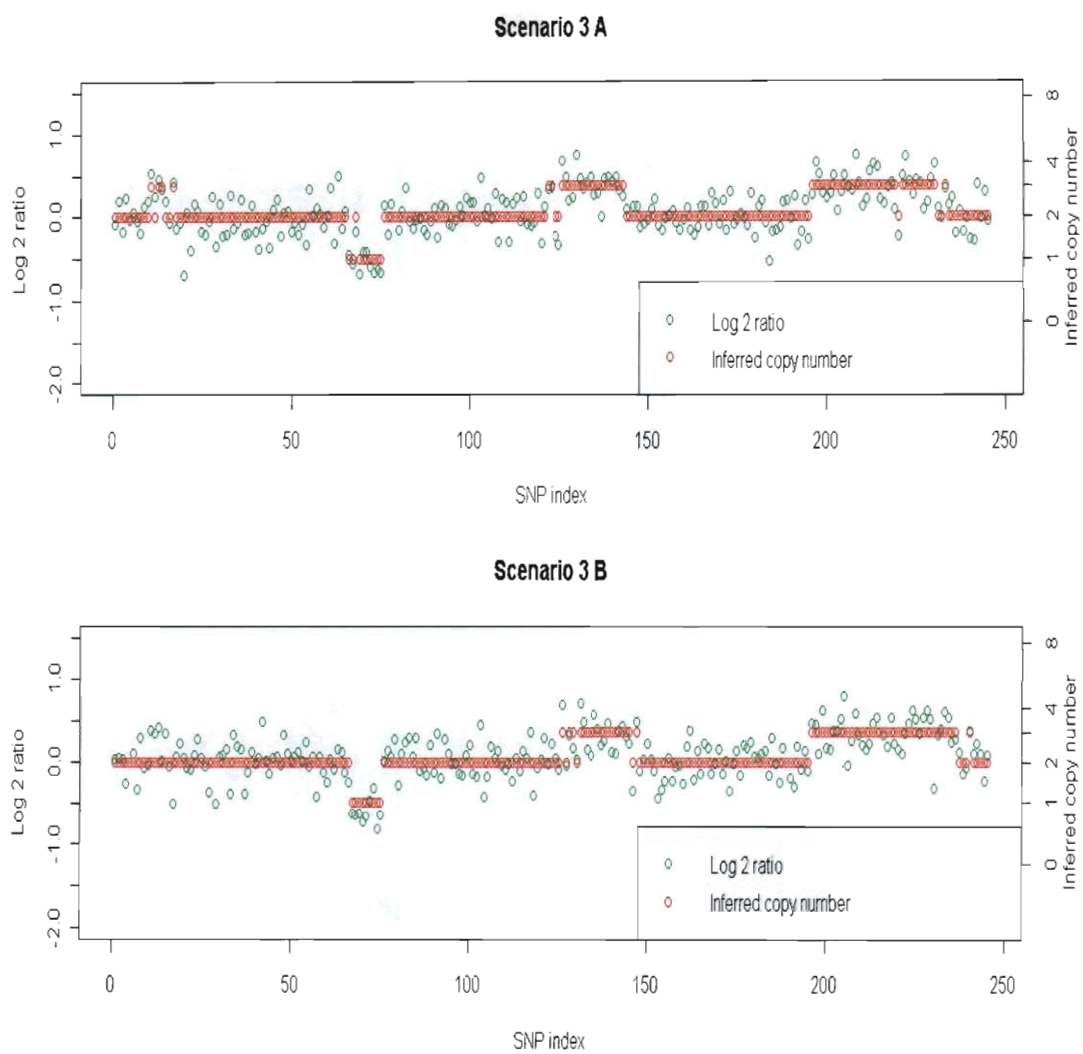


Figure 2.13: Scenario 3

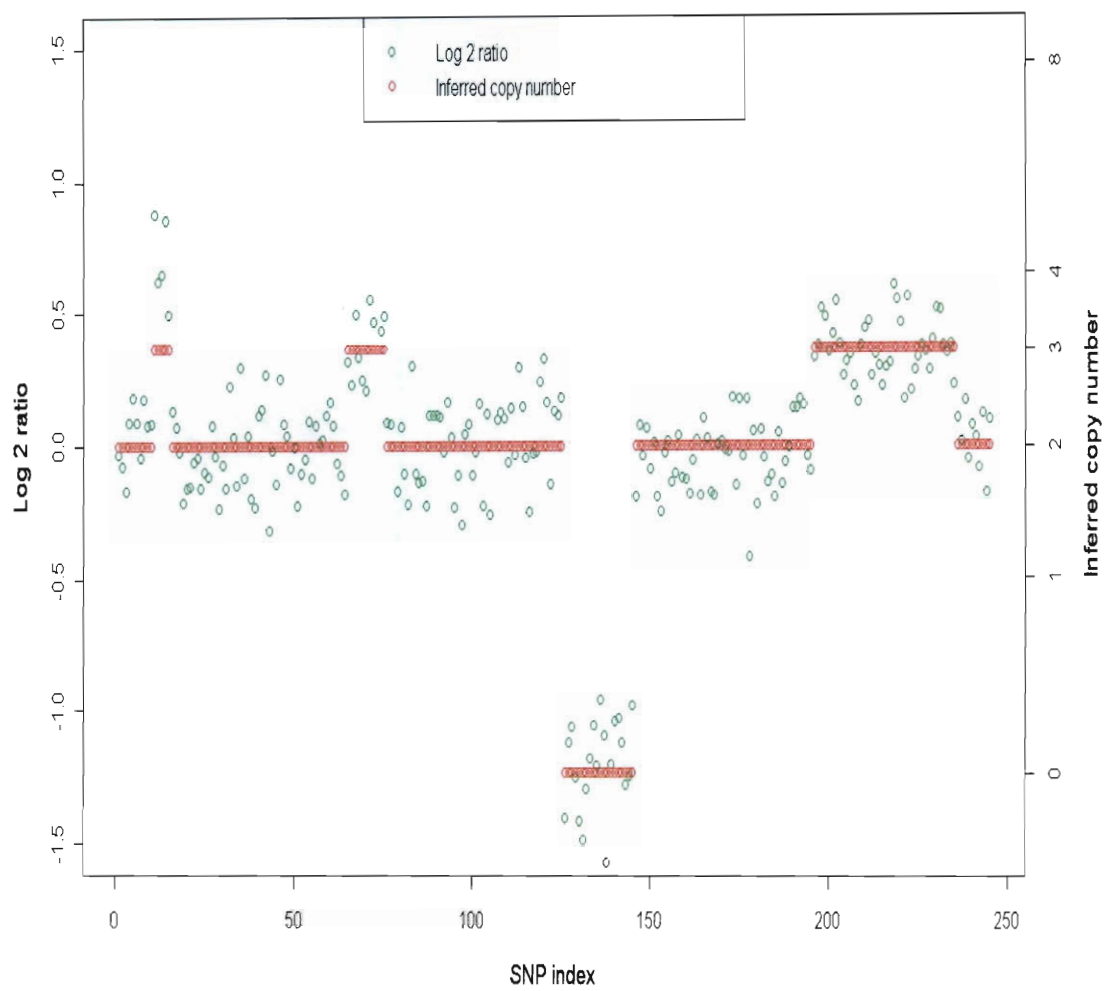


Figure 2.14: Scenario 8



		Scenario 1	Scenario 2	Scenario 3	Scenario 4
CN	# SNP	.05/7.3/0	.15/2.4/0	.2/1.8/0	.2/1.5/20
3	5	0	16	51	50
1	10	0	2	11	37
3	20	0	5	9	11
3	40	0	1	6	6

		Scenario 5	Scenario 6	Scenario 7	Scenario 8
CN	# SNP	.05/7.3/0	.15/2.4/0	.2/1.8/0	.2/1.5/20
4	5	6	87	77	98
3	10	0	14	31	31
0	20	0	0	0	0
3	40	0	9	17	4

Table 2.2: Misclassification rates from simulation study.

The entry is the misclassification rate over 50 replicates of one chromosome. Eight scenarios were simulated and defined by the given combination of true CN, number SNPs within the region of aberration, and SD/SNR/percent contamination. The SD and SNR are given on the log2 ratio scale under a true CN of 3. The true CN profile for Scenarios 1-4 is CN(#SNPs): 2(10), 3(5), 2(50), 1(10), 2(50), 3(20), 2(50), 3(40), 2(10). The true CN profile for Scenarios 5-8 is CN(#SNPs):2(10), 4(5), 2(50), 3(10), 2(50), 0(20), 2(50), 3(40), 2(10).

CN	# SNP	Scenario 1	Scenario 2	Scenario 3	Scenario 4
		.05/7.3/0	.15/2.4/0	.2/1.8/0	.2/1.5/20
3	5	0	16	47	50
1	10	0	2	11	37
3	20	0	5	5	11
3	40	0	1	3	6

CN	# SNP	Scenario 5	Scenario 6	Scenario 7	Scenario 8
		.05/7.3/0	.15/2.4/0	.2/1.8/0	.2/1.5/20
4	5	0	0	4	6
3	10	0	7	20	31
0	20	0	0	0	0
3	40	0	1	4	4

Table 2.3: False negative rates from simulation study.

The entry is the false-negative rate over 50 replicates of one chromosome. See Table 2.2 for details.

CN	# SNP	Scenario 1/5	Scenario 2/6	Scenario 3/7	Scenario 4/8
		.05/7.3/0	.15/2.4/0	.2/1.8/0	.2/1.5/20
2	10	0	3	7	9
2	50	0	1	2	2

Table 2.4: False positive rates from simulation study.

The entry is the false-positive rate over 50 replicates of one chromosome. See Table 2.2 for details.

## 2.9 Real data application

To further assess the Bayes mixture model, we analyzed Affymetrix 250K array data from two sets of patients: aneuploidy and ependymoma. Data were obtained from Texas Children's Hospital, Houston, TX. A brief description of the data is given below.

- Tumor samples with known aneuploidy: Anonymized tumor samples were collected from the clinical cytogenetics laboratory at Texas Childrens Hospital as discarded materials after clinical diagnostic evaluation was completed. All samples were chosen for aneuploidy identified by cytogenetics. The following are the cytogenetic diagnosis for each case: Sample 688 (47 XY: +X, del(6)(q12q21), 7p-/q+, del(9)(p21)(p16)): Sample 52 (51 XY, +5, +7, +8, +8, +13): Sample 406 (45 XX, -13): Sample 282 (47 XY, t(10;12)(q24;p13); +21).
- Ependymoma is the third most common malignant pediatric brain tumor, with over 50% of cases arising in children younger than 5 years of age. Numerical and structural chromosomal abnormalities in ependymoma were identified in early cytogenetic studies involving karyotypic analysis and comparative genomic hybridization. Common genetic abnormalities in ependymoma involve losses on chromosomes 1p, 3, 3q, 9p, 10q, 13q, 16p, 17, 21 and 22q and gains of 1q, 4q, 5, 7, 8, 9, 12q, and 20 (Taylor [2009]). CGH data and 250k Affymetrix SNP data were obtained from Texas Children's Hospital.

In addition to comparing our results with the popular CNAG software, we also provide biological validation. Some of these cases, in fact, had karyotyping and FISH data for validation. Others were validated using quantitative PCR on selected regions and CGH data. One important feature of the CNAG software used to estimate copy number is the fact that it adjusts the observed log-ratios for variation in GC content across the probes. Integer copy numbers are subsequently inferred from the GC adjusted log-ratios using a hidden Markov model. To make comparable comparisons

between the Bayes and the CNAG methods we applied the Bayes model to the GC adjusted log-ratios from CNAG. One relatively recent issue arising in the analysis copy number aberration detection is the so-called “genome wave” Marioni *et al* [2007], Diskin *et al* [2007], a genome-wide spatial autocorrelation pattern in signal intensity data that may be confounded with the copy number profile across a chromosome. As a result the genome wave may lead to inflated false-positive rates in copy number calls. The genome wave has been consistently detected in both CGH and SNP based platforms. Diskin *et al* [2007] and the references therein describe possible genomic features underlying the wave effect and pre-processing methods to remove the wave effect prior to the analysis of copy number. It has been fairly well established that an adjustment for GC content largely removes the wave effect from the signal intensities (Diskin *et al* [2007]). Since we are using GC adjusted log-ratios from CNAG for the real data application we did not expect to observe a wave effect in our data and indeed none was present as shown in Figures 2.15-2.9.

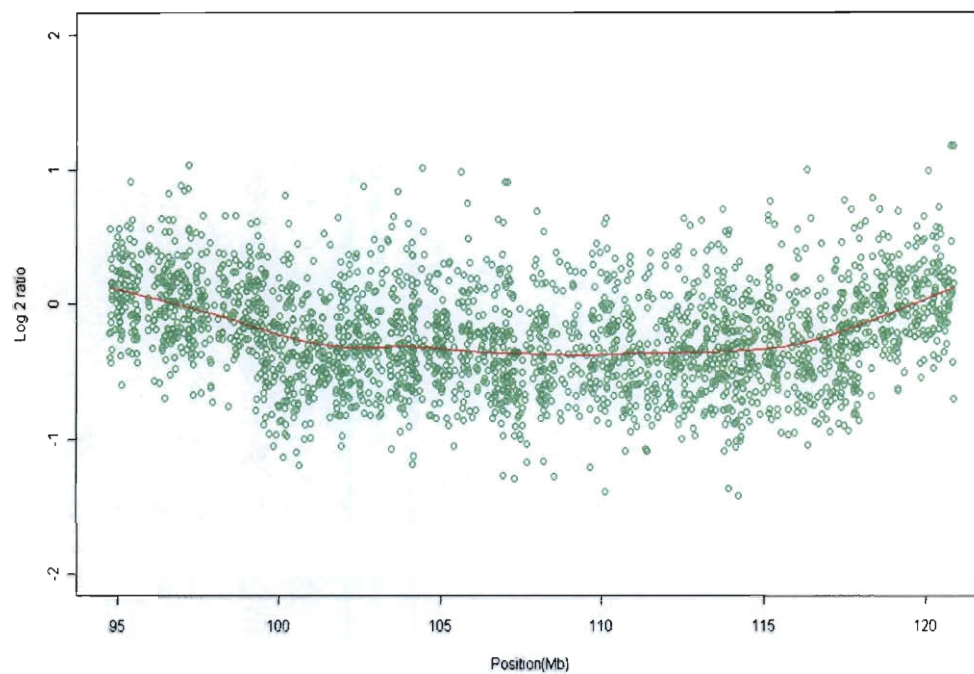


Figure 2.15: Case 1 chromosome 6. The red line is the loess curve with window size .3. There is no evidence of a genome wave.

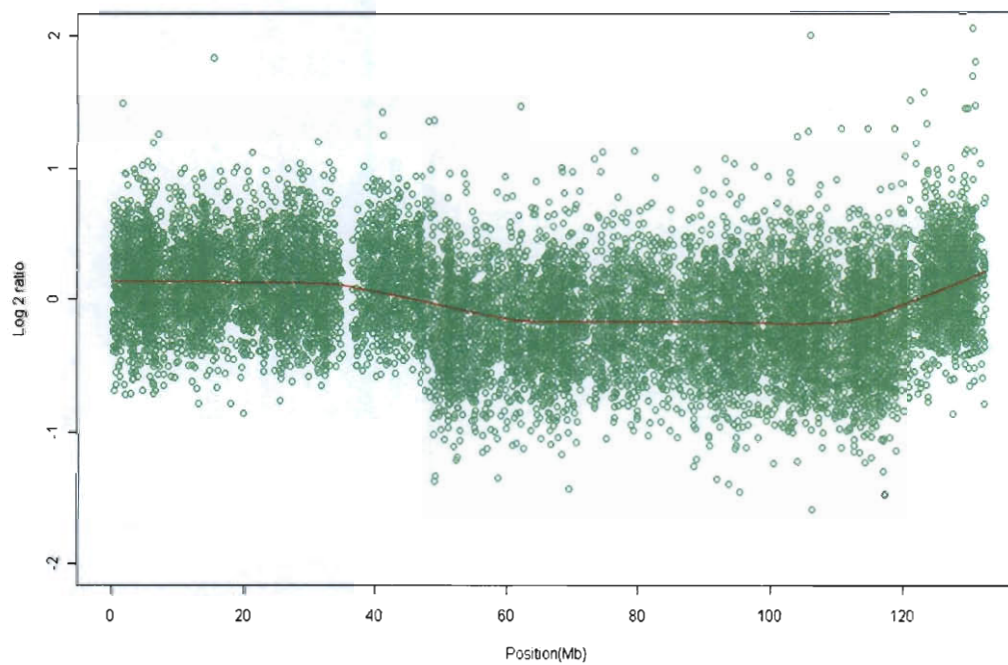


Figure 2.16: Case 2 chromosome 12. The red line is the loess curve with window size .3. There is no evidence of a genome wave.

Figure 2.17 shows normalized log-ratios by their genomic location over a segment of chromosome 6 from an aneuploidy case (Case 1). For this case, cytogenetics data show a loss at 6q1206q21. The inferred integer copy number is indicated by the red dots in Figure 2.17. The top panel shows results from CNAG, and the bottom panel shows result from the Bayesian model. One major difference in the two sets of results is that the Bayesian method gives smoother results as indicated by the longer stretches of the same inferred copy number, whereas the CNAG method varies more in the inferred copy number over the loss region. In particular, note the region from approximately 102Mb to 107Mb, covering 5Mb, which is assigned copy number 2 by CNAG. Although the reason for the misclassification is unknown, it would appear that it is not due to a small number of SNP loci in the region. A second difference between the two methods is that our Bayesian algorithm correctly detects the second change-point in going from a loss to a normal copy number at approximately 118Mb, while CNAG appears to miss it by about 1Mb; both methods correctly detect the first change-point at 99Mb.

In the next example we consider an ependymoma case (Case 2) for which there appears to be a relatively high degree of normal contamination: the mean log-ratio of the segment ranging from about 45Mb to 120Mb is  $-0.18$ , which is consistent with about 60% contamination. Figure 2.18 shows chromosome 12 for this case. The main difference between the results from the two algorithms centers on the segment from around 45Mb to around 120Mb. This segment is identified as CN=2 by CNAG and CN=1 by the Bayes algorithm. In order to validate this result we performed qPCR on two regions in this stretch. We chose a small region around position 50Mb (region 1) and a second one around 110Mb (region 2). The average copy numbers based on qPCR for these two regions resulted in the values 1.43, 1.55, respectively, with approximate 95% confidence intervals of (1.2, 1.71), and (1.33, 1.81), respectively. The validation results support a loss, in concordance with the Bayes method and in contrast to the inference based on CNAG indicating a normal copy number throughout



the stretch. To further evaluate this case we analyzed CGH data using the GLAD software available in Bioconductor. The aCGH platform we used was a BAC array platform, which contains 2,621 BAC clones and has a 3Mb resolution. This array is based on a 2-color competitive hybridization platform (Cy3/Cy5). The experiments were conducted by hybridizing the fluorescent-labeled tumor DNA with reference DNA on the array. Figure 2.19 shows results for chromosome 12. In the plot, each red and green dot stands for one BAC clone, i.e., a  $\sim 3\text{Mb}$  segment of DNA. Red dots represent gains and green dots represent losses. Yellow dots represent inferred copy numbers from CNAG, and purple dots represent inferred copy numbers from our mixture model. As the plot indicates, the region from around 45Mb to around 120Mb shows copy number loss, a finding that agrees with the result provided by our Bayesian model. Furthermore, the tail region of the chromosome indicates a copy number gain, which again confirms our findings and contradicts the CNAG result.

## 2.10 Conclusion

The array-based comparative genomic hybridization microarray has been the gold standard for estimating genomic copy number. As the CGH BACs are relatively large segments, the CGH estimates tend to be robust. On the other hand, the large segments do not allow detection of small CNVs. The SNP genotyping array provides an alternative to CGH, which is expected to identify genomic alterations with a higher resolution. Most SNP array algorithms use a hidden Markov model to infer integer copy numbers, and the component means tend to be set at the theoretical values. However, due to normal cell contamination, which occurs in most tumor samples, log-ratios can be shrunk toward zero, indicating a normal copy number. Consequently, in the presence of a high percentage of contamination, losses or gains may not be detectable. As of this writing, we are not aware of existing algorithms that account for this problem.

We have developed a Bayesian spatial normal mixture model to estimate copy number for SNP array platforms where the means of the components accommodate cell contamination. By using neighboring copy number information on either side of each SNP locus we can generate smoother maps than those based on HMMs. We have shown with a simulation study that our algorithm can detect both long and short segments quite precisely. Our results do not show sensitivity to different values of the scaling factor  $\phi$  in the prior model and to the number of neighbors as long as  $\phi$  is chosen to be small enough. By applying our method to real cancer data, we have demonstrated that our algorithm can do as well as CNAG, a very popular and accurate algorithm used with SNP arrays, and in certain cases performs better. In addition, our algorithm provides smoother realizations than CNAG. The Bayesian mixture model could be extended in a few ways. To more precisely smooth over neighboring probes, it would be helpful to account for inter-probe distance perhaps as a weighting factor when averaging neighboring information. The log-ratio copy number means could also be included as parameters with priors reflecting knowledge of normal cell contamination.

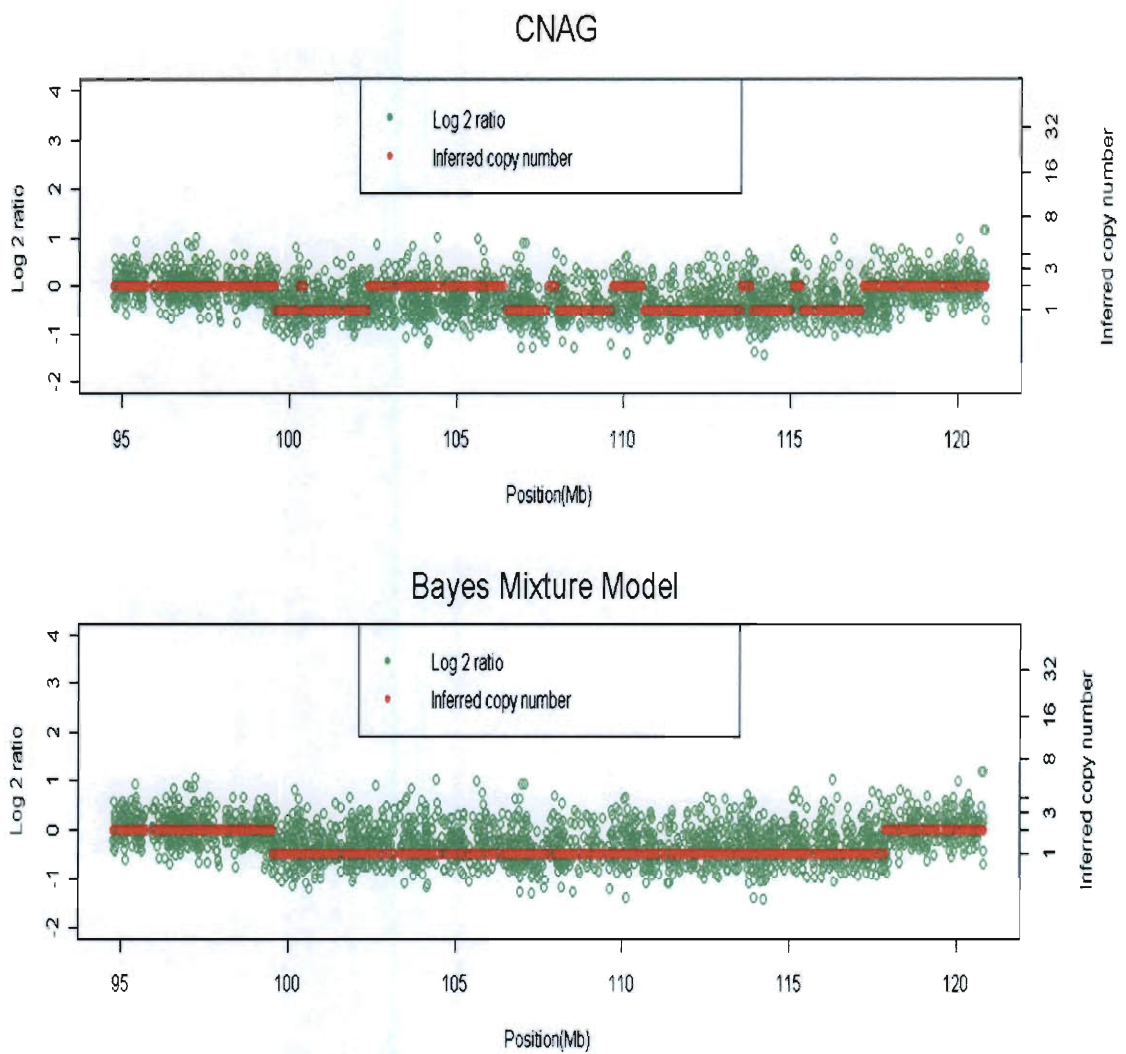


Figure 2.17: A segment of chromosome 6 from case 1. Cytogenetics data show a loss at 6q1206q21.

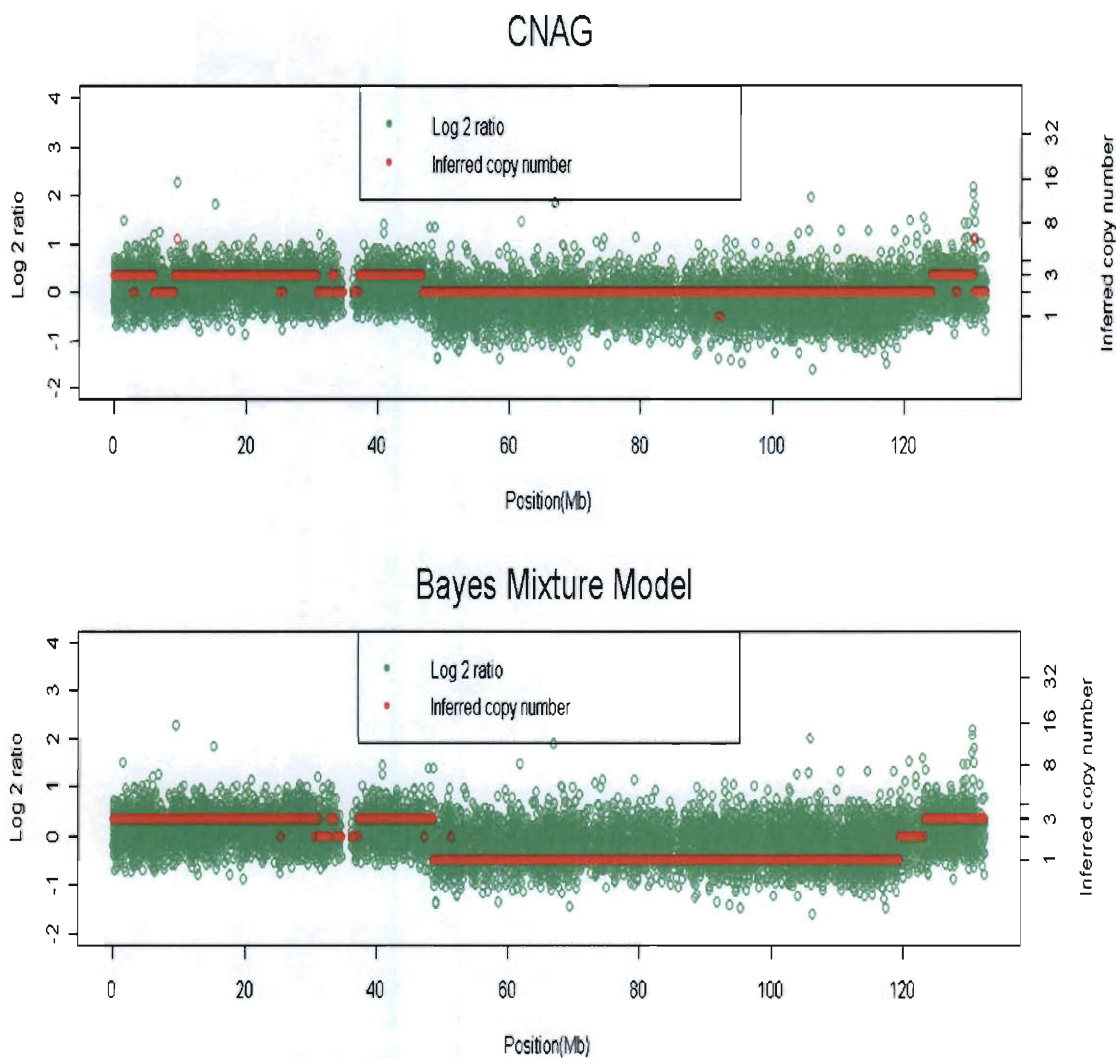


Figure 2.18: Chromosome 12 from case 2. We validated two regions using qPCR, around positions 50Mb and 110Mb. The average copy numbers based on qPCR for these two regions resulted in the values 1.43, 1.55, respectively, with approximate 95% confidence intervals of (1.2, 1.71), and (1.33, 1.81), respectively. The validation results support a loss, in concordance with the Bayes method and in contrast to the inference based on CNAG indicating a normal copy number throughout the stretch.

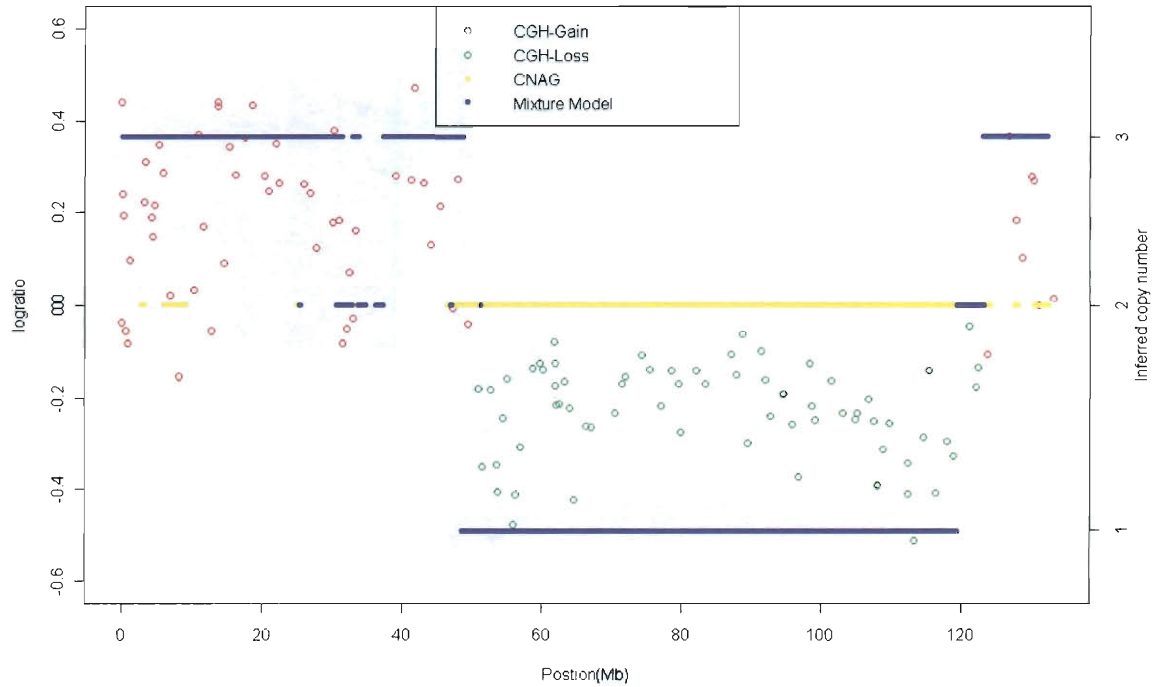


Figure 2.19: Case 2 CGH result analyzed using GLAD software in Bioconductor. The aCGH platform we used was a BAC array platform, which contains 2,621 BAC clones and has a 3Mb resolution. This array is based on a 2-color competitive hybridization platform (Cy3/Cy5). The experiments were conducted by hybridizing the fluorescent-labeled tumor DNA with reference DNA on the array. As the plot indicates, the region from around 45Mb to around 120Mb shows copy number loss, a finding that agrees with the result provided by our Bayesian model. Furthermore, the tail region of the chromosome indicates a copy number gain, which again confirms our findings and contradicts the CNAG result.

## Chapter 3

# Detection of Gene Interactions for Classification using Gene Expression Data

### 3.1 Introduction

In the study of human genetics, mapping of complex traits is a major challenge. In contrast to simple traits, which can be attributed to mutations of single genes, complex traits involve more than one factor. Complex traits are much more common in the population and include asthma, hypertension, heart disease, Alzheimer's disease, and diabetes. Recently high-density oligonucleotide microarrays have been important tools to map complex traits. They have become a major tool to study the differences between two types of samples, for example tumor and normal samples. Due to the large number of genes that are measured in microarray experiments, and the fact that the majority of them do not contribute to the class difference, an important step is to select 'informative' genes. A large number of methods have been proposed in the literature, and most of them are univariate approaches, for which we evaluate one gene at a time. The most popular one is the t-test. Dudoit [2002] compared the

performance of several discrimination methods (nearest neighbor classifiers, linear discriminant analysis, classification trees, bagging and boosting) for the classification ability, based on the preliminary selection of genes using the ratio of between-groups to within-groups sum of squares. At the end of the article they discussed their concern about these kinds of gene selection methods because they ignore interactions between genes. Two or more genes can interact in a way that there are no main effects, so methods to detect these kinds of interactions are needed. Understanding how interactions among genes contribute to the trait is having a large impact on biomedical research, agriculture and evolutionary biology. However, most current strategies are marginal approaches, which examines one gene at a time, so they ignore possible information contained in gene-gene or gene-environment interactions. In the second part of the thesis, I present a new method for detecting gene-gene interactions to improve classification using gene expression data from microarrays.

## 3.2 Biological background

William Bateson invented the term 'epistasis' around 100 years ago to describe the effect of one allele at a locus masking the effect of the allele at another locus (Bateson [1909]). Ever since then the term epistasis has expanded to describe any form of interactions among genetic elements, which causes much confusion in the literature. Patrick Phillips has classified them into three main categories: compositional epistasis, functional epistasis, and statistical epistasis (Phillips [2002]).

- **Compositional epistasis** refers to the original usage of epistasis that an allele blocks the effect of an allele at another locus.
- **Functional epistasis** describes the molecular interactions that genetic elements have with each other, whether these interactions consist of genetic ele-

ments that operate within the same pathway or of genetic elements that directly form complexes with one another.

- **Statistical epistasis** is attributed to Fisher [2005] which addresses the deviation from additivity in the effect of alleles at different genetic loci with respect to their contribution to a quantitative phenotype. Epistasis in this sense is close to the usual concept of statistical interaction: departure from a specific linear model describing the relationship between predictive factors.

Most of the current statistical methods address this problem in the third context: statistical epistasis, where a regression is usually formed with cross terms representing interactions (Musani *et al* [2007], Cordell [2002], Marchini *et al.* [2005], Hoh [2003]). In this work, we are interested in the "interaction" pattern shown in Figure 3.1. In the plot, the x-axis represents expression values of one gene and the y-axis represents expression values of another gene. Each dot represents a sample. Red and green represent the two classes. We can see that with only one gene, either one of the two, we cannot find a cutoff to separate the two classes. However, with them together, we can see a clear pattern: samples in the (low, low) or (high, high) corners are mostly class I and samples in the (low, high) or (high, low) corners are mostly class II. In this thesis we try to detect interaction like this in a non-parametric way using gene expression data. One narrow definition of gene-gene interaction for classification is: their correlation in one group has the opposite sign in the other group. A more general definition is the joint distribution in one group is different from that in another group. For practical purpose, we tend to use the narrow definition in our discussion and application. Further remarks are given in our applications.

The central dogma of molecular biology is the process of  $\text{DNA} \rightarrow \text{mRNA} \rightarrow \text{protein}$ , with the two intervening steps called transcription and translation, respectively. Gene expression is the abundance of mRNA a gene makes. If a gene makes any mRNA, it is said that this gene is expressed. Not every gene is expressed in each cell



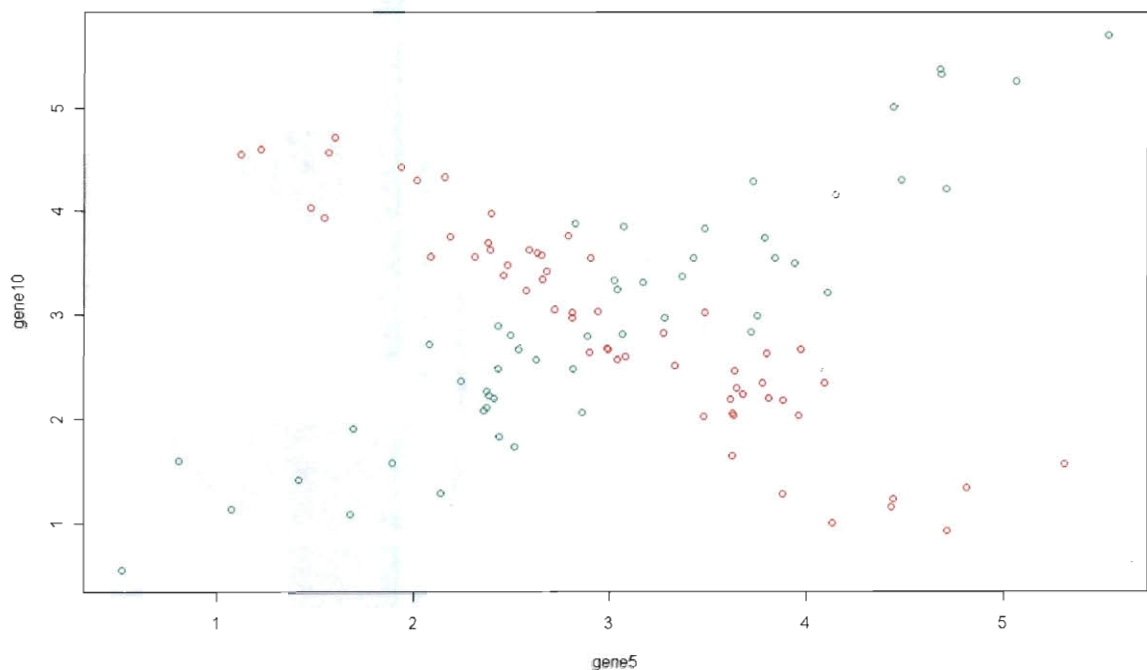


Figure 3.1: A hypothetical example of gene interactions

type. Each of the cells becomes specialized by obeying just some of the instructions in the DNA, resulting in blood, muscle, bone, liver, lungs, brain, etc. A mutated gene may signal disease.

With the advent of microarrays, expression levels of thousands of genes can be measured simultaneously. The technology of microarray for gene expression is similar to that for genotypes, except that here microarrays are used to measure changes in expression levels. So each probe on the microarray is a DNA fragment that represents specific gene coding regions. Sample and reference RNA is then fluorescently labeled and hybridized to the microarray. After washing off of non-specific bonding sequences, laser scanning is used to get the raw data. Figure 3.2 shows the technology of the

microarray experiments.

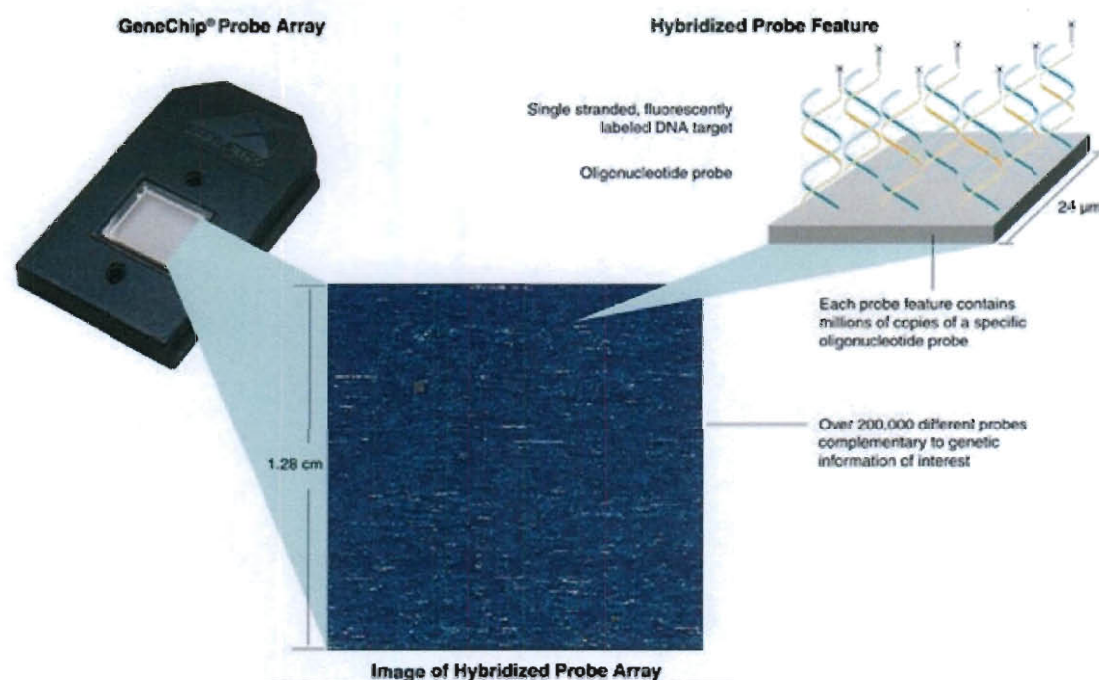


Figure 3.2: Overview of microarray technology.

Source: <http://www.microarrayworld.com/index.html>

### 3.3 Existing methods

In the literature, discussion of epistasis has mainly been based on genotype data, and there are very few papers for detection of gene-gene interactions using gene expression data (Yan *et al.* [2008]). Among the methods using gene expression data, the majority of them use regression with cross terms indicating interactions. Some use penalized regression and others first use certain threshold to select a small number of genes and then apply regression on these selected genes only. In this section, we will review one of the current techniques which is non-parametric. Before getting into it, we briefly describe the genetic algorithms, and why it makes sense to use in our context..

### 3.3.1 Multigene Expression Profile Model

Yan *et al.* [2008] developed two feature selection methods that evaluate the informativeness of a set of genes: the multigene profile association (MPAS) method and the signed multigene profile association (sMPAS) method. The authors claim that methods considering gene-gene interactions have better classification power (20% improvement) in gene expression analysis.

#### 3.3.1.1 Multigene profile association (MPAS) method

For each gene, we cluster the expression values in the training set with two classes pooled, into three states: high, normal, and low. For a specific set of  $K$  genes, a measure of association between this set with the class labels is defined as

$$MPD = \sum_{i=1}^T (\omega_1 * n_{i1} - \omega_2 * n_{i2})^2 \quad (3.1)$$

where  $T = 3^K$  corresponds to the total number of patterns,  $\omega_1 = n_2/(n_1 + n_2)$ ,  $\omega_2 = n_1/(n_1 + n_2)$ ,  $n_{i1}$  is the number of profile  $P_i$  observed among class I samples and  $n_{i2}$  is similarly defined for class II. In order to evaluate the importance of any gene  $G_i$  in the  $K$  genes, we recalculate a MPD score with  $G_i$  removed. The MPAS statistic is then defined as

$$MPAS(G_i | \text{current genes}) = \frac{1}{2} \Delta MPD(G_i \text{ removed}) + \delta \quad (3.2)$$

where  $\Delta MPD(G_i \text{ removed}) = MPD(G_i \text{ removed}) - MPD$  and  $\delta$  is an adjusting term so that MPAS has expected value of 0 under the null hypothesis that this gene has no association with the class difference. It is easy to see that MPAS measures the importance of each gene given current genes. Negative value of MPAS implies importance and positive value means that we can delete the gene without information loss.

Based on this statistic, a backward eliminating process is designed to select important genes. Generate  $B$  random subsets of genes  $\{S_b, b = 1, \dots, B\}$ . For each gene in  $S_b$ ,

compute MPAS value. If all genes in  $S_b$  have negative values, stop the current iteration. Otherwise, remove the gene with the highest positive value and iterate back. After the process is done for all  $B$  random subsets, compute the selection frequency of each gene. Finally, pick the  $p$  genes with the highest frequencies.

For the prediction part, the authors propose to use both marginal genes and gene pairs. A weighted sum of votes is used, with weights being a gene's (or gene pair's) level of importance and vote being the gene's (or gene pair's) prediction on a test sample. Once the  $p$  genes are selected, they are first used as marginal predictors, with marginal weights:

$$W_i^{(m)} = F_i / \sum_{i=1}^p F_i \quad (3.3)$$

Where  $F_i$  is the selection frequency of the  $i^{th}$  gene. For a test sample, the expression values of the  $p$  selected genes are first discretized using the k-means clustering result on the training data. Suppose for gene  $i$ , the test sample falls in state  $h$  ( $h$  takes values a, b, or c). The vote of gene  $i$  for this sample is then  $V_i^{(m)} = \omega_1 * Q_i^{h,1} / Q_i^h$  where  $Q_i^h = \omega_1 * Q_i^{h,1} + \omega_2 * Q_i^{h,2}$ , is the adjusted total number of training samples with gene  $i$ 's state being  $h$ , with  $Q_i^{h,1}$  and  $Q_i^{h,2}$  being the numbers of class I and II samples with gene  $i$ 's state being  $h$ , respectively. The marginal vote for this sample is then defined as:

$$p^{(m)}(x \in \text{class } I) = \sum_{i=1}^p W_i^{(m)} V_i^{(m)} \quad (3.4)$$

The test sample is classified to class I if the above vote is greater than .5, and to class II otherwise.

For a different approach using joint predictors, the MPAS screening procedure is run again on only the  $p$  selected genes. Then the selection frequency of each pair of genes is obtained and the top  $p^*$  pairs are used as joint predictors. The vote and weight of each selected pair are:

$$V_i^{(j)} = \omega_1 * \tilde{Q}_i^{h,1} / \tilde{Q}_i^h \quad (3.5)$$

$$W_i^{(j)} = \tilde{F}_i / \sum_{i=1}^{p^*} \tilde{F}_i \quad (3.6)$$

The joint vote is then defined as

$$P^{(j)}(x \in class I) = \sum_{i=1}^{p^*} W_i^{(j)} V_i^{(i)} \quad (3.7)$$

Finally we combine these two votes:

$$P(x \in class I) = \alpha p^{(m)}(x \in class I) + (1 - \alpha) P^{(j)}(x \in class I) \quad (3.8)$$

where  $0 \leq \alpha \leq 1$  is set to .75 to put more weight on marginal vote. For the application of this approach, we set  $B = 500000$ ,  $K = 10$ ,  $p = p^* = 50$ .

### 3.3.1.2 Signed multigene profile association (sMPAS) method

This second approach is proposed in order to avoid the discretization of the expression values, which depends on the number of states and would also result in information loss due to the conversion of continuous data into discrete values. This approach is derived from the methods for marked point processes (MPP) (Stoyan *et al.* [1995]). Considering the space of expression profiles spanned by several genes, the discriminant analysis between two classes is equivalent to the spatial segregation problem for two point processes with different labels. The basic idea is that for any  $K$  genes under study, define the sMPI value as the number of correct predictions of the nearest neighbor classifier for class I using leave-one-out cross validation. More specifically, for  $K$  genes under study, denote the expression profile of the  $j^{th}$  sample from class I as

$$X_j^{(I)} = (x_{1j}^{(I)}, \dots, x_{Kj}^{(I)})^t \quad (3.9)$$

The expression profiles for a sample from class II is defined as  $X_l^{(II)}$ . The two marked point processes to be segregated are denoted as  $X^{(I)}$  and  $X^{(II)}$ . The nearest neighbor

distance is a good indicator of separation between clusters of points (Dixon [1948], Ripley [1979]). Given a fixed point  $X_j^{(I)}$  in  $X^{(I)}$ , define  $\nu(X_j^{(I)})$  as its Euclidean distance to the nearest neighbor among points in class I. And define  $\tau(X_j^{(I)})$  as its Euclidean distance to the nearest neighbor among points in class II. Define the sMPI statistic as

$$sMPI(X^{(I)}) = \sum_{j=1}^{n_1} 1_{\{\nu(x_j^{(I)}) \geq \tau(x_j^{(I)})\}} \quad (3.10)$$

This is equivalent to counting the number of correct predictions using 1-nearest-neighbor classifier for class I using LOO.

Similarly as in the previous approach, for any gene in the current  $K$  genes, define the sMPAS value as the difference between the sMPI scores without and with this gene. Again, negative values of sMPAS indicate importance of this gene given current genes. The sMPAS score can be similarly defined for class II. Then a backward elimination screening algorithm is applied just like the procedure in the previous method, except that now we run the procedure twice, first using scores for class I and then using scores for class II. At the end of the process, we get the selection frequency of each pair of genes and select the top  $p$  pairs for prediction, half for class I and half for class II.

Once we have the  $p$  pairs of genes, we again use a weighted sum of votes to predict the class labels of the test samples. For any test sample  $x$ , let  $NND_i(x)$  denote the distance between  $x$  and its nearest neighbor in the space spanned by the  $i^{th}$  selected pair of genes. Then this pair of genes gives vote

$$V_i(x) = \text{sign}(NN) \frac{1}{1 + NND_i(x)} \quad (3.11)$$

where  $\text{sign}(NN)$  is 1 if the nearest neighbor is from class I and  $-1$  otherwise. For each vote's weight, we use the information score  $sMPI_i$ . For pairs selected for class

I, the weight is defined as

$$W_i = \sum_{k=1}^{sMPI_i} \binom{n_1}{k} \theta_1^k (1 - \theta_1)^{n_1-k} \quad (3.12)$$

where  $\theta_1 = (n_1 - 1)/(n - 1)$ . To understand this, suppose we have a random variable  $X$  which follows the Binomial distribution  $B(n_1, \theta_1)$ . Then  $W_i$  is one minus the p-value. Similarly for pairs selected for class II, define the weight as

$$W_i = \sum_{k=1}^{sMPI_i} \binom{n_2}{k} \theta_2^k (1 - \theta_2)^{n_2-k} \quad (3.13)$$

where  $\theta_2 = (n_2 - 1)/(n - 1)$ .

Finally, we classify the test sample  $x$  to class I if and only if

$$\sum_{i=1}^p W_i V_i(x) \geq 0 \quad (3.14)$$

To evaluate the performance of the proposed methods, the authors compared their approach to several existing methods using the breast cancer data studied by van't Veer *et al.* [2002] and Tibshirani *et al.* [2002]. Table 3.1 shows the misclassification rates of the evaluated methods on the breast cancer data. In the "corr" method (van't Veer *et al.* [2002]), the correlation coefficient computed between a gene's expression and the class label was directly used. In SAM (significance analysis of microarrays), a t-type score that computes the mean expression difference of a gene between two classes, standardized by a measure of within-class variability, is used to select informative genes. DLDA relies on maximum likelihood discriminant rules when the class densities have the same diagonal covariance matrix. In Golub [1999], a weighted gene voting scheme which turns out to be a minor variant of a special case of linear discriminant analysis is used. From the table, the three algorithms they propose do better than the four existing methods and sMPAS has the best overall performance. Therefore, the authors claim that methods considering gene-gene interactions have better classification power.

Gene Selection	Classifier	Misclassification (Top 50 genes)
sMPAS	sMPAS	0.295
MPAS	MPAS	0.308
MPAS	Marginal	0.346
Golub	Golub	0.385
Corr	Corr	0.385
P-value of t-test	DLDA	0.41
SAM	DLDA	0.423

Table 3.1: Misclassification on breast cancer data (van't Veer *et al.* [2002]). Source: Yan *et al.* [2008].

### 3.3.2 A Introduction to Genetic Algorithm

In the context of gene selection for classification using gene expression data, our goal is to select a subset of informative genes that jointly contribute to the discrimination between different classes of samples. Since it's a "small  $n$  (sample size)" and "large  $p$  (number of variables)" situation, many such subsets of genes may exist. The strategy we try to explore is to find a large number of such subsets and then assess the relative importance of genes by examining the selection frequencies of genes in these subsets. To use 'brute force' to compare all subsets of genes is not feasible. For example, there are approximately  $10^{100}$  ways to select 50 genes from 2000 genes. So we need a more efficient technique to go through fewer combinations to find optimal solutions. A genetic algorithm can be used in this context. Genetic algorithms have been used in many combinatorial problems involving high dimensional data (Clark [1996] and Forrest [1993]).

The genetic algorithm (GA) is an adaptive heuristic technique used to find exact or nearly optimal solutions to search or optimization problems. The basic concept of



GAs is designed to simulate genetic processes in natural evolution, specifically those that follow the principles of natural selection and "survival of the fittest" laid down by Charles Darwin.

Pioneered by John Holland (Holland [1975]), genetic algorithms have been widely applied in many fields of engineering. The genetic algorithm mimics natural evolution and selection. In nature, individuals compete with each other for a limited amount of resources and to attract a mate. Successful individuals will have larger numbers of offspring, while poorly performing individuals have few or even no offspring at all. In this way, genes from the 'good' individuals will spread to an increasing number of individuals in each successive generations. So species evolve to become more and more suited to their environment. The genetic algorithm uses an analogy. In GA, each individual represents a possible solution to the problem. Each individual is assigned a fitness score according to how good the solution is. The highly fit individuals are given opportunities to reproduce, and the least fit members are less likely to have offspring (Beasley *et al.* [1993]). In this way, the overall fitness of the new generation will be improved. Over many generations, the solutions presented by GA are more likely to be globally optimal.

There are five major components of a genetic algorithm: chromosome, fitness, selection, reproduction and termination.

- Chromosome: Each chromosome consists of a set of parameters that represent a potential solution to a problem. In the context of gene selection, a chromosome is a set of genes that are selected.
- Fitness: Fitness is problem dependent. It measures the quality of a particular chromosome. In our case, we want to find interactions which can help with the classification, so we can use classification accuracy as fitness function.

- Selection: During the reproduction phase, a proportion of the current population is selected to produce offspring. The selection is fitness-based, where the more fit individuals are more likely to get selected. Some 'good' individuals may be selected several times while 'poor' ones may not be selected at all.
- Reproduction: After selection and transmission, individuals produce offspring which comprise the next generation. There are two mechanisms: crossover and mutation.
  - Crossover: See Figure 3.3 for an illustration of single point crossover. Given two parents, cuts their chromosomes at some randomly chosen position, to produce two 'head' segments, and two 'tail' segments. The tail segments are then swapped over to produce two new full length chromosomes. Crossover may produce offspring of higher fitness. And, of course, it is also possible that it would produce offspring of lower fitness, but these 'bad' individuals are not likely to be selected in the next generation.
  - Mutation: See Figure 3.4 for an illustration of single point mutation. Mutation is applied to each child after crossover. The purpose of mutation in GAs is to preserve and introduce diversity. Mutations should allow the algorithm to avoid local minima by preventing the population of chromosomes from becoming too similar to each other, thus slowing or even stopping evolution.
- Termination: termination is also problem dependent. Common termination conditions are:
  - Fixed number of generations is reached
  - The overall fitness of the population becomes stable
  - A solution is found that satisfies some criteria
  - Allocated budget (computation time/money) is reached

- The highest ranking solution's fitness is reaching or has reached a plateau such that successive iterations no longer produce better results
- Manual inspection
- Combinations of the above

The first two are not recommended, because they don't guarantee convergence. The third and fourth one are commonly used.

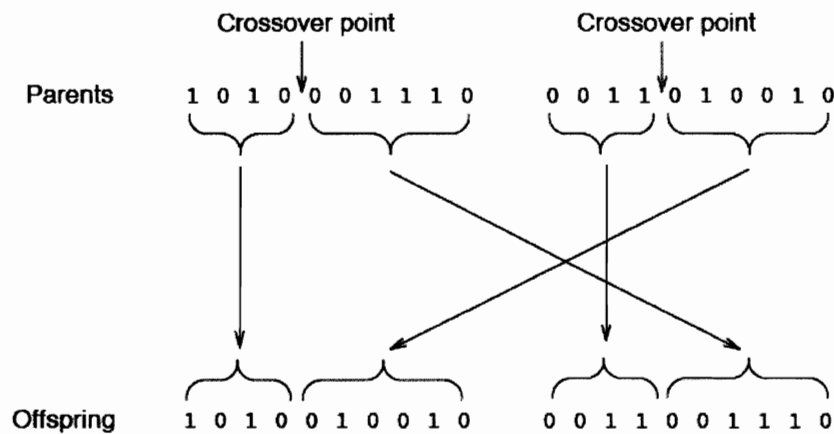


Figure 3.3: Single point Crossover

Source: [Beasley *et al.* [1993]]

The flowchart of the procedure is shown in Figure 3.5.

### 3.3.3 k-Nearest-Neighbor Classifiers

As mentioned earlier, the fitness in the genetic algorithm is classification accuracy. The classifier k-nearest-neighbor algorithm is used here. In classification procedure, there is a training set for which we know the class labels. The k-nearest-neighbor algorithm is a non-parametric method for classification. The idea is very simple.



Figure 3.4: A single mutation

Source: [Beasley *et al.* [1993]]

Given an object, we find the  $k$  training objects closest in distance to this object, and then classify using majority vote among the  $k$  neighbors. Despite its simplicity,  $k$ -nearest-neighbors has been successful in a large number of classification problems, including handwriting digits, satellite image scenes and EKG patterns. It is often successful where each class has many possible prototypes, and the decision boundary is very irregular (Hastie *et al.* [2001]). The classification accuracy is the percentage of samples which are correctly classified.

### 3.3.4 Genetic Algorithm/k-nearest-Neighbor Method

Li *et al.* [2001a] proposed to combine a Genetic Algorithm and the  $k$ -nearest neighbor classifier to assess the importance of genes for sample classification based on expression data. In their study, the genetic algorithm is used to identify a large number of subsets of 50 genes that can correctly classify the majority of the samples and then use the selection frequency of the genes to assess the relative importance of genes for sample classification. The authors applied it to expression data from normal versus tumor tissue from human colon. Two distinct clusters were observed, and the majority of the samples were classified correctly. This approach is a multivariate approach (samples are compared in multi-gene dimensions), however, the interaction informa-

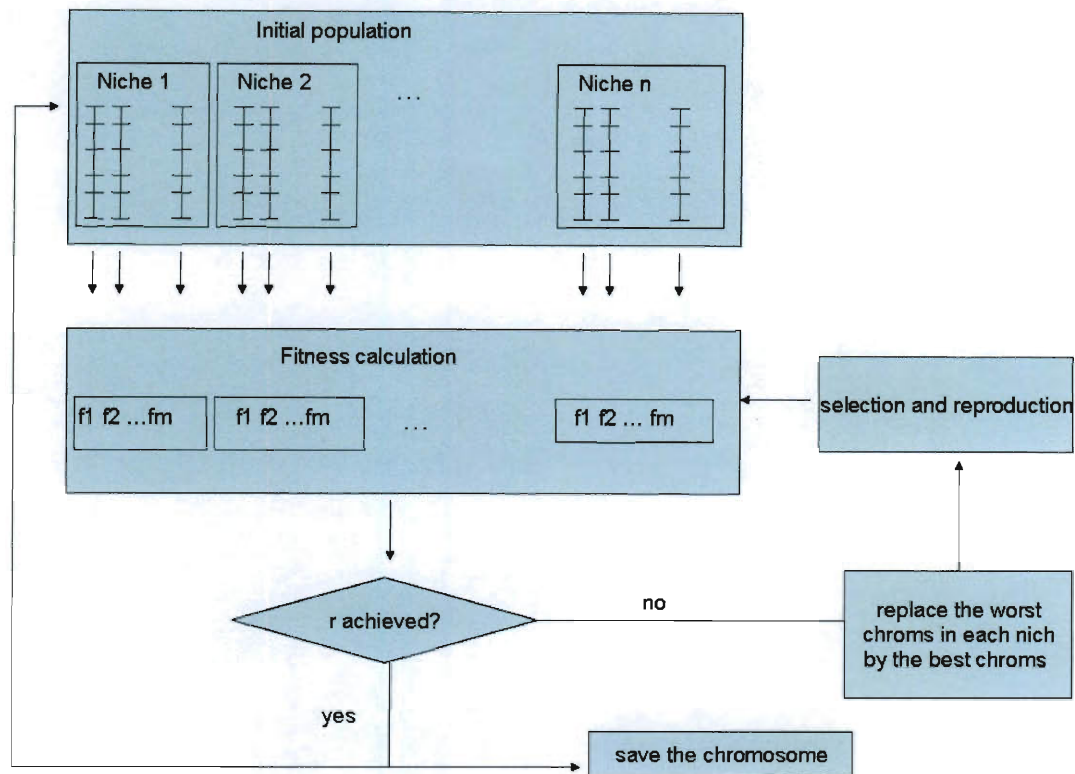


Figure 3.5: Flowchart of the GA procedure

tion is not extracted.

Here is the basic idea of Li *et al.* [2001a] (Figure 3.5). We start with  $n_{niche}$  sub-populations (or 'niches') where each contains  $n_{chrom}$  chromosomes. In a typical run,  $n_{niche} = 10$  and  $n_{chrom} = 100$ . Each chromosome consists of  $d$  genes, which are initially selected randomly from the gene pool. Each niche evolves independently, except that, the best chromosomes identified at each generation, one from each niche, are combined and used to replace the 10 least fit chromosomes in each niche in the next generation.

Each chromosome is given a fitness score, which measures the 'quality' of the chromosome. The  $k$ -nearest-neighbor classifier is used here to compute the fitness score. For the  $d$  genes in the chromosome, each sample is represented by its expression values of the  $d$  genes. We can compute the distance between a sample and each of the training samples using Euclidean distance in the  $d$ -dimensional space. A sample is classified according to the class membership of its  $k$  nearest neighbors: if all of the  $k$  nearest neighbors of a sample belong to the same class, the sample is classified to that class, otherwise, the sample is considered unclassifiable.  $k = 3$  is used in the algorithm. To assess the ability of classification of the chromosome, we calculate the number of correctly classified training samples and use it as the fitness score of the chromosome. Obviously, the larger this value is, the better the chromosome is.

We set a threshold,  $\gamma$ , which is the number of correct predictions we hope to achieve. The largest value of  $\gamma$  should be equal to the number of samples. If the current generation of chromosomes does not include any that achieve  $\gamma$ , then a new generation is formed. The selection of chromosomes to go into the next generation is based on the principle of survival-of-the-fittest. The single best chromosome (with the largest fitness score) from each niche is entered into the respective next generation for that niche deterministically, and the remaining positions are filled based on sampling that is weighted according to the relative fitness score of the chromosomes in the parent generation, probabilistically. Once a chromosome is chosen for transmission, we perform mutation, which is to substitute new genes into the chromosomes. The number of substitutions is assigned randomly between 1 and 5, with probabilities .53125, .25, .125, .0625, and .03125, respectively. In this way, a single replacement is given the highest probability while simultaneous multiple replacement has low probability. This strategy prevents the search from behaving as a random walk as it would if many genes were introduced at each generation. When  $d$  is small, say less than 10, then only one gene is selected for mutation with probability of 1. Once the number

is determined, these genes are selected randomly and are substituted by genes that are not already in the chromosome.

The above procedure is repeated until  $\gamma$  is achieved in any of the niches. We then save the selected 'best' chromosome. And the whole process is restarted. We stop the process when a large number (the author use 10000) of 'best' chromosomes are obtained.

After we get the large number of 'best' chromosomes, we calculate the frequency with which each gene is selected across all best chromosomes. The top 50 genes with the largest frequencies are then used to classify samples in the test set. Finally, we obtain classification accuracy based on the final set of genes.

In a subsequent paper (Li *et al.* [2001b]), the authors studied the sensitivity, reproducibility, and the stability to the choice of parameters of the algorithm. It turned out that the algorithm is highly repeatable with independently runs and are not sensitive to the parameters, e.g  $d$ . The authors successfully applied this algorithm to several datasets (colon cancer data, leukemia data, and lymphoma data) and achieve great performance. For example, the method correctly classified 33 of the 34 test samples of the acute myeloid leukemia datasets.

### 3.3.5 Comparison of the three algorithms

The genetic Algorithm is a stochastic search method. It avoids the comparison of all subsets of genes and provide an alternative way to go through fewer combinations to find the optimal solution. However, the genetic algorithm in Li *et al.* [2001a] can only detect genes with large main effects and thus ignores possible gene-gene interactions. The MPAS method of Yan *et al.* [2008] is said to consider joint effects by considering random subsets of genes. There are two disadvantages of this approach. First, it

depends on the discretization of the expression values of each gene into three clusters, which are very arbitrary. And some genes may have two or more than three clusters. Second, for the prediction part using joint predictors, the MPAS screening process is run a second time on the  $p$  selected genes only. This procedure will ignore interactions that include a gene having no or little main effects. There are two drawbacks to both MPAS and sMPAS. First, we need a huge  $B$  to start with, especially when the number of genes is large. Second, in the backward eliminating process, if a gene gets deleted, it cannot be recovered again.

To see if the genetic algorithm is comparable with other existing methods, I applied the GA version in Li *et al.* [2001a] to the breast cancer data mentioned above. Its misclassification rate is added to Table 3.1 (See Table 3.2). From the table, we can see that the genetic algorithm outperforms the four existing popular methods, which only evaluate the genes marginally and ignore possible information contained in gene interactions, but does worse than the methods proposed in Yan *et al.* [2008], which are said to consider joint effects. Therefore, we believe that models which consider joint effects (or interactions) can significantly improve the power of the model to select informative gene (or interacting gene pairs) and thus help the classification.

### 3.4 Proposed GA Method

As discussed earlier, most current approaches use regression to detect interactions. This kind of interaction is in the statistical sense, in which interaction means deviation from additivity, thus is very limited. Also, when the number of genes is large, there are a huge number of cross terms, which will cause some problems. Li *et al.* [2001a] cannot detect interactions. sMPAS in Yan *et al.* [2008] requires a huge number of random sets, especially when the number of genes is large (e.g., thousands), which is usually the case. And once the genes in each of the  $B$  subsets are chosen, there



Gene Selection	Classifier	Misclassification (Top 50 genes)
sMPAS	sMPAS	0.295
MPAS	MPAS	0.308
MPAS	Marginal	0.346
<b>GA</b>	<b>knn</b>	<b>0.372</b>
Corr	Corr	0.385
Golub	Golub	0.385
P-value of t-test	DLDA	0.41
SAM	DLDA	0.423

Table 3.2: Misclassification including GA on breast cancer data.

cannot be new genes introduced. However, in a GA, through mutation, new genes can replace useless genes, and thus make the algorithm converge faster.

In this section, we propose a new method of detecting gene-gene interactions useful for classification using a genetic algorithm. The genetic algorithm has a number of advantages. It can efficiently scan a vast solution set, so it's more likely to converge toward a global optimum instead of local optimum. More probable solutions are sampled more frequently than less probable ones. Bad solutions do not effect the end solution negatively as they are simply discarded. The inductive nature of the GA means that it does not have to know any rules of the problem.

Through simulations studies, we found that main effects tend to mask interaction. If some genes have large main effects, and if we don't exclude them, then all the pairs we get at the end of the algorithm will have one of them. We propose a two step model. In the first step, we directly use the method of Li *et al.* [2001a] to get the top  $p$  genes with highest selection frequencies as main effect genes. In the second step, we remove these  $p$  genes out from the original data, and use the remaining data to get

gene-gene interactions. In this way, our algorithm cannot directly detect interaction pairs with one gene having a large main effect. However we can look into them by examining all pairings that include the gene with large main effect.

For most algorithms aimed at detecting gene interactions, the frequencies of pairs of genes are calculated at the end of the algorithm. For example, in the genetic algorithm proposed by Li *et al.* [2001a], after we get a large number of nearly optimal solutions, we can get the selection frequency of each gene pair, and regard the top  $p$  gene pairs as interaction pairs. Take Yan *et al.* [2008] as another example, after the screening is done for a large number of subsets of genes, we calculate the frequency of each gene pair of the remaining genes. In this thesis, instead of looking for gene pairs at the end of the algorithm, we propose to directly search for them in the algorithm.

In our model, each component of a GA-chromosome is a pair of distinct genes, instead of a single gene as in Li *et al.* [2001a]. So now each GA-chromosome consists of  $d$  distinct pairs of gene pairs. If we have  $P$  genes, then we have  $\binom{P}{2}$  gene pairs. Each pair of genes in a chromosome corresponds to a potential interacting pair. If a pair of genes interact and contribute to the class difference, we would expect it to occur over and over again.

In order to use k-nearest-neighbor classifier for the fitness function, we need to define a new distance measure based on pairs of genes. For  $K$  pairs of genes under study, suppose two samples  $x$  and  $y$  have the expression profiles

$$x = ((x_{11}, x_{12}), (x_{21}, x_{22}), \dots, (x_{K1}, x_{K2})) \quad (3.15)$$

$$y = ((y_{11}, y_{12}), (y_{21}, y_{22}), \dots, (y_{K1}, y_{K2})) \quad (3.16)$$

where  $x_{ij}$  and  $y_{ij}$  are the expression values of the genes in the  $i^{th}$  pair of sample  $x$  and  $y$  respectively. Let  $d_i$  be the Euclidean distance between the  $i^{th}$  pair of points  $(x_{i1}, x_{i2})$  and  $(y_{i1}, y_{i2})$ . Then define the distance between the two samples as the average of the

Euclidean distances between each gene pair:

$$d(x, y) = \sum_{i=1}^K d_i / K \quad (3.17)$$

Once we have this new distance measure between samples of pairs, we can define the fitness function using the k-nearest-neighbor classifier accordingly. We deviate from the usage in Li *et al.* [2001a], where the voting should be unanimous, we propose to use a less strict criterion: a sample is classified through majority voting, i.e., if the majority of the  $k$  nearest neighbors belong to a class, then the sample is classified to that class. For  $k$ , a too large value would allow less flexibility in detecting subclusters and also would increase the computational burden. Too small a value would not be large enough to form tight clusters. We use  $k = 3$ .

We propose not to use cross-over in this study because we don't think it helps much for the task of finding interacting gene pairs. In our proposed algorithm, each pair of genes is considered as a single component in the GA chromosomes. Therefore, crossover will not produce new gene pairs and we decide to use mutations for simplicity.

For the mutation part, we treat the GA-chromosome of  $d$  gene pairs as an ordinary GA-chromosome consisting of  $2d$  genes. Once a chromosome is chosen for transmission, we randomly pick genes to mutate. In this way, either one gene or both of the two genes in a gene pair have an opportunity to get mutated.

This method can be easily extended to detection of interactions involving more than two genes. For example, if we are interested in finding interacting gene triples, we simply use three distinct genes as a component in each GA-chromosome and apply the same procedure.

In this algorithm, there are two parameters we need to set:  $d$  (the number of components in a chromosome) and  $\gamma$  (the threshold in GA). The algorithm may take a few minutes to many hours depending on the size of the data and how difficult the classes can be separated. For  $\gamma$ , we can start with some large value to achieve at least 80% accuracy; if the algorithm is too slow, we can reduce  $\gamma$  and run the program again. Generally, we find it works well with a value of  $\gamma$  corresponding to 70% to 80% accuracy. For  $d$ , too small a value would ignore the joint effects of interacting pairs, and too large a value would introduce higher level's of noise. In the simulation study in the next section, we report on the sensitivity of the results with varying values of  $d$ .

### 3.5 Simulation Study

In this section, we investigate the performance of the proposed algorithm on simulated data, and compare the results with those from Yan's approach. We simulated eight data sets representing eight different scenarios, with each scenario a case-control study with sample sizes 50 and 60 for cases and controls respectively. Each subject has 500 genes measured with nine of them involved in main effects and interactions. Other genes are independent.

For each of the 8 scenarios, gene 1 and gene 2 have large main effects, and there are four pairs of interacting genes: gene 1 with gene 3, gene 5 with gene 10, gene 20 with gene 40, and gene 50 with gene 60. The other 491 genes independently follow the standard normal distribution. The variance of any gene is 1.

- **Scenario 1:** For cases (with sample size 50), the mean vector of the expression values of the 9 genes is (7, 7, 3, 3, 3, 3, 3, 3, 3), the correlation vector of the four pairs is (.9, .9, .8, .7). For controls (with sample size 60), the mean vector of the expression values of the 9 genes is (0, 0, 3, 3, 3, 3, 3, 3, 3), the correlation

vector of the four pairs is  $(-.9, -.9, -.8, -.7)$ . This setting means that the four interacting pairs comprise genes with no main effects.

- **Scenario 2:** The setting is the same as Scenario 1 except that the correlation between gene 5 and gene 10 for the control group is now the same as that for the case group (both .9). So in this scenario, gene 5 and gene 10 do not interact. We generate this scenario to make sure the algorithm does not detect genes which do not interact.
- **Scenario 3:** For cases (with sample size 50), the mean vector of the expression values of the 9 genes is  $(7, 7, 3, 3, 3, 3, 3, 3, 3)$ , the correlation vector of the four pairs is  $(.9, .9, .8, .7)$ . For controls (with sample size 60), the mean vector of the expression values of the 9 genes is  $(0, 0, 4, 4, 4, 4, 4, 4, 4)$ , the correlation vector of the four pairs is  $(-.9, -.9, -.8, -.7)$ . This setting means that the four interacting pairs comprise genes with small main effects, since the means are 3 and 4 in cases and controls.
- **Scenario 4:** For cases (with sample size 50), the mean vector of the expression values of the 9 genes is  $(7, 7, 3, 3, 3, 3, 3, 3, 3)$ , the correlation vector of the four pairs is  $(.9, .9, .6, .5)$ . For controls (with sample size 60), the mean vector of the expression values of the 9 genes is  $(0, 0, 3, 3, 3, 3, 3, 3, 3)$ , the correlation vector of the four pairs is  $(-.9, -.9, -.6, -.5)$ . For this setting, there are still no main effects with genes in interacting pairs, but we decrease the correlations to see if the algorithm can still detect them.
- **Scenario 5:** The setting is the same as Scenario 1 except that the correlation between gene 5 and gene 10 for the control group is now .3 instead of  $-.9$ . So here we design correlations with the same sign, but smaller correlations, instead of having the opposite signs.
- **Scenario 6:** The setting is the same as Scenario 5, except that the mean vectors

for gene 5 and gene 10 are now  $(3, 3)$  for cases and  $(4, 4)$  for controls, which means we add small main effects to gene 5 and gene 10.

- **Scenario 7:** The setting is the same as Scenario 5, except that the mean vectors for gene 5 and gene 10 are now  $(3, 3)$  for cases and  $(3.5, 3.5)$  for controls.
- **Scenario 8:** The setting is the same as Scenario 5, except that the mean vectors for gene 5 and gene 10 are now  $(3, 3)$  for cases and  $(4.5, 4.5)$  for controls.

In all cases, we first use the GA algorithm in Li *et al.* [2001a] to get individual genes with large main effects. In each run, I use varying  $d$ 's ranging from 2 to 20,  $\gamma = .85$ , and the number of 'best' chromosomes set to be 5000.

In the second step, we first exclude gene 1 and gene 2 from the gene pool, and then use the proposed genetic algorithm to detect interacting gene pairs. For  $d$ , the number of pairs, I consider  $d = (1, 2, 3, 4, 5, 10)$ . For  $\gamma$ , I use .80 corresponding to around 73% accuracy. The number of 'best' chromosomes is set to be 10,000. In all cases we examined, we get the ordered lists of all gene pairs, with the top one being the pair with the largest selection frequency, the second one being the pair with the second largest frequency, etc. Then we report the ranks of the true interacting pairs in the lists.

We also applied the sMPAS method in Yan *et al.* [2008] to each of the eight scenarios for comparison. Since for this approach, we get two lists due to the two screening processes, we report the orders in both lists.

For both the genetic algorithm we proposed and Yan's approach, I did three independent runs for each parameter setup to see if the results are stable. I list the three sets of ranks from Yan's method in the tables below. For our algorithm, the ranks are very stable, with most of them the same or off by one or two. Therefore, I did

not include them in the tables.

Table 3.3 to Table 3.10 give the ranks of the four true interacting pairs with the two different algorithms corresponding to the eight scenarios respectively. 'NA' in the table means that the pair is selected 0 times.

**Scenario 1** For step 1, main effects detection, gene 1 and gene 2 have overwhelming selection frequencies. The selection frequencies are shown in Figure 3.6. We can see that genes 1 and 2 are selected about half of the time (2500 out of 5000), while other genes are with frequencies less than 100. Thus, in step 2 for interaction detection, we remove these two genes. For step 2, the ranks of the four true interacting pairs from both algorithms are shown in Table 3.3. The second and third columns give the ranks from Yan's method, with the three numbers separated by comma in each column representing the ranks from the three independent runs. Column 4 to column 9 give the ranks from our proposed GA with varying  $d$ 's. Since in our algorithm, we exclude the genes with large main effects, the pair with gene 1 and gene 3 is never selected. In Yan's method, however, these two genes are not removed, but this pair is not detected either. This is because gene 1 itself can correctly classify all the samples, so any other gene would not help to achieve better accuracy. For the other three pairs, we see that with GA, except when  $d = 1$ , the results are very good, with these three pairs being the top three pairs. With Yan's method, the results are also good, except that there are some variations across independent runs. For example, in Yan2, the pair 50 and 60 has rank 1 in one run, 16 in the second run, and 26 in the third run. If the rank is 26, then this pair will not be selected.

**Scenario 2** For step 1, again, gene 1 and gene 2 have large selection frequencies. See Figure 3.7. We remove these two genes for step 2. In the second step of interaction detection, the ranks are shown in Table 3.4. Here, the two pairs 1 and 3, 5

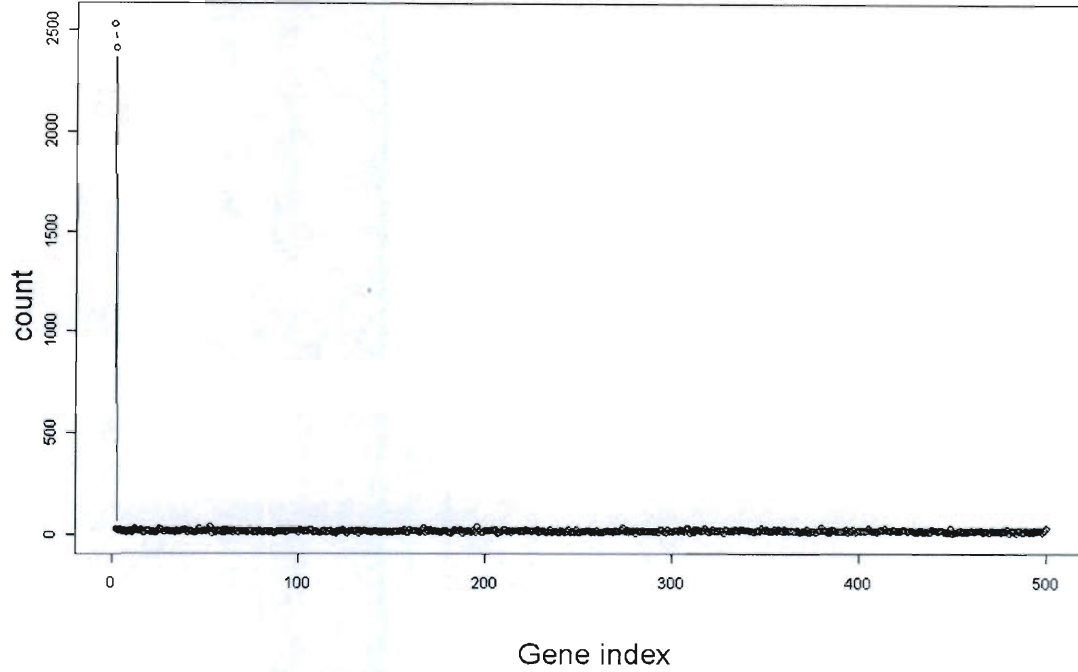


Figure 3.6: Selection frequencies of each gene for scenario 1

and 10 cannot be detected by either algorithm. And the two algorithms do about the same on the other two pairs.

**Scenario 3** For step 1, the results depend on  $d$ . With a smaller  $d$ , for example, when  $d = 2$ , or 3, or 4, the seven genes 5, 10, 20, 40, 50, 60 are detected, as well as gene 1 and gene 2. Although the selection frequencies of the seven are much lower than those of gene 1 and gene 2, they do pop out. However, when  $d$  is larger, for example when  $d \geq 10$ , only gene 1 and gene 2 are detected, these seven genes cannot be detected. See Figure 3.7 for the frequencies. The reason is following: in the algorithm, once a chromosome achieves the accuracy threshold,  $r$ , the current search is over and



Table 3.3: ranks of true interacting pairs for Scenario 1

true pair	YAN1	YAN2	GA (d=1)	GA (d=2)	GA (d=3)	GA (d=4)	GA (d=5)	GA (d=10)
g5 & g10	2,3,1	26,1,1	11	2	1	1	1	1
g20 & g40	1,7,5	2,2,3	12	3	3	3	3	3
g50 & g60	3,8,2	1,16,26	8	1	2	2	2	2
g1 & g3	NA	NA	NA	NA	NA	NA	NA	NA

Table 3.4: ranks of true interacting pairs for Scenario 2

true pair	YAN1	YAN2	GA (d=1)	GA (d=2)	GA (d=3)	GA (d=4)	GA (d=5)	GA (d=10)
g5 & g10	> 110000	> 100000	NA	NA	NA	NA	NA	NA
g20 & g40	1,4,1	11,39,7	6	1	1	1	1	1
g50 & g60	2,1,4	5,37,1	NA	10	8	23	20	42
g1 & g3	NA	NA	NA	NA	NA	NA	NA	NA

this chromosome is a best chromosome. When we use a larger  $d$ , we include more genes into the chromosomes, so it's more likely that gene 1 or gene 2 is selected. And once either of them is selected, the search is over. When we use a small  $d$ , it's less likely that gene 1 or gene 2 is included, so the search continues until a chromosome achieves  $r$ . Sometimes when some of the last seven genes are included, a chromosome may achieve  $r$ . When  $d$  equals 2, the number of best chromosomes which do not have 1 or 2 is 940, and when  $d$  equals 10, this number is only 70. In the second step of interaction detection, the ranks are shown in Table 3.5. In this scenario, again, the pair of 1 and 3 is not detected by either algorithm. For the other three pairs, GA does better than Yan's approach. Again, with GA, except when  $d = 1$ , the three true interacting pairs are the top three, except for an rank of 4 for the pair 20 and 40 when  $d = 2$ , which is still very good. With Yan's approach, the results are also good, except for some variations with independent runs. For example, in Yan2, the pair 50

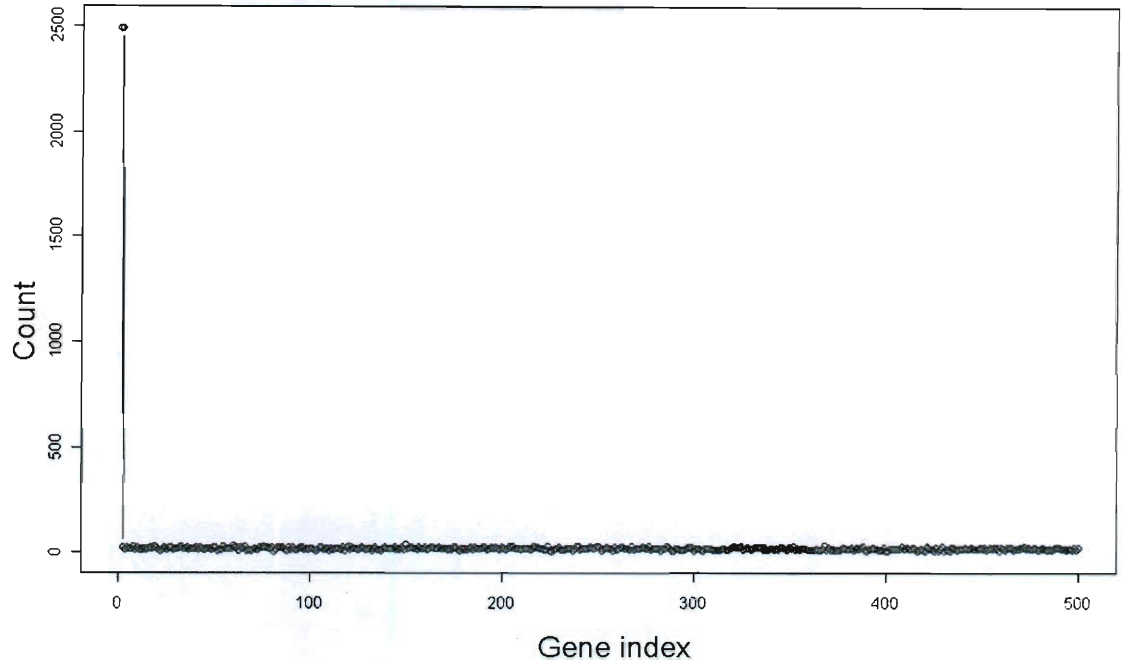


Figure 3.7: Selection frequencies of each gene for scenario 2

and 60 has rank 23 in one run, 11 in a second run, and 1 in the third run.

**Scenario 4** For step 1, the frequencies are about the same as in scenario 1: gene 1 and gene 2 have large selection frequencies, and the other 498 genes have frequencies less than 100. See Figure 3.8. In the second step of interaction detection after we remove gene 1 and gene 2, the ranks of the four true interacting pairs are shown in Table 3.6. Now, the two pairs 1 and 3, 50 and 60 cannot be detected by either algorithm (correlation between 50 and 60 is low). For the pair 5 and 10, the two algorithms perform about the same. For the pair 20 and 40, GA does better. Here again, we see huge variation. In YAN1, the pair 20 and 40 has ranks 2, 2, and 137 in the three runs respectively. In YAN2, this pair has ranks 8, 145, 211 in the three

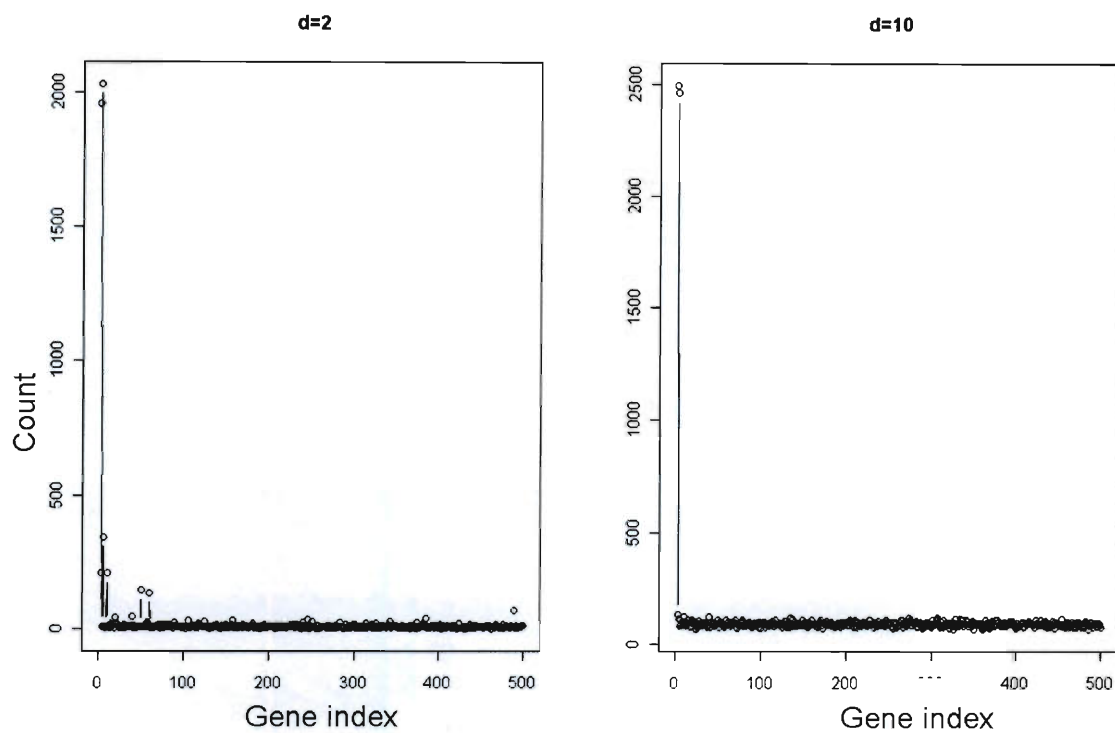


Figure 3.8: Selection frequencies of each gene for scenario 3

runs, respectively.

**Scenario 5** For step 1, again, gene 1 and gene 2 have large selection frequencies. See Figure 3.10. In the second step of interaction detection, the ranks are shown in Table 3.7. Now, the two pairs 1 and 3, 5 and 10 cannot be detected by either algorithm (the two correlations have the same sign). For the other two pairs, GA does much better. Notice the huge variation again in Yan's method. In YAN2, the pair 50 and 60 has ranks 4683, 69, 5 in the three runs respectively.

**Scenario 6** For step1, as in scenario 2, the detection of 5 and 10 depends on

Table 3.5: ranks of true interacting pairs for Scenario 3

true pair	YAN1	YAN2	GA (d=1)	GA (d=2)	GA (d=3)	GA (d=4)	GA (d=5)	GA (d=10)
g5 & g10	10,2,6	5,4,22	3	1	1	1	1	1
g20 & g40	5,8,4	1,7,12	15	4	3	2	2	2
g50 & g60	1,4,5	23,11,1	1	2	2	3	3	3
g1 & g3	NA	NA	NA	NA	NA	NA	NA	NA

Table 3.6: ranks of true interacting pairs for Scenario 4

true pair	YAN1	YAN2	GA (d=1)	GA (d=2)	GA (d=3)	GA (d=4)	GA (d=5)	GA (d=10)
g5 & g10	1,1,1	1,3,2	5	1	1	1	1	1
g20 & g40	137,2,2	211,145,8	NA	8	3	5	5	4
g50 & g60	> 185	> 739	NA	NA	NA	NA	NA	NA
g1 & g3	NA	NA	NA	NA	NA	NA	NA	NA

Table 3.7: ranks of true interacting pairs for Scenario 5

true pair	YAN1	YAN2	GA (d=1)	GA (d=2)	GA (d=3)	GA (d=4)	GA (d=5)	GA (d=10)
g5 & g10	2430,327,463	~ 123747	NA	NA	NA	NA	NA	NA
g20 & g40	22,86,2	4,19,378	7	1	1	1	1	1
g50 & g60	1,2,17	4683,69,5	NA	4	2	2	2	3
g1 & g3	NA	NA	NA	NA	NA	NA	NA	NA

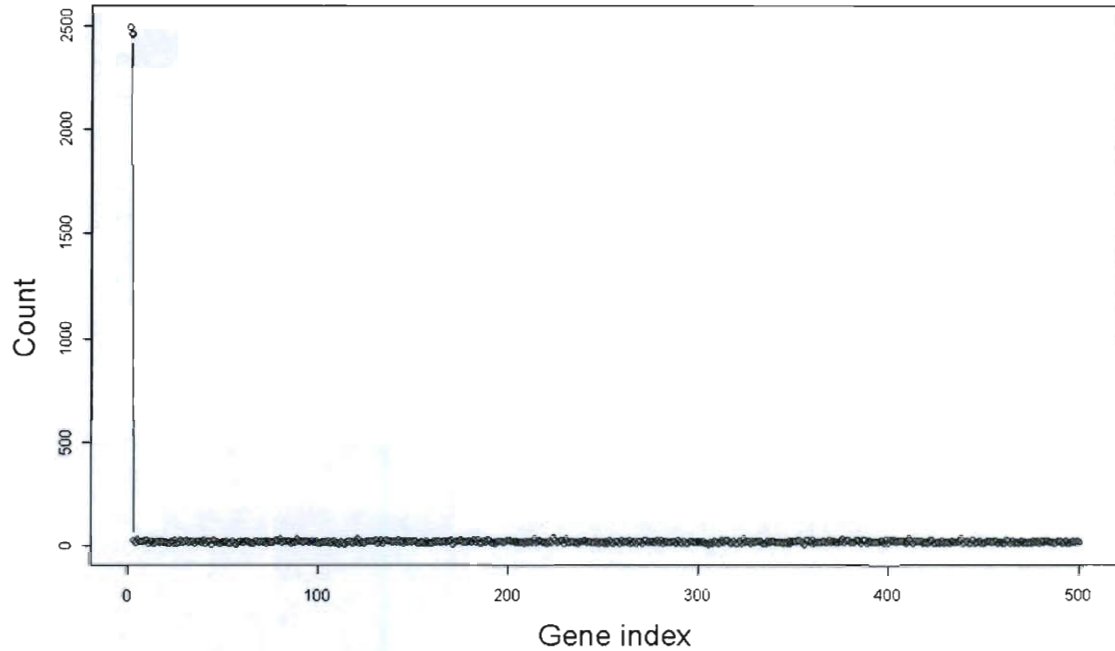


Figure 3.9: Selection frequencies of each gene for scenario4

$d$ . With a smaller  $d$ , we do detect them, and with a large  $d$ , we do not. See Figure 3.11. Again, we remove gene 1 and gene 2. In the second step of interaction detection, the ranks are shown in Table 3.8. In this scenario, for pairs 5 and 10, 20 and 40, GA does better. And for the pair 50 and 60, Yan's method does better. Notice that the pair 5 and 10 has very small rank in YAN1, and very large rank in YAN2.

**Scenario 7** For step 1, again, gene 1 and gene 2 have large selection frequencies. See Figure 3.12. In the second step of interaction detection, the ranks are shown in Table 3.9. Now, the two pairs 5 and 10 cannot be detected by Yan's method, but can be detected by GA when  $d = 2$  or 4. For the other two pairs, GA does better.

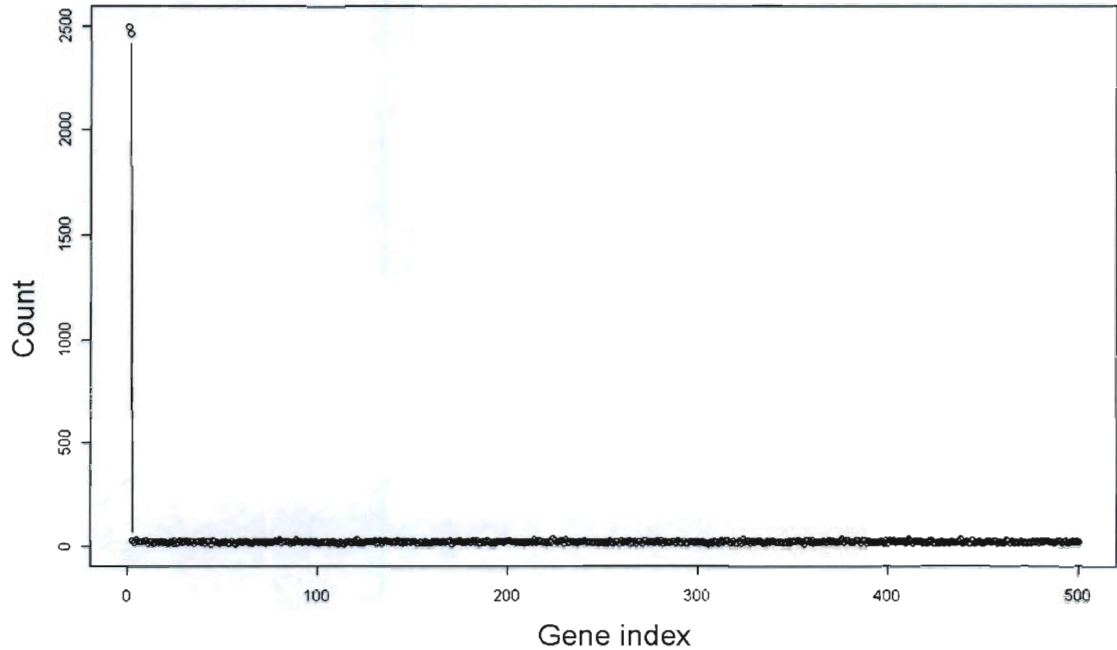


Figure 3.10: Selection frequencies of each gene for scenario 5

**Scenario 8** For step 1, the detection of 5 and 10 depend on  $d$ . See Figure 3.13. In the second step of interaction detection, the ranks are shown in Table 3.10. In this scenario, GA does better on the pair 5 and 10. Yan's method does better on the other two pairs. Because all the pairs detected by GA have either 5 or 10.

From the simulations studies, we can see that for both steps (step 1 of main effects detection, and step 2 of interaction detection), a small  $d$  ( $d = 2, 3, 4$ ) generally would be good. However,  $d = 1$  is not good because it ignores joint effects of the interacting pairs. When they are together in a GA-chromosome, the ability to classify is stronger. In most cases, GA does better than Yan's method. There are two disadvantages of Yan's method. First, the results are not stable across varying  $d$ 's. Second, the same

Table 3.8: ranks of true interacting pairs for Scenario 6

true pair	YAN1	YAN2	GA (d=1)	GA (d=2)	GA (d=3)	GA (d=4)	GA (d=5)	GA (d=10)
g5 & g10	1,4,3	> 11758	1	1	1	1	1	1
g20 & g40	2,2,6	3,45,2	112	16	7	5	38	336
g50 & g60	40,1,13	1,1,6	NA	163	303	202	608	302
g1 & g3	NA	NA	NA	NA	NA	NA	NA	NA

Table 3.9: ranks of true interacting pairs for Scenario 7

true pair	YAN1	YAN2	GA (d=1)	GA (d=2)	GA (d=3)	GA (d=4)	GA (d=5)	GA (d=10)
g5 & g10	> 581	> 100000	NA	30	73	48	142	91
g20 & g40	1,1,11	1,2,2	19	2	1	1	1	1
g50 & g60	2,2,8	3,1,16	NA	10	5	3	3	31
g1 & g3	NA	NA	NA	NA	NA	NA	NA	NA

Table 3.10: ranks of true interacting pairs for Scenario 8

true pair	YAN1	YAN2	GA (d=1)	GA (d=2)	GA (d=3)	GA (d=4)	GA (d=5)	GA (d=10)
g5 & g10	319,87,226	> 8292	22	9	55	1	1	1
g20 & g40	2,12,1	1,35,2	212	552	984	852	1371	1633
g50 & g60	1,2,5	6,52,1	350	414	293	435	569	461
g1 & g3	NA	NA	NA	NA	NA	NA	NA	NA

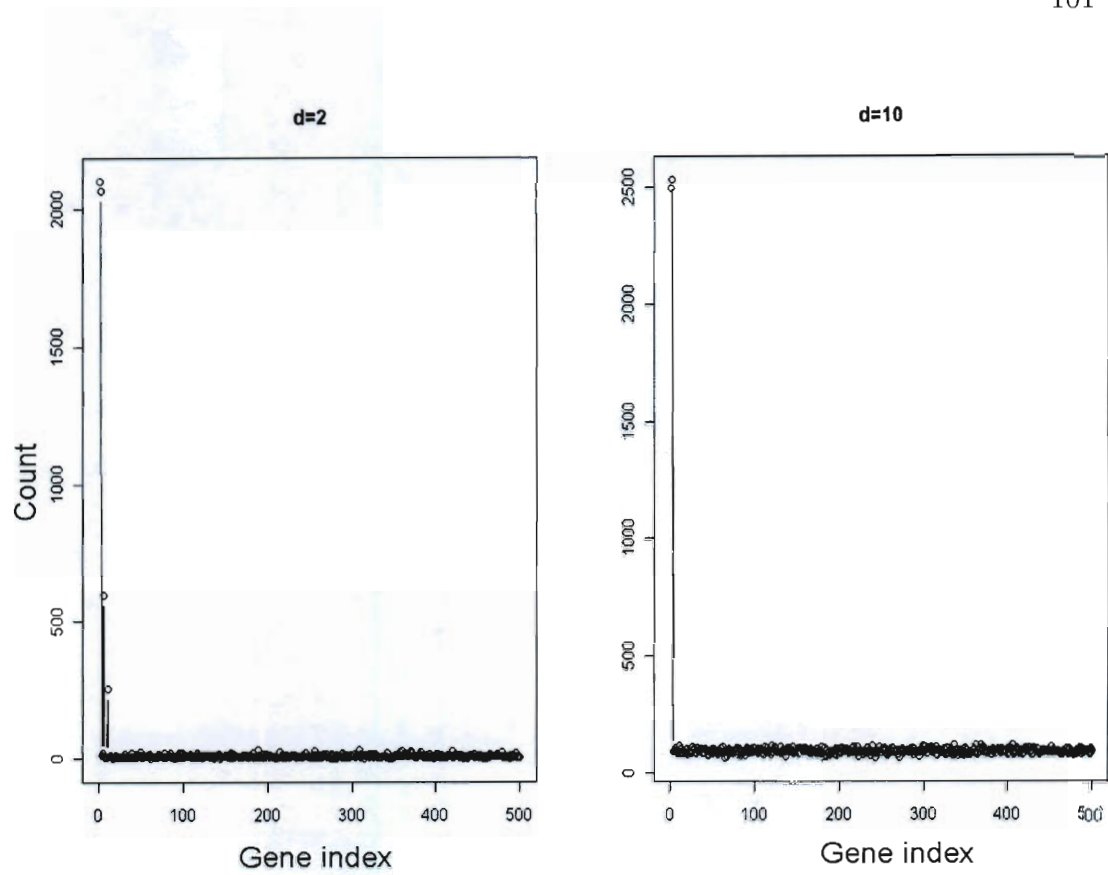


Figure 3.11: Selection frequencies of each gene for scenario 6

pair can have quite different ranks in Yan1 and Yan2, which is hard to interpret (like small false positive but high false negative, and vice-versa).



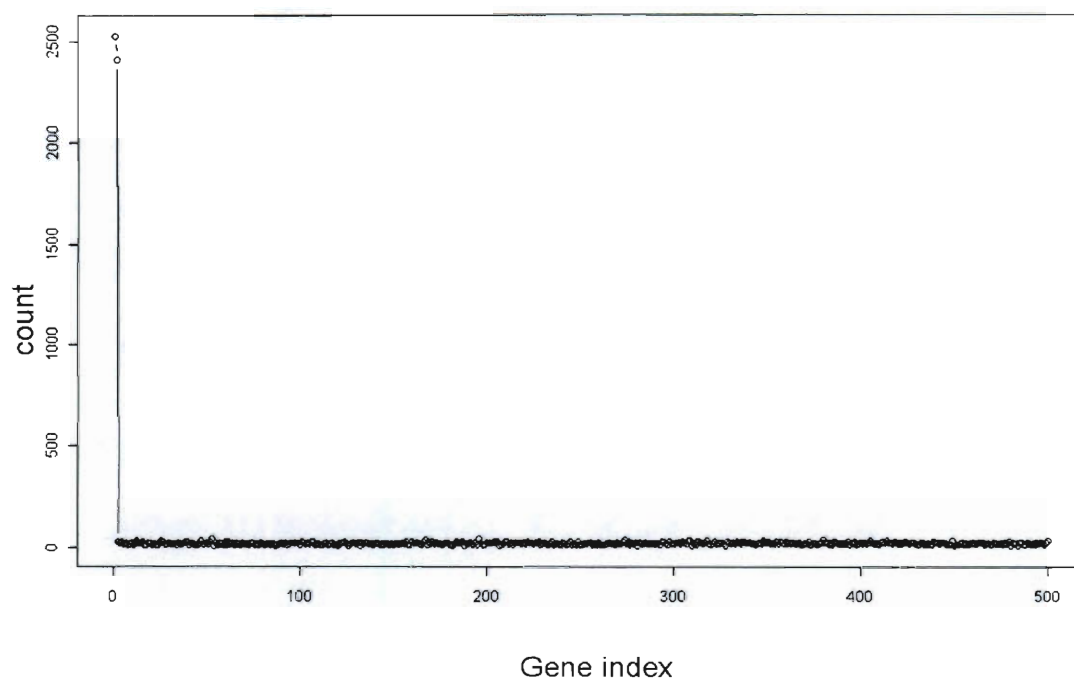


Figure 3.12: Selection frequencies of each gene for scenario 7

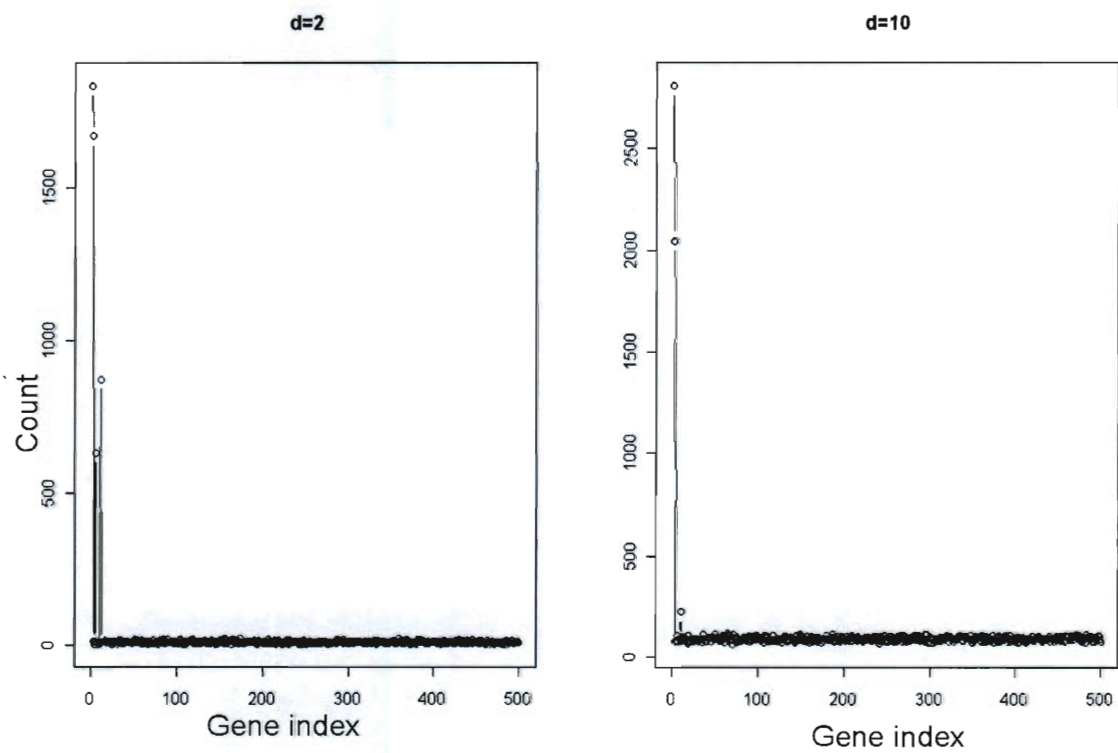


Figure 3.13: Selection frequencies of each gene for scenario 8

The above results are all based on one sample per scenario only. To see how the algorithm works in general, we did 10 replicates for each scenario. The ranks, as well as means and medians across replicates, are summarized in the Tables 3.11-3.15. Since for Yan's method, the results are highly variable across technical replicates, the sample variation will be worse and thus we decided not to do it.

Table 3.11 shows the ranks of the three pairs for scenario 1. For pairs 5&10 and 20&40, the ranks are exclusively 1, 2, or 3. For pair 50&60, most ranks are below 5, with the largest rank of 17. The median ranks are 1, 2, and 3 for the three pairs, (5, 10), (20, 40), (50, 60), respectively.

Table 3.12 shows the ranks of the three pairs for scenario 2. In this case, the ranks of pair 5&10 are consistently very large, while the ranks of the other pairs are mostly 1, 2, or 3 for the three pairs, (5, 10), (20, 40), (50, 60), respectively.

Table 3.13 shows the ranks of the three pairs for scenario 3. For all three pairs, the ranks are mostly below 5. The median ranks are 2, 3, and 2 for the three pairs, (5, 10), (20, 40), (50, 60), respectively.

Table 3.14 shows the ranks of the three pairs for scenario 4. The ranks of the pair 5&10 are consistently 1. The ranks of the other two pairs are high ( $\leq 10$ ) in some replicates, and low ( $\geq 100$ ) in other replicates.

Table 3.15 shows the ranks of the three pairs for scenario 5. In this scenario, the pair 5&10 is not detected in any replicate. For the other two pairs, the ranks are mostly 1 or 2. And the median ranks are 1 and 2.

Table 3.11: Ranks of true interacting pairs for 10 replicates for Scenario 1 under the proposed GA algorithm

replicates	5&10	20&40	50&60
1	1	3	2
2	1	2	4
3	1	3	7
4	1	3	2
5	2	1	17
6	1	2	3
7	1	2	3
8	1	2	3
9	1	2	3
10	1	2	4
Mean	1.1	2.2	4.8
Median	1	2	3

Table 3.12: Ranks of true interacting pairs for 10 replicates for Scenario 2 under the proposed GA algorithm

replicates	5&10	20&40	50&60
1	9722	1	8
2	50883	2	1
3	50202	2	1
4	9609	2	3
5	9445	1	3
6	50351	1	2
7	50622	1	3
8	50549	1	2
9	51028	2	1
10	51102	2	1
mean	38351.3	1.5	2.5
median	50450	1.5	2

Table 3.13: Ranks of true interacting pairs for 10 replicates for Scenario 3 under the proposed GA algorithm

replicates	5&10	20&40	50&60
1	1	3	2
2	3	4	5
3	2	1	3
4	2	3	1
5	1	3	2
6	3	1	2
7	1	2	7
8	2	9	1
9	2	3	1
10	2	1	4
mean	1.9	3	2.8
median	2	3	2

Table 3.14: ranks of true interacting pairs for 10 replicates for Scenario 4 under the proposed GA algorithm

replicates	5&10	20&40	50&60
1	1	3	24271
2	1	13	2435
3	1	17	4
4	1	5	40276
5	1	159	183
6	1	2	165
7	1	298	47
8	1	3	1162
9	1	6	12
10	1	1531	127
mean	1	203.7	6868.2
median	1	9.5	174

Table 3.15: ranks of true interacting pairs for 10 replicates for Scenario 5 under the proposed GA algorithm

replicates	5&10	20&40	50&60
1	10100	1	2
2	340	1	2
3	2948	1	2
4	50696	1	2
5	628	1	2
6	9296	2	1
7	50445	1	2
8	1164	1	25
9	2942	1	3
10	3034	2	1
mean	13159.3	1.2	4.2
median	2991	1	2



### 3.6 Real data application

We also applied the algorithm to the breast cancer data studied by van't Veer *et al.* [2002]. In this dataset, expression values of 24881 genes were measured for 44 good prognosis breast cancer samples and 34 poor prognosis breast cancer samples. We use the 4918 genes obtained by Tibshirani *et al.* [2002]. Each gene is standardized by its mean and standard deviation, so that a gene has mean 0 and variance 1 across individuals.

First, we directly applied the genetic algorithm in Li *et al.* [2001a] to select genes with main effects. We used chromosome length  $d = 3$ , threshold  $\gamma = 60$ , and number of best chromosomes 15000. The selection frequencies are shown in Figure 3.14.

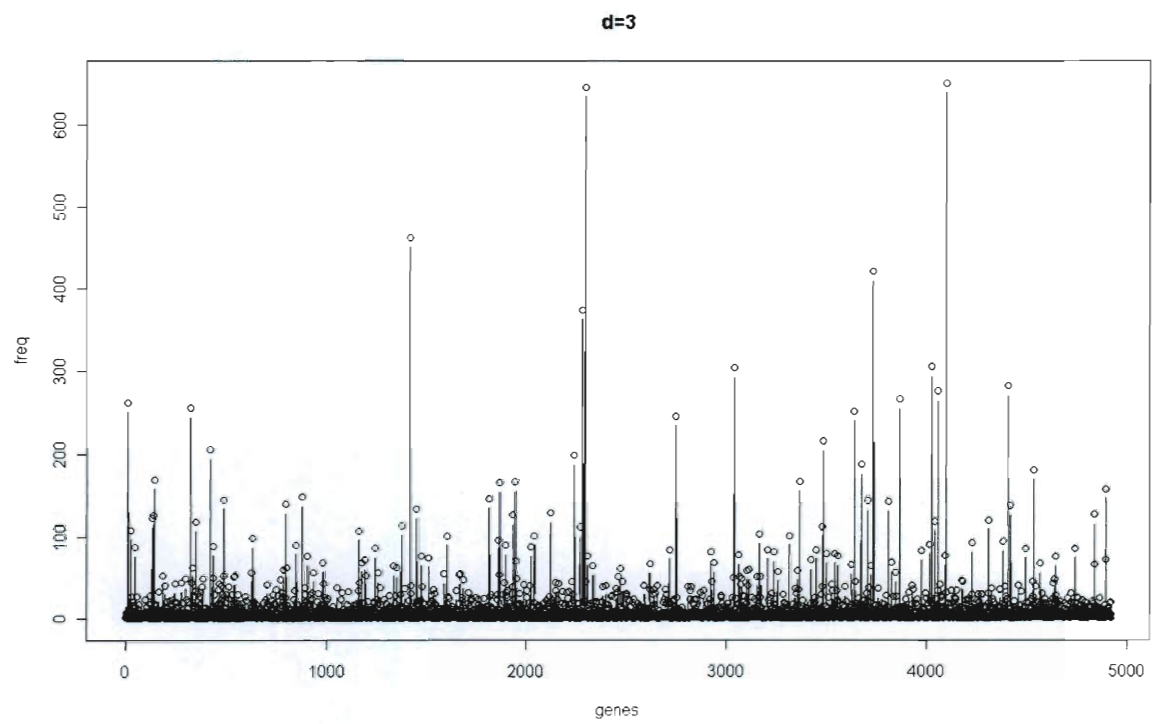


Figure 3.14: Selection frequencies of each gene for breast cancer data

Second, we removed the top 100 genes with largest selection frequencies, and ran the proposed method with  $\gamma = 61$ ,  $d = 3$ , and number of best chromosomes 150000. Figure 3.15 to Figure 3.22 show eight pairs of genes which appear in the top part of the ordered list. In each of the eight plots, red and green dots represent samples in the two classes. Red and green lines are the regression lines of the two sets of samples. Red and green curves are the lowess curves with window size .3. X-axis and y-axis names are gene names. Table 3.16 lists the pearson and spearman correlations for the two groups in each of the eight selected pairs of genes.

For pair 1, we can see in Figure 3.15 that the expression profiles show the cross pattern discussed earlier. The Pearson correlations for the two groups are  $-.347$  and  $.435$  respectively. The Spearman correlations are  $-.279$  and  $.416$  respectively. The two groups have correlations with opposite signs.

For pair 2, from Figure 3.16, except for some outliers, the red class has a very tight distribution centered around  $(-.5, -1)$ . The green class, however, is more spread out with a negative correlation. The Pearson correlations for the two groups are  $-.549$  and  $.268$  respectively. The Spearman correlations are  $-.618$  and  $.076$  respectively. The two groups have correlations with opposite signs.

For pair 3 shown in Figure 3.17, we can see that the two classes have very different distributions with different means and correlations. The Pearson correlations for the two groups are  $-.382$  and  $.231$  respectively. The Spearman correlations are  $-.464$  and  $.031$  respectively. The two groups have correlations with opposite signs.

For pair 4 in Figure 3.18, the green class has a clear pattern with positive correlation, while the red class has a tighter distribution. The Pearson correlations for the two groups are  $.626$  and  $0.076$  respectively. The Spearman correlations are  $.626$  and

−.23 respectively. The two groups have correlations with opposite signs.

Pair 5 shows cross pattern as seen in Figure 3.19. The Pearson correlations for the two groups are .471 and −.19 respectively. The Spearman correlations are .564 and −.152 respectively. The two groups have correlations with opposite signs.

Pair 6 in Figure 3.20 shows some cross pattern. The Pearson correlations for the two groups are .254 and −.717 respectively. The Spearman correlations are .22 and −.614 respectively. The two groups have correlations with opposite signs.

Pair 7 also kind of shows the cross pattern. The Pearson correlations for the two groups are .652 and −.354 respectively. The Spearman correlations are .544 and −.304 respectively. The two groups have correlations with opposite signs.

In Figure 3.22, pair 8 shows similar pattern as pair 2. The Pearson correlations for the two groups are −.429 and .271 respectively. The Spearman correlations are −.459 and .169 respectively. The two groups have correlations with opposite signs.

Table 3.16: Correlations of the two groups

	pair1	pair2	pair3	pair4	pair5	pair6	pair7	pair8
Pearson gp1	-0.347	-0.549	-0.382	0.626	0.471	0.254	0.652	-0.429
Spearman gp1	-0.279	-0.618	-0.464	0.626	0.564	0.220	0.544	-0.459
Pearson gp2	0.435	0.268	0.231	-0.076	-0.190	-0.717	-0.354	0.271
Spearman gp2	0.416	0.076	0.031	-0.230	-0.152	-0.614	-0.304	0.169

One thing to mention is that in scenario 4, where the correlations between gene pair 50 and 60 are .5 and −.5 for the two classes, this pair cannot be detected in most of the replicates. However, in real data application, we detected much weaker

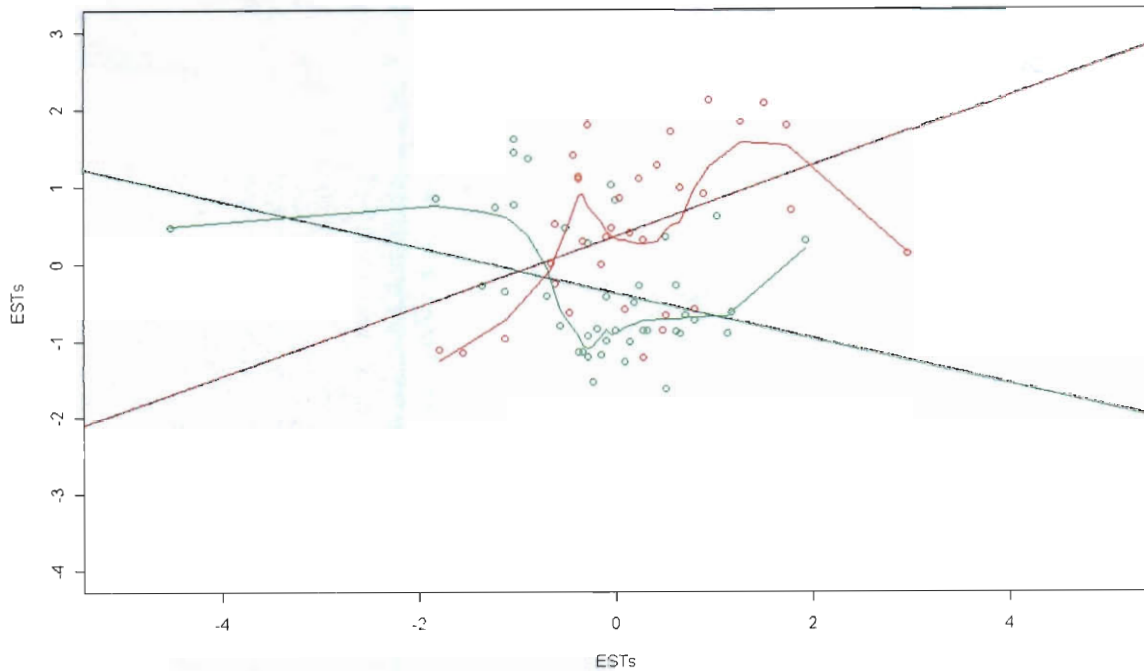


Figure 3.15: Joint pattern of gene pair 1 identified by GA

interactions, for example, pair 1 (with pearson correlations  $-.347$  and  $.435$ ) and pair 8 (with pearson correlations  $-.429$  and  $.271$ ). This suggests that whenever a pair has the strongest interaction, it will have high rank, no matter how strong it is. In scenario 4, genes 5 and 10 have the strongest interaction, so its rank is mostly 1. The pair 50 and 60 has much weaker interaction, so sometimes we cannot detect it. In scenario 2, gene 5 and gene 10 do not interact, so now the interaction between 20 and 40 is the strongest, so its rank is mostly 1.

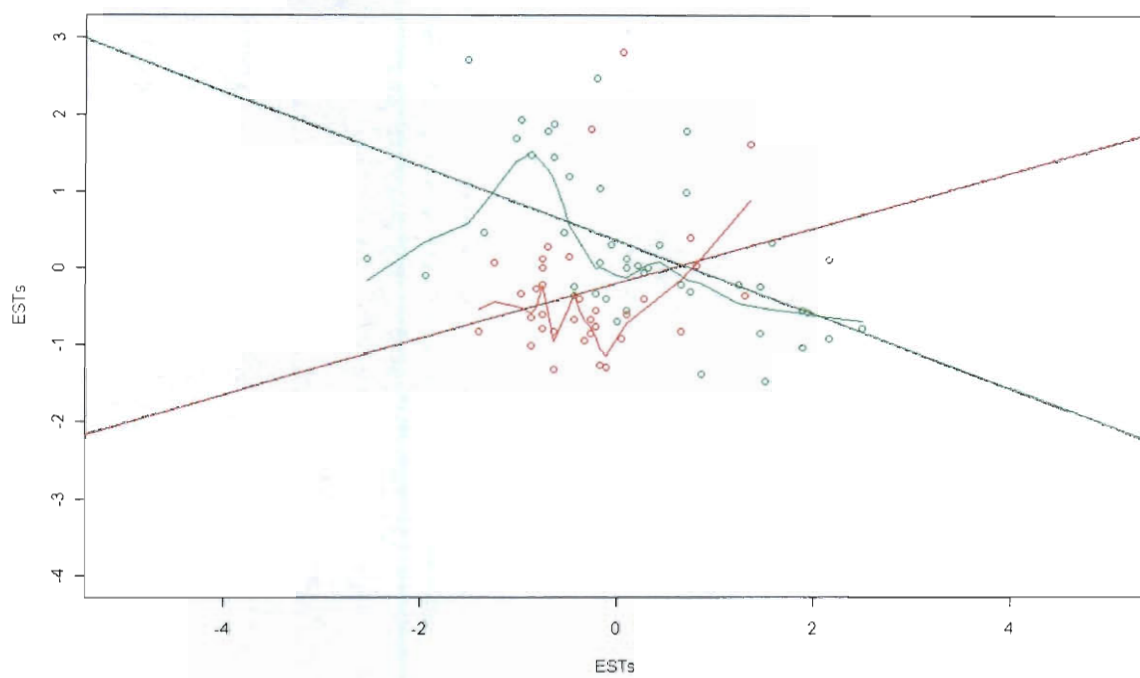


Figure 3.16: Joint pattern of gene pair 2 identified by GA

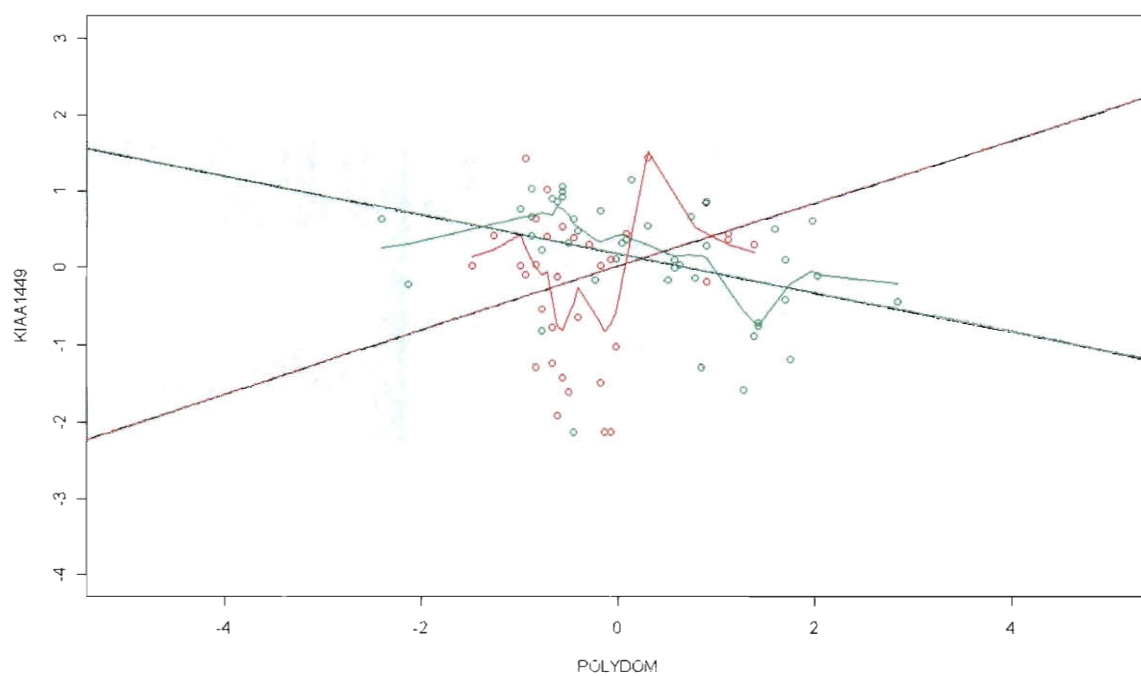


Figure 3.17: Joint pattern of gene pair 3 identified by GA

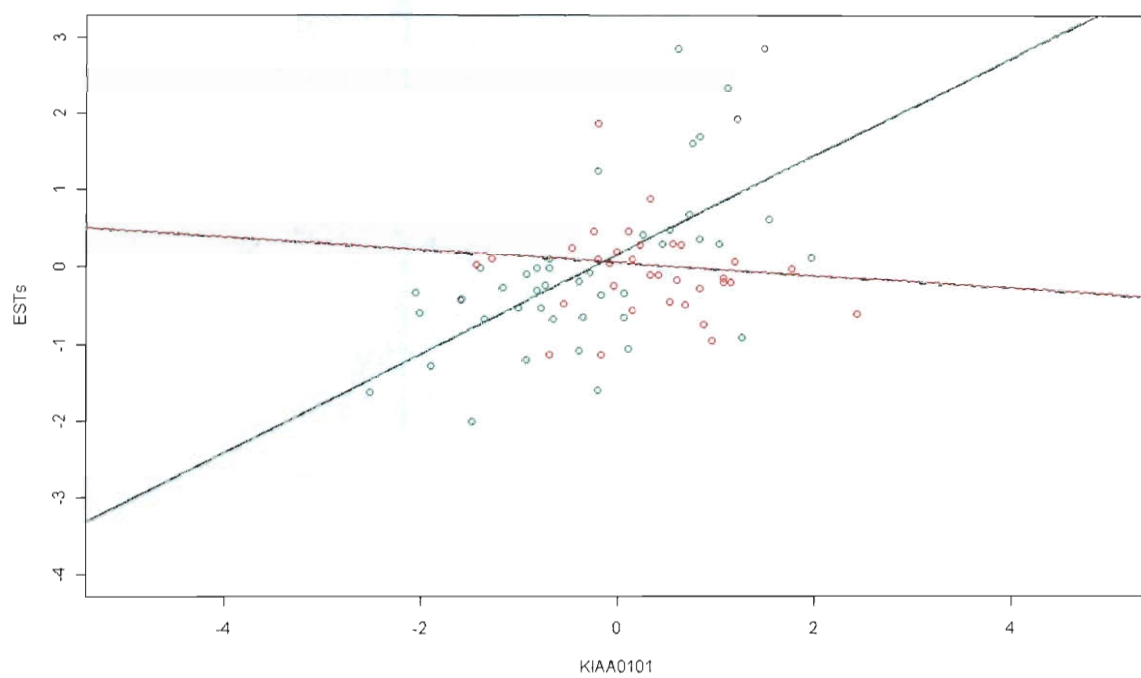


Figure 3.18: Joint pattern of gene pair 4 identified by GA



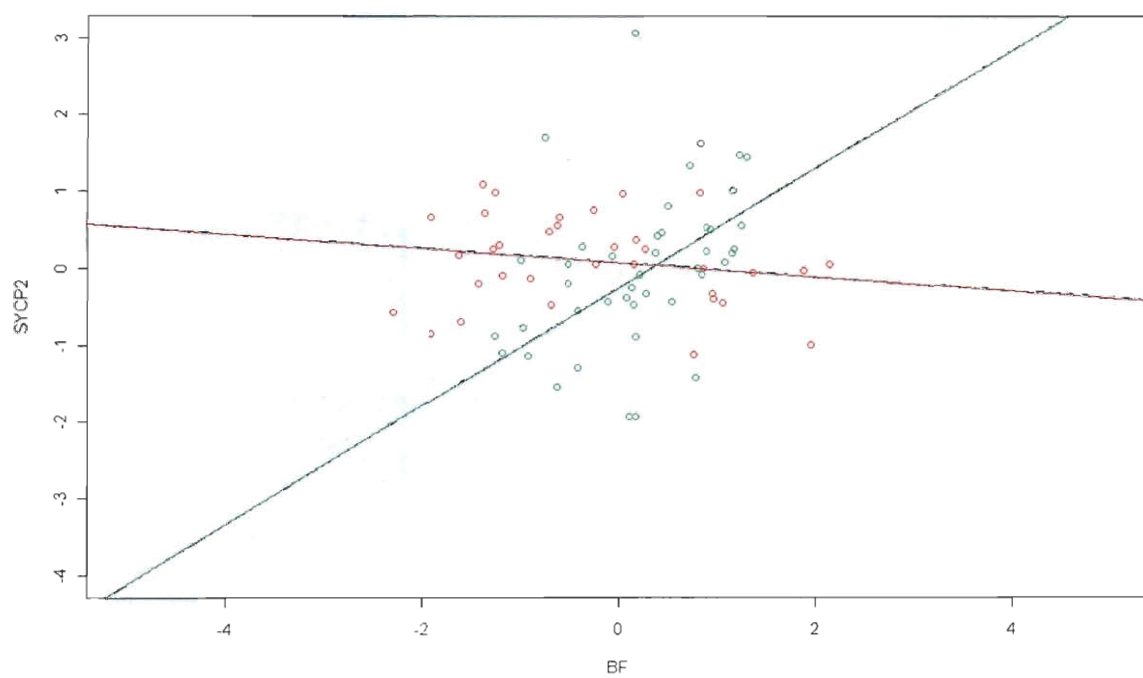


Figure 3.19: Joint pattern of gene pair 5 identified by GA

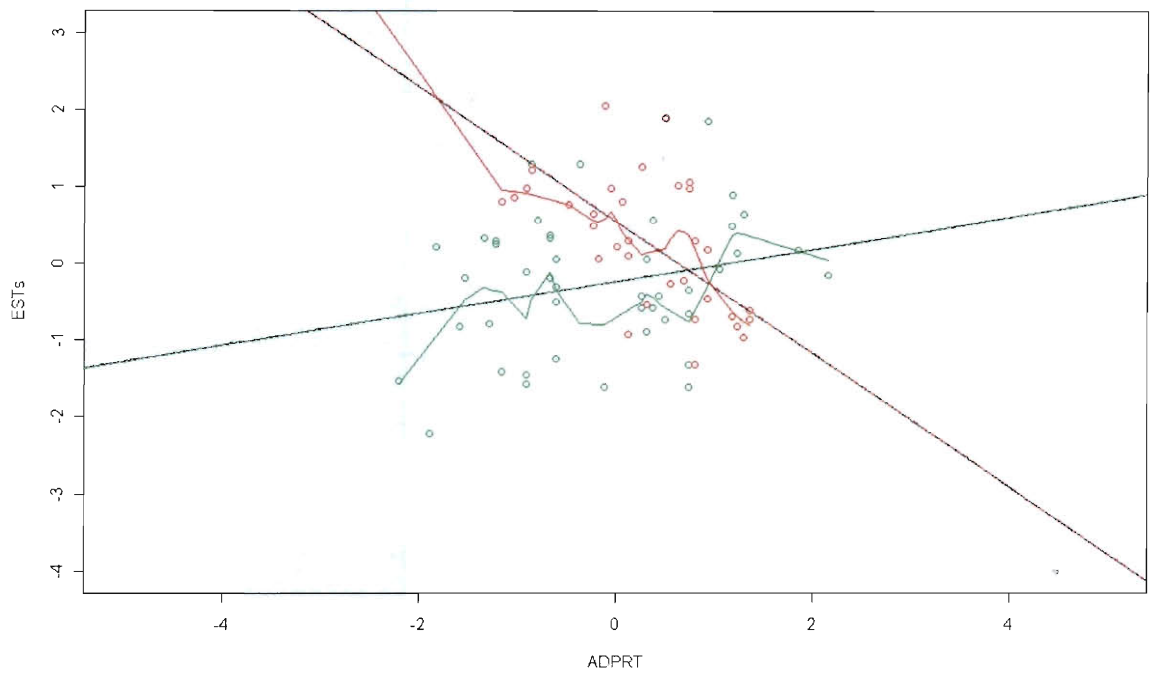


Figure 3.20: Joint pattern of gene pair 6 identified by GA

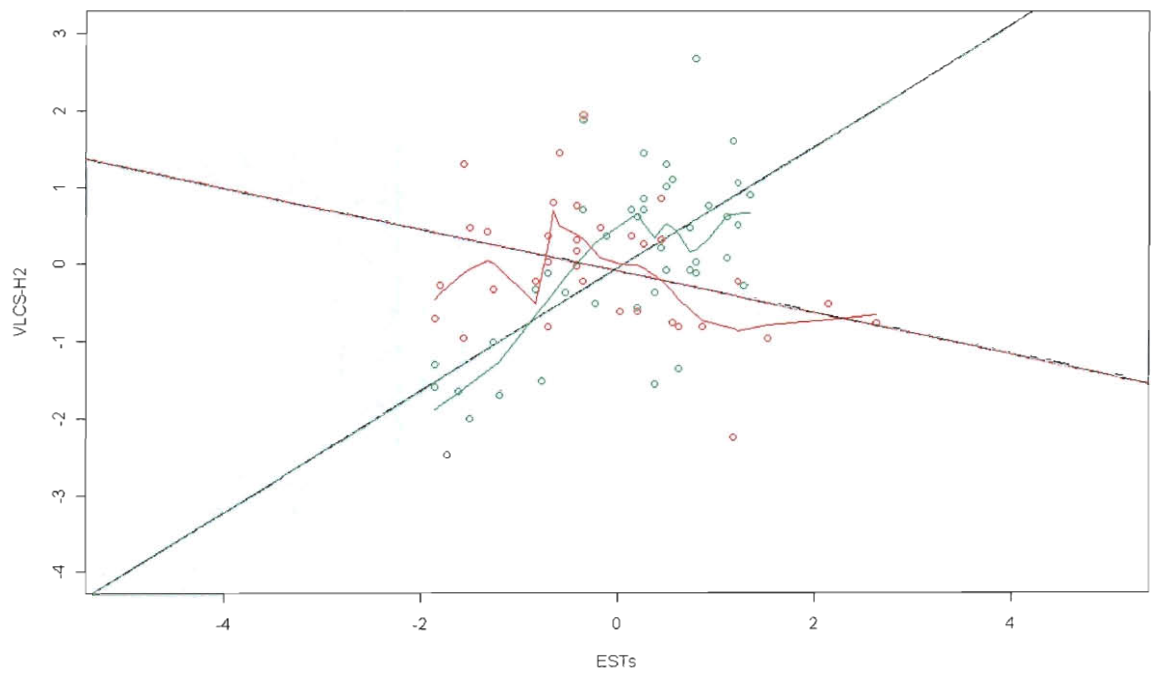


Figure 3.21: Joint pattern of gene pair 7 identified by GA

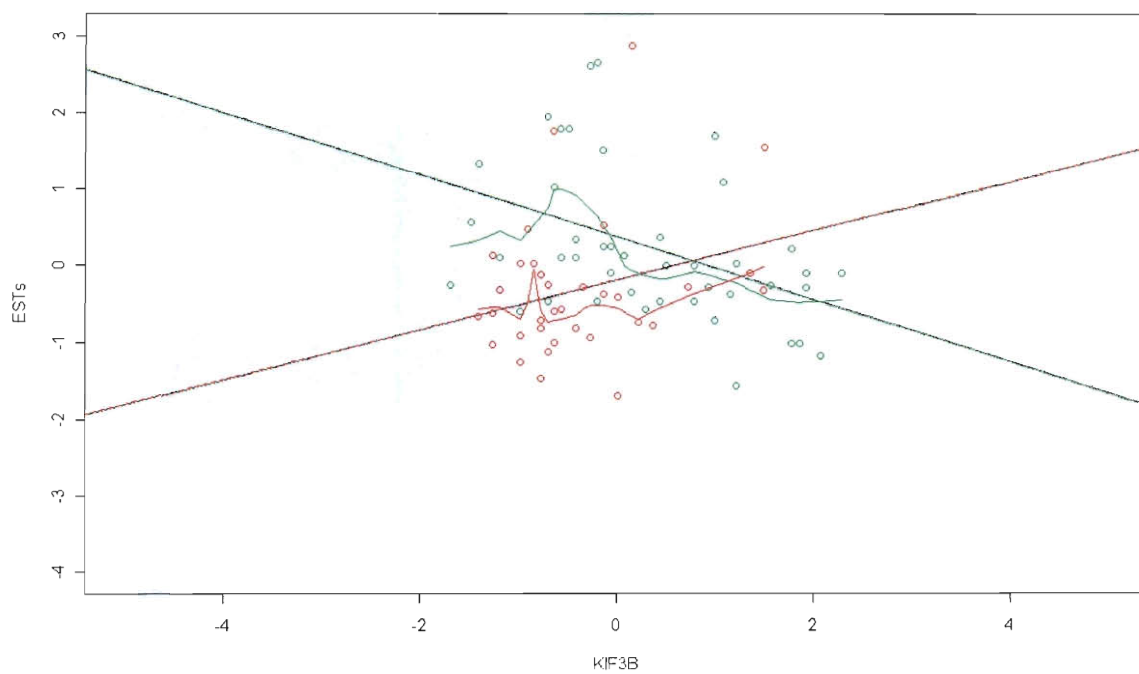


Figure 3.22: Joint pattern of gene pair 8 identified by GA

As a comparison, I also present here eight plots (Figure 3.23 to Figure 3.30) for eight pairs of genes with selection frequency of 0. The corresponding correlations are shown in Table 3.17. From the plots and correlations, we don't see any pattern in the expression profiles and the correlations of the two classes are very similar.

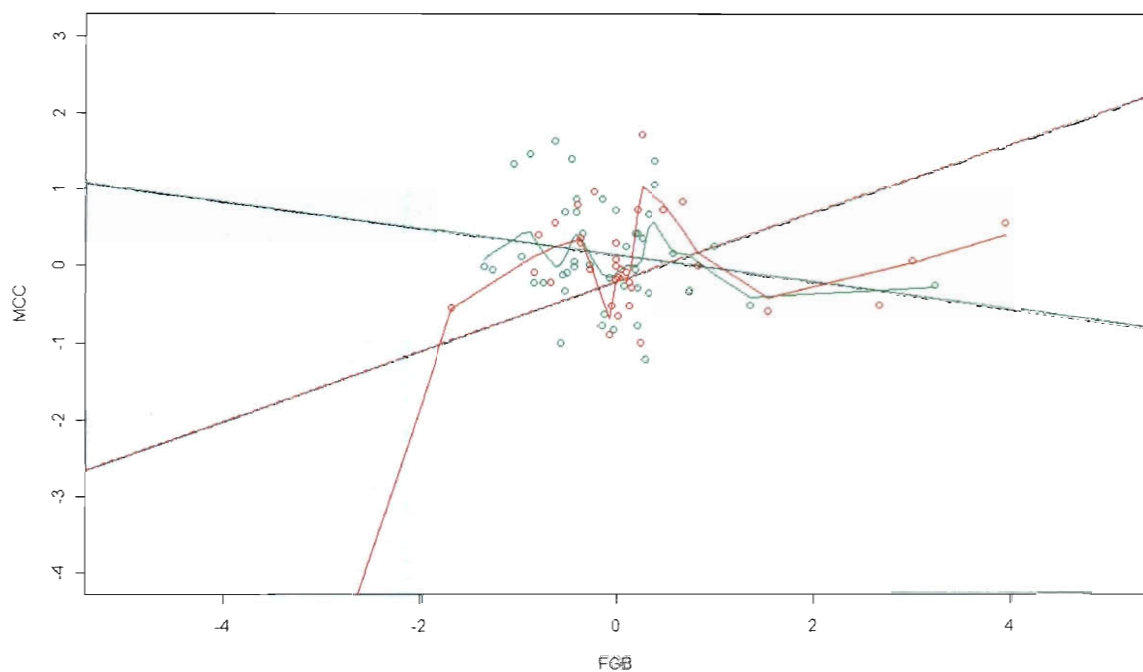


Figure 3.23: Joint pattern of gene pair 1 with selection frequency of 0

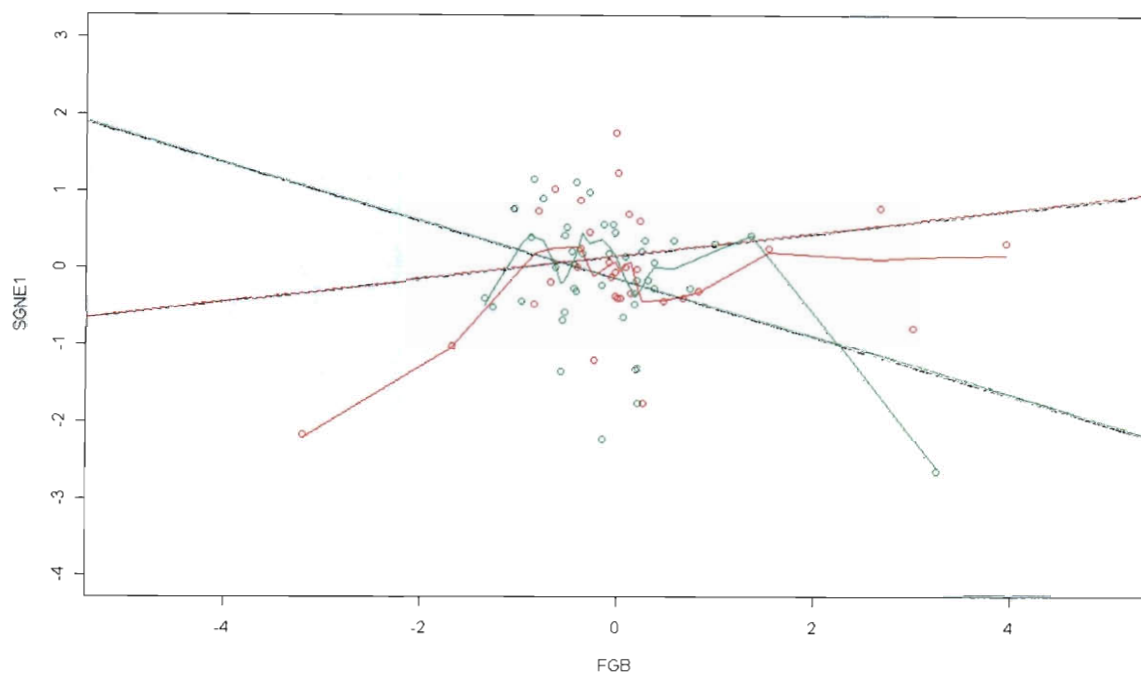


Figure 3.24: Joint pattern of gene pair 2 with selection frequency of 0

Table 3.17: Correlations of the two groups

	pair1	pair2	pair3	pair4	pair5	pair6	pair7	pair9
pearson gp1	-0.194	-0.360	-0.152	-0.040	0.258	-0.085	0.083	0.113
spearman gp1	-0.163	-0.097	-0.034	0.035	0.212	-0.156	-0.013	0.099
pearson gp2	0.433	0.154	0.035	0.101	-0.092	-0.374	0.146	-0.059
spearman gp2	0.080	0.030	0.078	0.019	-0.050	-0.187	0.055	-0.073

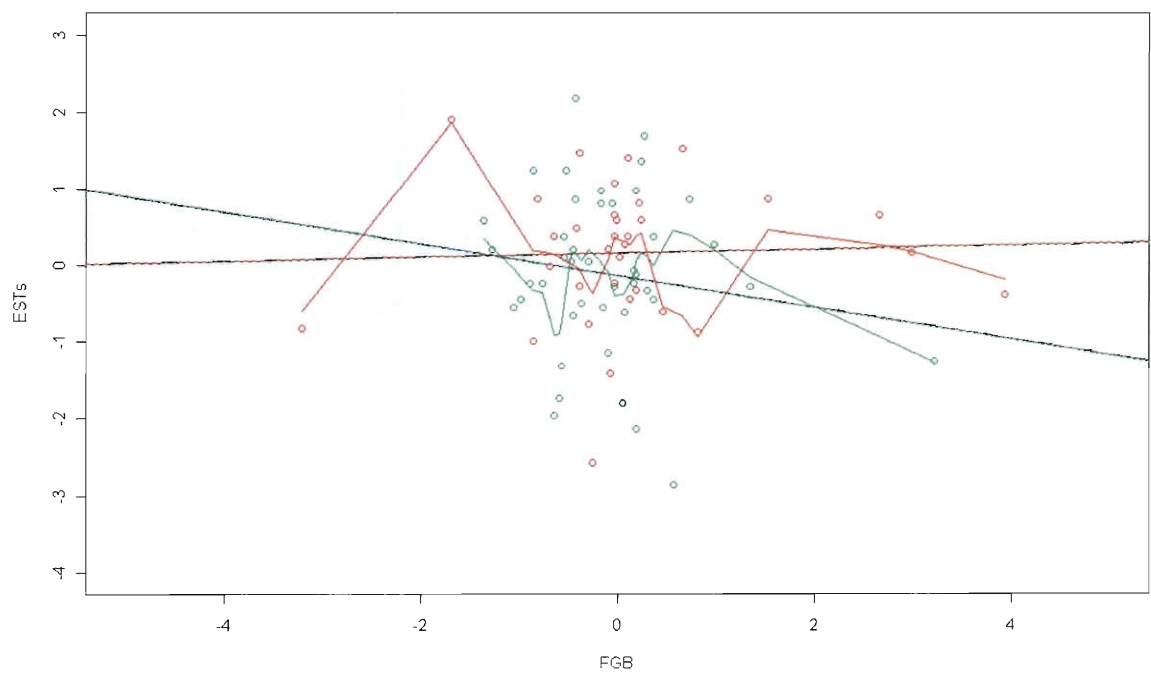


Figure 3.25: Joint pattern of gene pair 3 with selection frequency of 0

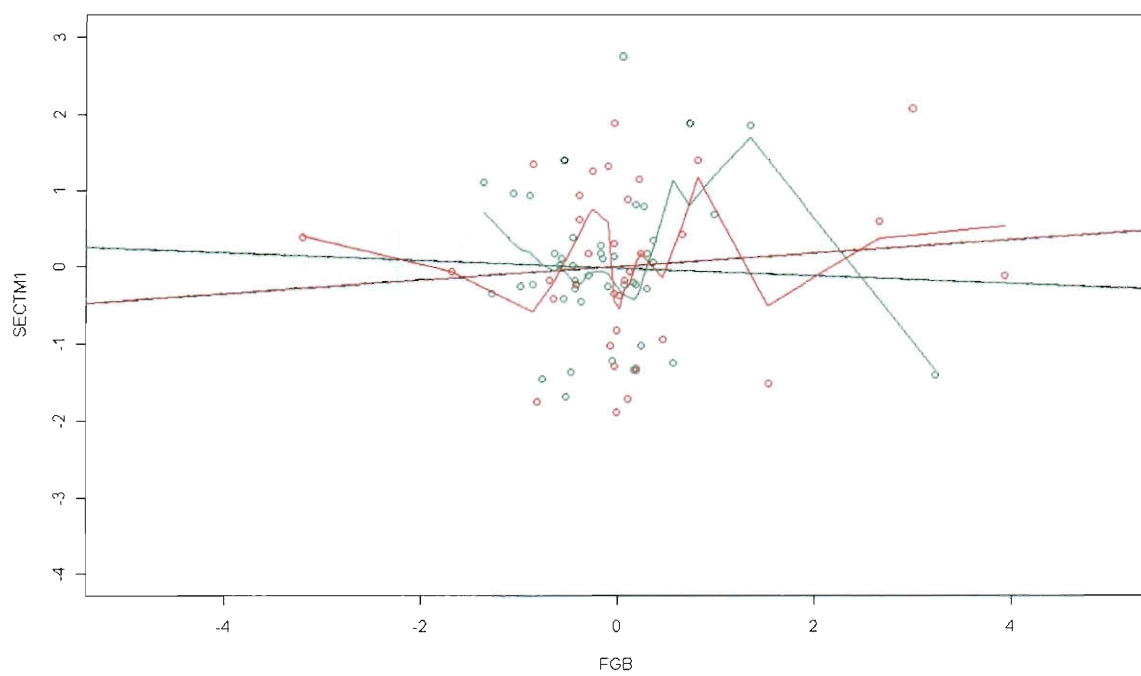


Figure 3.26: Joint pattern of gene pair 4 with selection frequency of 0



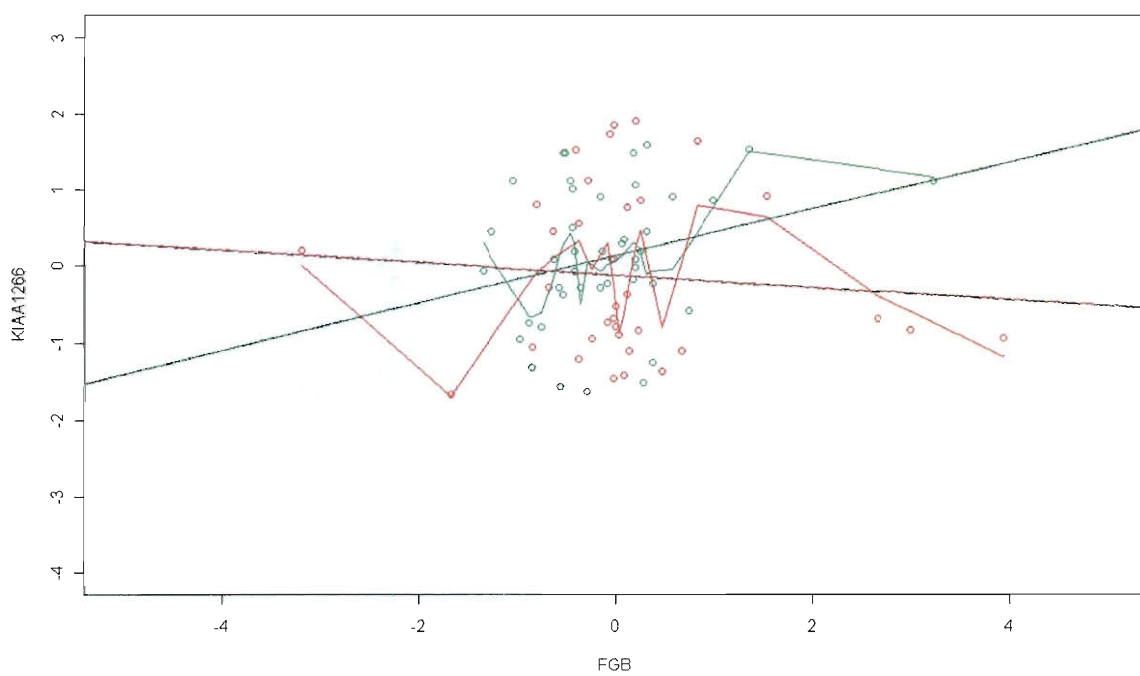


Figure 3.27: Joint pattern of gene pair 5 with selection frequency of 0

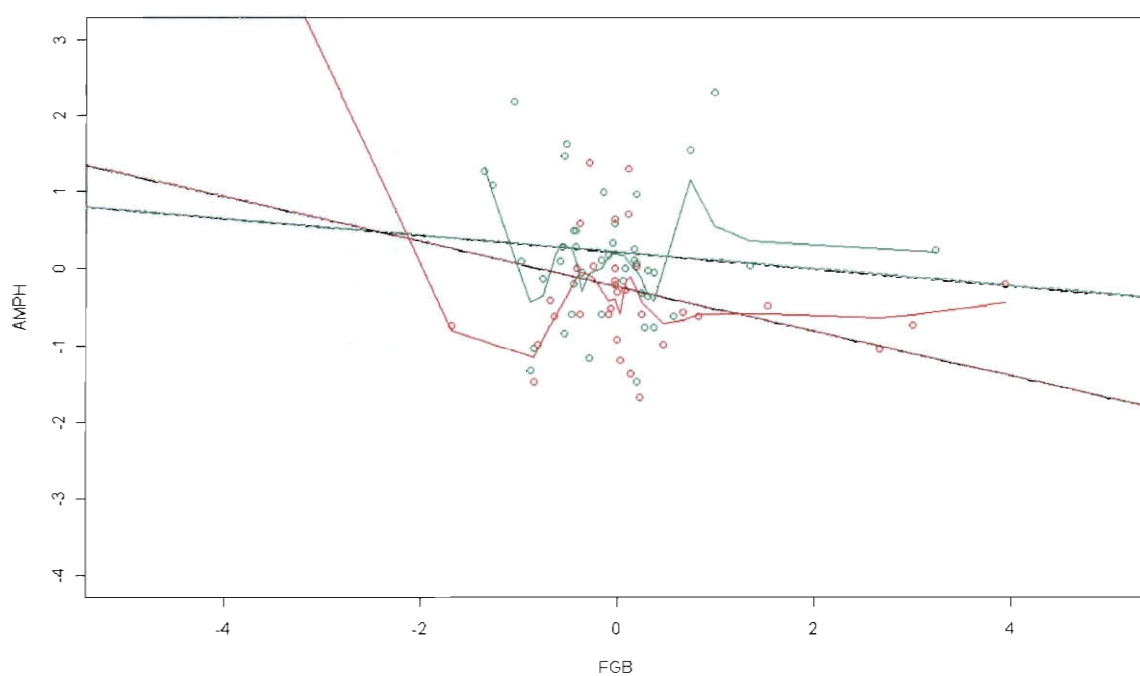


Figure 3.28: Joint pattern of gene pair 6 with selection frequency of 0

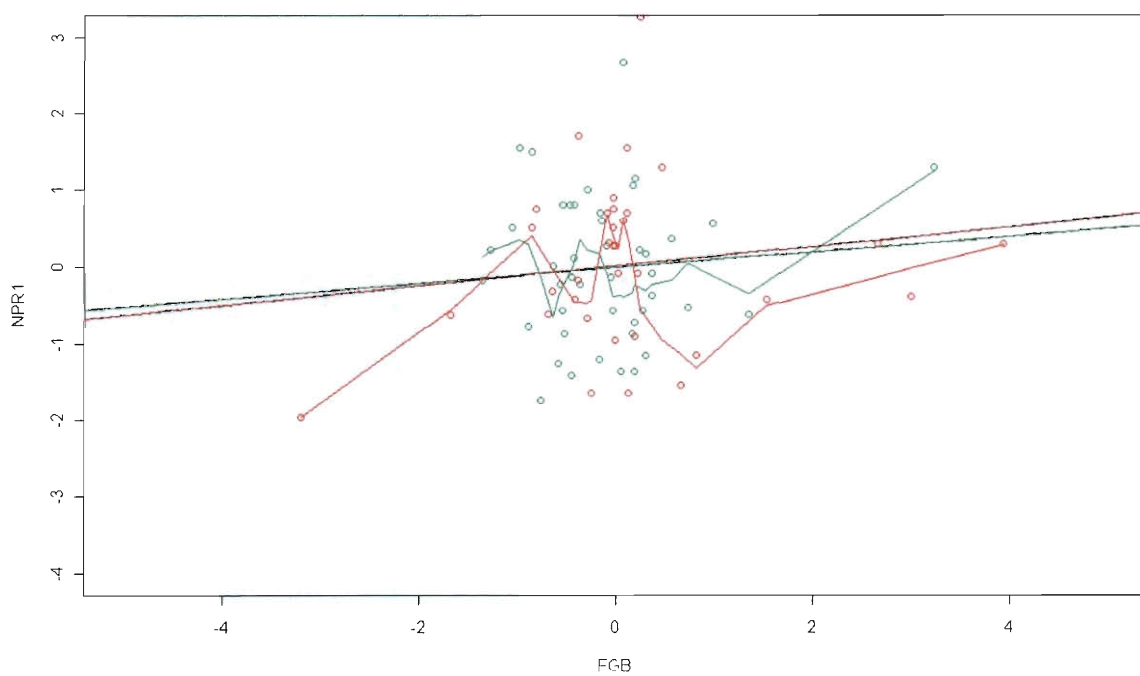


Figure 3.29: Joint pattern of gene pair 7 with selection frequency of 0

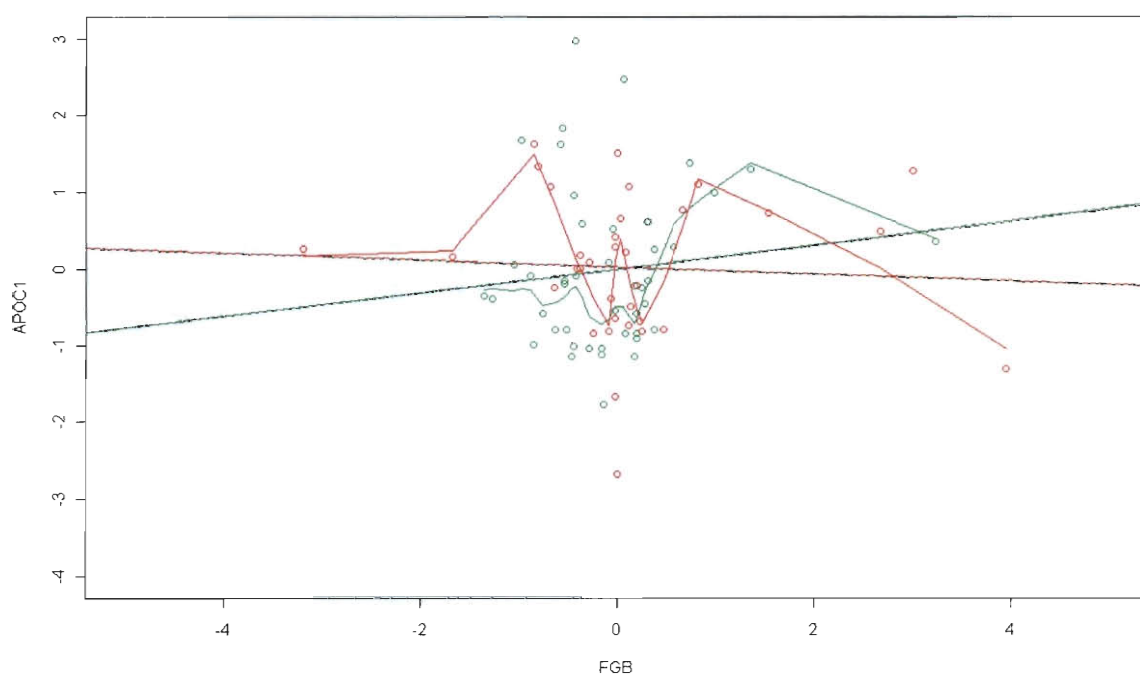


Figure 3.30: Joint pattern of gene pair 8 with selection frequency of 0

### 3.7 Conclusion

A reliable classification of cancer is very important to cancer diagnosis, treatment and prevention. A very important issue is to select a set of informative genes, and then use these genes to classify the samples. Recent technology of microarrays allows the expression levels of thousands of genes to be measured simultaneously. Most classification methods using gene expression data use some uni-variate approach to select informative genes, like t-test. This approach ignores class information contained in gene interactions. Most statistical methods aiming at detecting interactions are based on regression, with cross terms representing interactions. This is a parametric approach, and since the number of cross terms increases rapidly as the number of genes increases, most methods first use some threshold to select a small number of genes, and then apply regression on these selected genes. This way, interactions with no or mild main effects are ignored. We have proposed a new non-parametric method to detect gene-gene interaction for classification using genetic algorithm. We have shown with a simulation study that our algorithm can detect interacting gene pairs very precisely. Our results do not show sensitivity to different values of chromosome lengths ( $d$ ) as long as  $d$  is not too large. By applying our method to real cancer data, we have demonstrated that our method can detect interactions in the sense we defined earlier. The genes in these pairs do not have main effects, so they would escape the detection by uni-variate methods like t-test. This method could be extended in a few ways. Instead of removing a certain number of genes with main effects based on the selection frequency, some threshold can be calculated probabilistically to select genes to remove. Similarly, in the second step of detecting interactions, we can use some threshold to select pairs of genes. The current method cannot detect interacting gene pairs with one gene having large main effects. As a future direction, the method can be extended to detect these interactions as well.

# Bibliography

- Alvarez-Castro, J (2008) How to perform meaningful estimates of genetic effects. *PLoS genetics*, **4**, e1000062.
- Autio,R *et al* (2003) CGH-Plotter: MATLAB toolbox fo CGH-data analysis. *Bioinformatics*, **19**, 1714-1715.
- Bateson,W (1909) Mendel's principles of heredity. *Cambridge university press*.
- Beasley,D *et al.* (1993) An overview of Genetic Algorithms: Part 1, Fundamentals. *University Computing*, **15**, 58-69.
- Besag,J.,Kooperberg,C. (1995) On conditional and intrinsic autoregressions. *Biometrika*, **82**, 733-342
- Bignell,G *et al.* (2004) High-resolution analysis of DNA copy number using oligonucleotide microarrays. *Genome Res*, **14**, 287-95.
- Broet,P.,Richardson,S. (2006) Detection of gene copy number changes in CGH microarrays using a spatially correlated mixture model. *Bioinformatics*, doi:10.1093/bioinformatics/btl035.
- Consortium IH (2003) The international HapMap project. *Nature*, **426**, 789-96.
- Clark,D.E. *et al.* (1996) Evolutionary algorithms in computer aided molecular design. *J Comp Aided Mol Des*, **10**, 337-358.

- Colella, S. *et al* (2007) QuantiSNP: an objective Bayes hidden-Markov model to detect and accurately map copy number variation using SNP genotyping data. *Nucleic Acids Research*, **35**, 2013-2025.
- Cordell, H. (2002) Epistasis: what it means, what it doesn't mean, and statistical methods to detect it in humans. *Human molecular genetics*, **11**, 2463-2468.
- Demuth, J. *et al*. (2006) Experimental methods for measuring gene interactions. *Annu. Rev. Ecol. Evol. Syst.*, **37**, 289-316.
- Diskin, S.J. *et al* (2007) Adjustment of genomic wave in signal intensities from whole-genome SNP genotyping platforms. *Nuc Acids Res*, **36**, e126.
- Dixon, P. (1948) Testing spatial segregation using a nearest neighbor contingency table. *Ecology*, **75**, 1940-1948.
- Douglas, E. *et al* (2004) Array Comparative Genomic Hybridization Analysis of Colorectal Cancer Cell Lines and Primary Carcinomas. *Cancer Research*, **64**, 4817-4825.
- Donlon, T.A. *et al* (1986) Isolation of molecular probes associated with the chromosome 15 instability in the Prader-Willi syndrome. *Proc. Natl. Acad. Sci. USA*, **83**, 4408-4412.
- Dudoit, S. and Fridlyand, J. (2002) Comparison of discrimination methods for the classification of tumors using gene expression data. *J.R. Statist. Soc. B*, **97**, 77-87.
- Fernandez, C., Green, P. (2002) Modelling spatially correlated data via mixtures: a Bayesian approach. *Royal Statistical Society. B*, **64**, 805-826.
- Fisher, R.A. (1918) The correlation between relatives on the supposition of Mendelian inheritance. *Trans. R. Soc. Edin*, **52**, 399-433.
- Forrest, S. (1993) Genetic algorithms: principles of natural selection applied to computation. *Science*, **261**, 872-878.

- Golub, TR. (1999) Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, **286**, 531-537.
- Green,P. (1995) Reversible-Jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, **82**, 711-732.
- Hastie,T. *et al.* (2001) The elements of statistical learning: data mining, inference, and prediction. *Springer*.
- Hastings,W. (1970) Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57**, 97-109.
- Hodgson,G *et al* (2001) Genome scanning with array CGH delineates regional alterations in mouse islet carcinomas. *Nature Genetics*, **29**, 459-464.
- Hoh, J. (2003) Mathematical multi-locus approaches to localizing complex human trait genes. *Nat Rev*, **4**, 701-709.
- Holland,J.H. (1975) Adaptation in natural and artificial systems. *MIT press*.
- Hutter, M. (2007) Exact Bayesian regression of piecewise constant function. *J Bayesian Anal*, **2**, 1635-664.
- Izraeli, S (2005) Chromosome copy number and leukemia-lessions from Down's syndrome. *Hematology*, **10**, 164-6.
- Knuutila,S (1998) DNA copy number amplifications in human neoplasms: review of comparative genomic hybridization studies. *JAm J Pathol.*, **152**, 1107-23.
- Knuutila,S (1999) DNA copy number losses in human neoplasms. *JAm J Pathol.*, **155**, 683-94.
- Lai, WR *et al* (2005) Comparative analysis of algorithms for identifying amplifications and deletions in array CGH data. *Bioinformatics*, **19**, 3763-3770.



- Li,L *et al.* (2001a) Gene assessment and sample classification for gene expression data using a genetic algorithm/k-nearest neighbor method. *Combinatorial Chemistry and High Throughput Screening*, **4**, 727-739.
- Li,L *et al.* (2001b) Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. *Bioinformatics*, **17**, 1131-1142.
- Li,C., Wong, W.H. (2001) Model-based analysis of oligonucleotide arrays: expression index computation and outlier detection. *PNAS*, **98**, 31-36.
- Linn,S *et al* (2003) Gene expression patterns and gene copy number changes in dermatofibrosarcoma protuberans. *American Journal of Pathology*, **163**, 2383-2395.
- Marchini,J *et al.* (2005) Genome-wide strategies for detecting multiple loci that influence complex diseases. *Nature Genetics*, **37**, 413-417.
- Marioni, JC *et al* (2007) Breaking the waves: improved detection of copy number variation from microarray-based comparative genomic hybridization. *Genome Biol*, **8**, R228.
- McLachlan,G. and Peel,D. (2000) Finite mixture models. *Wiley, New York*.
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M.N., Teller, A.H., and Teller, E. (1953) Equations of state calculations by fast computing machine. *J.Chem.Phys.*, **21**, 1087-1091.
- Musani, S *et al* (2007) Detection of gene gene interactions in genome-wide association studies of human population data. *Human heredity*, **63**, 67-84.
- Nannya,Y. *et al.* (2005) A robust algorithm for copy number detection using high-density oligonucleotide single nucleotide polymorphism genotyping arrays. *Cancer Res*, **65**, 14.

- Olshen,A *et al* (2004) Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics*, **5**, 557-572.
- Phillips, P (2008) Epistasis-the essential role of gene interactions in the structure and evolution of genetic systems. *Nature reviews*, **9**, 855-867.
- Pique-Regi,R. *et al* (2008) Sparse representation and Bayesian detection of genome copy number alterations from microarray data. *Bioinformatics*, **24**, 309-318.
- Pinkel,D. *et al.* (1998) High resolution analysis of DNA copy number variation using comparative genomic hybridization to microarrays. *Nature America Inc*, **20**, 207-211.
- Pollack,J *et al* (2002) Microarray analysis reveals a major direct role of DNA copy number alteration in the transcriptional program of human breast tumors. *Proceedings of the National Academy of Sciences USA*, **99**, 12963-12968.
- Rabiner, L. (1989) A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, **77**, 257-286.
- Rancoita, P.MV *et al* (2000) Bayesian DNA copy number analysis. *BMC Bioinformatics*, **10**, doi:10.1186/1471-2105-10-10.
- Redon, R. *et al.* (2006) Global variation in copy number in the human genome. *Nature*, **444**, 444-454.
- Richardson,S.,Green,P. (1997) On Bayesian analysis of mixtures with and unknown number of components. *Royal Statistical Society.B*, **59**, 731-792.
- Ripley,BD. (1979) Tests of 'randomness' for spatial point patterns. *Journal of the Royal statistics society: Series B*, **41**, 368-374.
- Rue,H. (2001) Fast sampling of Gaussian Markov random fields. *J.R.Statist.Soc.B*, **63**, 57-75.

- Rueda, O. *et al* (2007) Flexible and accurate detection of genomic copy-number changes from aCGH. *PLOS computational biology*, **3**, 1115-1122.
- Shadeo, A. *et al* (2006) Comprehensive copy number profiles of breast cancer cell model genomes. *Breast Cancer Research*, **8**, R9.
- Shaw, C. *et al.* (2004) Non-recurrent 17p11.2 deletions are generated by homologous and non-homologous mechanisms. *Human Genetics*, **116**, 1-7.
- Snijders, A. *et al* (2003) Shaping of tumor and drug-resistant genomes by instability and selection. *Oncogene*, **22**, 4370-4379.
- Stoyan, D. *et al.* (1995) Stochastic geometry and its applications. *Chichester: John Wiley and Sons*.
- Stephen C. Mack. Michael D. Taylor (2009) The genetic and epigenetic basis of ependyoma. *Childs Nerv Syst*, DOI 10.1007/s00381-009-0928-1
- Tibshirani R. *et al.* (2002) Pre-validation and inference in microarrays. *Statistical applications in genetics and molecular biology*, Article I.
- Tipping, M. (2001) Sparse Bayesian learning and the relevance vector machine. *J Mach Learn Res*, **1**, 211-244.
- Toruner, G. *et al.* (2007) An oligonucleotide based array-CGH system for detection of genome wide copy number changes including subtelomeric regions for genetic evaluation of mental retardation. *Am J Med Genet A*, **143**, 824-9.
- van't Veer *et al.* (2002) Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, **415**, 530-536.
- Wang, K. *et al* (2007) PennCNV: an integrated hidden Markov model designed for high-resolution copy number variation detection in whole-genome SNP genotyping data. *Genome Research*, **17**, 1665-1674.

- Wang, J. *et al.* (2009) Genome-wide analysis of copy number variations in normal population identified by SNP arrays. *The Open Biology Journal*, **2**, 54-65.
- Weiss, M *et al* (2003) Determination of amplicon boundaries at 20q13.2 in tissue samples of human gastric adenocarcinomas by high-resolution microarray comparative genomic hybridization. *The journal of Pathology*, **200**, 320-326.
- Willenbrock, H. *et al* (2005) A comparison study: applying segmentation to array CGH data for downstream analysis. *Bioinformatics*, **21**, 4084-4091.
- Winchester, L. *et al* (2009) Comparing CNV detection methods for SNP arrays. *Brief Funct Genomics*, **8**, 353-366.
- Wolf, M. *et al* (2004) High-Resolution Analysis of Gene Copy Number Alterations in Human Prostate Cancer Using CGH on cDNA Microarrays: Impact of Copy Number on Gene Expression. *Neoplasia*, **6**, 240-247.
- Yan, X *et al.* (2007) Discriminant analysis using multigene expression profiles in classification of breast cancer. *Proceedings of the 2007 International Conference on Bioinformatics and Computational Biology* (BIO-COMP'07).
- Yan, X *et al.* (2008) Selecting informative genes for discriminant analysis using multigene expression profiles. *BMC Genomics*, **9**:S14.
- Zhang, B. and Horvath, S (2006) Ridge regression based hybrid genetic algorithms for multi-locus quantitative trait mapping. *Int.J.Bioinformatics Research and Applications*, **1**, 261-272.
- Zhao, X. *et al.* (2004) An integrated view of copy number and allelic alterations in the cancer genome using single nucleotide polymorphism arrays. *Cancer Res*, **64**, 3060-3071.
- Zheng, T *et al.* (2006) Backward genotype-trait associate (BGTA)-based dissection of complex traits in case-control designs. *Human Heredity*, **62**, 196-212.

# Appendix A

## Details of the derivations of the MCMC algorithm

The full details of the derivations for the MCMC algorithm are reported here.

1. Updating  $k$ : According to Richardson and Green (1997), a new component is accepted with probability  $\min\{1, \alpha\}$  where  $\alpha = \frac{p(\theta'|y)r_m(\theta')}{p(\theta|y)r_m(\theta)q(u)} \left| \frac{\partial \theta'}{\partial(\theta, u)} \right|$ . In our model  $\theta = (k, \mu, \sigma^2, \mathbf{x})$ . We have the following distributions

$$\frac{r_m(\theta')}{r_m(\theta)q(u)} = \frac{(1 - b_{k+1})\frac{1}{k+1}}{b_k \frac{1}{k_{max}-k} p(\mu_*) p(\sigma_*^2) p(\mathbf{x}_*)}$$

$$p(\theta'|y) \propto \left( \prod_{i=1}^n \sum_{j=1}^{k+1} \omega'_{ij} N(y_i | \mu'_j, \sigma_j'^2) \right) \left( \prod_{j=1}^{k+1} p(\mu'_j) \right) \left( \prod_{j=1}^{k+1} p(\sigma_j'^2) \right) \left( \prod_{j=1}^{k+1} p(\mathbf{x}'_j) \right) p(k+1)$$

$$p(\theta|y) \propto \left( \prod_{i=1}^n \sum_{j=1}^k \omega_{ij} N(y_i | \mu_j, \sigma_j^2) \right) \left( \prod_{j=1}^k p(\mu_j) \right) \left( \prod_{j=1}^k p(\sigma_j^2) \right) \left( \prod_{j=1}^k p(\mathbf{x}_j) \right) p(k)$$

Since we add a component to the original vector by an identity transformation our Jacobian is equal to 1. We therefore have

$$\alpha = \frac{(1 - b_{k+1})\frac{1}{k+1} p(k+1)}{b_k \frac{1}{k_{max}-k} p(k)} \prod_{i=1}^n \frac{\sum_{j=1}^{k+1} \omega'_{ij} N(y_i | \mu'_j, \sigma_j'^2)}{\sum_{j=1}^k \omega_{ij} N(y_i | \mu_j, \sigma_j^2)}$$

which gives (2.12). Similar derivations hold for (2.13).

2. **Updating  $\mathbf{x}$ :** For each location  $i$ , the full conditional of  $(x_{i1}, \dots, x_{ik})$  is

$$\left\{ \sum_{j=1}^k \omega_{ij} N(y_i | \mu_j, \sigma_j^2) \right\} \prod_{j=1}^k N\left(x_{ij} \mid \frac{h \sum_{i' \sim i} x_{i'j}}{1 + hn_i}, \frac{1}{1 + hn_i}\right)$$

where  $n_i$  is the number of neighbors at location  $i$ . We therefore use a proposal distribution of the type

$$\prod_{j=1}^k N\left(x'_{ij} \mid \frac{h \sum_{i' \sim i} x_{i'j}}{1 + hn_i}, \frac{1}{1 + hn_i}\right).$$

The acceptance probability is

$$\min\left(1, \frac{\sum_{j=1}^k \omega'_{ij} N(y_i | \mu_j, \sigma_j^2)}{\sum_{j=1}^k \omega_{ij} N(y_i | \mu_j, \sigma_j^2)}\right)$$

where  $\omega'$  are the weights associated to the proposed  $\mathbf{x}$ .

3. **Updating  $h$ :** The full conditional for  $h$  is

$$c(h)^k \exp\left(-\frac{h}{2} \sum_{j=1}^k \sum_{i \sim i'} (x_{ij} - x_{i'j})^2\right) I(0 \leq h \leq h_{max}).$$

We use a Metropolis-Hastings random walk with proposal a truncated normal distribution,  $h' \sim TN(h, \sigma_h^2) I(0 \leq h' \leq h_{max})$ . The acceptance probability is given by,

$$\min\left(1, \frac{c(h')^k \exp\{-\frac{h'}{2} \sum_{j=1}^k \sum_{i \sim i'} (x_{ij} - x_{i'j})^2\} \left(\Phi\left(\frac{h_{max}-h}{\sigma_h}\right) - \Phi\left(\frac{-h}{\sigma_h}\right)\right)}{c(h)^k \exp\{-\frac{h}{2} \sum_{j=1}^k \sum_{i \sim i'} (x_{ij} - x_{i'j})^2\} \left(\Phi\left(\frac{h_{max}-h'}{\sigma_h}\right) - \Phi\left(\frac{-h'}{\sigma_h}\right)\right)}\right)$$

4. **Updating  $\mu, \sigma^2$ :** The full conditional for  $(\mu_1, \dots, \mu_k)$  is

$$\prod_{j=1}^k N\left(\frac{\sum_{i: z_i=j} y_i}{n_j}, \frac{\sigma_j^2}{n_j}\right) I(a_{ji} < \mu_j < b_{ji})$$

The full conditional for  $\sigma_j^2$  is

$$\sigma_j^2 \sim \text{Inverse-Gamma}\left(\frac{1}{2}n_j + \alpha_{\sigma^2}, \frac{1}{2} \sum_{i: z_i=j} (y_i - \mu_j)^2 + \beta_{\sigma^2}\right)$$