SPARSE REPRESENTATION MODELS AND APPLICATIONS TO BIOINFORMATICS

by

Roger Pique-Regi

A Dissertation Presented to the
FACULTY OF THE GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(ELECTRICAL ENGINEERING)

August 2009

Copyright 2009 Roger Pique-Regi

Dedication

To my parents Merce and Joan Ramon, and my sister Mariona.

Acknowledgements

I would like to thank my advisor Prof. Antonio Ortega and Dr. Shahab Asgharzadeh for their guidance and support on my research. They have always been available and enthusiastic to discuss my research, open and patient to listen my ideas, and providing suggestions and criticism which greatly improved the quality of my work. My regular meetings with them have been very enjoyable and helped me to improve my communication and research skills. I owe a lot of my knowledge to their experience. Dr. Shahab Asgharzadeh had the patience to teach me the fundamentals of biology and oncology, and Prof. A. Ortega transmitted me his knowledge in signal processing. They have given me the tools necessary to propose the models I used in my research.

I would also like to thank Prof. Bart Kosko for his interest in my research and forming part of my dissertation committee, and Prof. Paul Marjoram and Prof. Keith Jenkins for their participation in my qualifying examination. It is indeed a great privilege to have their valuable comments and feedback on my work. I also want to acknowledge my collaborators in many of my research projects: Prof. D. C. Thomas, Prof. Paul Marjoram, Dr. Corina Sthir, Prof. Kim Siegmund, Lingyan Shen, Jordi Monso-Varona, Dr. Richard Sposto, Dr. Diana Abdueva and J. R. Gonzalez. I also want to thank all my professors that have given me the tools which are the foundation of my research.

Thanks also to all my friends for the happy moments we shared and for cheering me up when I was having a hard time: Jose Ramon, Mahesh, Mona, Yvonne, Jonathan, Julia, Ivona, Vicky, Stavros, Victor, Andreu, Jordi, Selina, Quimi, Anna and Yesim. I am especially grateful to my house mate and friend Cintia for her encouragement and support. I am also indebted to all the people in the signal and image processing group and especially Ivy, Zihong, Polin, InSuk, Sphinx, Carlos, JaeHoon, Hye-Yeon, Ngai Man, Hiusheng, Talya and Gloria. I also feel sorry for not having been able to be closer to my old friends Carlos, Joan, Juan, Jordi and Miguel in Barcelona.

Finally, I would like to express my deepest gratitude to my parents Joan Ramon and Merce and my sister Mariona for their unconditional love. You have always supported me to pursue my dreams and you encouraged me during my hardest times. I also feel indebted to the rest of my family and very especially to my grandparents Dolores, Senen and Rosita. It is always a regret not have been able to spend much time with you during these years. In the course of completing my studies I met Nadia who filled my heart with happiness. I cannot imagine how lonely I would feel without you. I am starting a new journey with you and I am very excited to pursue new dreams together.

Table of Contents

List Of Tables

List Of Figures

Abstract

Microarrays and new sequencing techniques offer a high throughput platform to study the whole genome with the unprecedented capability of measuring millions of genomic features on a single essay. This massive parallel measurement power has an enormous potential for research in Biology and Medicine with the ultimate objective of identifying and learning the biological processes occurring in different organisms and diseases. Existing model learning methods are severely limited by the reduced number of samples that are usually available compared to the measurements.

We propose that sparse signal representations can model these biological signals and we develop the analytical tools to accurately extract the relevant information. We exploit the underlying sparseness of the biological model to overcome some of the problems associated with analyzing these massive datasets. This work demonstrates the potential benefits of this approach by studying three different problems involving microarray data.

The first problem concerns the supervised design with a limited amount of training samples of a classification procedure to predict tumor progression. We propose a greedy search strategy to select a sparse feature subset with a block diagonal covariance matrix structure to build a linear discriminant analysis model for tumor prognosis. The second problem deals with the detection of copy number alterations. We develop a maximally

sparse representation for these copy number alterations, and a sparse Bayesian learning approach is optimized to detect these alterations from noisy microarray observations. The third problem involves finding genetic determinants of gene expression. In this case, we propose a linear regression model with a sparse Bayesian prior on the large matrix of the regression coefficients relating genome alterations to transcription levels.

Chapter 1

Introduction

1.1 Significance and scope of the research

In recent years we have witnessed a tremendous advance in technologies to extract biological data from living organisms. Automated DNA sequencing has made it possible to obtain a reference sequence for the human genome (Human Genome Project - HGP [16]) and many other organisms. From these DNA sequences 3 billions of base pairs (the AGCT genetic code) one can identify which portions are genes that are transcribed to RNA to produce a protein. It is estimated that there exist about 25.000 human genes, and these are used as a blueprint to build about 100.000 different proteins which are responsible for running the biological functions on human cells. Although 99.9% of the genome is identical among all humans, small differences in the form of Single Nucleotide Polymorphisms (SNP), inversions, and copy number alterations (CNA, e.g., deletions and duplications) give rise to the rich variability between individuals. New technologies such as microarrays provide the means to measure gene expression activity (RNA arrays) and genomic alterations (SNP and aCGH arrays) with millions of probes along

the genome [81]. Other techniques have been developed to study protein levels, protein structure, DNA methylation and many other biological processes occurring in the cell and living organisms.

These advances have contributed to the discovery of key new findings in Biology; new genes, new functionalities, alternative gene splicing, gene silencing, copy number polymorphisms and many other. These discoveries have also tremendously affected other related fields. In medicine [15,35], they have lead to new molecular diagnostic procedures, unveiled the underlying biological processes of some diseases, and guided the development of new drugs. Despite technical advances, severe noise degradation of the measurements due to cross-hybridization and other biological effects poses a challenge when trying to extract reliable information from these large sets of data.

Accurate and computationally efficient methods are essential for detecting genetic differences, genomic alterations, and identifying which genes are active or corregulated. The interdisciplinary field of *bioinformatics* has been growing quickly and has led to the development of increasingly efficient and reliable tools for the analysis of very large biological datasets. The key features of these datasets are i) *small* n number of samples ($n \sim$ hundreds), ii) *large* p number of measurements ($p \sim$ millions), and iii) *sparseness* of the underlying biological models. For example, a microarray can interrogate millions of positions along the genome with dozens of probes for all the known 25.000 genes of the human genome but only a very reduced set (about 10 to 100) will be active at a given time on some tissue. Gene activity is also regulated through networks with a sparse number of interactions. The alterations along the genome are also a rare event and only a very small subset have a role in in explaining differences in gene activity. In conclusion,

though the number of measurements is very large, the number of underlying variables that are involved in regulating the biological process is however quite small.

This research is one of the first to apply recent advances in sparse signal representations and machine learning to develop new bioinformatics tools. Sparse signal representations seek to minimize the number of elements chosen form a dictionary of signals necessary to approximate (e.g., as a linear combination) special cases of signals (e.g., smooth, piecewise constant, splines, etc.). Large efforts have been devoted to designing the most adequate bases (e.g., wavelets) or overcomplete collections of bases to build sparse representations for different classes of signals (e.g., images, speech and audio, econometric, biomedical) [19, 50, 62, 102]. Fast and accurate optimization algorithms for finding the optimal sparse representations have been proposed (e.g., matching pursuit [61, 70], basis pursuit [13] and sparse Bayesian learning [97, 104]). The theoretical properties of these methods are still under study but important results have been obtained [20, 98] for applications such as signal denoising [50] and compressive sensing [12, 48, 99].

New classifying techniques have been developed such as support vector machines and regularized linear discriminants with similar optimization algorithms to search the sparsest models (least number of features) with highest prediction accuracy. Examples of my research include developing *sparse* representations for detecting genome alterations and models for classifying cancer tumors. These methods exploit the *sparseness* of the underlying biological models to overcome the problem of extracting meaningful and reliable predictions from a dataset with a very large number of measurements compared to the number of observed samples, i.e., $p >> n$.

1.2 The Human Genome

This section provides a very short introduction to the human genome, and is not intended to provide a detailed description. The objective of the next two sections is to provide a very basic understanding of the biological processes and experimental techniques that are covered in this dissertation. A more detailed and precise description of the underlying biology can be found in molecular biology textbooks [2].

All the (somatic) cells of the body contain a full set of chromosomes, with identical genes, i.e., exactly the same DNA sequence that is known as *genome*. On 2003, the Human Genome Project completed the reference DNA sequence for the human genome, which can be browsed along with annotation information from many different sources like Ensembl [45] [1], the University of California Santa Cruz (UCSC) browser [52] [2], and the National Center for Biotechnology Information (NCBI)[3]. With the completed sequence, now the efforts are centered in finding and analyzing *polymorphisms*, i.e. common alterations of the reference sequence; and finding and studying *gene expression*, the pieces of the genome that carry out basic functions in the organism.

A *gene* is a section of the genome that encodes the information necessary to produce functional products, *proteins*, to accomplish some function in the cell or the organism. A gene is said to be activated if it is being read and generates the protein that it encodes in a process illustrated in Figure 1.1. On any given cell and any given time, only small subset of all the $\sim 25,000$ genes that compose the genome are active, i.e., they are being "expressed". The expression levels of the gene also change dynamically and are

[1] http://www.ensembl.org
[2] http://genome.ucsc.edu/
[3] http://www.ncbi.nlm.nih.gov/

regulated by a complex network of pathways in which other genes produce the RNA and proteins (transcription factors) that regulate other genes. The proteins and RNA can also interact with each other assembling into proteins complexes, reshaping their structure, and silencing the mRNA (miRNA and siRNA). Gene expression microarrays provide the technological means to measure transcription levels of hundreds of thousands of DNA fragments that are expressed.

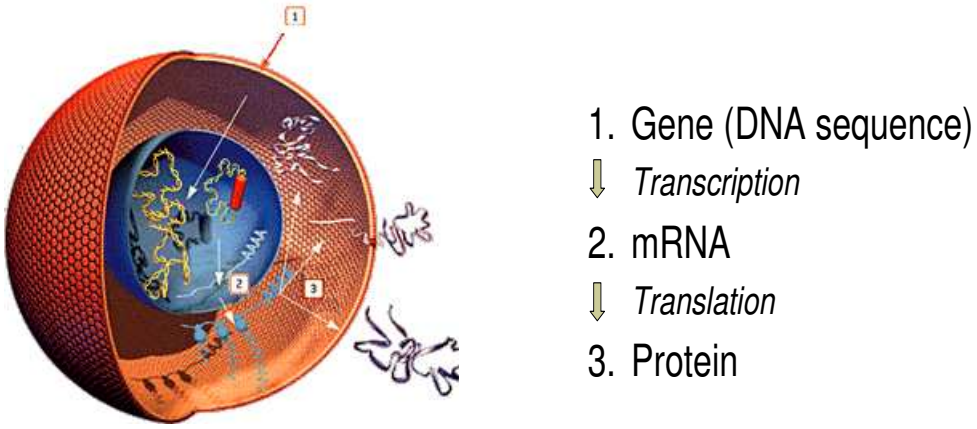


Figure 1.1: Gene expression in a cell (Permission for use: (c) Transgene S.A.). The DNA sequence of the gene (1) is copied into RNA sequences in process called *transcription*. Then this sequences are spliced and assembled into mRNA (2). This mRNA serves as a blueprint to build a protein (3) in a process called *translation*

The DNA sequence of the genome of any two given individuals is 99.9% identical. Differences in the sequences, are called *mutations*, and may or may not affect some of the many observable different traits that distinguish two persons. The most frequent of these sequence alterations are called *polymorphisms*; i.e., common variations of the reference sequence that are also found in a large amount of other individuals (see Figure 1.2 for an example). Each polymorphism is located in a particular position of the genome, *loci*, and the possible sequence variants are called *alleles*. Since for most of our genome we have

two copies of the DNA, the combination of the alleles of each copy define the *genotype*. The biological process of *meiosis* directs how the DNA material is replicated and the *alleles* transmitted to the offspring. Alleles that are close together in a chromosome are transmitted as a block, *haplotype*, and those that are in different chromosomes (or far away in the same chromosome) are independently transmitted. This haplotype structure is a consequence of *recombination* events during the meiosis in which the parental DNA copies are crossed over. Since genotypes of proximal loci are linked together, we can locate the position for rare events (e.g., disease traits) by association to genotypes of markers for which the positions are known. Genotyping microarrays are able to genotype thousands (now millions) of Single Nucleotide Polymorphisms (SNPs) markers scattered along the genome. These experiments can be used to locate and characterize this human genetic variation.

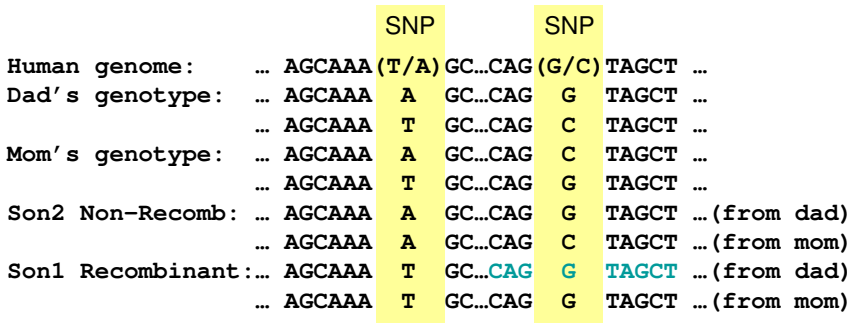


Figure 1.2: SNP allele transmission and recombination. A Single Nucleotide Polymorphism (SNP) is a type of genomic alteration that only affects one nucleotide of the DNA sequence and has only two possible states, i.e. alleles. Each descendant gets one copy of the autosomal genome of their parents and the combined alleles defines his genotype. Alleles of proximal SNP are linked together unless there is a recombination event.

1.3 Microarray technology

Microarrays are a new type of biological essays with very high throughput capabilities, i.e., the ability to perform a very large number of measurements in each experiment. This technology exploits the ability of RNA and DNA to bind specifically to, or hybridize to, the complementary DNA template from which it originated. The basic design consist of a solid support onto which relatively short DNA sequences (probes) from thousands of different sections of the genome are immobilized at fixed locations, see Figure 1.3. The DNA or RNA is extracted from the sample to study, fragmented, and fluorescently tagged; then, these tagged fragments preferably bind to target probes with the exact complementary DNA; and finally, the array is washed of unattached fragments and the fluorescent intensity of each probe is measured.

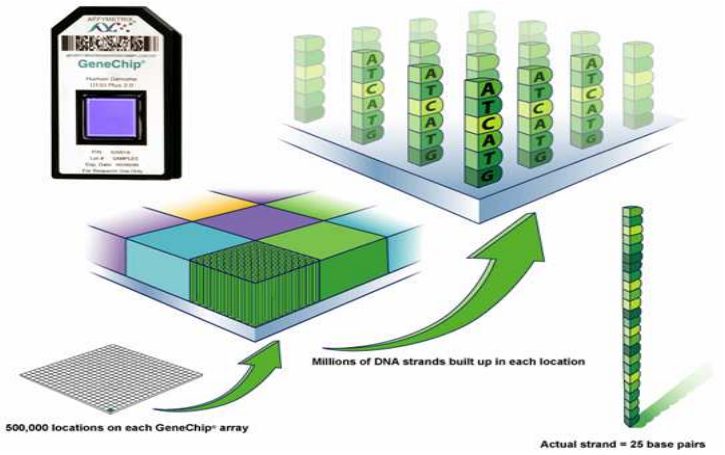


Figure 1.3: Affymetrix microarray design. (Image courtesy of Affymetrix. Permission for use: (c) Affymetrix 2007)

Different types of microarrays have been developed to measure genome features such as gene expression, genotypes, or copy number (see Table 1.1. The basic approach is the

same for all of them; the only change comes from the target DNA sequences that are on the chip and the type of genetic material that is extracted.

A gene expression array essay is depicted in Figure 1.4. In this case, the material extracted from the cell to study is RNA, and the microarray contains probes targeting RNA transcripts. The hybridization intensity of each probe gives a measure of the gene transcription level.

In SNP genotyping arrays, the DNA is the material extracted form the cells, then cut in known places by restriction enzymes, and fragmented into small pieces that contain only one SNP. The array now contains probes with the complementary DNA sequences targeting the two possible SNP alleles (see SNP in Figure 1.2). The hybridization levels associated with each allele can be used to infer the genotype or the copy number.

Table 1.1: Different types of microarrays

Type	Measurement	Application
SNP	Hybridization intensities of fragmented DNA to the two different SNP allele variants	Genotyping. Association studies. Drug response. Genome variation. Copy number alterations
aCGH	Hybridization intensities to large sections of DNA	Copy number alterations
Expression (Exon arrays)	Hybridization of fluorescently tagged RNA transcripts to its complementary coding DNA fragments	Gene expression. Gene regulation and function. Molecular profiling. (Alternative Gene Splicing)

1.3.1 Issues on analyzing microarray data

The high throughput capabilities of these microarrays have an enormous potential for research in Biology and Medicine. For example, microarray gene expression studies have been performed to find which genes are active (being transcribed) on different types of

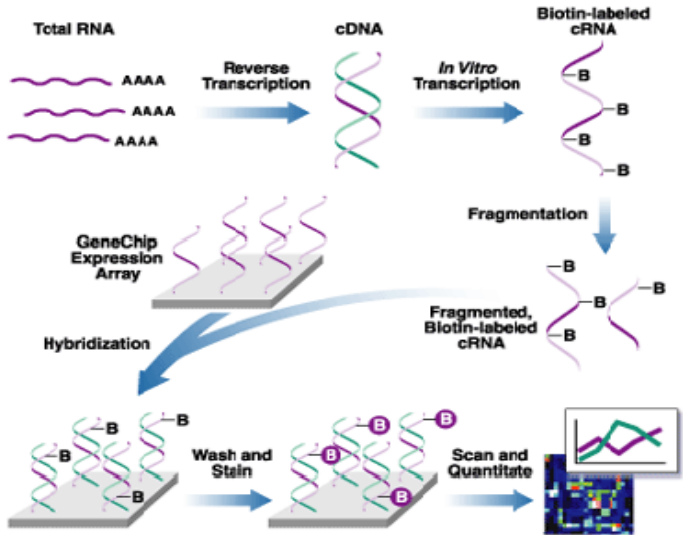


Figure 1.4: Affymetrix gene expression microarray essay. The microarray essay can be divided in three steps: a) sample preparation, b) hybridization, c) washing and scanning. In the first step a): the mRNA is extracted from the cells to study, converted to cDNA (reverse transcription), the cDNA is cut into small fragments of ~ 26 bases length, and this fragments are labeled with fluorescent tags. Then, in the hybridization step b), the fragments in the previous preparation specifically bind to the complementary short segments of DNA that are specially selected and immobilized on specific locations in the array. Finally in c), the array is washed and scanned giving a fluorescent intensity reading. Locations corresponding to *expressed* genes should have higher intensity that those corresponding to *non*-expressed genes. (Image courtesy of Affymetrix. Permission for use: (c) Affymetrix 2007)

tissues, different cell conditions, or in response to an administered drug or treatment. Other applications in medicine include development of new diagnosis and risk assessment procedures. Similarly, genotyping arrays have been used to define groups of high risk, to find a causal genetic locus for a disease, and to study differences in treatment response. The large volume of the data generated by microarrays pose new issues that require the development of new data analysis methods that are computationally efficient and statistically reliable. These issues are essentially three: i) normalization, ii) large number of variables, iii) small number of samples.

Normalization. The measurements obtained from microarrays are not perfect observations. Sample extraction and hybridization processes are affected by a large number of biological factors. Some of these factors can be modeled and corrected by appropriate *normalization* procedures. However, there may always be unknown effects like cross-hybridization, RNA degradation, or other sources of experimental error that are out of our control. In any case, it is important to define normalization procedures, that make the microarray measures comparable across different samples, and transform the data such that we can assume an appropriate distribution safely (e.g. Gaussian).

Large number of variables. Continuous improvements in array technology and sequencing are increasing the number of measurements, which are currently in the order of millions. Gene expression arrays and new exon arrays contain hundreds of thousands of probes targeting all the genes and their exon transcripts. The newest genotyping arrays cover nearly a million SNPs along the genome. However, it is expected that only a very small (*sparse*) subset of these variables will typically be related to the research questions

that we are trying to answer with the microarray experiment (e.g., those given as example in the beginning of the section). Thus, efficient as well as reliable statistical methods are required for screening these large sets variables to build models that are likely to be biologically meaningful.

Small number of samples. In contrast to the number of variables, the number of samples is typically very small ~ 100. This requires that the statistical procedures used for classification evaluation and for testing genes for association have to be very powerful, i.e. efficient in the number of samples. In classification, splitting the available samples in independent sets for model fitting, selection, and evaluation is very inefficient; and computationally intensive algorithms like cross-validation and bootstrapping approaches have to be used. Additionally, permutation based tests are used to assess the proportion of the genes that could be falsely deemed significant (i.e., the False Discovery Rate FDR) just by chance due to the large amount of noisy predictive variables that are being evaluated. In conclusion, the large number of variables together with the reduced amount of samples make the problem of robustly estimating the underlying biological models very challenging.

In order to overcome these challenges, prior knowledge about these biological models should be exploited. There exists a large number of databases that gather relevant information for a large number of genes, gene regulation properties, pathways, haplotype structures, and other potentially useful information that is continuously updated as ongoing research progresses. Currently, methods for combining these different sources of information are very limited, and only very simple models are being used due to the limited amount of samples that are available to fit the model. However, in the foreseeable

future it is very likely that more complex algorithms will exploit all this prior knowledge that is being gathered.

1.4 Research contributions

The research contributions of this work have been grouped into three major parts each one dealing with a different problem related to microarray analysis. The nexus among the three of them is that in all three cases linear models with sparseness constraints are adopted. In other words, the solution to the problem consist of finding a *sparse* linear combination that depends on only a small subset of all microarray probes. The adoption of sparse linear models is biologically supported by the underlying assumption that any basic cellular process is controlled by a very small portion of the genome.

In Chapter 2 we study the design and evaluation of molecular classifiers that are based on linear discriminant analysis (LDA) of gene expression microarrays. Different options to select the genes and to place *sparseness* constraints on LDA are studied. We start reviewing DLDA, which is a widely used method in which correlations are completely ignored (i.e., assumed to be 0), and we discuss the application of DLDA to analyze gene expression profiles for the prognosis of tumor progression in neuroblastomas [6]. Then, we consider a new gene selection algorithm SeqDLDA [78] that under the DLDA model selects the genes that are better modeled by the non-correlation assumption. Afterwards, SeqBDLDA [77] proposes an embedded approach in which the genes and a block diagonal covariance structure are jointly selected to fit a linear discriminant model. LDA and the developed feature selection procedures provide a flexible framework for microarray gene

expression analysis, in which models with different degrees of complexity can be adopted depending on the amount of available training samples or prior knowledge.

In Chapters 3, 4 and 5 we tackle the problem of detecting genome copy number alterations (CNAs) using microarray data [75]. Studying copy number alterations is important to understand the role of natural copy number variations in human genomes. It is also essential in understanding cancer cells, where genomic instability leads to large abnormalities in genome copy numbers. The hybridization intensities from SNP probes in genotyping arrays, or from specially chosen probes in aCGH, are correlated with the underlying number of DNA copies of their corresponding genome regions but are severely degraded by noise. In our approach, we model the genome copy number as a piece-wise constant (PWC) vector for which a sparse representation is formulated. Then, sparse Bayesian learning (SBL) is optimized for the proposed PWC representation and used to detect CNA breakpoints. Moreover, a backward elimination (BE) procedure is used to rank the inferred breakpoints; where a cut-off point can be adjusted to control the false discovery rate (FDR). The performance of our algorithm is evaluated using both simulated and real genome datasets and compared to other existing techniques. Our approach achieves the highest accuracy and lowest FDR, as compared to the state of the art, while improving computational speed by several orders of magnitude.

In Chapter 6, we describe a novel hierarchical Bayesian model for the influence of constitutional genotypes from a linkage scan on the expression of a large number of genes. This work can be considered one of the initial steps to find genetic determinants of gene expression at a genome-wide scale. The proposed model comprises linear regression models for the means in relation to genotypes and for the covariances between pairs

of related individuals in relation to their identity by descent estimates. The matrices of regression coefficients for all possible pairs of SNPs by all possible expressed genes are sparse, and modeled as a mixture of null values and a normal distribution of non-null values. The approach appears to be a promising way to address the huge multiple comparisons problem for relating genome-wide genotype-by-expression data.

Chapter 2

Sparse Linear Discriminant Analysis of gene expression microarrays

2.1 Introduction

Gene expression microarrays can measure the expression values of thousands of genes in the same experiment. A large number of fundamental biological and medical research questions involve identifying which genes and mechanisms are responsible in determining, for example, tumor progression, blood pressure or drug response. In a supervised learning approach we are given a training group of samples for which the outcome of the variable of interest is known. Thus, these examples, can be used to fit a discriminant function to predict the outcome. In microarray experiments the challenge is that the number of samples ($n \sim 100$) is very small compared to the number of features / probes ($p > 10000$). These discriminant methods are required to (i) *generalize well*, i.e., have a low prediction error for future samples; (ii) be *sparse*, i.e. be based on a small subset of features; and (iii) have low *False Discovery Rate*, i.e. have a large proportion of true relevant features that can also be confirmed by other studies.

Linear discriminant analysis (LDA) [22, 40] is a well known supervised classification technique in which the discriminant functions are linear combinations of the features for which we obtain a good separation between classes. If we consider a classification problem with two classes, the degree of separation can be measured by taking the squared distance between the centroids divided by the variance. In the context of microarrays, we are faced with a very large number of features, and very few training samples. Thus, the sample covariance matrix is singular and does not give a reliable estimate of the true covariance matrix to be used to fit the linear discriminant [105]. Two of the most popular techniques in microarray classification, diagonal linear discriminant analysis (DLDA) [23] and nearest shrunken centroids (NSC) [96], are based on LDA and solve this problem by imposing a diagonal structure to the covariance matrix and using only a small subset of the topmost discriminative features to build the classifier.

In this chapter we review DLDA and we apply the method to build a prediction model for the prognosis of neuroblastoma tumor progression [1] that incorporates a novel selection and evaluation method for DLDA based on nested cross-validation. This method is useful for deciding which genes to include in the DLDA model, and for accurately estimating the prediction error for future samples. The proposed model selection and evaluation methods have been positively received and have been suggested as useful tools for medical studies [92].

Afterwards, we introduce novel alternative strategies for constructing the linear discriminant function. First, SeqDLDA [2] in Section 2.3 keeps the DLDA model but proposes

[1] The application and evaluation of the DLDA model has been published as a part of a medical study for neuroblastoma in [6]

[2] The SeqDLDA model was proposed in [78]

a *wrapper* feature subset selection (FSS) approach that does not ignore correlations. In SeqDLDA, one gene is sequentially added and the linear discriminant recomputed using the DLDA model (i.e., a diagonal covariance matrix). Classical DLDA instead adds the gene with highest t-test score without checking the resulting model. In contrast, SeqDLDA will find the one gene that better improves class separation after recomputing the model.

SeqBDLDA [3] in Section 2.4 extends the DLDA model to consider a block diagonally structured covariance matrix. In this case we adopt a novel *embedded* FSS approach, in which each feature is added sequentially in the model either as an independent block or inside one of the previously existing blocks of the covariance matrix. At each step, the best feature and block model are decided by measuring the class separation after computing the resulting linear discriminant of that feature set and its corresponding block diagonal covariance matrix. This is the first time that such a joint design of the model with the feature selection has been proposed in the context of LDA. In order to reduce the complexity of exploring a large number of BDLDA models, an optimized repeated FSS (RFSS) search strategy[4] is also proposed.

These new contributions show considerable improvement in prediction accuracy both in simulated and real datasets, especially in the cases where more training samples are available (Section 2.5). Additionally, the more complex block diagonal modeling could become even more promising in the future since it could be used to exploit prior knowledge on gene corregulation; i.e., the block structure could be estimated from previous experiments or genome databases.

[3]The SeqBDLDA model was proposed in [78]
[4]This work was in collaboration with Lingyan Shen [91]

2.2 Background

2.2.1 Linear Discriminant Analysis

In statistical pattern recognition problems, Bayes decision techniques provide optimal classification performance as long as the distribution of the samples is known [22]. In many practical cases, these distributions are not known and they must be learned from training data. In the context of microarrays simple linear models are preferred because of the limited number of samples available, relative to the number of features.

Linear discriminant analysis (LDA) is a widely used technique for sample classification [22, 40]. For two classes, LDA is defined by a linear discriminant function $g(\mathbf{x})$:

$$g(\mathbf{x}) = \mathbf{w}^t\mathbf{x} - b \begin{cases} > 0 \Rightarrow \text{Class A} \\ < 0 \Rightarrow \text{Class B} \end{cases} \tag{2.1}$$

where $\mathbf{x}$ is the vector containing the gene expression of the sample to classify. $\mathbf{w}$ is a vector of weights orthogonal to the hyperplane that together with the scalar b define the decision boundary $g(\mathbf{x}) = 0$ that discriminates between the two classes to separate. If the samples are normally distributed with known means and variances: $f_A(\mathbf{x}) \sim \mathcal{N}(\mathbf{m}_A, \mathbf{K})$, $f_B(\mathbf{x}) \sim \mathcal{N}(\mathbf{m}_B, \mathbf{K})$; the optimal Bayes maximum a posteriori classifier is given by the LDA discriminant function with:

$$\mathbf{w} = \mathbf{K}^{-1}\mathbf{d} \qquad \mathbf{d} = \mathbf{m}_A - \mathbf{m}_B \tag{2.2}$$

$$b = \ln\left(\frac{\pi_A}{\pi_B}\right) - \mathbf{w}^t\frac{(\mathbf{m}_A + \mathbf{m}_B)}{2} \tag{2.3}$$

where π_A and π_B are the prior probabilities of each class; and $\mathbf{d}$ is the vector that links the two class centroids.

In a more general sense this solution is also reasonable (but not necessarily optimal) if distributions are symmetrically bell-shaped (like a normal) because it gives the direction in which the variance between/within classes ($J_{\mathbf{K}}(\mathbf{w})$) is maximized; i.e., when

$$J_{\mathbf{K}}(\mathbf{w}) = \frac{\left(\mathbf{d}^t\mathbf{w}\right)^2}{\mathbf{w}^t\mathbf{K}\mathbf{w}} \tag{2.4}$$

$$\mathbf{w} = \arg\max_{\mathbf{w}} J_{\mathbf{K}}(\mathbf{w}) = \mathbf{K}^{-1}\mathbf{d} \tag{2.5}$$

In practical applications, the mean vectors and the covariance matrix are not known and have to be estimated from training data, typically using maximum likelihood (ML) estimators. However, choosing the ML estimators it is only asymptotically optimal [30], i.e. when the number of training samples grows so $\hat{\mathbf{w}} \rightarrow \mathbf{w}$.

Since in our case the number of features p is larger than n this convergence will never happen. Indeed, in this scenario the regular ML estimates are very misleading, because i) the estimates are unreliable, and ii) the sample covariance matrix is singular. A $p \times p$ sample covariance matrix $\hat{\mathbf{K}}$ has rank at most $n-2$. Thus, the null space of this matrix has dimension at least $p-n+2$ giving the false impression that the natural variation is 0 in this subspace. Then, any spurious difference on the estimated class vector means on this subspace will be falsely regarded as very significant.

When n and p are comparable, different authors [30, 34, 39, 41] have proposed a regularized solution for the problem by assuming some structure in the covariance matrix (e.g., a diagonally dominant covariance matrix). This has the advantage of reducing

the effective number of parameters that need to be estimated and the risk of obtaining a sample covariance matrix that is singular. However, when n is much smaller than p regularization alone is not enough to achieve reliable classification and it is necessary to further simplify the model by discarding features, i.e., by selecting a reduced feature set. Feature selection is in fact almost always needed in the context of microarray genomic classification, where p is in the order of tens of thousands of genes while n corresponds to a few hundred tissue samples. Taking cancer as an example, it is typically expected that only a few genes will be associated with the disease. Thus, feature (i.e., gene) selection serves the dual purpose of i) reducing the effect of a small training set on classification performance, and ii) identifying substantial genes that are more likely to be associated with the disease.

2.2.2 Feature subset selection (FSS) approaches

There are three major approaches to classifier design and feature subset selection (FSS) [36]; namely, (i) *filter*, (ii) *wrapper*, and (iii) *embedded*. In *filter* approaches, features are first ranked using a statistical score, such as a t-test. Then the classifier is built by selecting the highest ranking features. This is the most popular method in microarray classification problems, due primarily to its simplicity (see Section 2.2.3). Note, however, that it completely ignores interactions among genes.

In *wrapper* approaches [53] a classifier is constructed with different candidate feature subsets, the performance is measured (using, for example, cross validation), and finally the feature subset that achieves the maximum performance is chosen. This is a combinatorial optimization problem and a full search would be very complex, requiring

2^p different evaluations, and prone to overfitting. For this reason, only greedy search strategies using different heuristics are feasible. In the context of microarrays and LDA, wrapper approaches have been proposed using full [105] or diagonal [10, 78] covariance matrices and different search strategies (see Section 2.3).

Finally, *embedded* approaches [58] consider jointly the classifier design and the FSS. This is in contrast to the wrapper approaches that consider the classifier as a black box that induces a prediction rule once the feature subset is chosen. Guyon et al. proposed an embedded approach [37] for Support Vector Machines. To the best of our knowledge, in the context of LDA we have been the first to propose an embedded design approach [77], see Section 2.4.

2.2.3 Diagonal Linear Discriminant Analysis (DLDA)

The diagonal linear discriminant analysis (DLDA) model is simply the LDA model with the covariance matrix constrained to be diagonal and a filter FSS approach. This approach, formally introduced in [23], is only optimal when the features are uncorrelated multivariate normal. However, with limited training data the DLDA models are more reliably trained and may achieve a better prediction accuracy than using the LDA with an unreliable full covariance matrix.

Under the diagonal assumption each variable in the DLDA model can be treated independently since in (2.2) we have that $w_i = d_i/(\sigma_i^2)$ where $\sigma_i^2 = K_{i,i}$. Thus, the linear discriminant (2.1) can be rewritten as linear combination of univariate classifiers:

$$g(\mathbf{x}^*) = \log\left(\frac{\pi_A}{\pi_B}\right) + \sum_{i=0}^{p} \underbrace{\left(\frac{\hat{m}_i^A - \hat{m}_i^B}{\hat{\sigma}_i}\right)}_{\hat{h}_i \text{ weight}} \underbrace{\left(\frac{x_i^* - \hat{m}_i}{\hat{\sigma}_i}\right)}_{\hat{v}_i \text{ vote}} \begin{cases} \geq 0 \Rightarrow \mathbf{x}^* \in \text{ClassA} \\ < 0 \Rightarrow \mathbf{x}^* \in \text{ClassB} \end{cases} \tag{2.6}$$

where the right term represents a vote,$\hat{v}_i$, a single feature discriminant that scores how likely it is that a new sample belongs to class A rather than class B; and, the left term is a weight $\hat{h}_i$ that scores how good a feature i is in discriminating two classes. It is easy to see that $\hat{h}_i$ is proportional to the t-statistic for the difference of two population means. This is illustrated by Figure 2.1 for a two dimensional example.

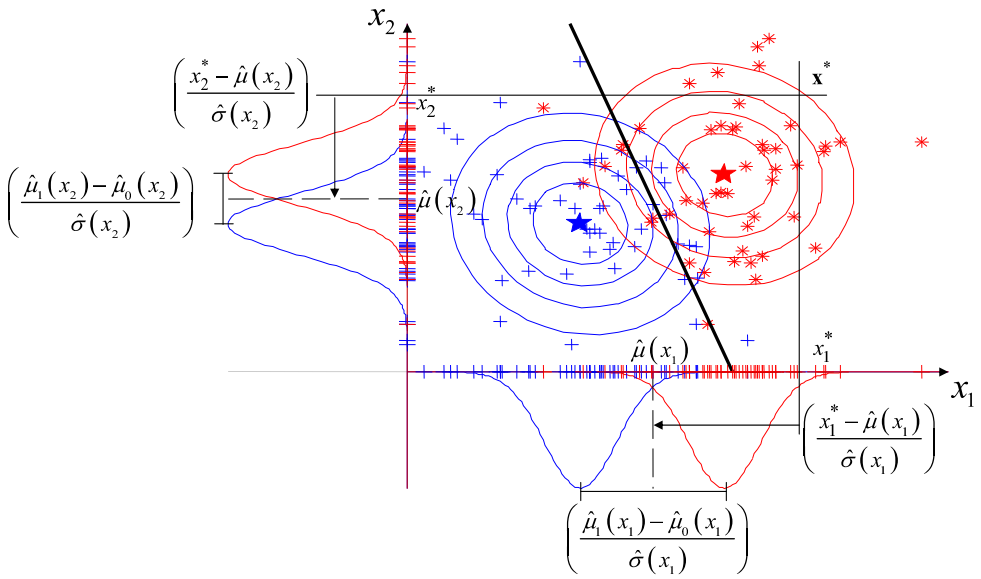
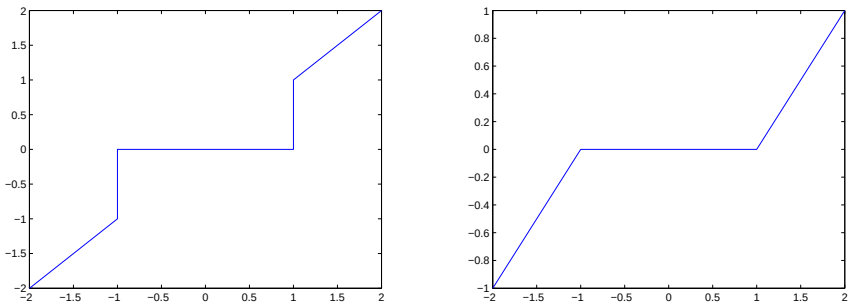


Figure 2.1: Graphical interpretation of the DLDA model (2.6) for two features. Each axis represents a feature with the corresponding univariate distributions. Feature x_1 has a larger separation between class centroids compared to the underlying noise than feature x_2. In DLDA we can classify a new sample $\boldsymbol{x}^*$ by a simple linear combination of single feature discriminants $(x*_1, x*_2, \ldots)$ weighed by the appropriate weights (2.6).

In this diagonal model, most of the genes will probably be irrelevant for the classification. Low scoring genes, can be seen as adding noise to (2.6) so it makes sense to remove the terms with lowest $\hat{h}_i$. This can be seen as a filter approach for feature selection (Section 2.2.2), since genes are first ranked using the statistical score, and then the discriminant function is built by selecting the highest ranking genes. The size of the model, i.e. the number of genes, is usually determined by cross-validation, see Section 2.2.4.

The nearest shrunken centroid (NSC) [96] and the weighed covariate (WC) approaches in [33] are tightly related to this DLDA procedure [23]. DLDA [23] and WC [33] can be seen to remove features by hard-thresholding $\hat{h}_i$ (Figure 2.2 a). In NSC instead, the $\hat{h}_i$ scores are continuously shrunk to 0 before they are completely removed, i.e., soft-thresholding (Figure 2.2 b). In NSC [96] the scores $\hat{h}_i$ are also made more robust; a dampening term is added to the standard deviation, $\hat{\sigma}_i' = \hat{\sigma}_i + \text{median}(\hat{\sigma}_i)$, to protect against very small $\hat{\sigma}_i$ occurring by chance. In our experience, hard-thresholding usually gives a better prediction with smaller feature sets.



(a) (b)

Figure 2.2: Hard (a) and Soft (b) thresholding functions. Both methods set small values ($< \tau = 1$) to zero, but (b) shrinks all the values by τ while (a) leaves the values above the threshold ($> \tau$) untouched.

Finally, the size of the model (i.e. the number of top scoring genes to add in the DLDA or NSC model) is specified as the one that maximizes the classifier prediction accuracy, which can be estimated by cross-validation (Section 2.2.4).

2.2.4 Model selection and evaluation with cross-validation

Before introducing the new extensions to LDA in the following sections, this background section will conclude by reviewing the cross-validation techniques that are used to evaluate the classification models and to guide their construction. The objective of the prediction accuracy evaluation is to assess the performance for future new samples not included in the training set. The prediction error computed on the training set, the training error, is a rather optimistic estimate of the generalization error [40]. For this reason the classifier has to be evaluated using an independent set of samples known as the test set.

In microarray classification problems the data available is very limited and we cannot afford to reserve a large number of samples solely for testing. In these situations, cross-validation (CV) provides an efficient method of iteratively splitting the available samples between a training set and an independent testing set. For example, in 10-fold CV, the training samples are randomly partitioned in 10 segments (balanced to preserve the class proportions), then for each of the 10 segments we train the model with the other nine and use the one reserved for testing for evaluating the performance. The advantage of using cross-validation over reserving a large fraction of samples for an independent testing set is that larger training sets can generate more reliable and more accurate models.

In this chapter we have adopted a 100X10-fold CV (100 repetitions of 10-fold CV) to select the size of the LDA model; i.e. the number of genes P in DLDA and SeqDLDA, and

the covariance structure in SeqBDLDA. The model is chosen as the one that minimizes the average cross-validation error. Some authors point out that to report the minimizing cross-validation error as the future error rate for new samples could introduce a bias again, the selection bias, and that a second external round of CV is necessary [5]. To further ensure that no selection bias is introduced, we use a nested cross-validation procedure as suggested in [5]. First, an internal 100X10-fold CV strategy is used to select the LDA model. Then, an external leave-one-out CV (LOOCV) is used to give an unbiased estimate of the model performance. The evaluation, model/gene selection, and training are performed as described in Algorithm 1

Algorithm 1 Nested Cross-validation

(Leave-one-out external cross-validation loop)
for sample $i = 1 \dots N$ **do**
  Leave sample i for external test set
  Use remaining $N - 1$ samples to form the external training set
  (Repetitions of internal cross-validation loop)
  for Repetition $r = 1 \dots R$ **do**
    External training set is randomly partitioned in N segments
    (Internal 10-fold cross-validation loop)
    for Segment $j = 1 \dots N$ **do**
      Leave segment j out for the internal test set
      Use remaining $N - 1$ segments for the internal training set
      for LDA model $m = 1 \dots M$ **do**
        Find features to include and fit the model using only internal training set
        Test the model on the internal test set and count the errors
      end for
    end for
  end for
  Find the model m^* that minimizes the average internal cross-validation error.
  Find the features and fit the model m^* using the complete external training set.
  Test the final external model on the left out sample i.
end for
Report the external error rate as the expected error rate for future samples.

In summary, the nested cross-validation approach provides an efficient reuse of samples to perform three tasks that would otherwise require three independent sets of samples: i) a training set to estimate the models, ii) a validation set to chose the best model, and iii) a testing set to evaluate the final performance of the validated model. If the number of samples is very limited, which is the case of most microarray studies, cross-validation has the advantage that more samples can be dedicated to build the model. In the nested cross validation approach, the selection bias is avoided by never using the samples reserved for testing in the inner loops. In the context of microarray studies, we are the first to employ this nested cross-validation strategy together with LDA based models that we discuss in this chapter. In a tumor risk prognosis study [6] using the DLDA model (see results in Section 2.5.1) this evaluation approach has been encouraged and received positive reviews [92].

2.3 SeqDLDA – Sequential Diagonal Linear Discriminant Analysis

In the background section (Section 2.2) we mentioned that for microarray applications we are usually forced to make assumptions about the covariance matrix structure because there is usually not enough data to accurately estimate all pair-wise correlations. In DLDA model (Section 2.2.3), if gene correlations are not zero (which can happen often), selecting all the top-scoring features ignoring their correlations is not the best strategy. As illustrated intuitively in Figure 2.3 we could select the features whose correlations are

better suited for the DLDA model. This is the idea behind sequential DLDA (SeqDLDA) described in this section [5].

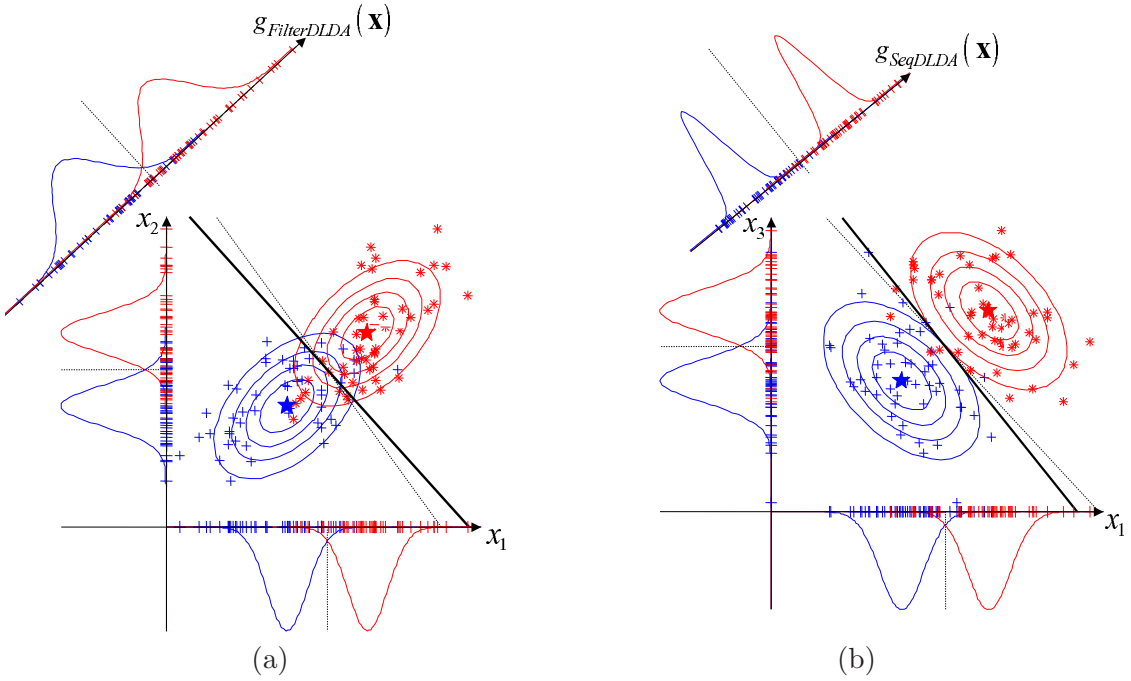


Figure 2.3: Two possible scenarios for selecting correlated features in a DLDA model. Feature x_1 is more discriminative than x_2, i.e., $H(x_1) > H(x_2)$, and $H(x_2) > H(x_3)$. The DLDA with a filter FSS approach (a) would choose x_1 and x_2, while if we check the resulting discriminant $H(g(x))$, the SeqDLDA approach (DLDA with wrapper FSS) in (b) has a better class separation $H(g_{SeqDLDA}(x)) > H(g_{FilterDLDA}(x))$.

The discriminant function in SeqDLDA is the same as in Filter-DLDA [23] (Section 2.2.3)

$$g_l(\mathbf{x}) = \log\left(\frac{\pi_A}{\pi_B}\right) + \sum_{i \in \mathcal{S}_l} H(x_i)\left(\frac{x_i - \hat{\mu}(x_i)}{\hat{\sigma}(x_i) + \hat{\sigma}_0}\right) \tag{2.7}$$

$$H(x) = \frac{\hat{\mu}_A(x) - \hat{\mu}_B(x)}{\hat{\sigma}(x) + \hat{\sigma}_0} \qquad \hat{\sigma}_0 = \underset{i=1..p}{\text{median}}\left(\hat{\sigma}(x_i)\right) \tag{2.8}$$

[5] SeqDLDA was initially proposed in [78]

where $H(x)$ in (2.8) measures class separation. In relation to (2.6) we have that $H(x_i) = \hat{h}_i$, $\hat{\mu}_A(x_i) = \hat{m}_i^A$, $\hat{\mu}_B(x_i) = \hat{m}_i^B$, and $\hat{\sigma}_i = \hat{\sigma}(x_i) + \hat{\sigma}_0$. The additional term $\hat{\sigma}_0$ protects against an unusually low σ produced by chance and makes the score $H(x)$ more robust (this strategy is also used in NSC [96]). The essential difference between SeqDLDA and FilterDLDA is how we choose the set of features $\mathcal{S}$.

Starting from an empty set of features $\mathcal{S}_0 = \emptyset$, at every iteration l, we add the one gene j that most increases $H(g_l(x))$ to the set of selected features $\mathcal{S}_l = \mathcal{S}_{l-1} \cup \{j\}$. The SeqDLDA approach can be seen as a wrapper FSS approach [53] (Section 2.2.2). Instead of measuring $H(g(x))$ for all possible combinations of features, we use a greedy search described in [53] as Forward-Selection/Hill-Climbing.

In contrast to SeqDLDA, regular filter DLDA adds the gene with highest score $H(x_i)$ without checking the resulting model. The number of computations is much higher in SeqDLDA because at each iteration we have to evaluate the resulting discriminants of all possible candidates to add into the model. The advantage is that in situations as those depicted in Figure 2.3, SeqDLDA can choose the features whose correlation structure works better for the DLDA model. An approach similar to SeqDLDA was also proposed in [10], but using a regular t-test which reduces the robustness of the model and the exploratory search resulting in a much lower performance. In Section 2.5.2 we can find the results that evaluate the SeqDLDA compared to other LDA approaches for microarray applications.

In SeqDLDA, maximizing $H(g_l(x))$ is also equivalent to maximizing $J_{\hat{\boldsymbol{K}}}(\boldsymbol{w})$ in (2.4), where the vector $\boldsymbol{w}$ is calculated using only the diagonal part of $\boldsymbol{K}$ in (2.5); i.e., $w_i =$

$d_i/(\sigma_i + \sigma_0)$. In the following section we will extend this approach to consider other choices for the covariance matrix using a similar greedy search strategy.

2.4 BDLDA – Block Diagonal Linear Discriminant Analysis

The DLDA models covered in previous sections ignore the correlation structure. The difference between filter-DLDA (Section 2.2.3) and SeqDLDA (Section 2.3) is in the feature selection: SeqDLDA checks which selected features give better results with the DLDA model. Model selection and feature selection are usually considered two separate tasks. For example, in a Linear Discriminant Analysis (LDA) setting, a modeling assumption is typically made first (e.g., a full or a diagonal covariance matrix can be chosen) and then with this model the feature subset providing the best prediction performance is selected. If limited training data is available, then the number of parameters of a model that can be reliably estimated will also be limited. In the context of LDA, model selection basically entails simplifying the covariance matrix by setting to zero some of these components. This leads to different *block diagonal* matrix structures (e.g., full / diagonal) which involve different sets of features and require different parameters to be estimated.

In this section, we argue that LDA feature and parameter selection should be done jointly; and we propose a greedy algorithm SeqBDLDA [6] for joint selection of features and of a *block diagonal* structure for the covariance matrix. To the best of our knowledge this is the first time such a joint design is proposed in the context of LDA. In more recent work, shrunken centroids regularized discriminant analysis (SCRDA) [34] proposes what can also be considered a joint feature/model selection for LDA. SCRDA, includes two

[6] SeqBDLDA was initially presented in [77]

shrinkage parameters that are adjusted by cross-validation: the first parameter regularizes the covariance matrix by shrinking the off-diagonal terms towards zero, the second parameter performs variable selection by shrinking each feature weight of the discriminant (as in NSC, see Section 2.2.3). In contrast to SCRDA, which only offers a trade-off between a full and a diagonal covariance matrix, the BDLDA framework introduced in this section considers a search across a large collection of different block diagonal options.

The choice of a block diagonal structure is motivated by microarray classification problems, where we have a very large amount of features (i.e., genes) that are expected to be corregulated in small blocks (groups of corregulated genes). Figure 2.1 illustrates a scenario with two features which could be modeled as one BDLDA block. In the context of gene expression analysis, this can be the case of two corregulated genes x_1 and x_2, where a very small change in x_2 triggers a larger change in x_1. The microarray measurement noise makes it very difficult to detect changes in x_2, but taking into account the correlation with x_1 results in a more discriminative LDA model.

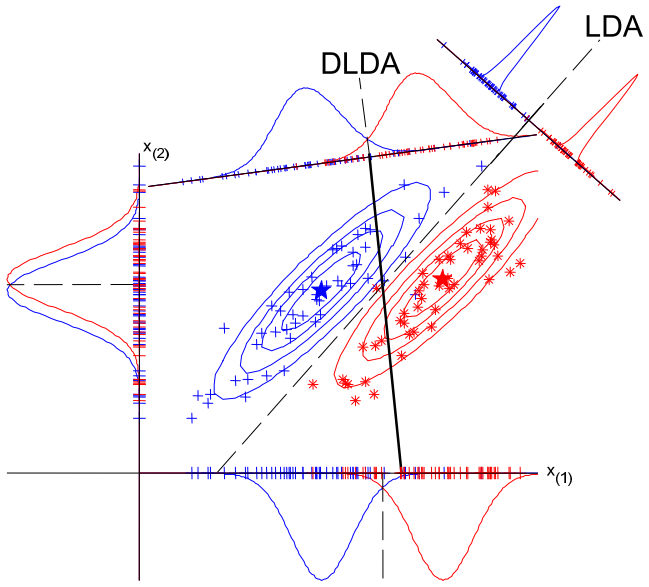
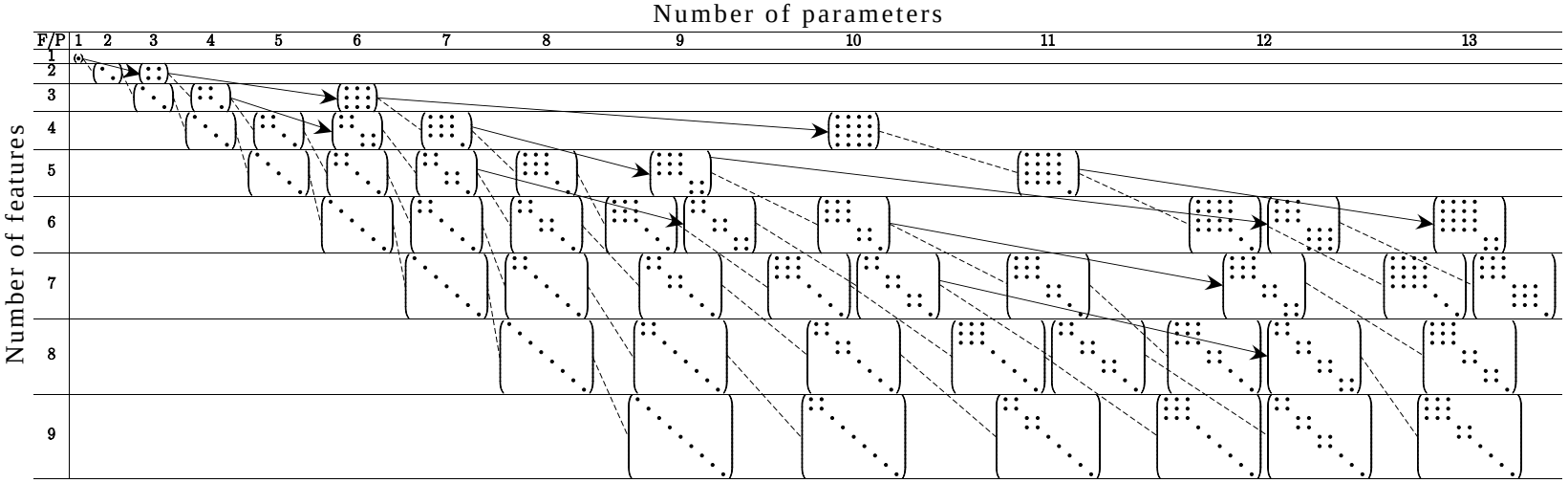


Figure 2.4: Linear discriminant scenario with two correlated features. Feature x_1 is more predictive than feature x_2 since it has better separation between classes. If the correlation between x_2 and x_1 is ignored, the resulting DLDA classifier assigns practically all the weight to x_1, so x_2 will be removed from the model if we use a filter (Section 2.2.3) or wrapper (2.3) approach. If we do not ignore the correlations, the resulting LDA model is more powerful in separating the two classes.

Table 2.1: Sequential generation of candidate covariance matrix models for LDA.



Starting with an empty list, we add one feature at a time (namely, the one that maximizes a statistical score) using two possible operations: (i) *Block expansion* (solid lines), where a new feature is added to an existing block grouping already chosen features in the correlation structure. (ii) *Independent feature addition* (dashed lines), where a feature is added ignoring correlations (i.e., independent of existing blocks of variables in the correlation structure). The best among all these models is selected using cross-validation.

2.4.1 Block Diagonal LDA – BDLDA

Model selection, in the LDA context, is essentially a choice of a structure for the covariance matrix. Thus a simple method would be to perform feature selection for both a diagonal and a full covariance matrix structure and select the best of the two. In a diagonal model, the number of parameters to estimate, l is $l = p$, while in a full matrix $l = \frac{1}{2}p\,(p+1)$. In this section, we propose to further increase the number of available models by including a whole range of block diagonal matrix structures, as shown in Table 2.1.

Applying the bias/variance trade-off principle [39] in this setting implies that the more parameters we estimate the less bias we will have, but at the cost of increasing the variance. For this reason, the LDA performance is limited primarily by the number of parameters to estimate (rather than by the number of features). We use this insight to develop novel efficient techniques to embed feature and model selection, which are based on searching for the best feature set and covariance model *for a given number of parameters.*

Thus, for a given number of parameters, more features can be used with a diagonal covariance model than with a full covariance matrix, but correlation among features will be completely ignored. For uncorrelated features this model will perform best, but there might be correlations present that could be exploited to get better performance with fewer features. Exploring all possible feature subsets and possible block diagonal structures is not feasible. Thus, we propose a sequential greedy algorithm, *SeqBDLDA*, for finding at the same time a feature subset and a block diagonal structure.

2.4.2 Greedy Feature and Model Selection for Block Diagonal LDA

Our proposed greedy algorithm for feature and model selection (see Algorithm 2), adds features to the model sequentially, one at a time. The process starts by selecting the best feature measured with the J score of (2.4). Then, at each stage, we have two options: i) adding one more feature, independent of all previously selected features, thus leading to a new block in the block-diagonal structure, and ii) growing the current block in the matrix structure by adding one more feature to it. These two options are marked with dashed and solid lines, respectively, in Table 2.1 and can be used alternatively to produce feature subsets with different block diagonal covariance structures. In both operations the current set of features, $\mathcal{A}$, is "inherited" from the parent node.

In order to determine which is the best new feature for a given structure we use the scoring procedure discussed in Section 2.4.2.1. After obtaining one feature subset $\mathcal{A}_m$ for each of the models in Table 2.1, we are interested in finding which is the more reliable model if the number of parameters is limited. To do so we use leave-one-out cross-validation (see Section 2.4.2.2).

Algorithm 2 Greedy feature subset and model construction

Create first model with best feature: $i = \arg\max_{j \in S} \frac{d_j}{\sigma_j}$
for all Model m in Table 2.1 **do**
  $\mathcal{A} \leftarrow$ Feature set of the parent node
  $j^* \leftarrow$ AddFeature($\mathcal{A}$,m) ▷ Find the best feature to add
  $\mathcal{A}_m \leftarrow \mathcal{A} \cup \{j^*\}$
  $\epsilon_m \leftarrow$ EvaluateModel($\mathcal{A}_m$,m) ▷ Using crossvalidation
end for
$l \leftarrow$ Number of parameters
$m^* \leftarrow \arg \max_{m:|m|=l} \epsilon_m$ ▷ Find the best model with l parameters

2.4.2.1 Feature addition scoring procedure

Assume that we have already chosen a subset of features $\mathcal{A}$, with sample covariance matrix $\hat{\mathbf{K}}_{\mathcal{A}}$ and difference of sample means $\hat{\mathbf{d}}_{\mathcal{A}}$. Then, from (2.2) the LDA classifier with a model m is constructed using the following weights:

$$\mathbf{w}_{\mathcal{A}} = \hat{\mathbf{K}}_{\mathcal{A},m}^{-1} \hat{\mathbf{d}}_{\mathcal{A}}, \tag{2.9}$$

where $\hat{\mathbf{K}}_{\mathcal{A},m}$ is obtained from $\hat{\mathbf{K}}_{\mathcal{A}}$ by zeroing out those terms that are zero in model m (see examples in Table 2.1). Then, using (2.5), the best new feature to add to the model $j \in \mathcal{A}^C$ (where $\mathcal{A}^C$ is the complement of $\mathcal{A}$ in the original feature set) will be:

$$j* = \arg\max_{j \in \mathcal{A}^C} \frac{\left(\hat{\mathbf{d}}_{\mathcal{A}_j}^t \mathbf{w}_{\mathcal{A}_j}\right)^2}{\mathbf{w}_{\mathcal{A}_j}^t \hat{\mathbf{K}}_{\mathcal{A}_j} \mathbf{w}_{\mathcal{A}_j}} \quad \mathcal{A}_j = \mathcal{A} \cup \{j\} \tag{2.10}$$

In our greedy procedure, the new feature is always added in the lower right corner of the matrix, either as an independent block (i.e., ignoring correlations), or by increasing the size of the lower right block by one. In finding the best feature, significant computational savings can be achieved by exploiting the block structure of the matrix in (2.9), and the fact that only certain blocks in vectors and matrices in (2.10) change with j.

2.4.2.2 Model selection with cross-validation

Since we used the J score (2.4) to guide the search for the feature subset we cannot use it to decide which model to select. This is because it is a biased estimate of performance of the classifier that can be used to compare alternative models with same number of

parameters and features, but does not provide a reliable way to compare models with different structures.

We used leave-one-out cross-validation (Section 2.2.4). to estimate the probability of error of a classifier without bias. In leave-one-out cross-validation, one sample is left out and we train with the remaining $n-1$ samples. Then the sample that has been left out is classified. The entire training procedure is repeated n times for each of the samples and the error rate ϵ_m is estimated as the total number of misclassified samples divided by n. In our case, if the number of parameters is limited to l, we will select the model in the column l of Table 2.1 with the lowest cross-validation error.

2.4.3 Relationship with other LDA methods and applications

Table 2.1 contains several models that have been proposed in the literature: models "grown" by following *only* solid lines, correspond to "full matrix" LDA with forward feature selection (SeqLDA, [105]). Alternatively models grown by following only dashed lines correspond to forward selection using the Diagonal LDA (SeqDLDA, Section 2.3) model. Thus both "full matrix" LDA and SeqDLDA are part of the space of solutions being searched. Note also that if some a priori knowledge was available about the structure of the covariance matrix this could be exploited to reduce the complexity of the search by removing some of the paths in Table 2.1 from consideration. For example, if it is believed that features will tend to be correlated in small groups, it is very easy to set limits on the maximum size of the blocks to be explored by our algorithm.

2.4.4 Repeated feature subset selection BDLDA (Rep-BDLDA)

Using cross validation in the model selection (Section 2.4.2.2) for BDLDA is very time consuming, thus not an appropriate algorithm for gene expression data, which has a large number of features and the number of possible models grows exponentially. As an alternative, we could exploit some underlying knowledge of the data to reduce the BDLDA parameter search space. This is the idea behind the repeated FSS (RFSS) search strategy[7] discussed in this section.

RFSS search method consist of repeating the model construction and feature selection N times. At each repetition, only a predefined maximum number of features $MaxFeature$ is permitted, with the features selected during previous iterations removed from the set of candidate features. Finally, the N models are combined by vector concatenating N means and block diagonally concatenating N covariance matrices. The feature sets in all N models are different and uncorrelated. The model construction is performed N times or stops when there are not enough candidate features. These heuristic search enables the algorithm to find more discriminating features without being influenced by previously selected models. The resulting approach, Rep-BDLDA is useful for gene expression data, because genes belonging to the same pathway tend to be have sparse correlations, but also more than one gene in the same pathway could independently lead to the same outcome we are trying to predict.

[7]This work was in collaboration with L. Shen, more details can be found in [91]

2.5 Results

Several simulated and real microarray datasets are used to evaluate the linear discriminant methods discussed in this chapter: DLDA (Section 2.2.3, SeqDLDA (Section 2.2.3) and SeqBDLDA (Section 2.2.3). The microarray data used in this section correspond to publicly available datasets from medical research in four different cancer classification studies (leukemia [33], colon [4], prostate [93] and neuroblastoma [6]). Cross-validation (Section 2.2.4) is used to properly fit and evaluate the models. In simulated datasets, where the underlying distributions are known, the true classification accuracy can be obtained either analytically or numerically (with Montecarlo Simulation).

2.5.1 Application of DLDA to neuroblastoma

The development of the classification techniques proposed in this chapter are part of a medical study for neuroblastoma tumor led by Dr. Asgharzadeh at Childrens Hospital Los Angeles. This section will describe the dataset that was collected, and will show how to apply and evaluate the DLDA model (as was done in [6]). Some of the evaluation techniques used here, will also be used in the following sections to evaluate the other LDA based techniques presented in this chapter.

Metastatic neuroblastomas lacking amplification of the MYCN proto-oncogene vary in their clinical behavior. Those diagnosed before one year of age are least aggressive and those diagnosed after two years are most aggressive. Age, however, is not always correlated with survival, and it is hypothesized that molecular classification of tumors at diagnosis using gene expression profiling would improve prediction of outcome.

Gene expression profiles of 102 untreated primary tumors from patients with stage 4 MYCN non-amplified neuroblastoma diagnosed at < 12, $12-24$, > 24 months of age were determined using Affymetrix microarrays. A supervised method using DLDA (Section 2.2.3) was devised to build a multi-gene model for predicting outcome.

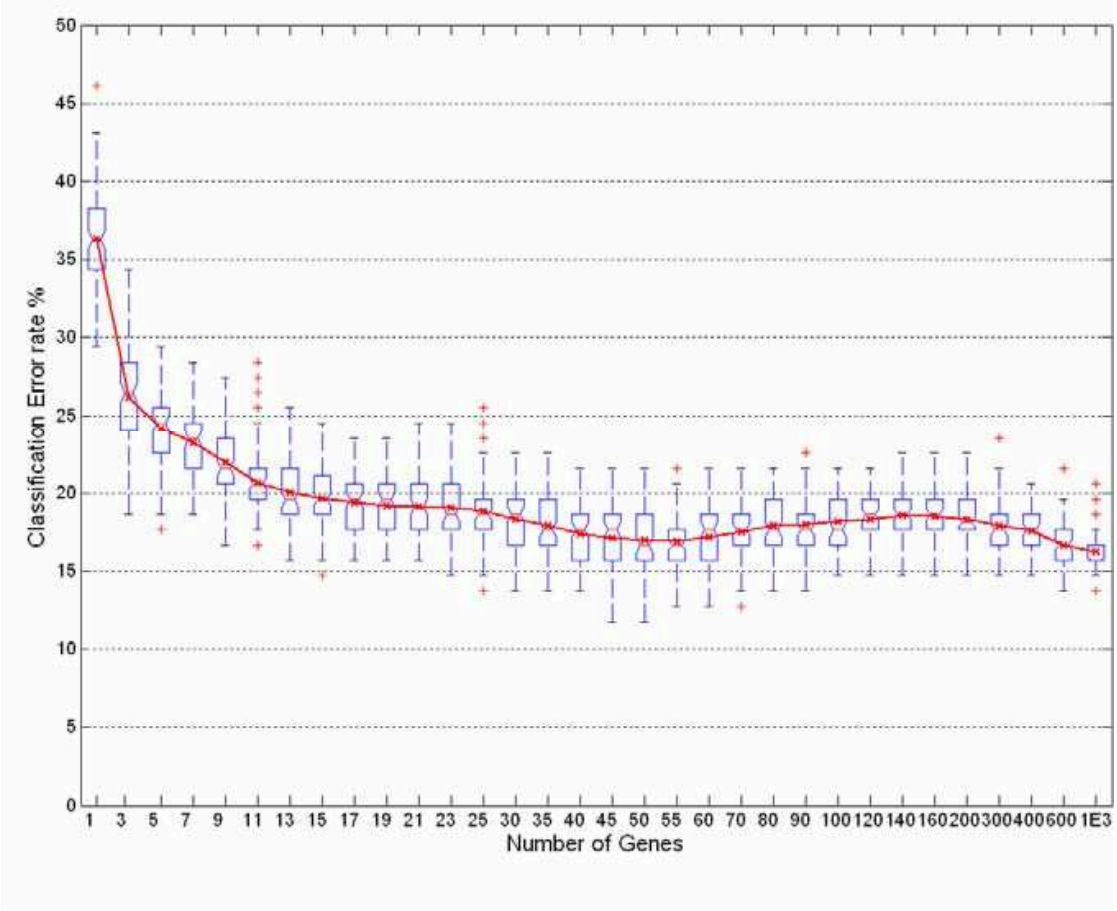


Figure 2.5: Classification error plot showing the percentage of neuroblastoma patients misclassified using DLDA models of different size (x-axis). Red line: mean error rate generated from 100 times 10-fold cross validations using the training set. Blue box plots: lower and upper bounds of the boxes represent the 25th and 75th quartiles of the cross-validation errors for a given gene model; whiskers represent data within the range of 1.5 times the upper and lower inter quartiles; outliers are shown as red crosses. The first minima of the curve occurred with 55 genes.

The average cross-validation error rate was first minimized for models that used probes for 55 genes (52 unique genes) as shown in Figure 2.5. As a validation strategy, we performed nested leave-one-out cross-validation of the entire process including the model selection. The leave-one-out cross-validation error-rate estimates from our nested algorithm in predicting progression status was 15.7% with permutation standard error of $\pm 3.95\%$ for all patients and $17.9\% \pm 5.09\%$ for those diagnosed after 12 months of age. Without cross-validation, the training error rate of 9.8% for 55 genes using all 102 tumors is probably an underestimate and reflects the need for cross-validation [5]. To evaluate if the selection of the multi-gene model that minimizes classification error rate occurred by chance, we permuted our samples by randomly mislabeling them with respect to their progression status and created 1000 sample sets. The error rates generated from multi-gene models in these permuted sample sets was found to be higher than that generated from our correctly labeled sample set ($P < 0.01$).

The gene expression signatures of tumors obtained at diagnosis from patients with clinically indistinguishable high-risk, metastatic neuroblastomas identify subgroups with quite different outcomes. Accurate identification of these subgroups with gene expression profiles will probably facilitate development, implementation, and analysis of clinical trials aimed at improving outcome. The list of candidate genes can also be used as candidate targets for future medical studies. The following sections will analyze in this and other microarray datasets if the other techniques introduced in this chapter can obtain a better performance in terms of prediction accuracy or in capturing better the correlations between the genes.

2.5.2 Evaluation of SeqDLDA in four microarray datasets

The proposed SeqDLDA algorithm (2.3) has been evaluated using 100 runs of 10-fold Cross-Validation on several 2-class datasets shown in Table 2.2. The leukemia [33] (n=72, p=7129), colon [4] (n=22, p=2000), prostate [93] (n=102, p=6033) datasets are publicly available and widely used in other studies. The neuroblastoma dataset (n=102, p=44298) has been described in Section 2.5.1. In all cases, the gene expression has been normalized by clipping values lower than 1 and taking a log-transform.

Using the same evaluation methods, the proposed SeqDLDA approach has been compared to DLDA [23], NSC [96], GP-DLDA [10], ULDA [106] and Linear SVM.

Table 2.2: Average Cross-validation error, number of selected genes and standard deviation (SD)

	Leukemia	Colon	Prostate	Neuroblastoma
Seq-DLDA	4.11%,180(1.32%)	12.06%,50(1.87%)	5.53%,26(0.90%)	13.87%,70(2.41%)
GP-DLDA	3.82%,18(0.77%)	13.08%,16(1.76%)	6.44%,20(0.70%)	15.77%,35(1.61%)
DLDA	3.38%,7(1.30%)	12.40%,3(1.44%)	6.99%,2(0.33%)	16.91%,55(1.54%)
NSC	4.18%,70(0.80%)	10.31,20(1.02%)	7.65%,6(0.42%)	17.98%,70(1.67%)
ULDA	3.39%,p(0.747%)	15.19%,p(2.72%)	8.53%,p(1.10%)	13.42%,p(1.55%)
Lin-SVM	2.61%,p(0.57%)	15.39%,p(2.17%)	8.01%,p(1.14%)	14.13%,p(1.45%)

In the studied datasets SeqDLDA obtains results very close to the best approach, and the best results for the prostate and neuroblastoma datasets. Additionally SeqDLDA performs gene selection, unlike ULDA and SVM whose classifier uses the whole set of genes. Gene selection is crucial in order to identify genomic targets that may explain the disease.

Classical DLDA filtering approaches [23, 96] provide similar results in the absence of gene correlations or inter-pair correlations in GP-DLDA [10]. For example, Nearest

Shrunken Centroid (NSC) obtains the best results in the colon dataset. However correlation among genes is generally present and the SeqDLDA method will allow us to choose genes that may have a lower score (under a diagonal correlation assumption) but can be shown to provide better classification performance when combined with the already selected genes. Additionally, we have also noticed that improvement in performance over DLDA is more noticeable when a larger number of training samples is available. In the neuroblastoma data set, the average misclassification rate of DLDA (16.91%) is significantly reduced to 13.87% using SeqDLDA.

2.5.3 Simulation results with SeqBDLDA

The SeqBDLDA has been first extensively analyzed with artificial data for two basic reasons. First this allows to control the covariance matrix and to evaluate the ability of the algorithm to select a model close to actual one. Second, evaluation is simplified, since for a given LDA-trained model we can exactly compute the probability of error without having to estimate it.

The training data is generated by drawing n samples with distributions $f_A(\mathbf{x}) \sim N(\mathbf{m}_A, \mathbf{K})$, $f_B(\mathbf{x}) \sim N(\mathbf{m}_B, \mathbf{K})$. The two basic generating parameters are $\mathbf{K}$, and $\mathbf{d} = \mathbf{m}_A - \mathbf{m}_B$. We have experimented with several covariance matrix structures and randomly permuted the features, so that in general two contiguous features are not necessarily correlated. In the experiments presented here $\mathbf{d}$ was fixed so that the SNR of the features is exponentially decreasing with parameter γ:

$$\left|\frac{d_j}{\sigma_j}\right| = e^{-\gamma j} \quad \left(\sigma_j^2\right)_j = \mathrm{diag}(\mathbf{K}) \tag{2.11}$$

The number of features that will be optimal for the classifier will usually be between $1/\gamma$ and $4/\gamma$ approximately, increasing with the sample size n and decreasing with p. When n and p are constant, if γ is small, a large number of features will be required for the classifier and a diagonal matrix model will be preferred over a full matrix one.

After training the weight vector $\mathbf{w}$, we can compute the exact probability of error

$$P_{e|\mathbf{w}} = 1 - \Phi\left(\frac{1}{2}\sqrt{\frac{J_{\mathbf{K}}(\mathbf{w})}{1+1/n}}\right) \quad J_{\mathbf{K}}(\mathbf{w}) = \frac{\left(\mathbf{d}^t\mathbf{w}\right)^2}{\mathbf{w}^t\mathbf{K}\mathbf{w}} \tag{2.12}$$

where $\Phi(x)$ is the standard normal cumulative distribution function and $1+1/n$ takes into account the cost of estimating the b parameter in (2.3). This is possible because we know the underlying distribution that generates the samples we want to classify. Using Montecarlo simulation, we repeat the training and evaluation T times and the average P_e is estimated as:

$$\hat{P}_e = \frac{1}{T}\sum_{t=1}^{T} P_{e|\hat{\mathbf{w}}_t} \tag{2.13}$$

These results are reported for our proposed algorithm (SeqBDLDA) along with the two related wrapper methods SeqDLDA and SeqLDA (see Section 2.4.3). Finally, 95% confidence intervals asses the statistical significance of our findings.

The first simulation example uses a Toeplitz symmetric covariance matrix. A Toeplitz symmetric matrix arises from AR processes, in which contiguous features are locally correlated. This is exploited by several classification algorithms [30], which will, however, fail if the features are permuted. Our proposed algorithm avoids this problem since

it is invariant to feature permutation. This comes indirectly from our original design assumption that no prior knowledge exists about correlation between features.

In our experiments the more diagonally dominant the matrix is, the better the diagonal model will be. If the training data is limited, the full-matrix approach quickly fails as we increase the number of parameters. Figure 2.6 illustrates this with the following covariance matrix:

$$\mathbf{K} = \begin{pmatrix} \frac{1}{4} & -\frac{1}{8} & \frac{1}{10} & 0 & \cdots \\ -\frac{1}{8} & \frac{1}{4} & -\frac{1}{8} & \frac{1}{10} & \ddots \\ \frac{1}{10} & -\frac{1}{8} & \frac{1}{4} & -\frac{1}{8} & \ddots \\ 0 & \frac{1}{10} & -\frac{1}{8} & \frac{1}{4} & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{pmatrix} \tag{2.14}$$

In general, the SeqBDLDA algorithm achieves a very good performance for a Toeplitz covariance matrix. Although a Toeplitz matrix it is not strictly block diagonal, it has a sparse number of correlations different than 0. Thus, even in the case that the underlying features covariances cannot be arranged in blocks for some permutation, SeqBDLDA still does a good job in reducing the number of parameters that have to be estimated to approximate the underlying Toeplitz structure.

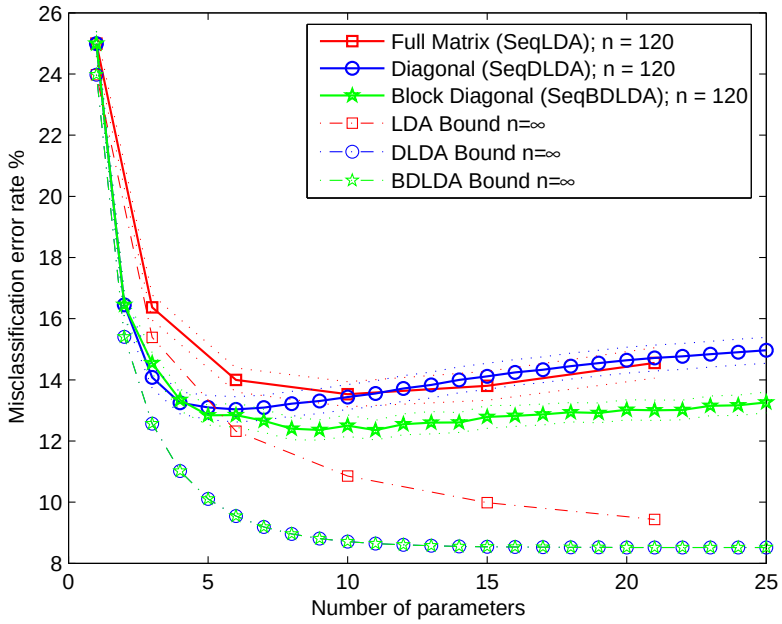


Figure 2.6: SeqBDLDA Classification performance, Toeplitz covariance matrix. $p = 200$, $n = 120$, $\mathbf{K}$ as in (2.14), $\gamma = 0.2$. Solid and dotted lines represent the mean $\hat{P}_e$ and its 95 % confidence interval for 100 trainings. In this example SeqBDLDA (in green) outperforms both SeqLDA (red) and SeqDLDA (blue).

The second experiment tests the algorithm with block diagonal matrices. Figure 2.7 shows the results for the following covariance matrix structure:

$$\mathbf{K} = \begin{pmatrix} \mathbf{A} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{D} \end{pmatrix} \tag{2.15}$$

$$\mathbf{A} = \begin{pmatrix} \frac{1}{2} & \frac{1}{3} & 0 \\ \frac{1}{3} & \frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{1}{2} & \frac{1}{2} \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} \end{pmatrix}$$

$$\mathbf{C} = \begin{pmatrix} \frac{1}{2} & 0 & -\frac{1}{3} & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 & 0 \\ -\frac{1}{3} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \end{pmatrix} \quad \mathbf{D} = \begin{pmatrix} \frac{1}{2} & 0 & \dots \\ 0 & \frac{1}{2} & \ddots \\ \vdots & \ddots & \frac{1}{2} \end{pmatrix}$$

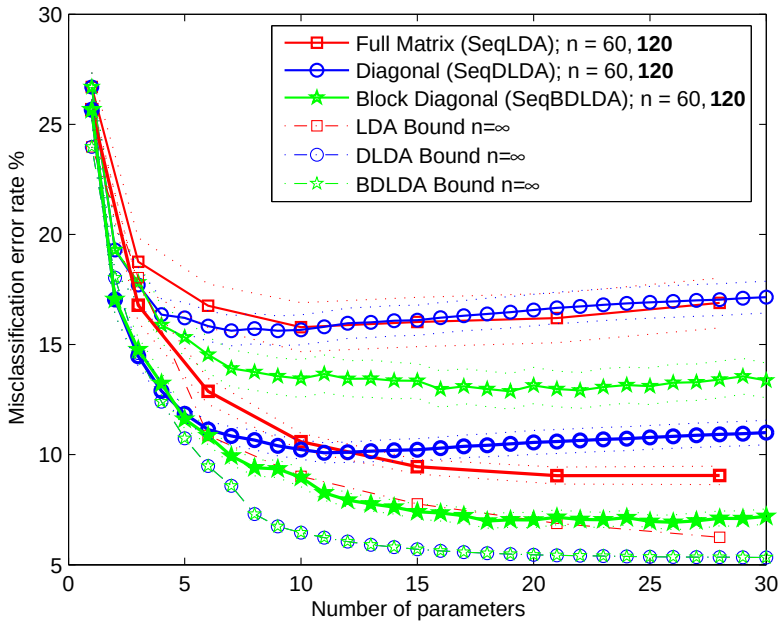


Figure 2.7: SeqBDLDA Classification performance, Block covariance matrix. $p = 200$, $n = 60$ (thin line),120 (thick line), $\mathbf{K}$ as in (2.15) , $\gamma = 0.1$. Solid and dotted lines represent the mean and its 95 % confidence interval for P_e of 100 trainings. This example illustrates that depending on the number of samples available, SeqBDLDA (in green) can chose an adequate block diagonal structure, that outperforms both SeqLDA and SeqDLDA.

Figure 2.7 shows that when training data is very limited, e.g., $n = 60$, a diagonal structure (SeqDLDA) outperforms a full matrix approach (SeqLDA), while as n increases

the full matrix approach becomes better. Our technique approach can default to SeqLDA or SeqDLDA for some number of parameters, but it is also capable of choosing other intermediate block-diagonal alternatives. This is why most of the time SeqBDLDA can outperform SeqLDA and SeqDLDA in a block matrix scenario.

2.5.4 Results with Rep-BDLDA on microarray data

In microarray datasets the SeqBDLDA algorithm evaluated in the previous section is very slow and we need mechanisms to avoid using cross-validation to select the model and heuristics for reducing the search of the best block structure. This can be achieved by Repeated FSS (RFSS) of small sets of BDLDA models as explained in Section 2.4.4 (more details and results also in [91]).

The Rep-BDLDA algorithm is tested on the colon, prostate and neuroblastoma datasets. The Rep-BDLDA algorithm was used with $N = 5$ repetitions of BDLDA models with maximum block size $MaxGrow = 3$ and $MaxFeature = 20$ features. The average 50X10-fold cross-validation error rates and standard deviations are shown in Table 2.3. In SCRDA [34], the cross-validation error rate corresponds to the minimum after adjusting two regularization parameters.

In all three real datasets, BDLDA has the lowest error rates. Among them, Neuroblastoma, with more than 40,000 features, is considered the most challenging. Our algorithm reduced the error rate by more than 2%, compared to the second best algorithm, SeqDLDA (Section 2.5.2).

Table 2.3: Average Error Rate (Standard Deviation) for microarray data

	Colon	Prostate	Neuro-blastoma
RepBDLDA	**10.06%**	**5.21%**	**10.61%**
	(1.15%)	(0.85%)	(1.29%)
SeqDLDA	12.06%	5.53%	13.87%
	(1.87%)	(0.9%)	(2.41%)
NSC	10.31%	7.65%	17.98%
	(1.02%)	(0.42%)	(1.67%)
SCRDA	11.41%	5.41%	14.22%
	(1.69%)	(0.89%)	(1.39%)

2.6 Conclusions

This chapter covered the design and evaluation of classifiers based on linear discriminant analysis (LDA) for microarray applications. The design of accurate classifiers is challenging due to the limited number of training samples compared to the large number of genes in microarray studies. Under these conditions, the estimation of parameters to fit an LDA model (the covariance matrix and the class centroids) is not robust. However, the underlying biological models tend to be sparse in the sense that: i) very few genes are normally relevant for the outcome the classifier is trying to predict, ii) genes are correlated in relatively small groups. Several modeling and feature selection approaches have been proposed using the LDA framework exploiting this underlying sparseness.

Starting from DLDA models that ignore gene correlations we investigate several searching approaches (filter DLDA and wrapper SeqDLDA) for selecting a subset of genes. This is then expanded to BDLDA models in which we search for a feature subset along with a block structure that models the interactions between genes. In other words,

we solve the problem of fitting an LDA model by searching the combination of genes and gene correlations that give best discrimination between classes.

Depending of the number of the training samples available, we can can search for a smaller or larger model in terms of number of parameters to estimate. The appropriate size of the model can be determined by crossvalidation. The embedded FSS search tries to give the best block structure and parameter/feature selection of a model of a given size. Results on simulated and real microarray data demonstrate that the proposed SeqDLDA, SeqBDLDA and specially RepBDLDA offer a very competitive performance compared to other state-of-the-art approaches such as NSC, SCRDA, GP-DLDA and SVM.

There is a steady increase in the knowledge about gene regulation and more and more microarray experiments are deposited into large public available databases every day. In future work, all this prior knowledge could be used in the FSS and BDLDA framework presented here. For example, groups of corregulated genes could be proposed to belong to the same block in BDLDA, or gene pathways could be used to guide the FSS search.

Chapter 3

Sparse representation and Bayesian detection of genome copy number alterations from microarray data

Genomic instability in cancer leads to abnormal genome copy number alterations (CNA) that are associated with the development and behavior of tumors. Advances in microarray technology have allowed for greater resolution in detection of DNA copy number changes (amplifications or deletions) across the genome. However, the increase in number of measured signals and accompanying noise from the array probes present a challenge in accurate and fast identification of breakpoints that define CNA. In this chapter we propose a novel detection technique that exploits the use of piece-wise constant (PWC) vectors to represent genome copy number and sparse Bayesian learning (SBL) to detect CNA breakpoints [1]. First, a compact linear algebra representation for the genome copy number is developed from normalized probe intensities. Second, SBL is applied and optimized to infer locations where copy number changes occur. Third, a backward elimination (BE) procedure is used to rank the inferred breakpoints; so that a cut-off point can be efficiently adjusted in this procedure to control for the false discovery rate (FDR). The performance

[1] Part of the work presented in this chapter has been published in [75, 79]

of our algorithm is evaluated using simulated and real genome datasets and compared to other existing techniques. Our approach achieves the highest accuracy and lowest FDR while improving computational speed by several orders of magnitude.

3.1 Introduction

Copy number alterations involving deletion or replication of entire chromosomes or chromosomal regions are known to occur in numerous genetic disorders (e.g., Down's syndrome, Klinefelter's syndrome), while replications of multiple chromosomes leading to states of hyperploidy are well known in cancer biology [3]. Similarly, regional CNA have been demonstrated in tumors, and linked to leading them to develop aggressive behavior. Examples include loss of RB tumor suppressor in retinoblastoma or MYCN proto-oncogene amplification in neuroblastoma. Recently, a large number of copy number variants (CNVs) have also been described in the human genome [82] and found across large numbers of individuals. These recurrent copy number changes, CNVs, tend to be much smaller in size and will be considered in more detail in Chapters 4 and 5. Array-based technologies use genetic material as sensors or probes to estimate copy number for the intended genomic regions. The resolution for detection of CNA depends on the number and type of probes placed on these arrays. Comparative genomic hybridization (CGH, [51]) is one of the earlier array platforms that uses large insert DNA fragments (kilobases) as probes in measuring DNA copy number. These probes, numbering typically in the order of hundreds of thousands to millions, allow co-hybridization to take place between a fluorescently tagged genome of interest and a normal reference genome. The

relative intensity at a given probe is directly proportional to the copy number for that region. More recently, platforms using short oligonucleotide probes ($\leq$60 bases), which allow placement of hundreds of thousands of probes on an array, have become more widely used [80]. The probes are only hybridized to a tagged genome of interest and the intensities are usually compared to those of reference set of arrays of normal genomes. The majority of these arrays use oligonucleotides that also probe for regions with genotype polymorphisms thus providing both copy number and genotype information [43, 71]. The increase in the probe density poses computational challenges to accurately and efficiently assess DNA copy number and identify altered regions.

Several algorithms have been proposed to detect CNA [11, 29, 42–44, 60, 64, 68, 69, 73, 80, 107]. Most of these algorithms rely on a fundamental characteristic, namely, that a genome is composed of relatively long segments, DNA sequences, that have a constant number of copies present. The genomic segments can be represented by m probes mapping to a specific position on the genome having c_m copies. The copy numbers c_m can be ordered and arranged as vectors $\mathbf{c}$ that have two key characteristics: i) they are *piecewise constant (PWC)* with very small number of breakpoints relative to the number of probes and ii) they *have discrete values (DIS)* (i.e., copy numbers can only be 0,1,2,3,...). However, these properties cannot be directly observed in the log-intensities y_m measured with microarrays, due to contamination by biological and technical noise; thus a widely used model is:

$$y_m = x_m + \epsilon_m \tag{3.1}$$

where x_m represents the average log intensity, and ϵ_m is an additive zero-mean white random process (see Figure 3.1).

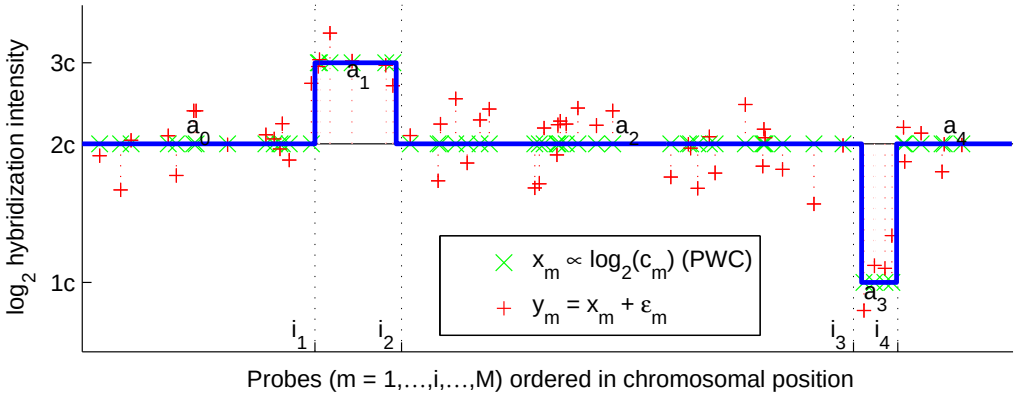


Figure 3.1: Graphical representation of the observation model (3.1) using a chromosome section with 2 alterations as an example (simulated data). The underlying mean hybridization intensity x_m is piece-wise constant (PWC) with breakpoints $\mathcal{I} = \{i_1, i_2, i_3, i_4\}$ that mark the starting probe of each segment, and amplitudes $\boldsymbol{a} = (a_0, a_1, a_2, a_3, a_4)$ that depend on the underlying number of copies (DIS). The observed probe hybridization intensities y_m do not follow this expected behavior due to degradation by hybridization noise ϵ_m.

Most techniques exploit the assumption that $x_m \propto log_2(c_m)$ and that properties PWC and DIS, as introduced above, are met. For example, one of the first and simplest techniques to exploit PWC consisted of applying a smoothing filter followed by a threshold [43, 80]. This has been improved upon by more specialized techniques such as wavelets [42], segmentation [60, 69, 73], or penalized least-squares [44]. Additionally, hidden Markov models (HMM) [29, 64, 68, 107] and Bayesian methods [11] exploit both PWC and DIS by assuming that each observation y_m comes from a probe in a particular hidden copy number state c_m to be inferred. Exploiting DIS can be difficult in cases where that state is rare (observed a small number of times), or in cases of specimens containing a heterogeneous population of cells with respect to DNA copy numbers. This

heterogeneous copy number state typically occurs in the case of tumor samples, where $x_m = log_2(\bar{c}_m)$ would correspond to the average copy number in the mixture.

Among all the previous methods, circular binary segmentation (CBS) [69] was found to be one of the most accurate methods for CNA detection by two independent comparative studies [57, 103], but was also one of the slowest. These studies used synthetic datasets where the CNA occur at known positions, the probes are uniformly spaced, and the hybridization noise is generated according to a white Gaussian distribution. More recently, new approaches [25, 85, 90] have extended previously proposed methods in order to target specific scenarios not considered by the CBS approach, e.g., presence of outliers [90], non-uniform probe spacing [85], and chromosomes with a reduced number of probes and non uniform variance [25]. In our case we focus on the default conditions and metrics proposed by [103] under which our results show that these new algorithms do not give better accuracy than that of CBS and are slower. The performance of these algorithms under different conditions that may arise on specific microarray platforms is discussed in Chapter 5. Recently, the computational performance of CBS algorithm has significantly improved with a new approximate version [101] with no significant loss of performance. However, the run-times of this new version and the other new algorithms are still very high, especially when applied to the new high density array platforms.

In contrast, we propose a novel modeling of genomic data using PWC vectors that can be efficiently exploited to build algorithms for CNA detection with very significant gain in computational speed. We also propose a new approach that we call genome alteration detection algorithm (GADA) for CNA detection from array data that combines the sparse Bayesian learning (SBL) technique introduced by [97] and a backward elimination (BE)

procedure that can efficiently adjust the accuracy trade-off between sensitivity and the FDR.

We evaluate our algorithm using the simulated array-CGH dataset proposed by [103], where the underlying positions of copy number changes are known and can be used as a benchmark to compare the accuracies of different algorithms. We also evaluate the performance of three algorithms [25, 85, 90] that appeared after the [103] comparative study, and the newer CBS implementation [101]. Using that benchmark dataset our GADA approach obtained one of the best accuracies, and the best performance in terms of computational speed, followed by CBS. Additionally we compare the results of our algorithm and CBS on data generated from several array types from two commercial manufacturers (Affymetrix and Illumina) using DNA from four different neuroblastoma cell lines. Our results indicate that our algorithm can analyze data efficiently from high density platforms and provide an accuracy similar or better than that of state of the art algorithms, but with reduced computation costs. On the new large array platforms, our algorithm is two orders of magnitude faster than one of the most accurate algorithms available to date, i.e., CBS [69].

This chapter is organized in to major parts. The first part (Sections 3.2 through 3.7) introduces the PWC models, the SBL and BE algorithms that compose the GADA approach; and also discusses the theoretical properties of these models and algorithms. The second part evaluates the proposed GADA approach in both simulated (Section 3.8) and real microarray data (Section 3.9) and presents the final conclusions.

3.2 PWC vector representation of Genomic Data

One of the major contributions of this work is the development of a compact description for the copy number along the chromosome using PWC vectors (green signal in Figure 3.1). Using simple linear algebra, any PWC vector $\boldsymbol{x}$ with K breakpoints $\mathcal{I} = \{i_1, \ldots, i_K\}$ can be compactly represented by a linear combination of K step vectors $\boldsymbol{f}_i$ (each with a single breakpoint i in $\mathcal{I}$, see Figure 3.2) plus a constant vector $\boldsymbol{f}_0$.

$$\boldsymbol{f}_i(m) = \begin{cases} -\sqrt{\frac{M-i}{iM}} & m \leq i \\ \sqrt{\frac{i}{M(M-i)}} & m > i \end{cases} \tag{3.2}$$

$$\boldsymbol{f}_0(m) = \frac{1}{\sqrt{M}} \tag{3.3}$$

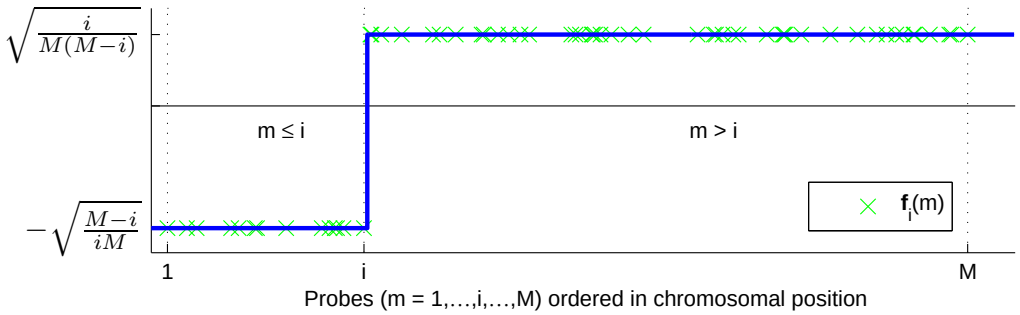


Figure 3.2: Step vector f_i with a breakpoint between probe i and $i+1$ as defined in (3.2). Notice that the step vectors have been normalized to have unit norm, $\sum_{m=1}^{M} (f_i(m))^2 = 1$, and average zero for $i > 0$, $\sum_{m=1}^{M} (f_i(m)) = 0$.

Therefore, in matrix notation, we can write this linear combination as:

$$\boldsymbol{x} = \boldsymbol{F}\boldsymbol{w} \tag{3.4}$$

where the columns of $\boldsymbol{F}$ are the step functions ($\boldsymbol{F} = [\boldsymbol{f}_0, \boldsymbol{f}_1, ..., \boldsymbol{f}_{M-1}]$); and, $\boldsymbol{w}$ is a *sparse* vector, i.e., there are only $K + 1$ non-zero components. Equivalently, we can remove the components of $\boldsymbol{w}$ that are zero and write:

$$\boldsymbol{x} \quad = \quad \boldsymbol{F}_{\mathcal{I}} \boldsymbol{w}_{\mathcal{I}} \tag{3.5}$$

where $\boldsymbol{F}_{\mathcal{I}} = [\boldsymbol{f}_0, \boldsymbol{f}_{i_1}, \ldots, \boldsymbol{f}_{i_K}]$ and $\boldsymbol{w}_{\mathcal{I}} = [w_0, w_{i_1}, \ldots, w_{i_K}]$. This representation has three very important properties that are proved in Section 3.2.1. First, the columns of $\boldsymbol{F}$ form a *basis* that can be used to represent any arbitrary vector. Second, it has a *nested structure*, and for each additional breakpoint i that the PWC vector may contain, we only require an additional weight w_i to be nonzero. Third, any arbitrary PWC vector with exactly K breakpoints can be represented with $K + 1$ non-zero components which is proved to be the minimum possible amount; i.e., *maximal sparseness*.

To the best of our knowledge, we are the first to explicitly propose this representation in the context of genome copy number variations [79] and to exploit its properties to develop a highly accurate and efficient detection technique that will be detailed in the following sections.

3.2.1 Properties of the PWC representation

This representation was initially inspired by the concept of wavelet footprints [21] where the more general case of piece-wise polynomial signals is considered from a wavelet analysis perspective. The *maximally sparse* representation for PWC signals demonstrated in wavelet footprints is reformulated here using standard linear algebra and extended to

arbitrary vector lengths. This also allows us to establish a correspondence between sets of breakpoints and a *nested structure* of vector subspaces which we use here to demonstrate the representation properties.

Mathematically, a PWC vector $\boldsymbol{x}$ can be completely characterized by its change locations (i.e., breakpoints) and the constant values of the regions in between (i.e., segment amplitudes):

Definition 1 (PWC vector)**.** *A piece-wise constant vector* $\boldsymbol{x} = (x_1, \ldots, x_M)^t$ *is characterized by an ordered set of* K *discontinuity locations* $\mathcal{I} = \{i_1 < i_2 \ldots < i_K\} \subset \{1, \ldots, M-1\}$ *(i_k denotes the beginning position of segment k) and a vector with the corresponding* $K+1$ *segment amplitudes* $\boldsymbol{a} = (a_0, \ldots, a_K)^t$*. Thus:*

$$\boldsymbol{x}^t = \Big(\underset{\uparrow 1}{a_0}, \ldots, a_0, \underset{\uparrow i_1}{a_1}, \ldots, a_{k-1}, a_{k-1}, \underset{\uparrow i_k}{a_k}, a_k, \ldots, \underset{\uparrow M}{a_K} \Big) \tag{3.6}$$

With this definition it is easy to show that the breakpoint sets $\mathcal{I}$s induce the following vector subspace properties:

Lemma 1 (PWC Vector Subspaces)**.** *Let* $\mathcal{S}_\mathcal{I}$ *be the set of all PWC vectors* $\mathbf{x}$ *that have breakpoint locations contained in* $\mathcal{I}$*, and segment amplitudes* $\boldsymbol{a} \in \mathbb{R}^{K+1}$*. Then, we have that:*

i) $\mathcal{S}_\mathcal{I}$ *is a subspace of* $\mathbb{R}^M$ *of dimension* $K+1$*.*

ii) $\mathcal{S}_{\mathcal{I}_1}$ *is a subspace of* $\mathcal{S}_{\mathcal{I}_2}$ *if and only if* $\mathcal{I}_1 \subset \mathcal{I}_2$

Proof. It is clear that i) holds since, first, for any $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ with breakpoints in $\mathcal{I}$, but different amplitudes $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$; we have that $\boldsymbol{x}_3 = \boldsymbol{x}_1 + \boldsymbol{x}_2$ may remove existing breakpoints but never create a breakpoint outside $\mathcal{I}$, thus $\boldsymbol{x}_3 \in \mathcal{S}_\mathcal{I}$ because it will always have

breakpoints contained in the same $\mathcal{I}$, and $\boldsymbol{a}_3 = \boldsymbol{a}_1 + \boldsymbol{a}_2$. Second, for any $\boldsymbol{x}_1 \in \mathcal{S}_\mathcal{I}$ and for all α, $\boldsymbol{x}_4 = \alpha \boldsymbol{x}_1$ will also have breakpoints contained in $\mathcal{I}$ with $\boldsymbol{a}_4 = \alpha \boldsymbol{a}_1$ and thus will belong to $\mathcal{S}_\mathcal{I}$. Furthermore, when $\mathcal{I}$ is fixed $\mathbf{x}$ and $\mathbf{a}$ vector spaces are isomorphic and hence $\mathcal{S}_\mathcal{I}$ has dimension $K+1$; thus, ii) readily follows from i). □

Part ii) of the lemma is equivalent to saying that any PWC vector $\mathbf{x} \in \mathcal{S}_\mathcal{I}$ can be represented as a linear combination of step vectors in $\mathcal{S}_{\{k\}}$, $k = i_1, \ldots, i_K$. With this principle in mind, we now introduce a basis for PWC signal representation that has some desirable properties.

Theorem 1 (PWC Basis)**.** *Define a matrix* $\boldsymbol{F} = [\boldsymbol{f}_0, \boldsymbol{f}_1, ..., \boldsymbol{f}_{M-1}]$*, with columns* $\boldsymbol{f}_i$ *defined as in* (3.2)*. Then, we have the following properties:*

i) (Complete Basis): *The columns of* $\mathbf{F}$ *are a basis for* $\mathbb{R}^M$*, i.e., for any* $\boldsymbol{x} \in \mathbb{R}^M$ *there exists a unique* $\mathbf{w}$ *such that* $\boldsymbol{x} = \boldsymbol{F}\boldsymbol{w}$*.*

ii) (Nested Structure): *The columns of* $\boldsymbol{F}_\mathcal{I} = [\boldsymbol{f}_0, \boldsymbol{f}_{i_1}, \ldots, \boldsymbol{f}_{i_K}]$ *, "nest" of* $\boldsymbol{F}$*, are a basis for the vector subspace* $\mathcal{S}_\mathcal{I}$*, formed by PWC vectors with breakpoints at* $\mathcal{I} = \{i_1 < i_2 \ldots < i_K\}$*.*

iii) (Maximal Sparseness): *Any PWC vector* $\boldsymbol{x} \in \mathcal{S}_\mathcal{I}$*, can be written as* $\mathbf{x} = \mathbf{F}\mathbf{w}$*, where* $\mathbf{w}$ *has as many as* $|\mathcal{I}|+1$ *non-zero entries, which is the minimal amount possible (maximal sparseness). Moreover, if the non-zero weights are* $\boldsymbol{w}_\mathcal{I} = [w_0, w_{i_1}, \ldots, w_{i_K}]$*, we can write* $\boldsymbol{x} = \boldsymbol{F}_\mathcal{I}\boldsymbol{w}_\mathcal{I}$*, where the subscript* $\mathcal{I}$ *denotes that only the columns of* $\mathbf{F}$ *(resp. components of* $\mathbf{w}$*) at the positions corresponding to the indices in* $\mathcal{I}$ *are included.*

Proof. In order to better understand the previous theorem we will give a proof by constructing the PWC basis using the nested structure. First, if $\mathbf{x}$ is a constant vector, i.e.,

it has no discontinuities, $\mathcal{I} = \emptyset$, the dimension of $\mathcal{S}_0 = \mathcal{S}_{\mathcal{I}=\emptyset}$ is one and can be spanned by the constant vector $\boldsymbol{f}_0$. Then, for $k = 1, \ldots, M-1$ the vector spaces $\mathcal{S}_k = \mathcal{S}_{\mathcal{I}=\{k\}}$ of PWC vectors with a single discontinuity between k and $k+1$ can be spanned by adding the element $\boldsymbol{f}_k$, a step vector with a breakpoint at that position. Moreover, the set of vectors now forms a complete basis: from Lemma 1.ii) any $\mathbf{x} \in \mathcal{S}_{\mathcal{I}}$ can be represented by linearly combining $\{\boldsymbol{f}_0, \boldsymbol{f}_{i_1}, \ldots, \boldsymbol{f}_{i_K}\}$. This basis construction proves ii) in the theorem, as well as i) when $\mathcal{I} = \{1, \ldots, M\}$.

Alternatively we can prove that i) holds, by simply checking that $\mathbf{F}$ is a square invertible matrix. The the rows of its inverse $\boldsymbol{F}^{-1}$ which form the *dual basis* are:

$$
\begin{aligned}
\boldsymbol{F}^{-1} &= \left[\tilde{\boldsymbol{f}}_0, \ldots, \tilde{\boldsymbol{f}}_{M-1}\right]^t \\
\tilde{\boldsymbol{f}}_0 &= \frac{1}{\sqrt{M}}\mathbf{1}_M \\
\tilde{\mathbf{f}}_k(m) &= \begin{cases} -\sqrt{\frac{k(N-k)}{N}} & m = k-1 \\ \sqrt{\frac{k(N-k)}{N}} & m = k \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
\tag{3.7}
$$

The most appealing property for our specific application is iii), since copy number vectors will have very few breakpoints ($K << M$) which makes $\mathbf{w}$ a sparse representation. We can prove iii) by the following argument. First, we cannot have less than $K+1$ non-zero elements because this is the minimum required to form a basis for $\mathcal{S}_{\mathcal{I}}$. Then, for all $m \notin \mathcal{I}$, we have that $x_m - x_{m-1} = 0$, and using the dual basis, $\boldsymbol{w} = \boldsymbol{F}^{-1}\boldsymbol{x}$, we have that for all $m > 0$ $w_m = 0$ if and only if $x_m - x_{m-1} = 0$. Thus, there are exactly $K+1$ non-zero elements, which is indeed the minimum (so the representation is maximally sparse). $\square$

3.3 Formulation of the breakpoint detection problem

The compact representation developed in the previous section can be used to facilitate estimating $\mathbf{x}$ from a degraded observation $\mathbf{y}$ generated as in model (3.1):

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{\epsilon} = \boldsymbol{F}\boldsymbol{w} + \boldsymbol{\epsilon}, \tag{3.8}$$

where $\boldsymbol{x}$ has been replaced by its representation in terms of the basis vectors, $\boldsymbol{F}\boldsymbol{w}$. Since the number of copy number changes is very small compared to the number of probes, $K << M$, then $\boldsymbol{x} = \boldsymbol{F}\boldsymbol{w}$ has a sparse representation in the $\mathbf{F}$ basis, while the noise $\boldsymbol{\epsilon}$ is not sparse in this representation. Under this scenario, the problem is formulated as that of finding $\hat{\boldsymbol{x}} = \boldsymbol{F}\hat{\boldsymbol{w}}$ that is closest to the observed $\boldsymbol{y}$ subject to having only K non-zero components of $\hat{\boldsymbol{w}}$.

$$\hat{\boldsymbol{w}} : \min_{\boldsymbol{w}} e\left(\boldsymbol{F}\boldsymbol{w}, \boldsymbol{y}\right) \text{ s.t. } s\left(\boldsymbol{w}\right) = K. \tag{3.9}$$

Different measures of *closeness* $e\left(.\right)$ and *sparseness* $s\left(.\right)$ can be used. For closeness, we will use the least squares error measure, since it is the most widely used for approximation and will facilitate comparison among algorithms, although it may be sensitive to outliers. For measuring sparseness we are especially interested in the l_0 norm (i.e., the number of $w_m \neq 0$), which best models the biological property that $K << M$ without imposing any restriction on the specific values of w_m.

Then, the optimization with these measures can be rewritten as follows:

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{F}\boldsymbol{w}\|_2 + \lambda \|\boldsymbol{w}\|_0 \tag{3.10}$$

where the l_p norm and the l_0 pseudo-norm are defined as:

$$\|\boldsymbol{w}\|_p = \sum_{m=1}^{M} |w_m|^p \qquad \|\boldsymbol{w}\|_{p\to 0} = \sum_{m=1}^{M} I\left(w_m \neq 0\right) \tag{3.11}$$

and with $\lambda > 0$ as a trade-off parameter between goodness of fit and sparseness.

Finding a solution for the problem of (3.10) would require solving $O(KM^2)$ least squares problems. This approach is intractable for chromosome lengths M and number of discontinuities K that are typical for our application. There exist several popular sub-optimal approaches both in the signal processing and in the statistics literature (see Table 3.1) that use a greedy search strategy or convex relaxation.

Table 3.1: Relationship between signal processing methods for overcomplete expansions and methods in statistics for variable selection in multiple regression

	Signal Processing	**Statistics**
Greedy Methods:		
MP-FS	Matching Pursuit [61]	Forward Selection [88]
OMP	Orthogonal Matching Pursuit [70]	
Relaxation methods:		
MoF-Ridge	Method of Frames [13]	Ridge regression [40]
BP-Lasso	Basis Pursuit [13]	Lasso [40]

Methods are paired when a similar version of equation (3.9) is solved (i.e., when the same metrics are chosen). But note that there will be differences in how λ is adjusted, and the size or types of design matrices $\mathbf{F}$ that are used.

The first class of strategies, greedy methods, consists of reducing the search space of all possible variable (breakpoint) subsets 2^M by assuming that the best set of K_1 variables

(breakpoints) will often be a subset of the best set of K_2 variables, for $K_2 > K_1$. If this assumption is correct, the set of best predictors can be constructed sequentially as in MP-FS; where we start selecting the vector (regressor) with largest projection (largest F-score), and keep adding the vector that most reduces the energy of the residual. This strategy is only optimal when $\mathbf{F}$ is orthogonal, or nearly optimal [20] when the coherence of $\mathbf{F}$ ($C = \max \langle \boldsymbol{f}_k, \boldsymbol{f}_j \rangle \; k \neq j$) is small and the signal is "sufficiently" sparse (i.e., $\|\boldsymbol{w}\|_0$ is small). It is important to note that this result cannot be applied to our case, since the coherence of $\mathbf{F}$ approaches 1, i.e., the set of vectors considered here is highly coherent.

The second class of strategies is based on replacing the l_0 sparseness measure $\|w\|_0$ by some other l_p measure, such that more efficient optimization methods (such as linear programming, projection or gradient methods) can be used. For example, for $p = 2$ (i.e., $\|w\|_2$), we would have a ridge regression in which the two square norms can be easily combined resulting in $\hat{\boldsymbol{w}}_{\text{rigde}} = \left(\boldsymbol{F}^t\boldsymbol{F} + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{F}^t\boldsymbol{y}$. However, $\hat{\boldsymbol{w}}_{\text{ridge}}$ is not sparse at all in l_0 sense, and thus we would be interested in using a p as small as possible. The l_1 norm is often used, because it is is the minimal one for which the constraints form a convex set and thus convex optimization or linear programming can be used to solve the problem. This is the strategy behind basis pursuit [13] and lasso [40], for which there exist a similar result as in MP [20] showing that if the coherence is small then minimizing for l_1 is equivalent to minimizing for l_0. Therefore, when $\mathbf{F}$ is highly coherent, as in our case, these techniques lead to sub-optimal performance and a new approach is needed.

In conclusion, the performance of the methods in Table 3.1 is severely limited by the high collinearity between the columns of $\mathbf{F}$ [20] (the inner product is almost 1, the maximum). On the other hand, sparse Bayesian learning (see next section) for the specific

application of CNA detection [79] can successfully exploit the collinearity structure of the PWC representation.

3.4 Sparse Bayesian Learning (SBL)

The optimization problem defined in (3.10) can be formulated from a Bayesian estimation point of view, as was done by [104], for the case where $\boldsymbol{F}$ is an arbitrary matrix, and solved using sparse Bayesian learning (SBL) [97], an empirical Bayes approach. Following [104], the problem in (3.10) can be cast as a maximum a posteriori (MAP) estimate:

$$\begin{aligned} \hat{\boldsymbol{w}}_{MAP} &= \arg\max_{\boldsymbol{w}} p\left(\boldsymbol{w}|\boldsymbol{y}\right) \\ &= \arg\max_{\boldsymbol{w}} p\left(\boldsymbol{y}|\boldsymbol{w}\right) p\left(\boldsymbol{w}\right) \\ &= \arg\min_{\boldsymbol{w}} -\log p\left(\boldsymbol{y}|\boldsymbol{w}\right) - \log p\left(\boldsymbol{w}\right) \end{aligned} \tag{3.12}$$

where the observation model $p\left(\boldsymbol{y}|\boldsymbol{w}\right)$ specifies the goodness of fit measure $e\left(.\right)$ and the prior distribution for the weights $p\left(\boldsymbol{w}\right)$ specifies the sparseness measure $s\left(.\right)$ in (3.9).

In SBL [97], the observation model is assumed normal:

$$p\left(\boldsymbol{y}|\boldsymbol{w}\right) \sim \mathcal{N}\left(\boldsymbol{F}\boldsymbol{w}, \sigma^2\boldsymbol{I}\right) \tag{3.13}$$

which leads to the mean square error as a measure of fit in (3.10). The limitations of this measure are basically the same as all the least square based methods. We will discuss robustness to extreme outliers in Section 3.8.1 and Chapter 4.

The sparseness measure in lasso/BP and Ridge-Tikhonov methods presented in Section 3.5 is equivalent to using the Laplacian and the normal distributions for the prior $p(w)$ respectively. In contrast, the prior distribution for the weights in SBL is specified as a hierarchical prior:

$$p\left(\boldsymbol{w}|\boldsymbol{\alpha}\right) = \prod_{m=1}^{M-1} \mathcal{N}\left(w_m|0,\alpha_m^{-1}\right), \tag{3.14}$$

where $\boldsymbol{\alpha}$ is a vector of hyperparameters that are distributed according to a gamma distribution:

$$p\left(\boldsymbol{\alpha}\right) = \prod_{m=1}^{M-1} \Gamma\left(\alpha_m|a,b\right). \tag{3.15}$$

The choice of $p(w)$ in convex relaxation methods is more relevant in terms of enforcing sparseness (approximating better the l_0 norm) in a computationally efficient manner than in terms of incorporating some existing prior knowledge on the actual $p(w)$ distribution. In this regard, the SBL prior has three useful features.

First, given the hyperparameters $\boldsymbol{\alpha}$, the conditional posterior weight distribution (3.16) is normal:

$$p\left(\boldsymbol{w}|\boldsymbol{y},\boldsymbol{\alpha},\sigma^2\right) = \mathcal{N}\left(\boldsymbol{w}|\boldsymbol{\mu},\boldsymbol{\Sigma}\right) \tag{3.16}$$

$$\boldsymbol{\Sigma} = \left(\sigma^{-2}\boldsymbol{F'F} + \mathrm{diag}(\boldsymbol{\alpha})\right)^{-1} \tag{3.17}$$

$$\boldsymbol{\mu} = \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{F^t y} \tag{3.18}$$

and, following [97], $p(\boldsymbol{w}|\boldsymbol{y})$ can be correctly approximated by point estimates as $p\left(\boldsymbol{w}|\boldsymbol{y}, \hat{\boldsymbol{\alpha}}, \hat{\sigma}^2\right)$; thus, the MAP estimate is given by the posterior mean $\hat{\boldsymbol{w}} = \boldsymbol{\mu}$.

Second, by treating the weights $\boldsymbol{w}$ as hidden variables, the maximum likelihood estimation for the hyperparameters $\boldsymbol{\alpha}$ can be obtained by the EM algorithm [97]. Thus, for each iteration l until convergence we would alternate:

$$E\,\text{Step}: \quad E_{\boldsymbol{w}|\boldsymbol{y},\boldsymbol{\alpha}^{(l)},\sigma^2}\left(w_m^2\right) = \Sigma_{mm} + \mu_m^2 \tag{3.19}$$

$$M\,\text{Step}: \quad \hat{\alpha}_m^{(l+1)} = \frac{1+2a}{\Sigma_{mm}+\mu_m^2+2b} \tag{3.20}$$

Finally, although this hierarchical prior does not appear to encourage sparseness, it has been demonstrated that indeed it has very good sparseness properties [97, 104]. This behavior can be unveiled by finding the marginal "effective" prior, $p(\boldsymbol{w})$, which is is an i.i.d. t-distribution with $2a$ degrees of freedom and a scale parameter of $\sqrt{a/b}$ (see Appendix A). When $b \to 0$ and a is small, this distribution peaks very sharply at 0, and has very thick flat tails that decay at (1+2a) rate in log-scale:

$$\log p(\boldsymbol{w}) \underset{b\to 0}{\to} C(a) + (1+2a)\sum_{m=0}^{M-1} \log|w_m| \tag{3.21}$$

Thus, as shown in Figure 3.3, with this prior we obtain a sparseness cost that more closely approximates the desired l_0 norm. In other words, this prior forces a very large number of weights to be 0 while the non-zero weights are free to take any value (in Figure 3.3 the sparseness penalty is almost constant for any $r > 0$), which matches well our underlying biological knowledge for copy number changes.

Although, the model contains several hyperparameters, the $\boldsymbol{\alpha}$ and σ parameters are estimated from the data while b is set to 0 (uninformative prior). Thus, sparseness is adjusted solely by the a parameter (Section 3.4.1 and Appendix A)

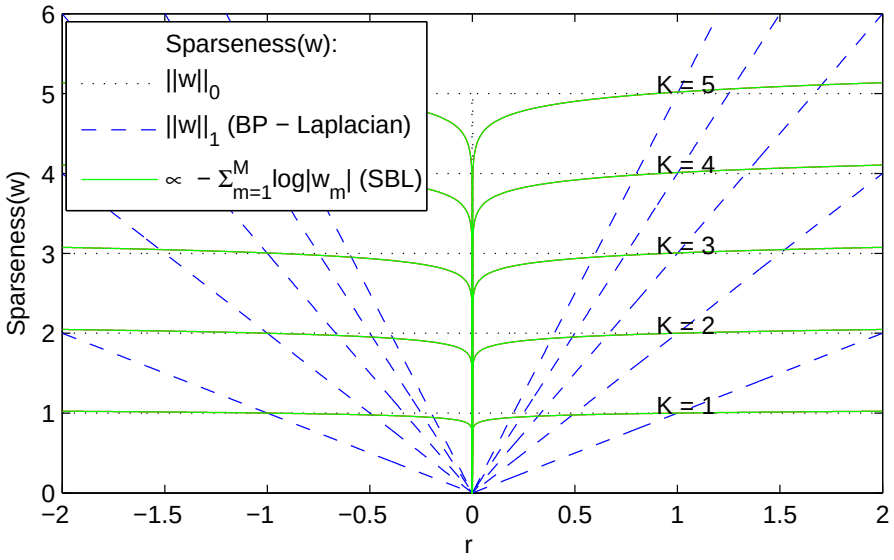


Figure 3.3: SBL and l_1 sparseness metrics compared to the desired l_0 norm (dotted line). Each curve represents the sparseness metric for an arbitrary vector w with only $K = 1, \ldots, 5$ non-zero coefficients coefficients at any position. All the non-zero weights are given the same magnitude r for different values of r on the x-axis. Ideally, we would like the sparseness metric to be inversely proportional to the the l_0 norm, which will be equal to the number of non-zero components (K) regardless of the value of the components themselves (i.e., r). Note that the SBL metric approximates better the l_0 norm, while l_1 norm deviates significantly from this ideal behavior.

3.4.1 Implementation of SBL to find copy number alterations

To the best of our knowledge this is the first time that SBL has been employed to find copy number alterations, and more specifically with the PWC representation that we propose, where $\boldsymbol{F}$ has a very special structure. One of our contributions [79] is the observation that SBL can function well in our situation where significant collinearity exists, unlike other standard methods in Section 3.3.

Additionally, SBL computational performance can be optimized for our PWC representation by exploiting the nested structure property. Direct computation of equations (3.17) and (3.20) for an arbitrary $\boldsymbol{F}$ would require $O(M^3)$ operations [97, 104]. However, for our particular $\boldsymbol{F}$ in (3.2), $\boldsymbol{H}_{\mathcal{I}} = \boldsymbol{G}_{\mathcal{I}}^{-1} = \left(\boldsymbol{F}_{\boldsymbol{\mathcal{I}}}^{\boldsymbol{t}} \boldsymbol{F}_{\boldsymbol{\mathcal{I}}}\right)^{-1}$ is, for all possible $\mathcal{I}$, a symmetric tridiagonal matrix, with main diagonal

$$\boldsymbol{h}_0(j) = \frac{(M - i_j)\, i_j}{M} \frac{(i_{j+1} - i_{j-1})}{(i_{j+1} - i_j)\,(i_j - i_{j-1})} \tag{3.22}$$

and upper/lower diagonal

$$\boldsymbol{h}_1(j) = \frac{\sqrt{(M - i_j)\, i_j\, (M - i_{j+1})\, i_{j+1}}}{M\,(i_{j+1} - i_j)} \tag{3.23}$$

This structure can be used to efficiently compute Σ_{mm} and μ_m for each EM step (3.19) in $O(M)$ steps (see lines 9-14 in Algorithm 3).

Additional computational savings are achieved through removal of columns of $\boldsymbol{F}$ that correspond to the breakpoints whose weights w are very likely to become 0 (lines 15-19 in Algorithm 3). This column removal strategy was used by [97] for the general $\boldsymbol{F}$ case, but, when combined with the tridiagonal structure exploited here, each EM step is solved more rapidly; complexity is $O(|\mathcal{I}|)$, so that the speed increases as the number of remaining breakpoints $|\mathcal{I}|$ decreases.

In our implementation, σ^2 is estimated from the data. The parameter σ^2 in the previous work [97, 104] is usually jointly estimated by the EM algorithm. However, since each chromosome in the genome is analyzed independently, and σ^2 is assumed to be the

Algorithm 3 Sparse Bayesian Learning SBL for PWC

Input: $\boldsymbol{y}, a, \sigma^2$

$\bar{\boldsymbol{y}} \leftarrow \frac{1}{M}\sum_{m=1}^{M} y_m$

$\boldsymbol{y} \leftarrow \boldsymbol{y} - \bar{\boldsymbol{y}}$ ▷ Removing $\bar{\boldsymbol{y}}$ allows us to remove $\boldsymbol{f}_0$ from $\boldsymbol{F}$

$\boldsymbol{\alpha} \leftarrow \mathbf{0}_M$ ▷ Initialize to a vector containing M zeros

$\mathcal{I} \leftarrow \{1, \ldots, M-1\}$ ▷ Initially every possible location can be a breakpoint

$[\boldsymbol{h}_0, \boldsymbol{h}_1] \leftarrow \boldsymbol{G}_{\mathcal{I}}^{-1}$ ▷ Inverse is tridiagonal symmetric with $\boldsymbol{h}_i = i$th-diagonal

$\boldsymbol{w}_0 \leftarrow \boldsymbol{F}^{-1}\boldsymbol{y}$ ▷ $\boldsymbol{F}^{-1}$ is bidiagonal

$\mathbf{z} \leftarrow \boldsymbol{F}^t\boldsymbol{y}$ ▷ Computed by solving the tridiagonal system $\boldsymbol{G}^{-1}\boldsymbol{z} = \boldsymbol{w}_0$

repeat

  $[\boldsymbol{t}_0, \boldsymbol{t}_1] \leftarrow \boldsymbol{T} = \left(\sigma^2\boldsymbol{G}_{\mathcal{I}}^{-1}\boldsymbol{\Lambda} + \boldsymbol{I}\right)$ ▷ $\boldsymbol{G}_{\mathcal{I}}^{-1} = [\boldsymbol{h}_0, \boldsymbol{h}_1]$ and $\boldsymbol{\Lambda}$ =diag(α)

  $\boldsymbol{w} \leftarrow$Solve the tridiagonal system $\boldsymbol{T}\boldsymbol{w} = \boldsymbol{w}_0$ ▷ E-Step

  Obtain diagonal of $\Sigma = \sigma^2\boldsymbol{T}^{-1}\boldsymbol{G}_{\mathcal{I}}^{-1}$ ▷ Only need tridiagonal band of $\boldsymbol{T}^{-1}$

  for $j = 1 \ldots |\mathcal{I}|$ **do**

    $\alpha_j \leftarrow \frac{1+2a}{w_j^2 + \Sigma_{jj}}$ ▷ M-Step

  end for

  (Optional reduction by removing very unlikely breakpoints)

  if $\exists i \in \mathcal{I} : \alpha_i > \tau = 1E8$ **then**

    $\mathcal{I} \leftarrow \{i \in \mathcal{I} : \alpha_i \leq \tau\}$

    $[\boldsymbol{h}_0, \boldsymbol{h}_1] \leftarrow \boldsymbol{G}_{\mathcal{I}}^{-1}$ ▷ Recompute only terms that change

    $\boldsymbol{w}_0 \leftarrow \boldsymbol{G}_{\mathcal{I}}^{-1}\boldsymbol{z}\,(\mathcal{I})$ ▷ $\boldsymbol{G}_{\mathcal{I}}^{-1}$ is tridiagonal

  end if

until $\boldsymbol{w}$ has converged ($\|\boldsymbol{w}_{old} - \boldsymbol{w}_{new}\| \leq \epsilon$)

Output: $\boldsymbol{w}_{\mathcal{I}}, \mathcal{I}$

same for all chromosomes, it is more robust to estimate σ^2 for the entire genome before applying the EM algorithm in each chromosome. In this chapter, σ^2 is estimated as

$$\hat{\sigma}^2 = \frac{1}{2M} \sum_{m=1}^{M} (y_m - y_{m-1})^2 \tag{3.24}$$

in which the difference $y_m - y_{m-1}$ removes the baseline PWC component and is distributed as $\mathcal{N}(0, 2\sigma^2)$ except for the breakpoints, which can be removed in the sum by replacing the mean by a trimmed mean. Similar estimates have also been widely employed in signal denoising approaches [21]. In the context of multiples samples, as will be discussed in Chapters 4 and 5, a probe specific variance terms could also be used.

Finally, the EM algorithm is guaranteed to improve the solution after each step and will always converge [104], but it may converge to a local minimum instead of the global minimum. However, these local minima are indeed always sparse (see Theorem 2 in [104]). The degree of sparseness in the SBL algorithm is controlled by the parameter a, as can be seen from (3.21) and Figure A.1, whereby an increase in a causes a sharper peak at zero with faster tail decay and leads to a sparser solution. The a parameter also controls the convergence rate of the EM algorithm, with larger a leading to faster convergence. However, larger values of a are not always desirable and lead to suboptimal placement of breakpoints because of rapid convergence of the EM algorithm to a local minimum. The EM local minimum problem can be corrected by checking the statistical evidence for each breakpoint after obtaining a set of breakpoints at an appropriate a level. The statistical significance test can be performed by a backward elimination procedure described in next section, which also allows more flexibility in setting the final desired degree of sparseness.

3.5 Breakpoint ranking by Backward Elimination

Not all breakpoints found by SBL have the same statistical significance since noise may make areas without any underlying alteration appear similar to those areas corresponding to actual alterations. Some breakpoints mark the separation between two long segments (i.e., such that each segment includes many probes) and are such that the difference between the estimated amplitudes of the two segments is large. Such breakpoints are more likely to correspond to true underlying changes in copy number, and therefore will have have higher statistical score $t_j = |\hat{w}_j| / \sqrt{\sigma^2 \boldsymbol{h}_0(j)}$ (see Appendix B). This score depends on the two contiguous breakpoints, and thus significance scores will change every time a breakpoint is removed (i.e. two segments are merged).

Instead of testing all the possible breakpoint combinations (i.e., segmentations), we have adopted a sub-optimal backward elimination (BE) strategy, in which we recursively eliminate the breakpoint with lowest statistical evidence t_j. Although the procedure is suboptimal, since we may eliminate breakpoints that would be more significant in a later stage, it is much less sensitive than forward selection [53]. The BE procedure can be stopped when all the remaining breakpoints have t_j higher than a specified T, the BE critical value. Moreover, with $\mathcal{I}_K$ being the breakpoint set obtained from SBL, the procedure creates a sequence of nested subsets $\mathcal{I}_1 \subset \mathcal{I}_2 \ldots \subset \mathcal{I}_K$, which are obtained backwards, and such that successive subsets differ only in one discontinuity: this directly provides a breakpoint ranking. This ranking $\mathbf{r}$ is obtained efficiently by Algorithm 4 in $O(|\mathcal{I}|)$, where we exploit the fact that removing one discontinuity at a time only affects the two neighboring breakpoints (lines 9 and 12).

Algorithm 4 starts with the set of breakpoints $\mathcal{I}$ (given by the SBL algorithm), the original array observations $\boldsymbol{y}$ and the noise variance estimate σ^2. Lines 2 and 3 find the breakpoint weights $\boldsymbol{w}_{\mathcal{I}}$ (i.e., the projection coefficients of $\boldsymbol{y}$ onto $\mathcal{S}_{\mathcal{I}}$) exploiting the structure of the PWC representation instead of using equation (B.1) directly. On the loop (lines 5-17) we sequentially remove the breakpoints with least statistical score t_j (line 6) until all breakpoints are removed. When we remove a breakpoint j^*, we do not need to compute all the weights again but only those of the left (line 9) and right (line 12) neighbors.

Algorithm 4 Breakpoint Ranking by Backward Elimination

Input: $\boldsymbol{y}, \mathcal{I}, \sigma^2$

1: Compute $\boldsymbol{H}_{\mathcal{I}}$, i.e. $[\boldsymbol{h}_0, \boldsymbol{h}_1]$, using (3.22) and (3.23)
2: $\mathbf{z} \leftarrow \boldsymbol{F}^t \boldsymbol{y}$ ▷ Computed by solving bidiagonal system $(\boldsymbol{F}^t)^{-1}\boldsymbol{z} = \boldsymbol{y}$
3: $\boldsymbol{w}_{\mathcal{I}} \leftarrow \boldsymbol{H}_{\mathcal{I}} \boldsymbol{z}\,(\mathcal{I})$ ▷ $\boldsymbol{H}_{\mathcal{I}}$ is tridiagonal
4: Compute scores $\boldsymbol{t}$, $t_j = \boldsymbol{w}_{\mathcal{I}}(j) \,/\, \sqrt{\sigma^2 \boldsymbol{h}_0(j)}$
5: **for** $k = |\mathcal{I}|, \ldots, 1$ **do**
6: $\quad j^* = \min_{i_j \in \mathcal{I}} |t_j|$ ▷ Find the least significant breakpoint for removal
7: $\quad r_k \leftarrow (i_{j^*}, t_{j^*})$ ▷ Give breakpoint the k-th rank
8: $\quad$ **if** $j^* > 1$ **then**
9: $\quad\quad \boldsymbol{w}_{\mathcal{I}}(j^*-1) \leftarrow \boldsymbol{w}_{\mathcal{I}}(j^*-1) + \sqrt{\frac{(M-i_{j^*-1})i_{j^*-1}}{(M-i_{j^*})i_{j^*}} \frac{(i_{j^*+1}-i_{j^*})}{(i_{j^*+1}-i_{j^*-1})}} \boldsymbol{w}_{\mathcal{I}}(j^*)$ ▷ Recompute left breakpoint
10: $\quad$ **end if**
11: $\quad$ **if** $j^* < |\mathcal{I}|$ **then**
12: $\quad\quad \boldsymbol{w}_{\mathcal{I}}(j^*+1) \leftarrow \boldsymbol{w}_{\mathcal{I}}(j^*+1) + \sqrt{\frac{(M-i_{j^*+1})i_{j^*+1}}{(M-i_{j^*})i_{j^*}} \frac{(i_{j^*}-i_{j^*-1})}{(i_{j^*+1}-i_{j^*-1})}} \boldsymbol{w}_{\mathcal{I}}(j^*)$ ▷ Recompute right breakpoint
13: $\quad$ **end if**
14: $\quad \mathcal{I} \leftarrow \mathcal{I} - \{i_{j^*}\}$ ▷ Remove breakpoint from the set
15: $\quad \boldsymbol{w}_{\mathcal{I}} \leftarrow \boldsymbol{w}_{\mathcal{I}}(\mathcal{I})$ ▷ Remove j^* component
16: $\quad$ Recompute $\boldsymbol{h}_0$, and $\boldsymbol{t}$ for new $\mathcal{I}$ ▷ Only j^*-1 and j^*+1 change (3.22)
17: **end for**

Output: $\boldsymbol{r}$

Finally, with the ranking of breakpoints $\mathbf{r}$, we can adjust the final breakpoint list to any critical value of T with no additional computational cost. This provides great flexibility in adjusting the final breakpoint set. The expected false discovery rate (FDR) is monotonically decreasing with T, thus we can obtain a list of breakpoints with lower FDR by increasing threshold T.

3.5.1 The role of the T parameter in BE ranking

The ranking provided by the backward elimination procedure, Algorithm 4, can be used to quickly return a breakpoint set with different degrees of sparseness that contains the breakpoints with the strongest evidence. This is done by cutting the ranking r at some specified threshold T, such that all the remaining breakpoints have a $|t_j| \geq T$. Both true positives and false positives will decrease with increasing level of sparseness (i.e., higher T) but if $P(|t_j| \geq T | w_j = 0) < P(|t_j| \geq T | w_j \neq 0)$, the expected proportion of false breakpoints on the returned set (i.e., the false discovery rate FDR) will be monotonically decreasing with T. The previous condition is true for Gaussian noise but will also be true for other symmetrically bell shaped noise distributions.

Additionally, we can associate a p-value for any particular value of t, or a significance cutoff $\alpha = P(|t_j| \geq T | w_j = 0)$ for any T, if we assume the noise is normal, using (B.9). If the noise is not Gaussian, the p-value will still be a good a approximation for the breakpoints with large flanking segments (i.e., the two neighboring breakpoints are far apart), since t will converge to a normal distribution under the null hypothesis (for any noise with zero mean, finite variance and small correlation). Alternatively, for small

segments, we could estimate the p-value by a resampling method (e.g. bootstrap [24]) or replace the t score by a non-parametric ranksum test.

It is important to notice that the aforementioned p-value is associated with a single breakpoint in one of the many possible segmentations. Thus, it does not take into account all the possible segmentations that are effectively tested during the algorithm, i.e. multiple hypothesis testing or multiple comparison problem. Commonly used tools to solve this problem (Bonferroni, Benjamini-Hochberg [8] and Benjamini-Yettukeli [9]) are not recommended here because they do not take into account the special correlation structure that exists between the t scores of overlapping or neighboring segmentations, and the independence between the t scores separated by one breakpoint or more. Solving the problem of the multiple testing in this scenario, in the sense of being able to provide a T that controls for the FDR being below some bound is still an open problem. However, since the FDR is monotonically decreasing with T, we can adjust it to achieve a particular degree of sparseness, and then estimate the FDR that corresponds to that T either using results from multiple samples, replicates or by a resampling procedure.

3.6 Segment Alteration Detection

The SBL and BE procedures are segmentation approaches that make no assumptions about the amplitude of the reconstructed segments. The objective is to provide a nearly optimal set of amplitudes and breakpoint positions that best fits the hybridization intensities observed in the array as described in (3.10). Once the breakpoints are fixed, in order to achieve the minimal residual error RSS, the amplitude corresponding to each

segment is given by the average hybridization level of all the probes that fall inside that segment. Because of this model, segments that correspond to the same underlying copy number state may be given a different reconstruction amplitude. Thus, an additional step has to be done to classify these segments into a copy number (0, 1, 2, 3, 4, ...) or alteration status (*Non-Altered*, *Gain* and *Loss*).

There are two popular alternatives to perform this additional step, since it is also required in other segmentation procedures such as DNAcopy [69] and CGHseg [73]. The first alternative, also used in smoothing and thresholding methods [43, 80], assumes or estimates a baseline *Non-Altered* mean hybridization level and classifies all the segments whose average amplitude is significantly above (below) that level as *Gain* (*Loss*), otherwise *Non-Altered* is assigned. The second alternative is the MergeLevels algorithm [103], which reduces the number of different reconstruction amplitudes by recursively merging those that are the least significantly different. The final smaller set of levels may be associated with a copy number state (0, 1, 2, 3, 4, ...).

Other CNA/CNV detection approaches, especially those that are based on HMMs automatically incorporate a classification into the different states of a hidden variable associated with each probe. However, as we discussed in Section 3.1, this may not be a good model when the number of hidden states that has been assumed does not match the true number of underlying mean hybridization levels. This is especially likely to occur when analyzing tumor samples that represent mixtures of cells with different copy number state, because cancer genomes are inherently unstable and heterogeneous [31].

3.7 GADA approach to CNA detection

We now summarize our algorithm for detecting CNA, which we call GADA, and consists of two steps. First, we apply SBL, which will provide a set of breakpoints with a specified initial level of sparseness controlled by the prior hyperparameter a. Then, the second step ranks the breakpoints provided by SBL by using a BE procedure, where the critical value T is used to establish the final degree of desired sparseness. The combination of these two approaches provides greater accuracy and flexibility.

First, this combination provides greater accuracy because each step minimizes the impact of the assumptions made by the other. SBL provides a better search strategy because effective removal of breakpoints is accomplished in several EM iterations. However, the breakpoint set detected by SBL may still include some spurious breakpoints (see Section 3.4.1). These 'false' breakpoints are then removed using the BE procedure (Section 3.5). The BE approach is greedy and fast, and it benefits from starting from a smaller set of breakpoints provided by the SBL, since fewer errors will accumulate with a smaller set (see Appendix B).

Second, it provides greater flexibility in adjusting the final breakpoint set. Both a and T can adjust sparseness in an equivalent way. We have shown that breakpoints obtained with higher sparseness settings in SBL (i.e. larger a values) tend to be subsets of those obtained with lower sparseness settings when evaluated using the same T value in BE (see Appendix C.1). Moreover, adjusting T can be done at no additional computational cost. Thus, SBL will be used with a small a, which gives a high initial sensitivity, and BE adjusts the final level of FDR.

The foreseeable usage by a practitioner of the GADA approach in detecting CNA (or CNV) would start by analyzing a large collection of microarray samples with a small initial a. This a can be obtained by analyzing a small subset of samples and/or chromosomes. However, we have found by analyzing simulated and real datasets on platforms ranging from 50K to 550K probes that $a = 0.2$ is small enough to give the necessary initial level of sensitivity (see Appendix C for more information). Following analysis of samples with SBL, the user can adjust T to obtain the final breakpoint set. A significance value $\alpha = P(|t| > T|w = 0)$ can be computed if the array noise is considered normal ($t \sim N(0, 1)$), or estimated using a resampling procedure. Any of the procedures that are typically used to control for FDR as was mentioned in Section 3.5.1 are not recommended for adjusting T because they do not take into account the dependence structure among the breakpoints. However, if replicate samples are available, the FDR can be estimated at a given T.

3.8 Simulation Results

3.8.1 Simulated CGH Data and evaluation metrics

The datasets used to compare the algorithms' rates of accuracy (sensitivity and FDR) are those proposed by [103]. To further assess these metrics in CNA occurring in genomes with differing complexities, we generated six additional simulated datasets containing 200 genomes each with 20 chromosomes. All datasets were generated in Matlab forming chromosomes of length 200 probes and sampling the CNA from the same empirical distribution used by [103], but were categorized by the number and length of CNA (see Table 3.2). These categories include: 1) no breakpoints, 2) only one breakpoint at any

position (uniformly distributed), 3-6) generated as in [103] but categorized by the number of breakpoints and the length of the altered segments.

Table 3.2: Simulated datasets categorized on the number of breakpoints and segment lengths

Category	Number of breaks	Segment lengths
1)	0	—
2)	1	Any length
3)	Few (2 to 4)	Large (10-150)
4)	Few (2 to 4)	Small (1-9)
5)	Many (5 to 10)	Large (10-150)
6)	Many (5 to 10)	Small (1-9)

All experiments consist of 200 samples with 20 chromosomes containing 200 probes. Each row represents a set of samples with different genomic complexity.

Table 3.3 shows definitions of the accuracy metrics used in the analyses of simulated data (sensitivity and FDR). A breakpoint is claimed to have been detected correctly only if it is placed within a distance of δ probes from the true breakpoint. In evaluating the performance of the algorithms, an algorithm is considered to perform better than another if 1) the algorithm's FDR is smaller with same sensitivity, or 2) if its sensitivity is higher with same FDR, or 3) if it has both lower FDR and higher sensitivity. All other cases are considered uninformative (e.g., similar FDR and sensitivity or discordant FDR and sensitivity). For each sample in a given simulated data set, the performance (FDR and sensitivity) of the algorithms was measured. The proportion of times that an algorithm performed better was obtained using only the informative cases. The two-sample test for binomial proportions (or McNemar's test) was used then to assess differences in the performance of the algorithms.

Concordance between algorithms was measured as $|\mathcal{A} \cap \mathcal{B}| \,/\, |\mathcal{A} \cup \mathcal{B}|$ [56]; where $\mathcal{A}$ and $\mathcal{B}$ are the breakpoint sets returned by each algorithm. Breakpoints belong to the

intersection (i.e., are considered to be the same), if they are separated by less than $\delta = 2$ probes.

Table 3.3: Possible outcomes for each candidate breakpoint position

Breakpoint	Not detected	Detected
Present	FN	TP
Not present	TN	FP

Performance metrics:

$$\underset{\text{or Recall}}{\text{Sensitivity}} = E\left[\frac{TP}{FN+TP}\right] \qquad \underset{\text{1-Precision}}{\text{FDR or}} = E\left[\frac{FP}{FP+TP}\right]$$

Note: A True Positive (TP) only occurs if the breakpoint that has been detected is within a distance of δ probes from a true breakpoint. If there is more than one breakpoint detected within this vicinity, only the closest one is considered TP and the remainders are False Positives (FP). The true breakpoint positions that are not detected are False Negatives (FN). The regions without a breakpoint where no breakpoints have been detected are True Negatives (TN). If the array has M probes, $M-1$ is the number of candidate breakpoints (i.e. $M - 1 = TP+FP+TN+FN$). The number of breakpoints falling in each of these categories are random numbers obtained on each simulated sample; thus expected values can be obtained for False Discovery Rate (FDR = 1- Precision) and Sensitivity (Recall) by taking the average over all the simulated samples.

3.8.2 GADA approach compared to greedy search methods

In this section we compare the GADA approach to the other popular methods in Table 3.1 that could be used with our PWC formulation. Compared to GADA, most of these existing methods are severely limited by the high collinearity/coherence between the columns of $\mathbf{F}$ (see Section 3.3).

Using the performance evaluation procedure described in Section 3.8.3 [103] the precision-recall operating curves (PROC) were generated for each approach. The sensitivity and FDR for detected CNA in the simulated CGH dataset is obtained at each operating point (Figure 3.4). SBL had the best performing PROC curve as compared to other approaches for all given values of δ (data shown only for $\delta = 2$).

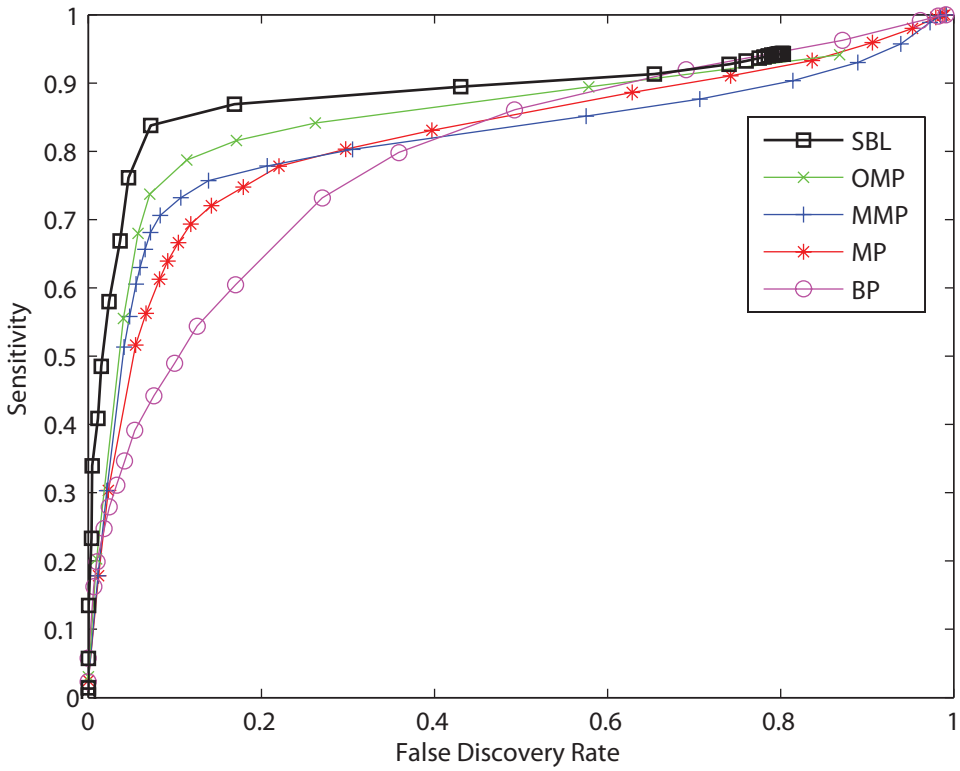


Figure 3.4: PROC operational curves for sensitivity vs. false discovery rate in detecting real copy number changes within a $\delta = 2$ sample precision window in the dataset introduced by [103].

3.8.3 GADA approach compared to other CNA detection methods

We evaluated the performance of the proposed algorithm and compared the results with other published algorithms that are publicly available; including CBS [69, 101], SWARRAY [54], HMM [29], RHMM [90], PL [25], RJaCGH [85], and GLAD [46]. We employed a simulated array-CGH dataset with known CNA positions [103], where the accuracy in detecting breakpoints was measured in terms of sensitivity and FDR as defined in Section 3.8.1.

The performance in terms of accuracy for all the analyzed algorithms (using their respective default parameters) is reported in Figure 3.5. Three of the methods, CBS,

HMM, and GLAD were previously analyzed by [103] and results are identical to those reported previously The faster new CBS [101] was also evaluated with results matching those from the previous implementation [69]. For RJaCGH, due to the long computational running time of the algorithm ($>$ 1 day), the segmentation results were obtained directly from the authors and then evaluated with the same metrics. GADA, CBS, RJaCGH and RHMM are the most accurate algorithms in terms of both sensitivity or FDR, while the remaining algorithms clearly show poorer accuracy in both metrics. Among these top four algorithms, considering the times required to analyze the entire dataset, GADA (48 seconds) is fastest, followed by CBS (625 seconds), RHMM (41 minutes) and RJaCGH ($>$ 1 day).

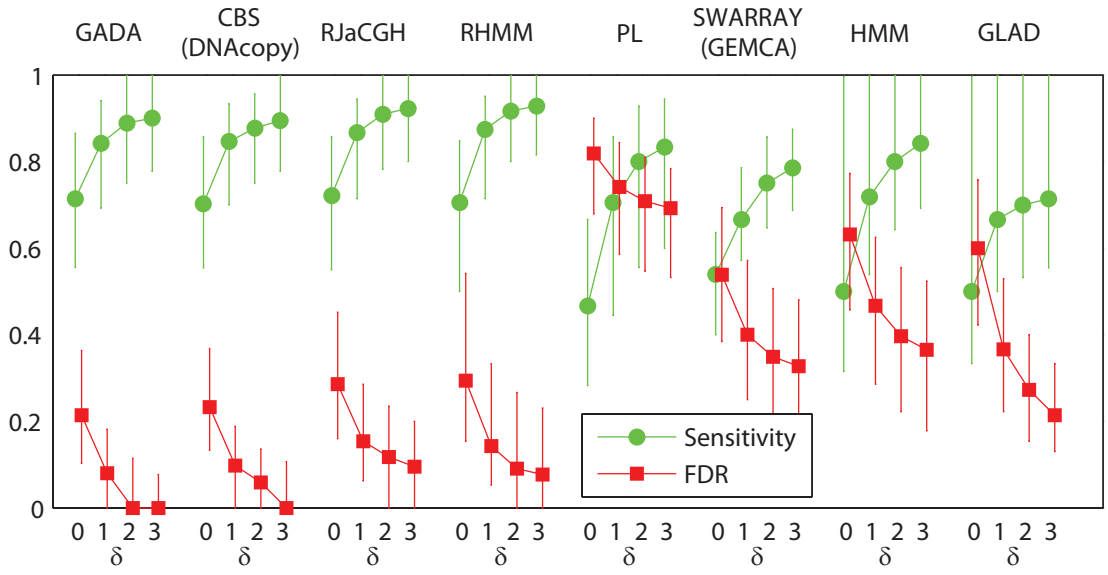


Figure 3.5: Median sensitivity and FDR for detecting known copy number changes within a probe window of δ length ($\delta = 0 - 3$). The results are obtained using the default parameters settings in each algorithm (in GADA this is a = 0.2 and T=4). The median and the interquartile range (IQR) are taken across the 500 samples.

In Figure 3.6, the parameters that control the trade-off between sensitivity and FDR are adjusted in GADA and CBS to generate the precision versus recall operation curves

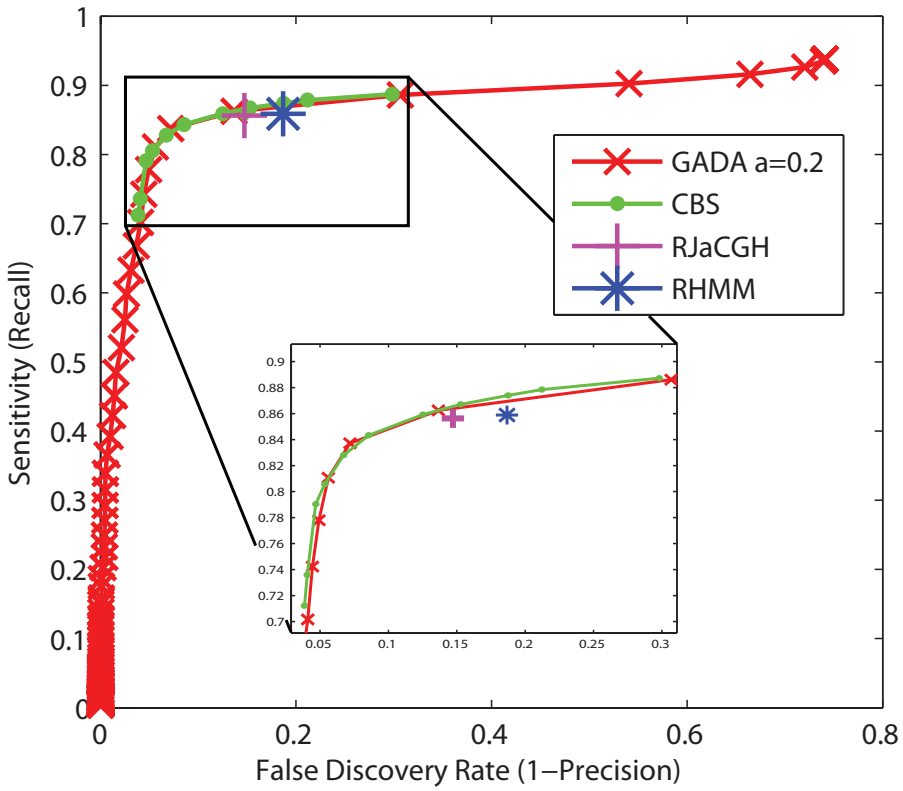


Figure 3.6: PROC operational curves for the mean sensitivity vs. FDR in detecting real copy number changes within a $\delta = 2$ sample precision window in the dataset introduced by [103] (averages taken across the 500 samples). RJaCGH and RHMM results are obtained using the default parameter settings and provide a single point. CBS operation points are obtained by varying the α, while GADA operating points were obtained by varying the T parameters with the default a=0.2.

(PROC). The single operating points generated by RJaCGH and RHMM algorithms (using their default parameters) are also shown for comparison. The results show no significant differences in performance among these four algorithms. The GADA results presented in this section are also not sensitive to different choices of the a parameter. Figure C.2 shows that essentially the same results as in Figure 3.6 are obtained for a range of a parameters. As discussed in Section 3.7, GADA is a two step procedure controlled by two parameters a and T. Setting a higher a simply makes the PROC curve

shorter (i.e., it starts further to the left and to the bottom) since all the breakpoints that would be removed by BE are instead eliminated in the SBL step. It should also be noted that RJaCGH, RHMM and PL are reported to have a better accuracy than CBS in situations different than the ones evaluated by the employed dataset, which may include: non-uniform probe spacing, chromosomes with a reduced number of probes, non uniform variance, and presence of outliers. In Chapters 4 and 5 we will study the impact of some of these situations on GADA performance, as well possible extensions to GADA in order to handle them.

In what follows we focus on comparing GADA to CBS, the baseline algorithm that most of the recent approaches use for comparison. The newer algorithms are not included in this analysis as they do not show significant improvements over CBS using the standard evaluation methods designed by [103] and have considerably slower running-times.

In Section 3.8.1 we observed that the majority of the simulated genomes by [103] have few breakpoints with large altered regions. To further assess the performance of GADA and CBS on genomes with complex patterns of CNA, typical of those observed in tumors, we generated six additional simulated datasets. These data sets contained varying complexity of CNA and were derived using the same procedure proposed by [103]. The datasets included both 'quiet' genomes (0-1 breakpoint) and complex genomes involving few or multiple breakpoints resulting in small or large CNA regions. The performances of GADA and CBS on these six datasets are provided in Table 3.4. Both algorithms work well for finding a small number of discontinuities within large segments, but there is significant evidence that GADA outperforms CBS for the complex cases. However, the magnitude of the overall differences in sensitivity and FDR between GADA and CBS is

relatively small ($< 3\%$), and the main advantage of our approach is in its flexibility and computational speed when analyzing large density arrays.

Table 3.4: Sensitivity and FDR dependence on datasets of different complexity

Compl.	FDR %			Sensitivity %			Both terms
Categ.	GADA	CBS	(<,>)	GADA	CBS	(>,<)	(GADA,CBS) p-val.
1)	0.00	0.00	(19,54)	—	—	—	—
2)	5.00	5.00	(76,62)	95.00	95.00	(43,41)	(74,52) 0.025
3)	4.04	5.61	(91,70)	95.96	97.83	(24,95)	(60,69) 0.21
4)	3.85	3.48	(84,77)	80.39	77.78	(129,30)	(94,34) 6E-8
5)	2.97	5.56	(162,30)	95.28	96.23	(50,92)	(100,30) 4E-10
6)	2.15	2.84	(119,62)	77.23	76.07	(155,38)	(114,20) 2E-16

For each of the simulated datasets of different complexity in Table 3.2 we compare the performance of GADA and the CBS algorithms. For all the cases, the GADA algorithm is set to $T = 4.0$, and CBS to $\alpha = 0.01$, since this provides comparable performance points in the PROC curves, and allows to comparison to other cases. The median sensitivity and False Discovery Rate % in breakpoint detection within 2 probes $\delta = 2$ are evaluated. The FDR and sensitivity of GADA and CBS are also compared for each sample in a given dataset and the number of times where FDR and sensitivity are smaller or larger (<,>) between the two algorithms are reported. The value pairs in parentheses are arranged so that if the left value is higher than the right one indicates a better performance of GADA. The rightmost column counts the number of times GADA is performing better than CBS either in terms of smaller FDR or higher Sensitivity (a p-value is computed as described in the Section 3.8.1).

3.9 Evaluation with microarray data

3.9.1 Neuroblastoma Genomic Data from Array Platforms

Four neuroblastoma cell lines, two with known MYCN oncogene amplification (SK-N-BE2, SMS-KAN) and two lacking MYCN amplification (LAN-6, CHLA-20) were grown in RPMI medium with 10% FCS to confluence prior to extraction of DNA using STAT60 (Tel-Test, Inc.). The same stock of DNA was used to perform whole genome analysis for CNA using Affymetrix SNP arrays 50K Xba, 250K Sty, and 250K Nsp and Illumina GoldenGate 550K SNP array based on their respective protocols. The raw data obtained

from the Affymetrix platform arrays were normalized using routines employed in Copy Number Analysis Tool version 3.0 (`www.affymetrix.com`) where log2ratios of the intensity of the probes are calculated after fitting a regression model generated from a normal set of diploid samples. The Illumina platform data were normalized and summarized using the BeadStudio Genotype analysis software and the log-R-ratio data were exported for further analyses. Data from 60 NCI cell lines generated using Affymetrix 50K Hind and 50K Xba [31] were also used to assess the computational speed of the algorithm (GEO repository accession number: GSE2520).

3.9.2 Computational speed in commercial microarray platforms

We recorded the time required to analyze using GADA and CBS copy number data generated on Affymetrix or Illumina platforms from neuroblastoma cell lines and NCI cell lines. Results are summarized in Table 3.5. The GADA algorithm was on average 100 times faster than the latest implementation of CBS. The GADA algorithm provides an additional advantage by identifying all breakpoints corresponding to all the operating points of the PROC curve within the time frames shown in Table 3.5. This allows real-time control of the final adjustment of the representation of CNA regions corresponding to different choices of the critical value T with no additional computational time. In contrast, in the current implementation of CBS, the entire procedure needs to be repeated in order to obtain sets of breakpoints at different values of the α parameter.

The computational complexity of SBL has been greatly optimized by exploiting the properties of the PWC representation as described in Section 3.4.1. The EM algorithm converges very fast, and each EM step is solved in a linear number of operations $O(M)$,

Table 3.5: Average analysis time (seconds) for Affymetrix and Illumina microarrays

	50K	100K	250K	500K	Illumina
GADA	**1.5**	**2.98**	**7.10**	**15.95**	**20.49**
CBS	197.7	444.9	597.72	1262.40	2665

Average time required to analyze the data in seconds per chip (only the time spent by the detection algorithm is counted). The 100K and the 500K columns correspond to the analysis of the combination of the two 50K (Hind/Xba) and two 250K (Nsp/Sty) chips respectively.

resulting in an overall running time that, as confirmed in Table 3.5, increases linearly with the array size M. In contrast, the computational complexity of CBS is composed of two parts; i) the circular binary segmentation optimization with $O\left(M^2\right)$ operations, and ii) the hybrid permutation test [101] that decides whether or not to proceed with the recursive segmentation with $O\left(MP\right)$ operations (P is the number of permutations). The hybrid permutation test in CBS has been improved as compared to the previous implementation [69], which required $O\left(M^2P\right)$ operations; however, the overall complexity is still limited by i), the circular segmentation taking $O\left(M^2\right)$ operations.

3.9.3 Comparison of neuroblastoma CNA detection using different array platforms

The DNA from two neuroblastoma cell lines with (SK-N-BE2, SMS-KAN) and without (CHLA-20, LAN-6) MYCN oncogene amplification were analyzed for DNA copy number alterations. Three Affymetrix genotyping arrays (50K Xba, 250K Nsp, 250K Sty) and Illumina's humanhap550 genotyping beadchip were used to generate the copy number data. A total of 105 breakpoints were identified for at least two of the platforms using the SBL algorithm and were used for further analysis (Tables 3.7, 3.8 and 3.6). Figure 3.7

shows graphical output of the algorithm on representative chromosomes where significant CNA are known to be associated with neuroblastoma.

Of the 105 breakpoints identified, 68 (65%) were identified on all platforms using GADA (Tables 3.7 and 3.8). The lowest density platform Xba, detected 78 (75%) of the 105 breakpoints, while the highest density platforms detected all (100%) the breakpoints. The detected alterations include the correct identification of the MYCN oncogene in the two cell lines with known MYCN amplification status and other common alterations found in neuroblastoma genome: loss of proximal region of 1p, gain of 17q, loss of distal region of 11q. Although the SK-N-BE2 showed copy number of two for chromosome 1p (Figure 3.7), genotype information revealed loss of heterozygosity (LOH) in this region (i.e. uniparental disomy - data not shown) with gain of 1q not reflecting any significant change in the rate of heterozygosity. There was also no gain of 17q in this cell line but there was loss of 17p and LOH for this region. Finally, we compared GADA and CBS detection performance in this real data set. The concordance rate between GADA and CBS for breakpoints that were detected by at least two platforms was 93% (Array specific concordances: Xba 97%, Nsp 90%, Sty 98%, Nsp+Sty 90%, Illumina 95%). There was also no significant difference between CBS and GADA in the distribution of distances for concordant breakpoints identified across the array platforms (Table 3.10).

3.10 Conclusions

We have introduced a new representation for genome copy number data and methodologies to detect CNA. The proposed PWC representation provides very useful properties

such as sparseness, embeddedness, and computational efficiency. This representation was exploited using a novel combination of two algorithms. The first one is based on sparse Bayesian learning (SBL), and the second one is a stepwise backward elimination (BE) procedure. Combination of these approaches results in an accurate and fast methodology, which we call GADA, to detect CNA. To the best of our knowledge, this is the first report that applies SBL to detect copy number changes or to estimate PWC representations in any application.

In simulated datasets, the GADA approach obtained the best performance in accurately detecting CNA when compared to other approaches. We have also demonstrated its applicability to two different commercial microarray platforms (Affymetrix and Illumina). The fast computational speeds obtained in analyzing these large arrays should allow further development of our algorithm in analyzing large cohorts of samples.

Although inclusion of allele specific copy number data has not been addressed in this work, the Bayesian framework in our algorithm could be extended to include the genotype data to improve placement of breakpoint positions. The genotype data and population heterozygosity frequencies could be used to jointly estimate loss of heterozygosity and allele specific copy number alterations. The advantage of such an approach is evident in our analyzed data of tumor cell lines with copy neutral LOH of chromosome 1p.

The performance of the proposed GADA approach has been studied and evaluated assuming that hybridization noise is additive white Gaussian [103]. However, real microarray probe hybridization intensities may be affected by a wide range of platform specific effects like regional trends, non-uniform variance and outliers. Normalization of the microarray probe intensities can correct or minimize the impact of some of these

effects in a pre-processing step to ensure that the data follows closely the model. Additionally, there exist several statistical tests (e.g. White test, Breusch-Pagan test or Kolmogorov-Smirnov) that could be performed on the residuals of the resulting segmentations to check for presence of the effects ignored by the model. Chapter 5 will evaluate the impact on the accuracy of GADA based on these different possible departures from the assumed model, and consider how these departures could be included in the Bayesian approach described here.

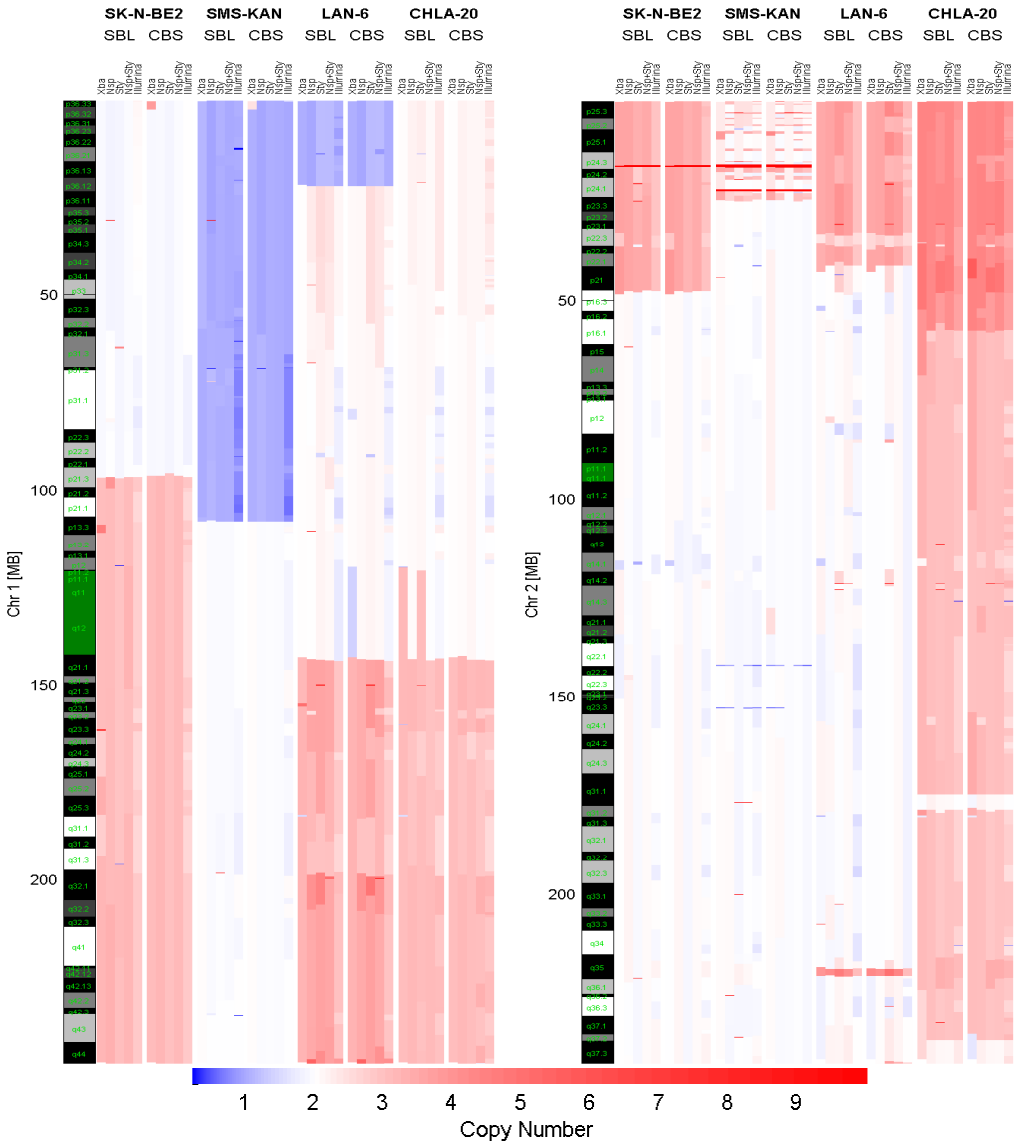


Figure 3.7: Inferred copy numbers from neuroblastoma cell-lines SK-N-BE2, SMS-KAN, LAN-6 and CHLA-20. Cell-lines were analyzed using Affymetrix's genotyping arrays 50K Xba, 250K Nsp and 250K Sty and Illumina's humanhap550 genotyping beadchip. The output of our software GADA(SBL) used the critical value of $T = 4.8$ and is compared to DNAcopy (CBS) with $\alpha = 0.01$. T was adjusted to the point where an increase on T removed concordant CNA between samples and platforms, and a decrease on T did not provide additional concordant CNA regions. Blue color tones indicate loss of genetic material, and red color tones amplification

Table 3.6: Significant copy number alterations found in four neuroblastoma cell lines

Chr.	**SK-N-BE2**	**SMS-KAN**	**LAN-6**	**CHLA-20**
1:		–(pEnd–p13.3)	–(pEnd–p36.12)	
	+(p21.3–qEnd)		+(p21.1–qEnd)	+(p21.1–qEnd)
2:	+(pEnd–p21)	+(pEnd–p24.1)	+(pEnd–p22.1)	++(pEnd–p16.1)
	++p24.3 **MYCN**	++p24.3 **MYCN**		+(p16.1–q31.1)
		++p24.1	+q35	
		–q22.1		
		–q23.3		+(q32.2–q37.2)
3:	–(pEnd–p14.2)		–(pEnd–p14.3)	
		+(p12.1–p11.1)		
4:				–(p16.1–p15.33)
		–(q12–q22.1)		–p24 **KLHL5**
		–q22.3	+(q35.1–q35.2)	+(q34.1–qEnd)
5:			–(q35.3–qEnd)	+q11.2
6:			–(q12–p16.3)	
			–(q22.31–qEnd)	
			– – q26 **PARK**	
7:			+(pEnd–p15.1)	
			–q21.1 **AHR**	+7
			–(p14.3–q11.21)	
			–(p11.21–q11.22)	–q33 **SEC8L1**
8:			–(pEnd–p12)	
			–(q22.1–q23.3)	+q21.3
	– –q24.23			+(q22.2–q24.1)
9:			–p24.2 **GLIS3**	
			–(p23–p21.2)	
			– – p21.3**MTAP**	
			–p13.3**RECK**	
10:		+(pEnd–p11.23)		
		–q22.3 **PTPRE**		
11:	+(q13.1–qEnd)	–(q14.2–q23.3)	+(q13.4–q25)	–q14.1
			– –(q25–qEnd)	+q22.1 **CNTN5**
12:			+(q23.3–q24.33)	+12
			++ q24.33	
13:			–q31.1	
14:			–(q23.2–qEnd)	
			+0(q31.3) **TTC8**	
15:				
16:		+16q	+(pEnd–p13.3) **LEP**	
17:			– –p11.2 **EPN2**	
	-(pEnd–q11.2)		-(pEnd–q11.2)	+17
			+(q21.2-qEnd)	
18:	-18			+(p11.23-qEnd)
19:		–(q13.2–q13.33)		+19p
20:	-p13			
21:				
22:				
X:	X	XX	X	XX

Table listing the most significant copy number alterations $T = 5$, that have been found on at least two of the platforms (Xba,Nsp,Sty,Nsp+Sty) being analyzed.

Table 3.7: Copy number breakpoints found on all platforms (SK-N-BE2 and SMS-KAN)

Cell-line name	Chr & Cytoband	GADA Position [BP]					CBS Position [BP]				
		Xba	Nsp	Sty	Nsp+Sty	Illumina	Xba	Nsp	Sty	Nsp+Sty	Illumina
SK-N-BE2	1p21.3	97045920	96895983	97183094	96895983	96808701	96602215	96564172	95918556	96486927	96808701
SK-N-BE2	2p24.3	15977810	15978001	15977810	15978001	15979864	15977810	15978001	15977810	15978001	15979864
SK-N-BE2	2p24.3	16419609	16453092	16462002	16462002	16463522	16419609	16453092	16462002	16462002	16465097
SK-N-BE2	2p21	48447814	47722197	48071628	47629563	47840828	48447814	47728339	48071628	47738916	47840828
SK-N-BE2	3p14.2	61381385	61301823	61423021	61159361	61312730	61227509	61301823	61137444	61241447	61312730
SK-N-BE2	8q24.23	137748993	137757306	137735555	137747078	137747933	137748993	137746403	137735555	137746403	137747933
SK-N-BE2	8q24.23	137892295	137955330	137924208	137924208	137919630	137892295	137931617	137924208	137931617	137919630
SK-N-BE2	11q13.1	64310154	64977325	65339642	65010150	65335248	64310154	64977325	65339642	65339642	65193464
SK-N-BE2	17q11.2	28109086	28263828	28164827	28283675	28247634	28109086	28263828	28164827	28266739	28214976
SK-N-BE2	20p13	2406160	2773972	3036010	3036010	2987115	2406160	2773972	3036010	3036010	2987115
SMS-KAN	1p13.3	108157301	107888773	108203626	108218567	108177825	108157301	108208349	108186644	108178227	108177825
SMS-KAN	2p24.3	15721907	15868241	15853157	15868241	15869663	15721907	15868241	15853157	15868241	15869663
SMS-KAN	2p24.3	16419609	16576876	16551640	16576876	16578889	16419609	16576876	16551640	16576876	16578889
SMS-KAN	2p24.1	21887032	21974989	22013932	22013932	21992435	21887032	21974989	22013932	22013932	21992435
SMS-KAN	2p24.1	22466771	22475341	22470539	22475341	22475673	22466771	22475341	22470539	22475341	22475673
SMS-KAN	3p12.1	83843045	84210511	84386931	83873910	84165323	85157384	84137907	84386931	83960187	84165323
SMS-KAN	3p11.1	88163152	90346746	97369003	90346746	90472437	88163152	90346746	96620438	90346746	90472437
SMS-KAN	4q12	55018889	55049094	55058835	55049094	55040244	55152302	55049094	55056941	55049094	55040244
SMS-KAN	4q22.2	94894653	94792613	94948871	94957181	94937901	94894653	94833508	94919018	94872717	94940338
SMS-KAN	10p11.23	30297356	30435753	30721148	30685964	30559838	30297356	30552253	30721148	30721148	30551022
SMS-KAN	10q26.2	129582283	129682011	129741611	129741611	129689191	129582283	129686948		129693110	129689191
SMS-KAN	10q26.2	130102560	130079710	130148252	130148252	130172807	130095140	130168048		130148252	130172807
SMS-KAN	11q14.2	85287454	85435237	85383124	85383124	85381622	85287454	85367689	85383124	85383124	85381622
SMS-KAN	11q23.3	117735474	117791205	118235879	118235879	117802601	117735474	117791205	117823148	117823148	117802601
SMS-KAN	19q13.2	45796700	45571967	45879279	45879279	45910451	45796700	45571967	45879279	45879279	45910451
SMS-KAN	19q13.33	57641156	57395071	57400799	57395071	57374324	57641156	57395071	57322747	57395071	57374324

Table listing the locations for the copy number breakpoints detected by GADA and CBS on the neuroblastoma cell-lines SK-N-BE2 and SMS-KAN that have also been found on all array platforms (Xba, Nsp, Sty, Nsp+Sty, Illumina).

Table 3.8: Copy number breakpoints found on all platforms (LAN-6, CHLA-20)

Cell-line name	Chr & Cytoband	GADA Position [BP]					CBS Position [BP]				
		Xba	Nsp	Sty	Nsp+Sty	Illumina	Xba	Nsp	Sty	Nsp+Sty	Illumina
LAN-6	1p36.33	21940846	22438012	22476248	22476248	22480144	22445846	22438012	22476248	22476248	22480144
LAN-6	1q21.1	142661525	143607676	143798352	143887291	144106312	142661525	143607676	143798352	143798352	144106312
LAN-6	2p22.1	42656869	41405461	33909326	41340562	41358977	42656869	41405461	41335143	41340562	41358977
LAN-6	2q36	218185198	218577787	218754682	218756027	218628755	218395997	218577787	218754682	218754682	218628755
LAN-6	2q36	220406369	220398375	220629809	220629809	220449867	220406369	220534849	220475305	220480390	220449867
LAN-6	3p14.3	57324905	57032227	57238434	57238434	57215650	57324905	57032227	57238434	57238434	57215650
LAN-6	6q12	68022441	68353881	68154792	68095015	68197504	68022441	68095015	67263660	68095015	68197504
LAN-6	6p16.3	106344198	105090334	105177437	105110313	105135994	105275051	105090334	105110313	105110313	105112419
LAN-6	6q22.31	123295546	123710384	123731764	123710384	123710826	123295546	123710384	123605395	123710384	123710826
LAN-6	7p15.1	27784099	24892359	24256367	24892359	24919337	27784099	31129325	25064458	24892359	24919337
LAN-6	7p14.3	31126902	31129325	31135758	31135758	31145274	31126902	31129325	31135758	31135758	31145274
LAN-6	7q11.21	62171000	62343621	63050316	63050316	62336389	62171000	62910986	62283233	62934268	62336389
LAN-6	7q11.21	63993279	63940919	63137057	63940919	63963370	63993279	63940919	65748163	63940919	63963370
LAN-6	7q11.22	69847573	69835520	68907809	69835520	69831960	69738283	69835520	69807385	69835520	69831960
LAN-6	8p12	39331612	37835460	37192319	38093651	37869447	37458281	37835460	38093651	38093651	37869447
LAN-6	8q22.1	93948610	95445117	95274323	95445117	95414304	96630283	95445117	95372632	93495648	95414304
LAN-6	8q23.3	116469762	116142633	116478611	116140376	116480735	116519714	116472717	116471437	116472717	116480735
LAN-6	9p24.2	3394434	3579281	3316679	3579281	3585674	3394434	3579281	3316679	3579281	3585674
LAN-6	9p24.2	4947650	4635935	4685068	4648449	4647040	4947650	4630212	4624827	4624827	4647040
LAN-6	9p23	12741741	12643846	13524659	12649691	12706172	12741741	12643846	12164190	12754534	12716962
LAN-6	9p21.3	21451790	21460464	21460997	21460997	21468318	21451790	21460464	21460997	21460997	21484643
LAN-6	9p21.3	22185820	22158464	22197037	22197037	22197037	22404973	22158464	22197037	22197037	22197037
LAN-6	9p21.2	28417657	28860162	28929272	28820009	28857478	28765609	28853202	28742971	28820009	28844830
LAN-6	11q13.4	71592372	71549242	71591974	71591974	71607855	71592372	71549242	71591974	71591974	71634231
LAN-6	12q23.3	105993065	106086197	106095939	106095939	106074551	105993065	106086197	106232150	106095939	106074551
LAN-6	14q23.1	60468351	60341891	60393691	60343301	60386017	60468351	60436455	60393691	60393691	60386017
LAN-6	17q11.2	25755541	24749129	24833230	24836351	24865310	25755541	24836351	24833230	24836351	24865310
LAN-6	17q21.2	36391251	35264341	36690164	35852756	35294289	36095487	35264341	36556209	36393487	35294289
LAN-6	17q22	50618719	51862430	51475956	51862430	51856911	50618719	51862430	51475956	51862430	51855630
CHLA-20	1q21.1	120089986	143607676	120928505	143887291	143328536	142661525	142756696	143798352	143657867	14802010
CHLA-20	2p16.1	68885099	57662314	55736176	57577846	57662175	57629406	57662314	58097954	57625311	57583694
CHLA-20	2q31.1	174726584	174705263	174793287	174717018	174730921	174726584	174705263	174717018	174717018	174730921
CHLA-20	2q32.2	178651035	178581699	179377095	178574520	178576047	178214924	178574520	179028748	178574520	178576047
CHLA-20	4p16.1	5722378	5842107	5913372	5842107	5844271	5722378	5842107	5913372	5842107	5842107
CHLA-20	4p15.33	12551237	12300938	12392275	12321237	12316278	12551237	12300938	12391799	12321237	12300938
CHLA-20	4q34.1	175185953	174895669	175097390	174895669	174897540	175185953	174895669	174890618	174895669	174895669
CHLA-20	8q21.3	87234127	87640754	87583206	87583206	87594384	87858742	87640368	87583206	87583206	87594384
CHLA-20	8q21.3	90473708	90386811	90407394	90407394	90366715	90138115	90372039	90407394	90376886	90366715
CHLA-20	8q22.2	100574413	99817691	99940387	99940387	99638911	99803560	99817691	99940387	99940387	99638911
CHLA-20	8q24.1	127917518	128903451	128922998	128922998	128913903	127917518	128903451	128922998	128922998	128913903
CHLA-20	18p11.23	8617957	8510927	8200848	8510927	8392640	8617957	8393289	8309751	8448484	8392640
CHLA-20	19q12	33171613	32761177	23876259	32690406	24095263	33171613	32761177	24165666	32690406	24053526

Table listing the locations for the copy number breakpoints detected by GADA and CBS on the neuroblastoma cell-lines LAN-6, CHLA-20 that have also been found on all array platforms (Xba, Nsp, Sty, Nsp+Sty, Illumina).

Table 3.9: Additional copy number breakpoints found by at least two platforms

Cell-line name	Chr & Cytoband	GADA Position [BP]					CBS Position [BP]				
		Xba	Nsp	Sty	Nsp+Sty	Illumina	Xba	Nsp	Sty	Nsp+Sty	Illumina
SMS-KAN	2q22.1	141962960	142006622		142006622	141996840	141962960	141991258		141991258	141996840
SMS-KAN	2q22.1	142284086	142419855		142419855	142418322	142284086	142419855		142386408	142418322
SMS-KAN	2q23.3	152928225	152959722		152945591	152947104	152928225	152959722			
SMS-KAN	2q23.3	153172826	153233532		153356905	153182040	153172826	153231102			
SMS-KAN	4q22.3	98081595	97971534		97971534	97946136	98081595	97971534		97971534	97946136
SMS-KAN	4q22.3	98389453	98794422		98492192	98515047	98389453	98489669		98489669	98515047
LAN-6	4q35.1		187001741	187043679	187043679	187037031		187001741	187248120	186997226	187037031
LAN-6	4q35.2		189693227	189400592	189537964	189715209		189693227	189400592	189693227	189715209
LAN-6	5q35.3		178389593	178435944	178439675	178388353		178389593	178435944	178439675	178388353
LAN-6	6q26		162768919	162672040	162783990	162769931		162768919	162770133	162770133	162769931
LAN-6	6q26		163042210	162863051	163042210	163069487		163073363	162948280	163042210	163069487
LAN-6	7p21.1	17042942	17048506		17048506	17079506					17079506
LAN-6	7p21.1	17194089	17293268		17293268	17187461					17194647
LAN-6	9p13.3		35932406		35934224	35923323					35923323
LAN-6	9p13.3		36095264		36117196	36036596					36036596
LAN-6	11q25		134393784		134408260	134410991					134410991
LAN-6	12q24.32		125117158		125475975	125319087		125257058	124609076	125131448	125319087
LAN-6	12q24.33		127723245	127722879	127723245	127723245		127723245	127835197	127723245	127723245
LAN-6	13q31.1				82988642	82996585					82996585
LAN-6	13q31.1				83045936	83063672					83055928
LAN-6	14q31.3	88300117	88381272		88443831	88310402	88300117	88381272		88443831	
LAN-6	14q31.3	88499809	88623396		88625132	88647502	88499809	88623396		88634883	
LAN-6	16p13.3		5755300	5775884	5775884	5679682		5669239	5775884	5802165	6023611
LAN-6	16q23.3			82340991	82342624	82374996		86364648	82340991	82342624	82502789
LAN-6	17		19109505		19117656	19120783				19117656	19120783
LAN-6	17		19175068		19175068	19145456				19175068	19145456
CHLA-20	2q37.2		236702079	236768287	236768287	236738515		236702079	236844697	236768287	236765691
CHLA-20	4p14		38741486		38741486	38758076		38741486		38741486	38752396
CHLA-20	4p14		38996761		38996761	39068335		38996761		38996761	39006763
CHLA-20	5q11.2		50870182		50824363	50850389		50824363		50824363	50850389
CHLA-20	5q11.2		51529692		51532772	51532772		51529692		51532772	51532772
CHLA-20	7q33		132875721		132875721	132884795		132875721		132875721	132875949
CHLA-20	7q33		133004505		133004505	132996066		133004505		133004505	132996066
CHLA-20	11q14.1		78683236	78694008	78694008	78691521			78694008	78694008	78691521
CHLA-20	11q14.1		78805883	78801531	78801531	78814667			78801531	78801531	78818346
CHLA-20	11q22.1	99371373	99366936		99366936	99378927	99371373	99240362		99366485	99378927
CHLA-20	11q22.1	100243669	100322922		100322922	100299086	100243669	100319819		100322922	100299086

Table listing the locations for the copy number breakpoints detected by GADA and CBS on the four neuroblastoma cell-lines (SK-N-BE2, SMS-KAN, LAN-6, CHLA-20) that have also been found on at two of the array platforms analyzed (Xba, Nsp, Sty, Nsp+Sty, Illumina).

Table 3.10: Differences in copy number breakpoint placing between chips

Chips compared	# cases	MAD [BP]		K-S p-value	
		GADA	CBS	GADA larger	CBS larger
i) $\min(\lvert Xba - Sty\rvert, \lvert Xba - Nsp\rvert)$:	59	95670	93132	0.54	0.76
ii) $\lvert Nsp - Sty\rvert$:	61	88024	71265	0.35	1.0
iii) $\lvert (Nsp\&Sty) - Illumina\rvert$:	91	22784	21388	0.58	0.95

For the confirmed breakpoints and excluding those near the centromere, we computed the median absolute difference in breakpoint location between chips (units in base pairs [BP]), and the p-value associated with the Kolmogorov-Smirnoff test for the hypothesis that differences are stochastically larger (i.e. less accurate) in one algorithm vs. the other. The chips compared are: i) the Xba to the Sty and Nsp separately, ii) the Nsp to Sty chips, and iii) the combined Nsp and Sty chips to the Illumina chips. No significant changes in accuracy have been found between chips of similar size on number of probes.

Chapter 4

Bayesian detection of recurrent copy number alterations across multiple array samples

Copy number changes (CNAs or CNVs) affecting small portions of chromosomes are difficult to identify. Advances in microarray technology now allow very high resolution scans of large cohorts of samples but at the price of severe noise degradation. Our proposed genome alteration detection algorithm (GADA) has been shown to be a highly accurate and efficient approach to analyze a single array sample. In this chapter, the sparse Bayesian learning (SBL) used in GADA is extended to model CNA on multiple samples that share breakpoint positions but may have different magnitude of alteration[1]. Our model is especially well suited to analyze sample replicates, i.e., multiple arrays from the same specimen. Our results show that replicates greatly improve the accuracy and robustness in detection. In some cases, a single replicate sample offers an accuracy equivalent to a 2-fold increase in the signal to noise ratio, while reducing by up to a 50% the detection of false CNA caused by outliers. The computational cost of the algorithm is essentially linear, $\mathcal{O}(NM)$, in the number of microarray probes, M, and samples, N.

[1]Parts of this work has been presented in [74]

In conclusion, the multiple sample GADA (N-GADA) presented here appears to be a promising tool for finely locating small CNAs that are shared across multiple samples.

4.1 Introduction

Recent advances in the microarray technology enabling high resolution genomic scans of large cohort of individuals have revealed presence of short copy number variation CNVs that are repeated across normal genomes (i.e., polymorphic CNAs) [82] constituting a completely new source of unstudied natural genetic variation. Small alterations are the most difficult to detect and the ones most likely to lead to false detections because of severe noise degradation. A joint analysis of many samples would undoubtedly increase the performance in detecting small CNAs, but nearly all currently available algorithms only analyze one sample at a time.

In Chapter3 (see also [75, 79]) we developed a copy number detection approach called GADA (genome alteration detection algorithm) that achieved excellent performance in single-sample CNA detection. Compared to other state-of-the-art methods, using standard evaluation datasets and benchmarks [103], GADA obtained the highest accuracy and was at least 100 times faster. GADA is based on a compact linear algebra representation of the array probe intensities as a piece-wise constant (PWC) vector and makes use of a two step detection approach. In the first step, sparse Bayesian learning (SBL [97, 104]) identifies all potentially interesting breakpoints that delimitate the CNA. The second step uses a backward elimination (BE) procedure to statistically rank the identified breakpoints, allowing a flexible control of the false discovery rate (FDR).

In this chapter we extend GADA to detect CNA across multiple samples (N-GADA). The method is especially suited to detect CNAs from sample replicates, since the underlying breakpoint locations should be the same, but the mean magnitude of the array probe measurements may be different. These differences may be due to sample contamination, different initial DNA concentrations, or other uncontrolled effects that cannot be corrected. Compared to the large number of algorithms proposed for single-sample CNA analysis, there are very few approaches dealing with the multiple sample problem [17, 60, 84, 89]. Two of them [17, 84] are post-processing techniques to refine the results obtained by a given single-sample algorithm and do not propose a joint model. The other two approaches [60, 89] propose models that only encourage overlap among CNAs across samples. In contrast, our approach is unique in the sense that it encourages recurrent breakpoint positions.

More precisely, the SBL hierarchical prior is modified in this chapter to encourage the selection of breakpoints delimiting CNA at exactly the same positions across the samples under analysis. We hypothesize that this may be a more powerful model when there is underlying evidence that the alterations start and end at recurrent positions, as is the case for sample replicates and possibly for CNA polymorphisms. In order to evaluate N-GADA we used simulation and real datasets of pairs of replicate samples with the same underlying copy number profile. The results of the new approach show that replicates greatly improve the accuracy and robustness of detection while maintaining a very good computational efficiency.

This chapter is structured as follows. The extended N-GADA approach and its implementation are presented in Section 4.2. Section 4.3 is devoted to presenting the results, and conclusions are discussed in Section 4.4.

4.2 N-GADA for multiple samples with shared breakpoints

In this section we extend the GADA approach so that it can handle multiple samples. First, we extend the SBL hierarchical prior to model sparse breakpoints occurring at similar locations across multiple samples. Then, we describe how to efficiently fit the resulting model using a expectation maximization (EM) procedure [66]. Finally, we detail the new multiple sample implementation of the BE procedure to control for the false discovery rate (FDR).

4.2.1 Sparse Bayesian Learning for multiple samples with shared breakpoints

As was shown in Section 3.4, the CNA detection can be formulated using SBL as the problem of finding the maximum a posteriori (MAP) estimate:

$$\begin{aligned} \hat{\boldsymbol{w}}_{MAP} &= \arg\max_{\boldsymbol{w}} p\left(\boldsymbol{w}|\boldsymbol{y}\right) = \arg\max_{\boldsymbol{w}} p\left(\boldsymbol{y}|\boldsymbol{w}\right) p\left(\boldsymbol{w}\right) \\ &= \arg\min_{\boldsymbol{w}} -\log p\left(\boldsymbol{y}|\boldsymbol{w}\right) - \log p\left(\boldsymbol{w}\right) \end{aligned} \tag{4.1}$$

where the observation model $p\left(\boldsymbol{y}|\boldsymbol{w}\right)$ specifies a goodness of fit measure and the prior distribution for the weights $p\left(\boldsymbol{w}\right)$ specifies the sparseness constraints. Here, we extend our previously proposed model in Section 3.4 to multiple samples. Assuming that the

noise is normal and independent across probes m and samples n, for a given underlying CNA profile for each sample, $\boldsymbol{x}^n = \boldsymbol{F}\boldsymbol{w}^n$, the observation model would be:

$$p\left(\boldsymbol{y}^1, \ldots, \boldsymbol{y}^N | \boldsymbol{w}^1, \ldots, \boldsymbol{w}^N\right) = \prod_{n=1}^{N} \mathcal{N}\left(\boldsymbol{F}\boldsymbol{w}^n, \sigma_n^2 \boldsymbol{I}\right) \tag{4.2}$$

and the prior distribution for the weights is specified as a hierarchical prior:

$$p\left(\boldsymbol{w}^1, \ldots, \boldsymbol{w}^N | \boldsymbol{\alpha}\right) = \prod_{n=1}^{N} \prod_{m=1}^{M-1} \mathcal{N}\left(w_m^n | 0, \alpha_m^{-1}\right) \tag{4.3}$$

where $\boldsymbol{\alpha}$ is a vector of hyperparameters that are distributed according to a gamma distribution:

$$p\left(\boldsymbol{\alpha}\right) = \prod_{m=1}^{M-1} \Gamma\left(\alpha_m | a, b\right). \tag{4.4}$$

Notice that here the $\boldsymbol{\alpha}$ hyperparameters are shared across multiple samples. This is in contrast to the application of SBL in 1-GADA, which implies that a different set of a hyperparameters is used for each sample. The role of the hyperparameter α_m is to control the likelihood of the presence of a breakpoint at a particular position of the genome but without imposing any restriction on the actual magnitude of the breakpoint w_m^n and its corresponding CNA.

The mathematical procedures to fit this multiple sample model and to infer the CNA breakpoints are basically the same as in 1-GADA (Chapter 3). We also use the EM algorithm, exploiting the conjugacy properties between the gamma and normal distributions, as well as the properties of our PWC representation (i.e., the matrix structure for $\boldsymbol{F}$).

The E-step is the same as before but repeated for each of the samples; i.e., finding the posterior distribution given the hyperparameters and the observation:

$$p\left(\boldsymbol{w}^n|\boldsymbol{y}^n,\boldsymbol{\alpha},\sigma_n^2\right) = \mathcal{N}\left(\boldsymbol{w}^n|\boldsymbol{\mu}^n,\boldsymbol{\Sigma}_n\right) \tag{4.5}$$

$$\boldsymbol{\Sigma}_n = \left(\sigma_n^{-2}\boldsymbol{F}^t\boldsymbol{F} + \mathrm{diag}(\boldsymbol{\alpha})\right)^{-1} \tag{4.6}$$

$$\boldsymbol{\mu}_n = \sigma_n^{-2}\boldsymbol{\Sigma}_n\boldsymbol{F}^t\boldsymbol{y}_n \tag{4.7}$$

The M-step, on the other hand, takes all the samples into account in computing the $\boldsymbol{\alpha}$ hyperparameters:

$$\hat{\alpha}_m = \frac{2a+N}{\sum_n\left(\Sigma_{mm}^n + \left(\mu_m^n\right)^2\right) + 2b} \tag{4.8}$$

The EM algorithm requires very few iterations to converge in our experiments; and all required operations in each iteration can be performed in a linear number of steps $\mathcal{O}\left(NM\right)$. This is clear for the M-step, and we already demonstrated in Chapter 3 that the operations required to compute $\boldsymbol{\mu}$ (4.7) and the diagonal of $\boldsymbol{\Sigma}$ (4.6) is $\mathcal{O}(M)$ for each sample, since we can exploit the fact that $(\boldsymbol{F}^t\boldsymbol{F})^{-1}$ is a tridiagonal matrix.

4.2.2 Backward Elimination for multiple samples with shared breakpoints

In Chapter 3 the statistical significances of breakpoints returned by SBL were ranked by a simple BE procedure using a standard linear regression model. Here, this is done within the SBL algorithm but taking into account the statistical evidence observed across multiple samples. For a single sample, both approaches are essentially equivalent; but

the new approach can exploit better the information gathered by SBL about the multiple samples (i.e., the α hyper-parameters). In this new BE procedure, after the SBL has converged for the first time to a set of breakpoints with high sensitivity, each breakpoint is statistically scored as

$$t_m = \sqrt{\sum_n \frac{{\mu_m^n}^2}{\Sigma_{mm}^n}} \tag{4.9}$$

The squared of this score can be seen as the sum across the individual squared scores (Section 3.5) for each individual sample, n, at position m. Using Appendix B the t_m^2 score also represents the total increase in the residual sum of squares (RSS) after removing breakpoint m of all samples. The lowest scoring breakpoint is recursively eliminated from the model by setting $w_m = 0$ and repeating the EM algorithm described in Section 4.2.1. Eliminating the lowest t_m increases the sparseness by removing the breakpoint that minimally increases the RSS, i.e, more likely to be a false positive (noise). Repeating the EM to re-estimate the hyperparameters α_m seems to lead to better results with the model proposed in this chapter because the α_m are shared across samples while it does not make a difference when the α_m are different across samples (Chapters 3 and 5). Finally, as in Section 3.5, the sensitivity vs. FDR trade-off is controlled by stopping the BE procedure when all the remaining breakpoints have a score higher than a critical value T.

4.3 Results

In this section we evaluate the proposed N-GADA algorithm for the case where $N = 2$ replicates are available, but results extend to other N. We employed the same artificial

dataset it is used in Section 3.8.1, which consists of 500 samples of 20 chromosomes with 100 probes where the underlying CNA are known and the noise is i.i.d. Gaussian. We generated the sample replicates using the same ground truth but with an independent new noise realization $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$, with uniformly distributed noise power $\sigma \sim U(0.1, 0.2)$ and tissue mixture $p \sim U(0.3, 0.7)$ parameters. These kind of simulations [103] may not reflect all possible scenarios, but constitute the most widely used method for quantitative evaluation.

These 2×500 samples are used to compare the performance of N-GADA to two other alternatives (Figure 4.1). The algorithms that combine both samples, i.e., 2-GADA and naive averaging, greatly improve the accuracy in breakpoint detection in comparison to the case in which no replicates are available (1-GADA). Roughly, a sample replicate would be equivalent to a two fold increase of the signal to noise ratio of a single sample. The results obtained by naive averaging are slightly worse than those of the 2-GADA approach; because the former assumes that breakpoints and segment reconstruction levels are the same while in the latter only the breakpoints are the same. On this simulation dataset, the reconstruction levels for each sample in the pair change depending on the tissue mixture parameter p.

In order to further assess the performance in terms of robustness, we randomly introduced single probe outliers (extreme values) in only one of the samples in each pair in a simulation dataset. Ideally, we would like to avoid false detections that are only supported by one of the samples. The single-sample algorithm and the one based on sample averaging cannot distinguish these outliers and nearly all of them will cause false

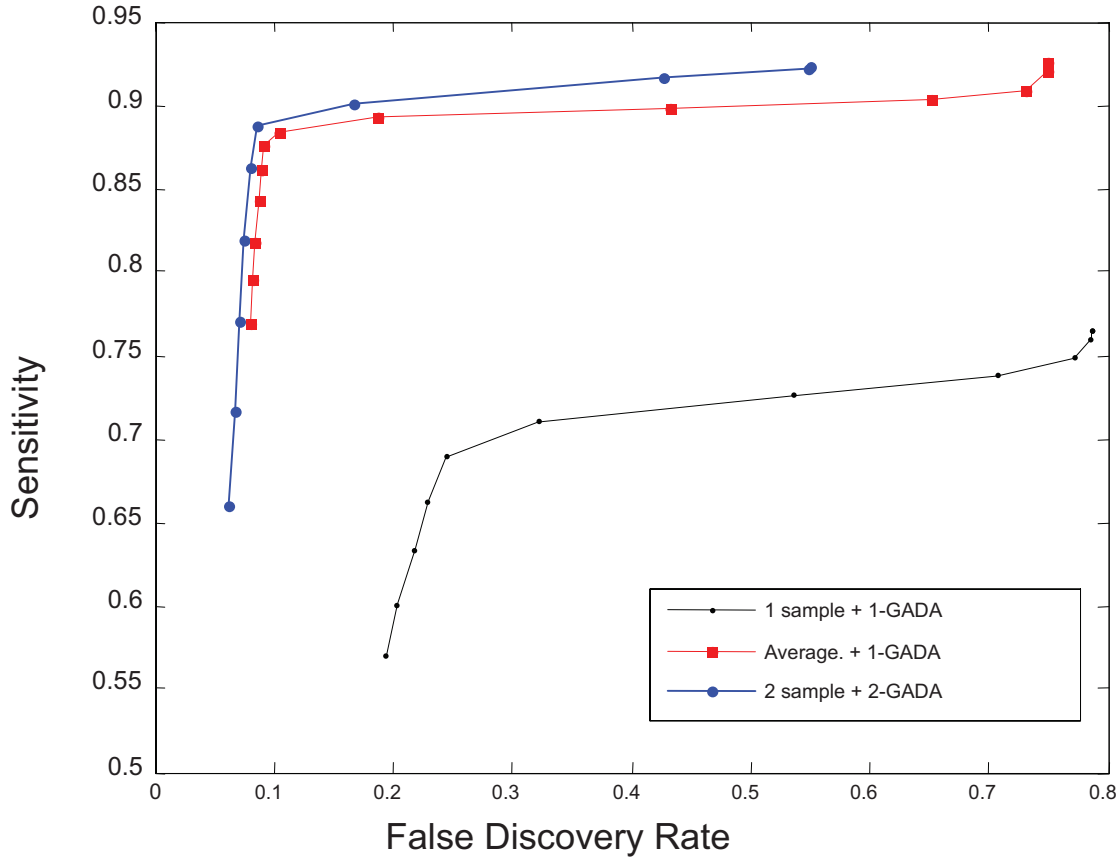


Figure 4.1: PROC operational curves for the mean sensitivity vs. FDR in detecting real copy number changes at their exact location. Black curve consist of applying 1-GADA to each of the two samples independently. Red curve combines the two samples by a weighted average into a single sample which is analyzed by 1-GADA. Blue curve is the proposed M-GADA approach. The benchmark metrics sensitivity and FDR are the same as originally defined in [103] in terms of CNA breakpoint detection.

detection. On the other hand, 2-GADA reduces false detection caused by these outliers by about 50%.

Our results on real data are also in accordance to the findings obtained using simulation data. Figure 4.2 shows a visual representation of some of the CNA detected on 3 different FDR operating points (T settings) for a pair of replicate samples (S1, S2) analyzed with Affymetrix 500K platform. The CNA found are very short segments because the samples are from a healthy human subject (NA01416). We can observe the higher

sensitivity of the 2-GADA approach on the deletion on q35; the CNA is retained for a higher significance setting $T = 7$ while it is removed on the single-sample approaches. This higher sensitivity can also be achieved by the sample averaging procedure, but this naive combination may cause more spurious false CNA (see 3rd column, $T = 4$). On the other hand, the 2-GADA approach is more robust since it retains the information of the origin of each observation. This can also be seen on an S2 outlier in q21.13 $T = 5$; 2-GADA eliminates this false alteration since it is not supported on S1, one of the two samples, while in naive averaging this outlier causes a false detection. In terms of computational speed, the 2-GADA approach performance is very competitive, with computational complexity linear in the number of probes M and samples N.

4.4 Conclusions

This chapter presents a novel approach N-GADA to solve the problem of finding CNA with breakpoints at recurrent positions across multiple samples. N-GADA extends the single-sample algorithm GADA presented in Chapter 3 using a Bayes hierarchical prior for the breakpoints that is shared across all the samples. Simulation and real data results show that the proposed approach achieves a higher accuracy and robustness to outliers when sample replicates are available. The resulting approach retains a linear complexity in the number of samples and probes. Thus, the approach can be considered a promising tool to discover small alterations that are recurrent across many samples.

In this chapter, the noise is considered independent across samples, but microarrays can also share a significant amount of noisy artifacts across samples. These array artifacts

would look similar to true recurrent CNV if not corrected. Chapter 5 will propose a method for correcting shared artifacts if multiple samples are available. This new method can be used in combination with the shared breakpoint method proposed in this chapter.

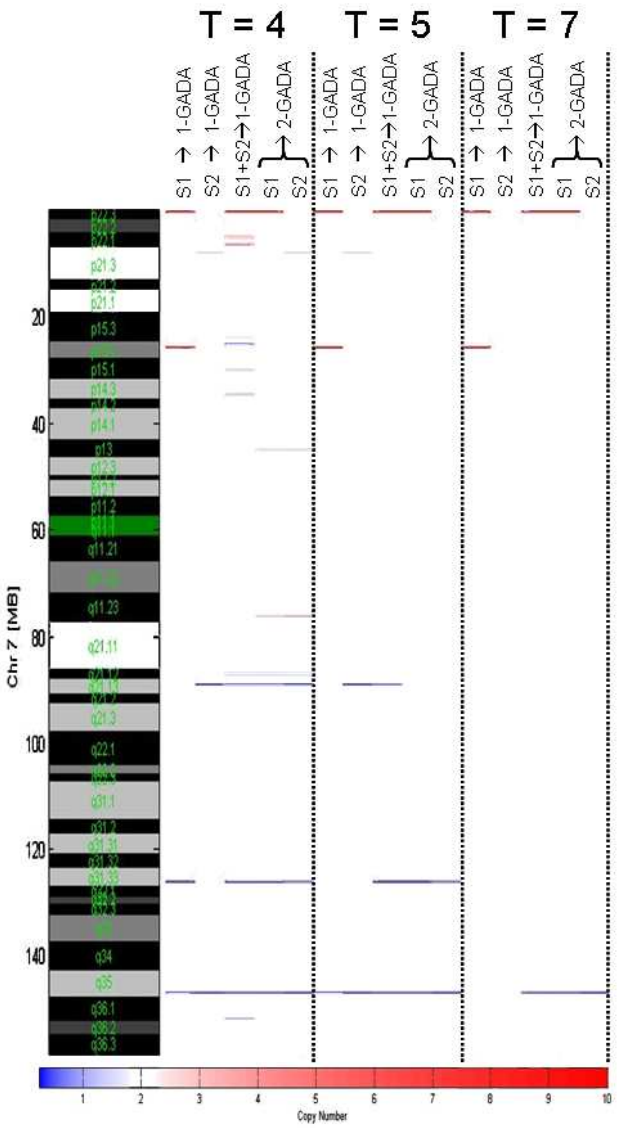


Figure 4.2: Visual representation of the detected CNA using different algorithms and settings (columns) on two replicates (S1 and S2) of a normal human sample (NA01416) analyzed using Affymetrix 500K (Nsp) platform. Columns are divided into three sections, each representing a different threshold T used for CNA detection. In each group, the first two columns correspond to the independent analysis of S1 and S2 using 1-GADA, the third column is the result of applying 1-GADA to the S1 and S2 weighted average, and the last two columns in each group (4th and 5th) are the outputs corresponding to S1 and S2 resulting of the 2-GADA joint analysis. For each claimed CNA, red tones represent amplification and blue tones loss of genetic material.

Chapter 5

Joint estimation of copy number variation and reference intensities on multiple DNA arrays using GADA

The complexity of a large number of recently discovered copy number polymorphisms is much higher than initially thought, thus making it more difficult to detect them in the presence of significant measurement noise. In this scenario, separate normalization and segmentation is prone to lead to many false detections of changes in copy number. New approaches capable of jointly modeling the copy number and the non-copy number (noise) hybridization effects across multiple samples will potentially lead to more accurate results. In this chapter, the genome alteration detection analysis (GADA) approach introduced previously is extended to a multiple sample model. The copy number component is independent for each sample and uses a sparse Bayesian prior, while the reference hybridization level is not necessarily sparse but identical on all samples. The EM algorithm used to fit the model iteratively determines whether the observed hybridization levels are more likely due to a copy number variation or to a shared hybridization bias. The new proposed approach is compared to the currently used strategy of separate normalization followed by independent segmentation of each array. Real microarray data obtained

from HapMap samples are randomly partitioned to create different reference sets. Using the new approach, copy number and reference intensity estimates are significantly less variable if the reference set changes; and a higher consistency on copy numbers detected within HapMap family trios is obtained. Finally, the running time to fit the model grows only linearly in the number samples and probes.

5.1 Introduction

The DNA copy number alterations (CNAs) introduced in Chapter 3 focused mainly the relatively large chromosomal aberrations that are commonly seen in cancer tumor cells. In recent years, when the resolution of DNA microarrays increased, it become apparent that changes in DNA copy number were also widespread across normal genomes. This natural DNA copy number variation (CNV) can be seen in many places across the genome of healthy individuals but the altered DNA segments are much shorter in size compared to those generally seen in cancer (CNAs) [26–28, 47, 82]. In this chapter we will extend the results of Chapter 3 to the detection of copy number variations (CNVs).

Currently, some known CNV regions (CNVRs) catalogued in the database of genome variants (DGV) [47] tend to be large and not very clearly delimitated regions and also miss smaller CNV (∼10Kbases) due to the lower resolution of the technologies used in the past [65,72]. Moreover, the copy number structure of some of the most highly polymorphic regions has a much higher complexity than initially thought [72]. In order to understand the role of CNVs as a genetic determinant we still require a better characterization and

delimitation of these polymorphisms along the entire genome using higher resolution arrays and more accurate detection algorithms.

There are several array platforms that can be used to measure CNV. In this work, we focus on the latest high density array platforms that use millions of short oligonucleotide probes distributed along the genome. The high resolution of these arrays is particularly well suited to detect new short CNVs and more accurately delimitate the position and structure of known CNVs. Additionally, some of these probes are also used to target the possible allelic variants of a single nucleotide polymorphism (SNP), making it possible to obtain with the same experiment the SNP and CNV genetic profile of a sample. Commercially available platforms with these characteristics include Affymetrix SNP 6.0 and Illumina Human 1M microarrays. The basic premise for copy number detection is that the hybridization intensities of probes falling under a CNV region will have higher (or lower) values than those expected on a non copy number variant (non-CNV) region. However, probe hybridization intensities also depend on other non-copy number related events, which can be regarded as experimental noise that makes CNV detection a challenging problem. Correct estimation of a *reference* hybridization intensity expected on a non-CNV probe is essential to consider that the noise has zero mean.

Existing CNV detection algorithms assume that the measurements are unbiased (i.e., the noise distribution is centered at zero) and require a separate *pre-processing* step for normalization and reference extraction (Figure 5.1A). Different normalization approaches have been proposed, including CNAT [43], dChip [107], CNAG [68], GEMCA [54], BeadStudio [71], CRMA [7] and ITALICS [83]. Examples of non-copynumber sources of variation that are targeted by some of these procedures include: allele cross-hybridization

(GEMCA, BeadStudio and CRMA) and, in Affymetrix chips, fragment length and GC content effect (CNAG, GEMCA, CRMA and ITALICS). These and other methods proposed by [18, 63] can remove probe hybridization biases that have spatial correlation (i.e. "wave-like") or correlation with the GC content or fragment length. While the spatially correlated portion of the bias can be removed by these pre-processing methods (Global Effects Normalization in Figure 5.1 A) there is still a probe specific bias due to its own binding affinity that needs to be corrected. This is usually done by taking a robust average (trimmed mean, median, clustering) across a set of *reference* samples. However, if the aim is to identify CNVs that are frequent in population, the application of median normalization on a set of reference samples, where the non-CNV regions are not known a priori, would lead to biased results. This has been already pointed out as potentially problematic by several authors [7, 54, 83], and there is no clearly defined methodology currently available to establish a good reference in regions of the genome with complex CNV patterns.

In this chapter, we propose a new model for joint estimation of CNVs and the *reference* hybridization intensity associated with the non-CNV state. This new model extends our previous work with the genome alteration detection analysis (GADA) approach (Chapter 3, [75, 79]) that achieved excellent accuracy in normalized samples. Specifically, we incorporate in the model a vector parameter for the *reference* intensities in addition to the CNV component. The copy number component, as in our previous work, is represented by a sparse Bayesian learning (SBL) prior, which favors estimates where each sample has a small number of CNV regions, but is uninformative with regard to the position and magnitude of these regions. Extending this representation, our proposed hybridization

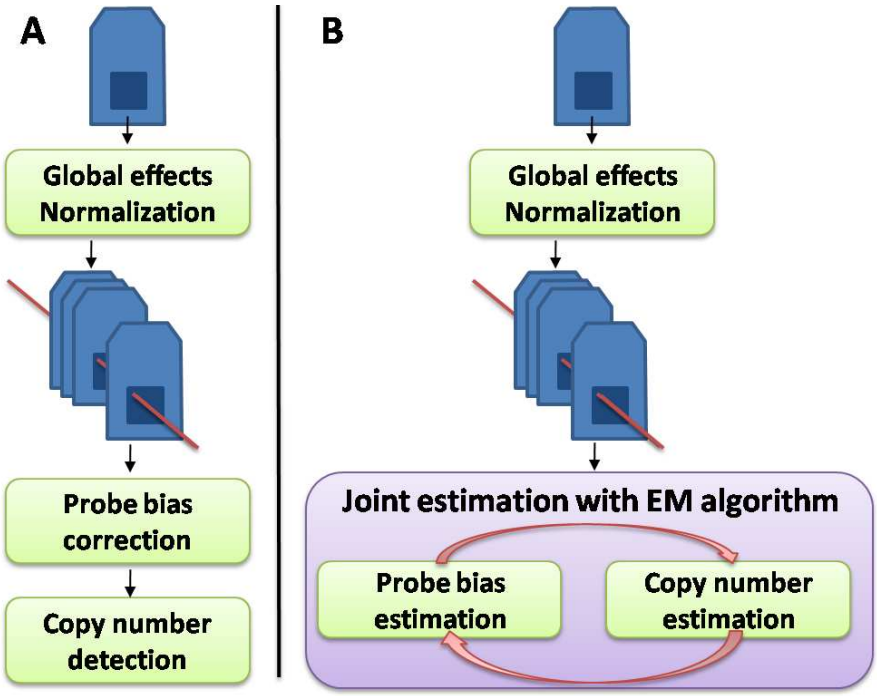


Figure 5.1: Copy number detection block diagrams: A) the typical workflow used to analyze copy number with separate pre-processing. B) the new proposed workflow using a joint estimation model for copy number variations and the probe hybridization intensities.

reference intensity has a flat prior, but the effects are shared among all samples. The EM algorithm is used to fit the model, and simultaneously estimates the *copy number* and the *reference* parameters in a given set of observed samples.

The new approach is evaluated in both simulated datasets and microarray data obtained from a pool of HapMap specimens [38] using the Affymetrix SNP 6.0 array platform. We compared the new GADA with joint reference normalization (GADA-JRN), with the currently used approach of using the median to compute the reference hybridization (GADA with separate median normalization, GADA-SMN) and, with the Affymetrix Genotyping Console GTC3.0.1. with GC correction. The presented results demonstrate that the detected CNV are significantly more consistent within the HapMap family trios.

CNV detected by the new approach are also less variable if we change the set of samples used to create the reference. Finally, the computation time to fit the new model GADA-JRN is very competitive, since the resulting algorithm (as was the case for GADA-SMN) has complexity that grows linearly with the number of samples and probes.

5.2 GADA model with separate median normalization (GADA-SMN)

In chapters 3 and 4 we assumed that there was no remaining bias after log ratio extraction (LRE); i.e.,

$$\tilde{y}_m = x_m + \epsilon_m, \tag{5.1}$$

In contrast, this chapter assumes that for a collection of microarray experiments the following holds true:

$$y_{mn} = x_{mn} + r_m + \epsilon_{mn} \tag{5.2}$$

where y_{mn} represents the $\log_2$ of the hybridization intensity observed by probe m on array n; x_{mn} represents change in hybridization intensity due to altered copy number, r_m is the *reference* probe hybridization intensity expected for a non-CNV state, and ϵ_{mn} is a zero-mean array noise.

In other words, comparing the models in (5.2) and (5.1), if the reference r_m were known, we could move it to the left side and $\tilde{y}_{mn} = y_{mn} - r_m$ would become the log-ratio

intensities with only two remaining variations: the CNV effect x_{mn}, and the zero-mean hybridization noise ϵ_{mn}. Once this effect has been removed, the n in the notation can be dropped because each sample n can be analyzed separately.

In general, the probe hybridization bias is not known and it has to be estimated and corrected. The typical approach to estimate this bias is to use the median or some other robust estimate of the mean and then perform copy number analysis independently in each sample. Throughout this chapter, GADA-SMN refers to the resulting approach of combining a separate median normalization followed by GADA (as in Chapter 3).

5.3 GADA model with joint reference normalization (GADA-JRN)

The new proposed model does not assume that the probe hybridization bias for each probe r_m in (5.2) has been removed, it is instead estimated jointly with the copy number from a large number samples. In vector form, the observation model for the log-hybridization of each probe on sample n can be rewritten as:

$$\boldsymbol{y}^n = \boldsymbol{x}^n + \boldsymbol{r} + \boldsymbol{\epsilon}^n \tag{5.3}$$

There are two basic premises that allow the joint estimation of the reference $\boldsymbol{r}$ and the copy number $\boldsymbol{x}^n$ component. First, the copy number component $\boldsymbol{x}^n$ on each sample is piecewise constant with a small number of breakpoints K and can be efficiently represented with as $\boldsymbol{x}^n = \boldsymbol{F}\boldsymbol{w}^n$. It should be noted that the number and position of those breakpoints is in general different for each sample. Second, the probe hybridization bias (or non-CNV

reference intensity) $\boldsymbol{r}$ is not necessarily PWC, but is exactly the same across multiple arrays. Our model can also be extended for cases in which the amplitude of $\boldsymbol{r}$ may change, i.e. $y_{mn} = x_{mn} + \rho_n r_m + \epsilon_{mn}$ (see Section 5.3.2).

The copy number component is modeled using and independent sparse Bayesian learning (SBL) hierarchical prior for each breakpoint m and sample n:

$$p\left(\boldsymbol{w}^1, \cdots, \boldsymbol{w}^N | \boldsymbol{\alpha}^1, \cdots, \boldsymbol{\alpha}^N\right) = \prod_{n=1}^{N} \prod_{m=1}^{M-1} \mathcal{N}\left(w_{mn} | 0, \alpha_{mn}^{-1}\right) \tag{5.4}$$

$$p\left(\boldsymbol{\alpha}^1, \cdots, \boldsymbol{\alpha}^N\right) = \prod_{n=1}^{N} \prod_{m=1}^{M-1} \Gamma\left(\alpha_{mn} | a, b\right) \tag{5.5}$$

The properties of the SBL prior are detailed in [75,97,104] and also in Chapter 3. Setting $b = 0$ and a to be small encourages a sparse number of breakpoints, but is uninformative with respect the position and magnitude of the corresponding CNV regions. Compared to the single sample model $N = 1$, this model includes an SBL prior independent for each sample n, which means that independent CNV locations can be chosen across samples. The only parameter that is shared is the hyperparameter a that controls the expected degree of sparseness (number of CNV) in each sample. The noise ϵ, as in Chapter 3, is assumed normal $p\left(\boldsymbol{\epsilon}^n\right) \sim \mathcal{N}\left(0, \sigma_n^2 \boldsymbol{I}\right)$ using a different variance parameter (with a flat prior) for each array experiment. Finally, the new $\boldsymbol{r}$ component is modeled using an uninformative flat prior.

5.3.1 Fitting the model with the EM algorithm

The model parameters are estimated by finding the maximum a posteriori (MAP) using a similar evidence maximization procedure as was introduced in Chapter 3. The EM

algorithm starts by setting α_m and r_m to zero; then, it iterates the following E and M steps:

$$E\,\text{Step}: \quad \boldsymbol{\Sigma}^n = \left(\sigma_n^{-2}\boldsymbol{F}'\boldsymbol{F} + \text{diag}(\boldsymbol{\alpha})\right)^{-1} \tag{5.6}$$

$$\boldsymbol{\mu}^n = \sigma_n^{-2}\boldsymbol{\Sigma}^n\boldsymbol{F}'(\boldsymbol{y}^n - \boldsymbol{r}) \tag{5.7}$$

$$\hat{\boldsymbol{x}}^n = \boldsymbol{F}\boldsymbol{\mu}^n \tag{5.8}$$

$$M\,\text{Step}: \quad \hat{\alpha}_m^n = \frac{1+2a}{\Sigma_{mm} + \mu_m^2 + 2b} \tag{5.9}$$

$$\hat{\sigma}_n^2 = \frac{1}{M}\left(\|\boldsymbol{y}^n - \hat{\boldsymbol{x}}^n - \boldsymbol{r}\|^2 + \sigma_n^2 \sum_m \left(1 - \boldsymbol{\Sigma}_{mm}^n \alpha_m\right)\right) \tag{5.10}$$

$$\boldsymbol{r} = \frac{1}{N}\sum_n (\boldsymbol{y}^n - \hat{\boldsymbol{x}}^n) \tag{5.11}$$

where $P\left(\boldsymbol{w}^n|\boldsymbol{y}^n, \boldsymbol{\alpha}^n, \boldsymbol{r}, \sigma_n^2\right) = \mathcal{N}\left(\boldsymbol{w}^n|\boldsymbol{\mu}^n, \boldsymbol{\Sigma}^n\right)$ exploiting the conjugacy properties between the gamma and normal distributions. The same notation as in Chapter 3 is used and the super/sub-scripts n and m are added to identify the parameters that correspond to each sample and probe respectively. For example, Σ_{mm}^n refers to the diagonal terms of the covariance matrix for the breakpoint weights $\boldsymbol{w}^n$ posterior distribution of sample n. Convergence of the model is reached with very few EM iterations; and all required operations in each iteration can be performed in a linear number of steps $\mathcal{O}(MN)$ exploiting the properties of our PWC representation (i.e., the matrix structure for $\boldsymbol{F}$, see Chapter 3 for details).

In relation to the previous GADA [75] model in Chapter 3, if the new $\boldsymbol{r}$ component was modeled by a fixed point estimate (e.g. the median across samples), then the entire model will be completely equivalent to processing each sample independently using GADA (i.e.,

GADA-SMN). This can also be seen on the EM steps, if $\boldsymbol{r}$ is a fix point estimate then (5.6 - 5.10) can be updated separately for each n. Therefore, in GADA-JRN the CNV are placed independently in each sample and the coupling is only through the estimation of $\boldsymbol{r}$.

During the EM algorithm most of the weight parameters $\hat{w}_m = \mu_m$ are driven to 0 to fulfill the sparseness constraints imposed by the hierarchical prior. Upon convergence to zero, the corresponding weight parameter and hyperparameter can be eliminated from the model and the EM algorithm can continue with a model of reduced dimensions. This elimination makes the algorithm run faster since it has to update fewer parameters (i.e., only those that are non-zero).

5.3.2 GADA-JRN model with a scale parameter for the bias

The main model proposed in this chapter assumes that the probe hybridization bias for each probe r_m in (5.2) has exactly the same magnitude across samples. However, we can adapt the model to include a different amplitude term ρ_n for each sample: In vector form, the observation model for the log-hybridization of each probe on sample n can be rewritten as:

$$\boldsymbol{y}^n = \boldsymbol{x}^n + \rho_n \boldsymbol{r} + \boldsymbol{\epsilon}^n \tag{5.12}$$

The same two basic premises that allowed the joint estimation of the reference $\boldsymbol{r}$ and the copy number $\boldsymbol{x}^n$ component also extend here with the inclusion of the ρ_n term. Notice that the probe hybridization bias (or non-CNV reference intensity) $\boldsymbol{r}$ is exactly the same

across multiple arrays but the addition of the ρ_n allows a change in the amplitude and the sign of $\boldsymbol{r}$. The copy number component and the noise is modeled exactly the same as in Section 5.3 and the new ρ_n parameters are given an uninformative flat prior.

The new resulting EM algorithm is:

E Step :

$$\boldsymbol{\Sigma}^n = \left(\sigma_n^{-2}\boldsymbol{F'F} + \mathrm{diag}(\boldsymbol{\alpha})\right)^{-1} \tag{5.13}$$

$$\boldsymbol{\mu}^n = \sigma_n^{-2}\boldsymbol{\Sigma}^n\boldsymbol{F}'(\boldsymbol{y}^n - \rho_n\boldsymbol{r}) \tag{5.14}$$

$$\hat{\boldsymbol{x}}^n = \boldsymbol{F}\boldsymbol{\mu}^n \tag{5.15}$$

M Step :

$$\hat{\alpha}_m^n = \frac{1+2a}{\Sigma_{mm} + \mu_m^2 + 2b} \tag{5.16}$$

$$\hat{\sigma}_n^2 = \frac{1}{M}\left(\|\boldsymbol{y}^n - \hat{\boldsymbol{x}}^n - \rho_n\boldsymbol{r}\|^2 + \sigma_n^2\sum_m\left(1 - \boldsymbol{\Sigma}_{mm}^n\alpha_m\right)\right) \tag{5.17}$$

$$\rho_n = \frac{\sum_m\left(y_{mn}r_m\right)}{\sum_m r_m^2} \tag{5.18}$$

$$\boldsymbol{r} = \frac{\sum_n \rho_n\left(\boldsymbol{y}^n - \hat{\boldsymbol{x}}^n\right)}{\sum_n \rho_n^2} \tag{5.19}$$

where we notice that if we fix $\rho_n = 1$ we obtain the same algorithm as before.

5.4 Backward Elimination

The sensitivity vs. false discovery rate (FDR) trade-off of our model is controlled by the hyper-parameter a. For higher values of a a more sparse solution with fewer CNVs is obtained, reducing both the FDR and the sensitivity (see Appendix A). In order

to efficiently explore solutions with different level of sparseness without having to run the algorithm all-over again, the same backward elimination (BE) strategy described in Section 3.5 is employed. Separately, for each of the samples the breakpoints with the lowest score t_m are recursively removed:

$$t_m = \sqrt{\frac{{\mu_m}^2}{\Sigma_{mm}}} \tag{5.20}$$

The sensitivity vs. FDR trade-off is controlled by stopping the procedure when all the remaining breakpoints have a score higher than a given critical value T. Therefore, as the algorithm continues to remove all the breakpoints and keeps track of which score, T, a particular breakpoint is eliminated, the breakpoints can be rapidly adjusted to any desired level based on their rank. The final result is reported as a set of segment breakpoints and amplitudes that represent the copy number variations. The parameter T is physically more informative than the parameter a because it can be interpreted as the standard error the user is willing to tolerate to call a CNV significant.

5.5 Performance metrics and evaluation methods

In order to compare the performance of the different approaches for combining normalization of the non-CNV probe reference intensities and copy number detection, the following methods and performance metrics are introduced. 270 samples from the Affymetrix dataset (Section 5.6.2) are randomly partitioned into reference sets of different sizes (10, 20, 30, 45, 70, 90 and 135 samples). In each partition, one given sample would have been grouped with a different set of samples. For example, if we create 10 random partitions

into 9 groups of size 30, each sample will have been grouped randomly with 29 different samples of the remaining 269.

The Affymetrix dataset used in this analysis was processed under ideal conditions (i.e. processed on the same day using three plates; personal communications). Thus, we expect copy number estimates $\hat{\boldsymbol{x}}_g^n$ would be very similar regardless of the subgroup chosen as the reference (see relevant results in Section 5.6.7). Although, significant changes in $\hat{\boldsymbol{r}}_g$ are not observed in this dataset, it is noteworthy that the $\hat{\boldsymbol{r}}_g$ will likely change with various laboratory conditions or across different batches. Section 5.6.7 will address possible methods to analyze samples from different batches. Variance in non-copy number and copy number estimates (V_r and V_x) across different subgroups g can be used to assess the performance, with smaller variance indicating better performance.

$$V_r = \underset{m}{\text{median}} \frac{1}{G} \sum_{g=1}^{G} (\boldsymbol{r}_g - \bar{\boldsymbol{r}}_g)^2 \tag{5.21}$$

$$V_x = \underset{n}{\text{median}} \left[\underset{m}{\text{median}} \frac{1}{G} \sum_{g=1}^{G} \left(\boldsymbol{x}_{|\boldsymbol{r}_g} - \bar{\boldsymbol{x}}_{|\boldsymbol{r}_g} \right)^2 \right] \tag{5.22}$$

Since this dataset contains 180 samples that are related in family trios, we also propose an additional measure of trio consistency. Identified CNVs in each HapMap trio are classified for each probe as in Table 5.1. Then, the *failed trio consistency rate* ($FTCR$) metric is defined as the ratio of inconsistent CNV probes in a trio among all identified CNV probes (except those considered uninformative):

$$FTCR = \frac{I}{C + I} \tag{5.23}$$

A smaller $FTCR$ value indicates that copy numbers within a family trio are more consistent. This measure ignores less frequent scenarios; e.g., if both mother and father have one chromosome with a CNV gain and the other chromosome without variation, it will still be possible (25% of the time) for the son to not inherit the CNV. In order to assess the validity of the $FTCR$ measure, the $FTCR$ scores of true trios are compared to those obtained from randomly grouping unrelated samples into trios.

Table 5.1: Consistency on HapMap trios

	Father and Mother pairs					
Offspring	(G,G)	(G,L)	(G,N)	(L,L)	(L,N)	(N,N)
Gain	C	C	C	I	I	I
Loss	I	C	I	C	C	I
Neutral	I	–	C	I	C	–

For each variation detected as (G)ain, (L)oss or (N)eutral (no variation detected) there are 18 possible CNV possible outcomes for each trio. Each outcome is labeled with 'C', 'I' and '–' indicating whether they are consistent, inconsistent or uninformative.

5.6 Results

All the samples used to evaluate the methods described in this chapter are publicly available as the Affymetrix SNP 6.0 Dataset [1]. The implementation for the new GADA method is publicly available at `http://biron.usc.edu/~piquereg/GADA`. The algorithm has been implemented in C and tested using Matlab.

5.6.1 Simulation datasets

An artificial dataset in which the underlying model is known (i.e., copy number, array artifacts and noise) was created to assess the proposed approach under specific scenarios.

Using the model in (5.2) we created an artificial microarray dataset with $N = 20$ samples and one single chromosome with $M = 10000$ probes. We generated a copy number profile x_{mn} containing two CNV regions (CNVRs) of different type (Figure 5.2A). The first one is a region with aligned CNVs (breakpoints occurring at same position across samples). The second type is a region with non-aligned CNVs. The non-CNV portion of the genome, has an expected log_2-ratio $x_{mn} = 0$. The first variation is chosen to represent a loss $x_{mn} = -1$ (copy number 1), and the second variation is chosen to have copy number 3 ($x_{mn} = log_2(3) - 1 = 0.58$). The reference, or systematic bias to remove, is chosen to be a sinusoid $r_m = sin(2\pi 0.001m)$, which represents a wavy effect similar to what has been observed in some actual array experiments [63]. The noise introduced in the dataset is white i.i.d Gaussian $\epsilon_{mn} \sim \mathcal{N}(0, 1)$. The major difference of this simulation model as compared to others [57, 103] is the inclusion of the probe hybridization bias effect $r_m = 0.5sin(2\pi 0.001m) + \mathcal{N}(0, 0.25)$ with two main components: i) a sinusoidal wave with spatial correlation similar to what has been observed in some actual array experiments [63], and ii) a noise wave without spatial correlation simulating each probe specific affinity. The proposed methods have also been evaluated with other simulation scenarios which include:

1. "Wave" only bias shared on all the samples $r_m = 0.5sin(2\pi 0.001m)$ (Figure 5.3 Top).

2. "Non-Wave" probe specific bias shared on all the samples $r_m \sim \mathcal{N}(0,1)$ (Figure 5.3 Bottom).

3. "Non-Wave+Wave" with a different amplitude in each sample, (Figure 5.10).

4. "Non-Wave" bias different in two subgroups (batch effect). $r_m^1 \sim \mathcal{N}(0,1)$ independent of $r_m^2 \sim \mathcal{N}(0,1)$ (Figure 5.14).

5.6.2 Affymetrix SNP 6.0 data set description and normalization

The Affymetrix dataset contains 270 samples from the International HapMap Project consisting of 30 CEPH trios (CEU), 30 Yoruban trios (YRI), 45 unrelated Han Chinese samples (CHB) and 45 unrelated Japanese samples (JPT). The Affymetrix Genome-Wide Human SNP Array 6.0 integrates about 1.9 million probesets (931,946 SNPs, 946,000 CN).

The preprocessing software packages that are available for the Affymetrix SNP Array 6.0 platform include: the Affymetrix Genotyping Console 3.0 (GTC3) with a normalization similar to CNAG [68], dChip with invariant set normalization [86], and Aroma.Affymetrix which implements the CRMA [7]. The CRMA method was chosen, since it has been shown to be more robust and accurate than other methods ([7]). The Aroma.Affymetrix package performs the following 4 correction steps: 1) allelic crosstalk calibration, *ACC*, for SNP probes; 2) probe level modeling *PLM*, which gives a single signal for the SNP probes; 3) fragment length normalization *FLN*, which corrects the differences in the PCR reaction due to the length and GC content of the fragment; and finally, 4) log ratio extraction, *LRE*, which calculates the log hybridization intensity relative to the expected diploid signal reference intensity using the median. In Figure 5.1A, the steps ACC, PLM

and FLN are grouped together as *global effects normalization*, and LRE is the *probe bias correction*.

In this chapter, we argue that while steps 1-3 may be performed safely during preprocessing, finding the non-CNV reference intensity is problematic in regions rich in copy number polymorphisms. In normal samples, most of the probes ($> 90\%$) will fall on regions without CNV which implies that the normalization model parameters which have a global effect on all probes can be safely estimated using robust strategies as those employed by CRMA [7]. On the other hand, in any given region of the genome containing a highly polymorphic CNV it is not known a priori which samples do not have a CNV. Ideally, this CNV effect should be removed before estimating the probe hybridization intensity associated with the non-CNV state. In this work, we will extract the corrected probe intensities after steps 1-3, and use our new proposed model (GADA-JRN) to jointly estimate the reference and the copy number component (Figure 5.1B). Results from Affymetrix GTC software with GC correction are also obtained for comparison.

5.6.3 Results with simulated data

The artificially generated data (Section 5.6.1) illustrates a scenario in which there are two relatively high frequency CNVs (Figure 5.2 A) with a bias on the hybridization measurement from the array experiment (Figure 5.2 B). If the probe hybridization bias $\boldsymbol{r}$ is not removed from the data, the results will be contaminated with a large number of spurious segments not associated with true CNVs (Figure 5.2 C). While other approaches [7,18,63] can correct the "smooth-wave" (GC correlated) part of the bias (see next section), GADA-JRN can also correct the non-smooth (uncorrelated) probe specific bias. The currently

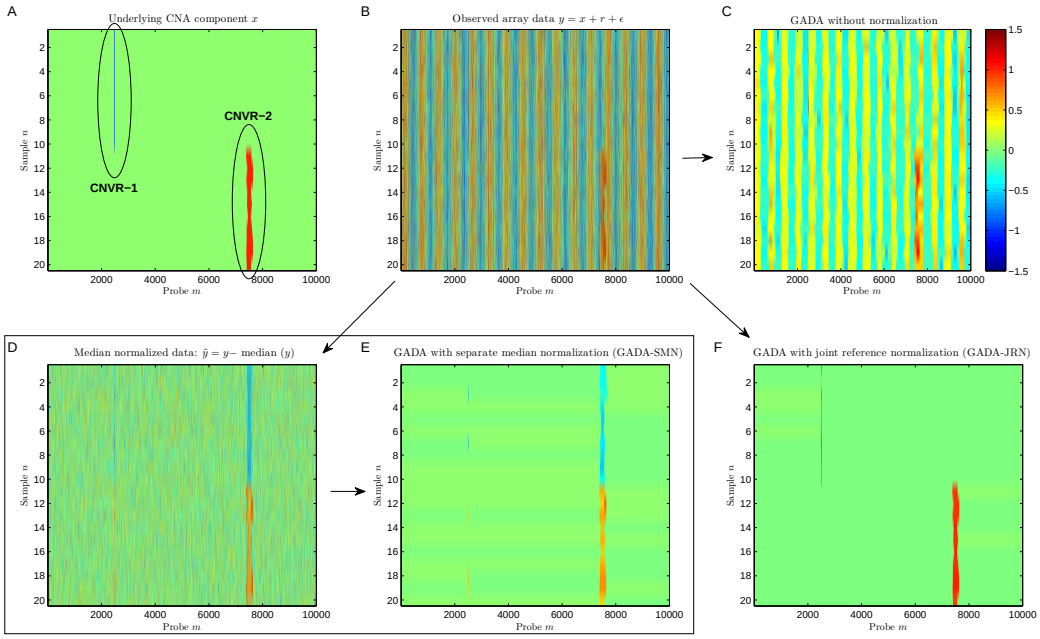


Figure 5.2: Illustration of the observation model. Colors represent the observed hybridization intensities and the relative copy number change (blue - loss '-1', red - gain '+1', green - neutral '0'). A) The true underlying CNV component with two CNV regions (CNVR-1 around m=2500 and CNVR-2 around m=7500). B) Simulated array hybridization intensities degraded by noise $\boldsymbol{\epsilon}^n$ and a systematic measurement bias $\boldsymbol{r}$. C) Copy number profile using GADA on non normalized data. D) Data after reference subtraction estimated by separate median pre-processing (SMN). E) Copy number profile using GADA with separate median normalization (GADA-SMN). F) Copy number profile estimated using GADA with joint reference normalization (GADA-JRN).

used approach of separate pre-processing is based on estimating r_m as the median across a set of reference samples (here the simulated samples themselves) before extracting the CNVs. This can eliminate r_m in the areas of the genome without CNVs ($x_{mn} = 0$ regions); but it is problematic in CNV regions containing a large amount of CNV across samples (Figure 5.2 D and E). The new joint reference normalization approach (GADA-JRN in Figure 5.2 F) can correctly delimitate the CNV on the samples with CNV on the region CNVR-2, while the separate median normalization (GADA-SMN in Figure 5.2 E) incorrectly reports CNV on samples $n = 1, \ldots, 10$. Additionally, GADA-JRN can better detect the small CNV on CNVR-1, in which the separate median normalization tends

to make the amplitude of the variation smaller and thus more difficult to detect. These results are not affected if r_m has different types of wave as illustrated with more examples in Figure 5.3.

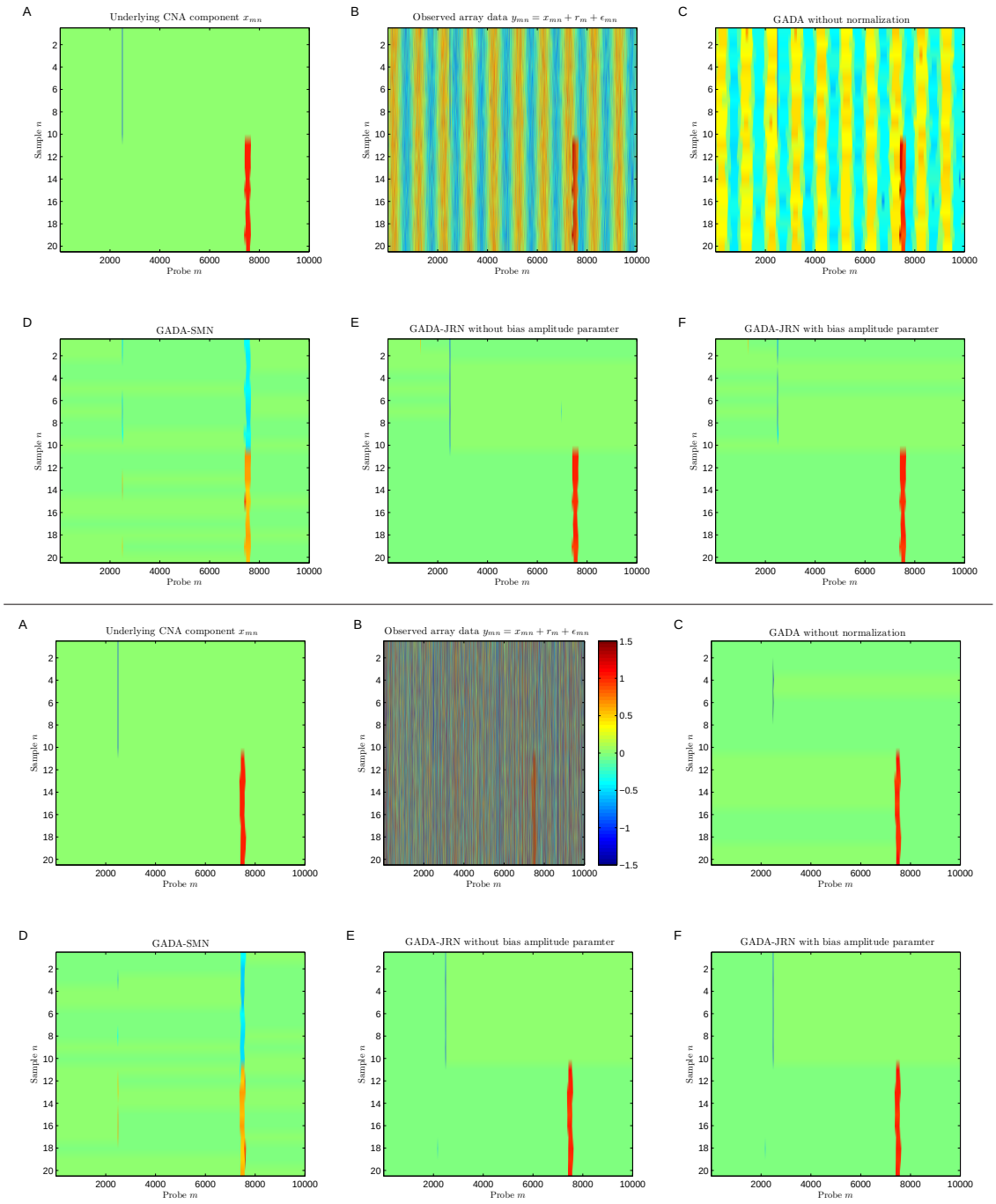


Figure 5.3: Simulation model with measurement bias of only one type: Top $r_m = 0.5sin(2\pi 0.001m)$, Bottom $r_m \sim \mathcal{N}(0, 1.0)$. The colors represent the observed hybridization intensity and the copy number (blue - loss, red - gain, green - neutral). A) The true underlying CNV component with two CNV regions (CNVR-1 around m=2500 and CNVR-2 around m=7500). B) Simulated array hybridization intensities degraded by noise $\boldsymbol{\epsilon}^n$ and a systematic measurement bias $\boldsymbol{r}$. C) Copy number profile using GADA on non normalized data. D) Copy number profile using GADA with separate median normalization (GADA-SMN). E) and F) Copy number profile estimated using GADA-JRN with or without a scaling parameter ρ_n for the bias.

5.6.4 Results with Affymetrix microarray data

The hybridization intensities are obtained after using ACC and applying FLN corrections from Aroma.Affymetrix, on the 270 HapMap samples analyzed with Affymetrix SNP 6.0 arrays. This pre-processing step appropriately scales and centers the data removing the spatially correlated part of the bias. We next compare GADA-JRN and GADA-SMN employing the evaluation methods introduced in Section 5.5.

Using the randomly chosen reference sets of different size, we evaluated the variability V_r and V_x in the reference intensities and the copy number estimates. Figure 5.4 shows that, as the number of samples in the reference set increases, the variability on CNV estimates V_x decreases. More importantly, we can see that using GADA-JRN achieves a considerably better performance when compared to GADA-SMN. In some cases GADA-JRN requires about half of the samples in order to obtain estimates of similar accuracy to those achieved with GADA-SMN. In terms of variance of the reference intensity estimates, V_r, GADA-JRN also achieves a significantly smaller values ($p < 1E-7$ Kolmogorov-Smirnov test, data not shown).

This improvement in performance can also be observed using the trio consistency measure ($FTCR$) described in Section 5.5. In Figure 5.5, the trio consistency improves (i.e., $FTCR$ decreases) with the size of the reference set, and GADA-JRN also achieves significantly better consistency. The results are also similar with change on the sparseness parameters a and T that set the trade-off between sensitivity and FDR. Figure 5.6 illustrates for one of the reference sets (90 CEU samples) the consistency that is obtained for different settings of the parameter T, which controls the backward elimination (BE)

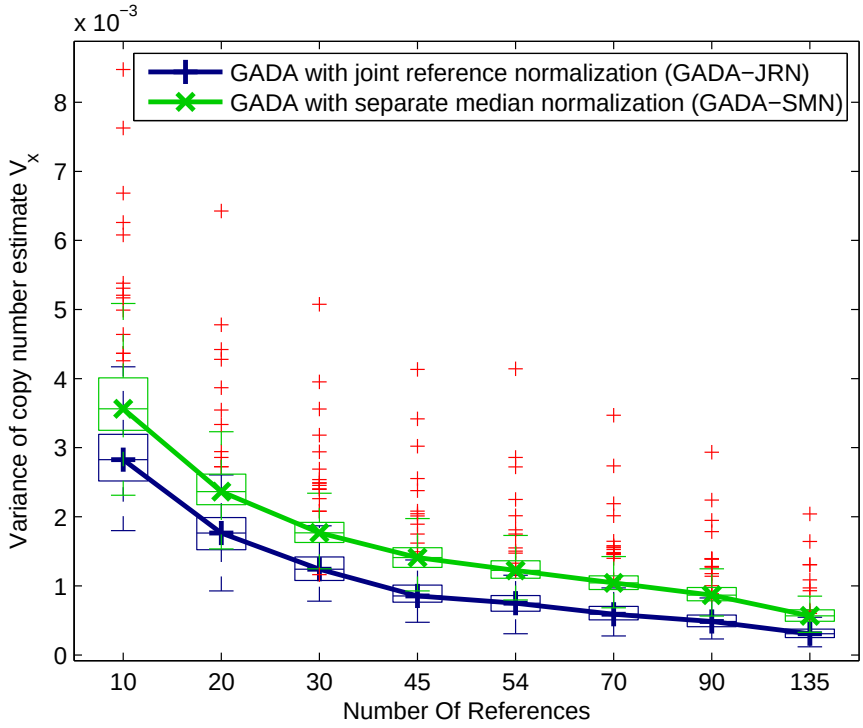


Figure 5.4: Variability on the copy number estimates if the set of reference samples changes.

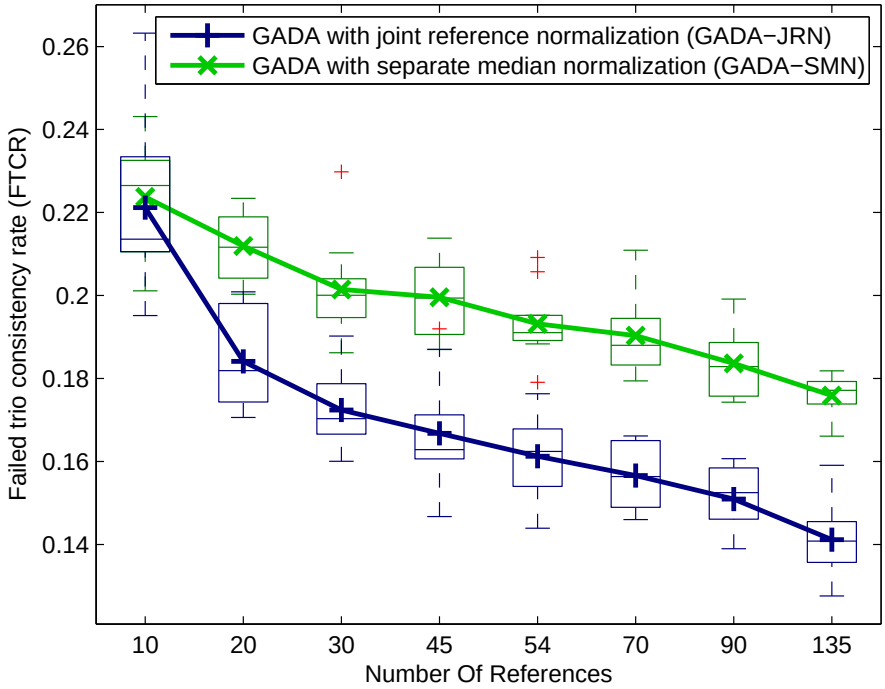


Figure 5.5: Consistency of the copy number estimates on HapMap Trios if the set of reference samples changes.

step. The $FTCR$ measure improves (decreases) with increasing T, since the number of detected false CNVs (i.e. FDR) decreases; but for higher values of T (e.g., $T > 8$) true CNVs may also fail to be detected on the offspring (i.e. lower sensitivity), and thus $FTCR$ stops decreasing. The $FTCR$ obtained on randomly formed trios of unrelated samples assures that this metric is actually capturing the increase on shared CNVs within a family ($p < 0.01$) and can be used to compare different normalization and copy number detection approaches. In Table 5.2 GADA-JRN obtains a better $FTCR$ (16.7%) than GADA-SMN (19.5%) and GTC (19.45%) when using 90 reference samples. On a larger reference set of 180 samples, GADA-SMN ($FTCR = 18.3\%$) and GTC ($FTCR = 17.31\%$) improve but GADA-JRN still retains a better $FTCR$ (16.5%). Overall, the new approach is especially indicated for small reference sets and for regions with highly polymorphic CNVs. Figure 5.7 and 5.8 shows the copy number estimates obtained on an already known highly polymorphic region of chromosome 17. The predicted gains and losses of GADA-JRN are retained when the reference set of 90 reference (CEU) samples is enlarged to include 180 (CEU+YRI) samples. On the other hand, GADA-SMN is less consistent in delimiting the CNV boundaries.

Table 5.2: Comparison on HapMap trio consistency $FTCR$

	Number of Reference Samples			
	90 CEU		180 CEU+YRI	
	FTCR	# CNVs	FTCR	# CNVs
GADA-JRN	**16.7%**	85 (T=10)	**16.5%**	127 (T=9.0)
GADA-SMN	19.2%	94 (T=9.0)	18.3%	125 (T=8.0)
GTC	18.45%	86	17.31%	127

Table with the Failed Trio Consistency Rate ($FTCR$) and the median number of CNVs per sample when using three different algorithms GADA-JRN, GADA-SMN, and GTC. The threshold values for GADA-JRN and GADA-SMN were chosen to obtain approximately equal number of CNVs among the algorithms.

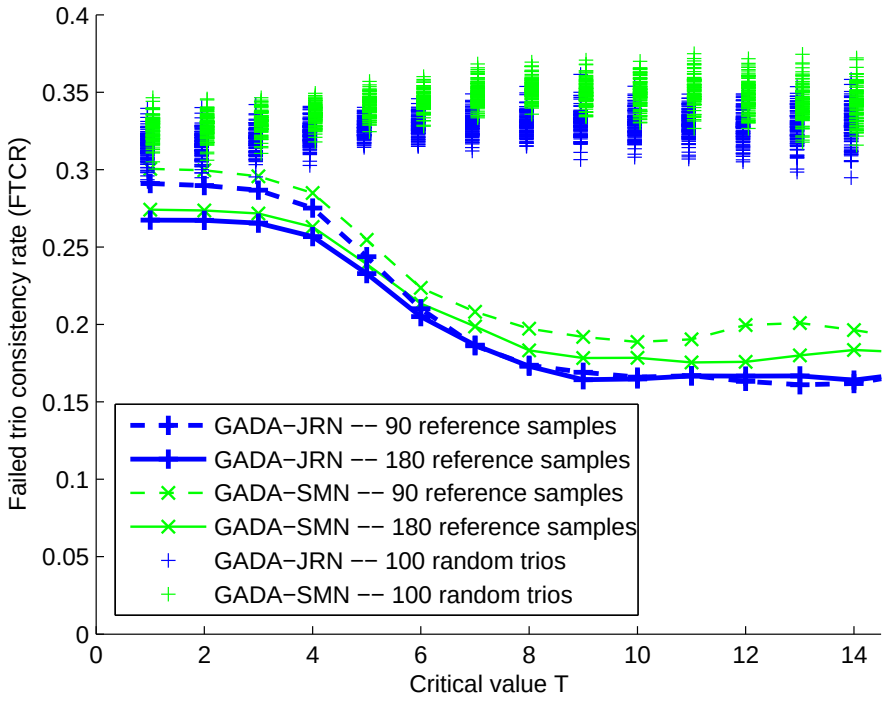


Figure 5.6: Consistency within HapMap trios using a different sparseness setting T. The dashed and solid lines correspond to a 90 (CEU) and 180 (CEU+YRI) sample reference set respectively. The cloud of points are the $FTCR$ values obtained from 100 randomly formed trios. The $FTCR$ values of GADA-JRN (blue) are smaller than those of GADA-SMN (green).

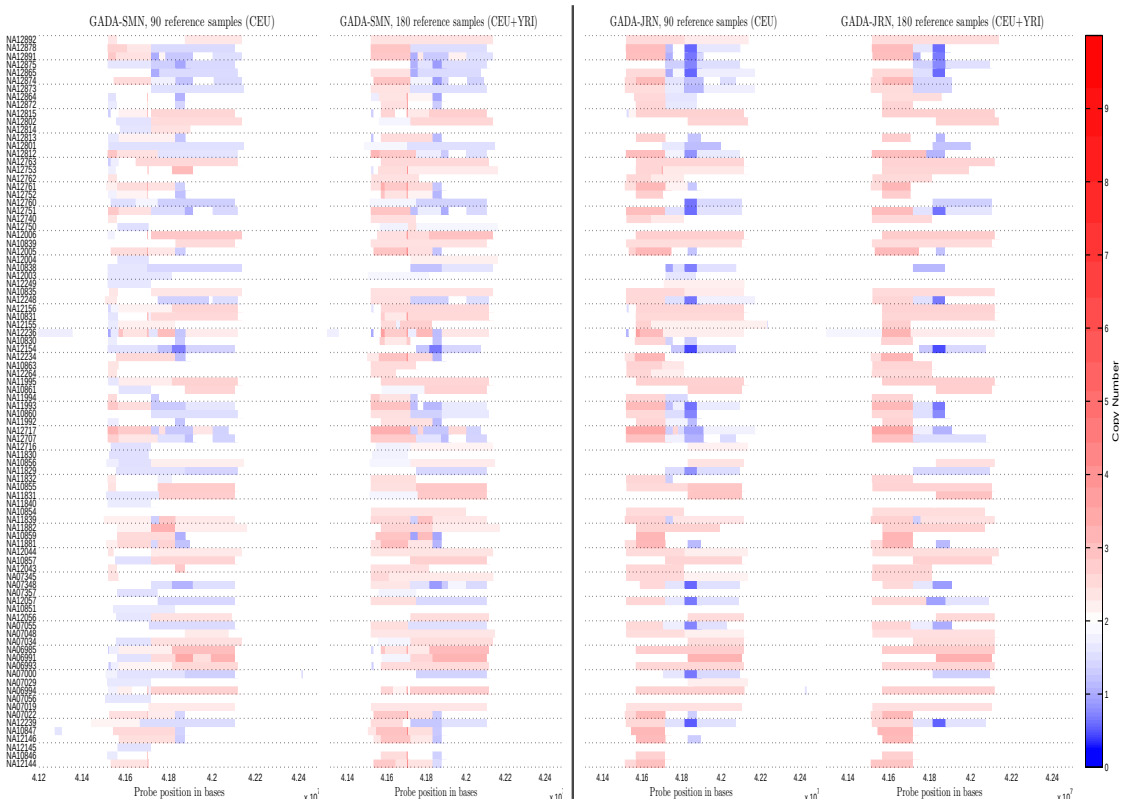


Figure 5.7: Section of the chromosome 17 that contains an already known CNV. Each row corresponds to one of the 90 CEU HapMap samples and are grouped in trios (father, son/daughter, mother) delimited by horizontal dotted lines. On the left of the thick vertical line are shown the CNVs estimated using GADA with separate median normalization using a reference set of 90 and 180 reference samples. On the right, copy number estimated using GADA with joint reference normalization show a higher consistency when the reference set is changed as well as within HapMap trios.

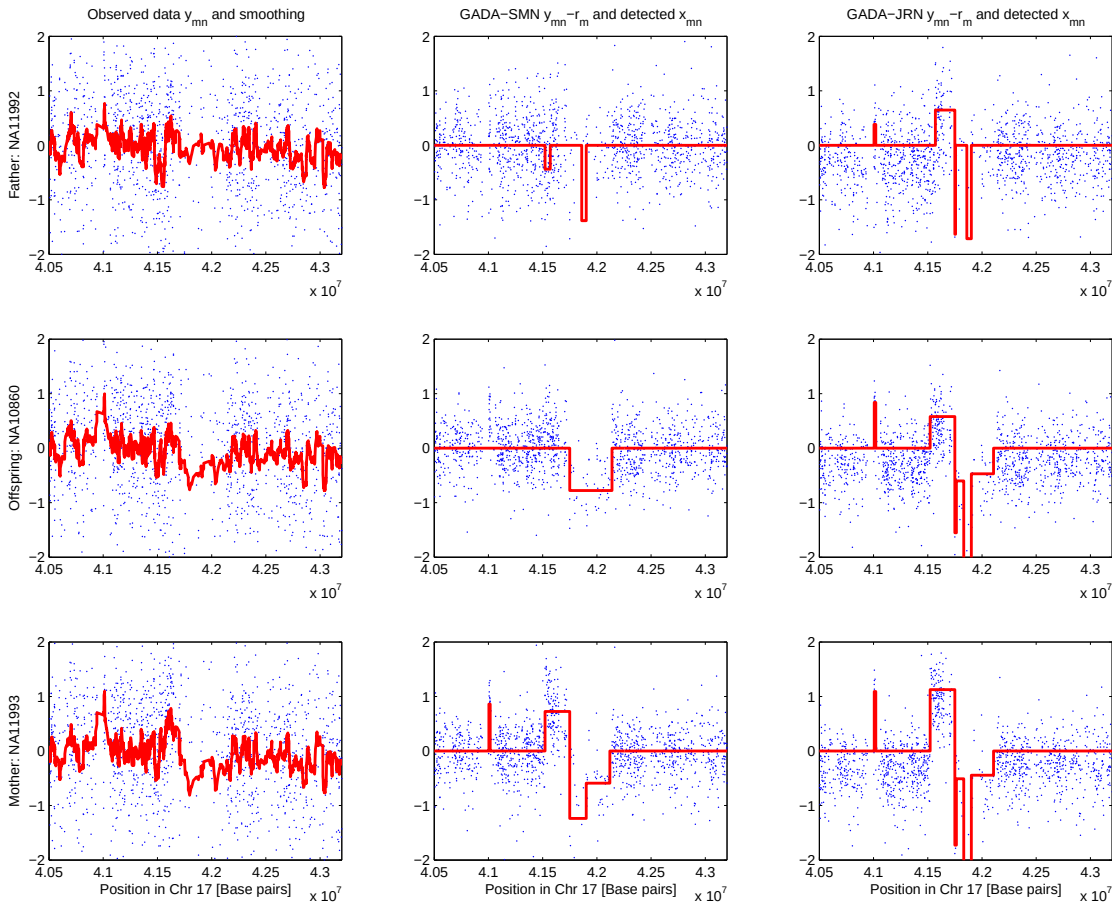


Figure 5.8: Example of a complex copy number section of Chr. 17 within a HapMap trio (Top row, father; Center row, son/daughter; Bottom row, mother). Left column shows the original observed array values and a smoothed version that shows that there is a hybridization bias correlated along the samples. Center column shows the median normalized intensities and the detected altered segments in red (GADA-SMN). Right column shows the resulting intensities corrected with the reference estimated by GADA-JRN and the corresponding detected segments in red. This result visually depicts that the detected alterations are more consistent within trios with the new GADA-JRN model.

Finally the computational time required for fitting the model is longer on the new approach, but still retains a very competitive linear complexity in the number of probes and samples (see Figure 5.9).

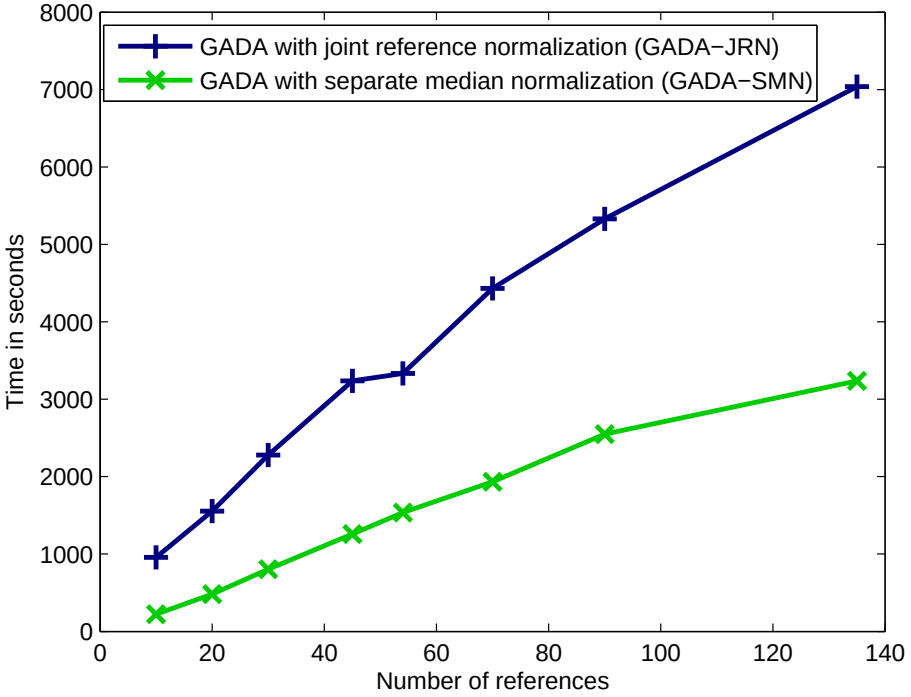


Figure 5.9: Computational time required to fit the models GADA-JRN and GADA-SMN. The time required to fit the model is linear on the number of samples for both approaches. Execution times required to process the models are measured on the same machine.

5.6.5 Simulation results with a scale effect

In order to evaluate this model which includes a scale parameter ρ_n, a new simulated dataset is created. The same underlying copy number profile $\boldsymbol{x}^n$ is used (Figure 5.10 A) but the wave r that is added to each sample is multiplied by a random amplitude $\rho_n \sim U(-1, 1)$ (uniformly distributed between -1.0 and 1.0). The bias wave used in this experiment is $r_m = 0.5sin(2\pi 0.001m) + \mathcal{N}(0, 0.25)$ and has two main ingredients: i) a

sinusoidal wave with spatial correlation, ii) a noise wave without spatial correlation. The resulting observed array intensities $y_m n$ have three components (Figure 5.10 B): a) the copy number component which is piecewise constant independent for each sample, b) the wave that is correlated across samples n and possibly across probes m, and c) the noise uncorrelated across samples n and m probes with variance 0.2.

The new joint reference normalization approach (GADA-JRN in Figure 5.10 F) can correctly delimitate the CNVs on the samples and extract the probe hybridization bias and its magnitude from each sample.

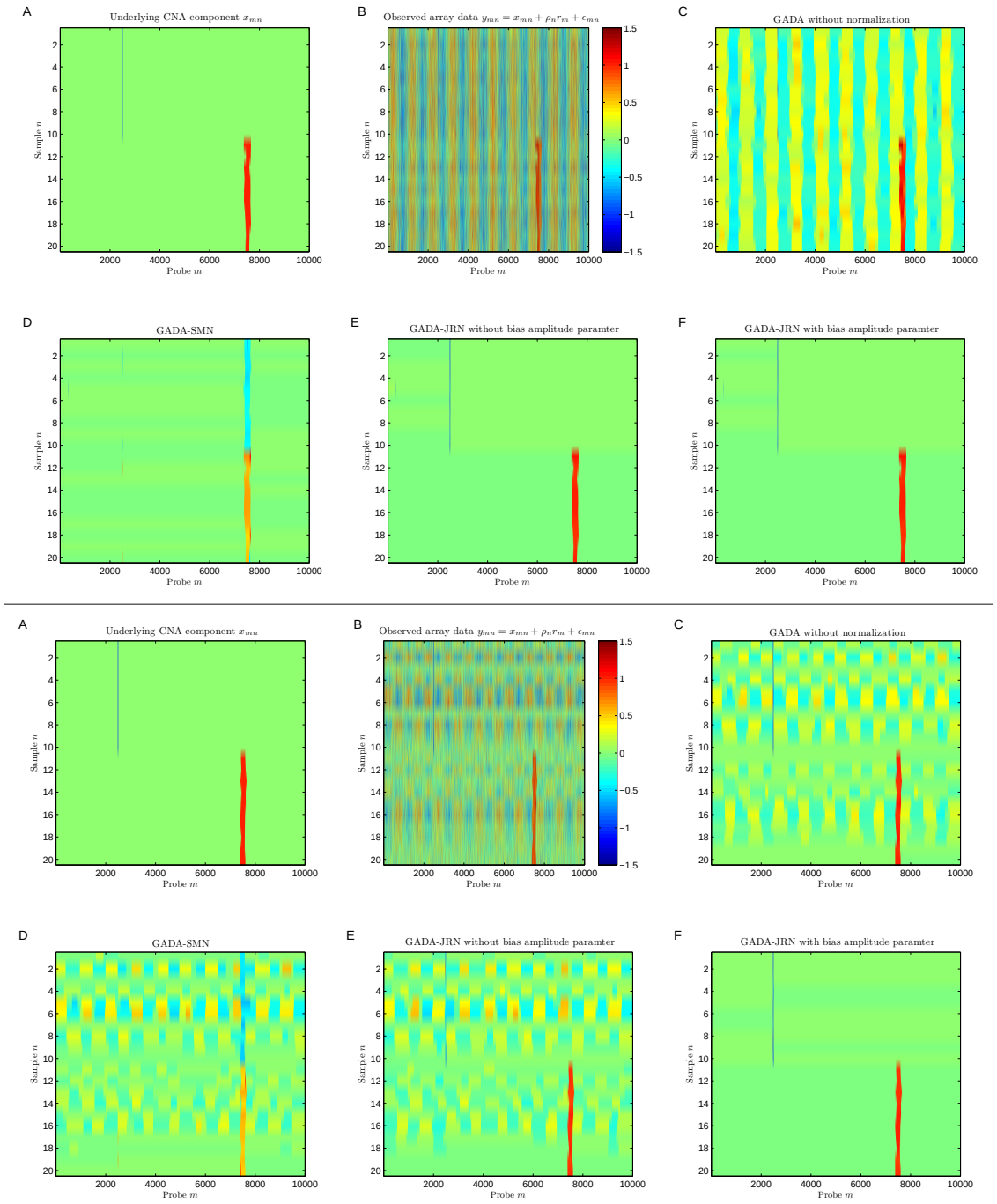


Figure 5.10: Simulation model with measurement bias with different amplitudes: Top $\rho_n \sim U(0.7, 1.0)$, Bottom $\rho_n \sim U(-1.0, 1.0)$. The colors represent the observed hybridization intensity and the copy number (blue - loss, red - gain, green - neutral). A) The true underlying CNV component with two CNV regions (CNVR-1 around m=2500 and CNVR-2 around m=7500). B) Simulated array hybridization intensities degraded by noise $\boldsymbol{\epsilon}^n$ and a systematic measurement bias $\boldsymbol{r}$. C) Copy number profile using GADA on non normalized data. D) Copy number profile using GADA with separate median normalization (GADA-SMN). E) and F) Copy number profile estimated using GADA-JRN with or without a scaling parameter ρ_n for the bias.

5.6.6 Scale effect on the Affymetrix data

The same pre-processed data as in Section 5.6.4 is used to evaluate the new model with a bias amplitude parameter. In real data, the change of the amplitude of the bias is likely to be a consequence of the differences on the initial amounts of DNA material as was shown by [18]. The Aroma.Affymetrix pre-processing can make the amplitudes ρ_n the same across samples using the appropriate scaling and correction to each of the Sty and Nsp fragments. For this reason, the results do not differ much whether or not this new parameter is added to the model (Figure 5.11).

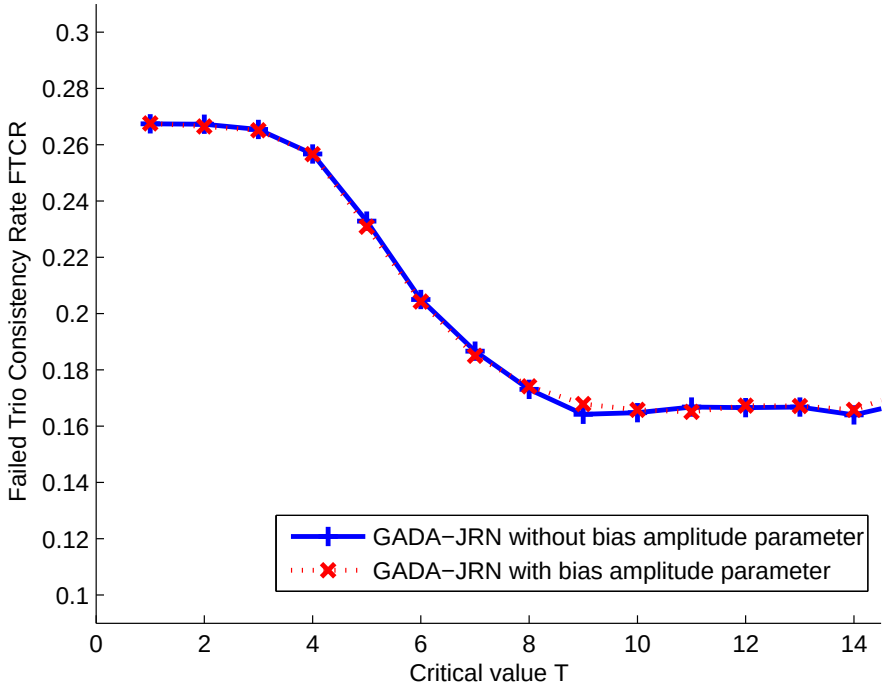


Figure 5.11: Consistency within HapMap trios using a different sparseness setting T. Using 180 HapMap samples (CEU+YRI) the GADA+JRN algorithm does not show a significant improvement if we add a bias amplitude parameter on this set of samples after Aroma.Affymetrix normalization.

5.6.7 Impact of the batch effects on the Affymetrix dataset

All the samples in the Affymetrix dataset were generated on the same laboratory at the same time and analyzed on three plates with each ethnic group (CEU, YRI, JPT+CHB) analyzed in separate plates. The batch effects that would be expected if the plates were analyzed in separate days or labs would be much higher than what it is actually seen in these samples, see simulated example in Figure 5.14. For this reason, we do not observe any significant difference in FTCR with the new algorithm if the two batches are analyzed separately or combined together with (GADA-JRN) in Figure 5.12. The performance of separate median normalization independently in each plate is worse because the median in not robust in a small set of samples (especially in this dataset where each plate has a different ethnic group).

If the subgroups (or batches) are not known a priori the residuals after running GADA can be useful to discover subgroups of samples that share a common artifact. In Figure 5.13B, two blocks can be visually appreciated that do not appear on Figure 5.13C. The average magnitude of these correlations is small (< 0.05) compared to the variability within each block (< 0.25) because the shift in r is not substantially large. Less than $< 0.36\%$ probes have a shift in r_m larger than a true copy number change ($|r_m^{CEU} - r_m^{YRI}| > 0.585 = log_2(3) - 1$). This remaining probe hybridization bias if spatially uncorrelated will be regarded as an additional white noise in our model (higher noise variance estimate) and are unlikely to cause false detections but a decreased sensitivity to true changes. These shifts are likely to be considerably larger if the batches

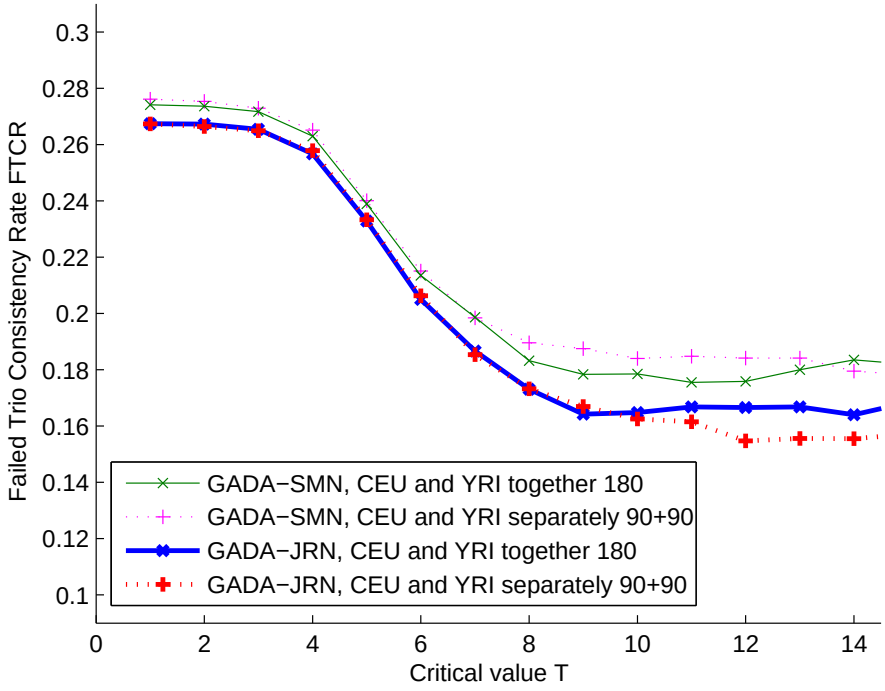


Figure 5.12: Consistency within HapMap trios when two plates (CEU and YRI) are analyzed separately or together using the GADA-SMN and GADA-JRN algorithms. In terms of FTCR we only obtain a small improvement if we separate each batch using the GADA-JRN, but a decreased performance in terms of the median because it requires more than 90 samples to become a more robust estimator.

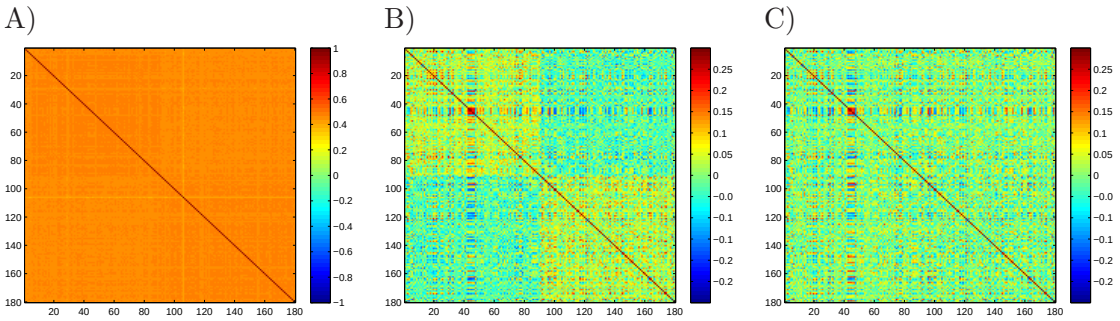


Figure 5.13: Pairwise Spearman's Correlations between different signals: A) the original microarray observed intensities, B) the residual intensities after processing together CEU+YRI with GADA-JRN, and C) the residual intensities after processing separately CEU and YRI groups with GADA-JRN. Samples 1 to 90 correspond to the CEU ethnic group and 91 to 180 to the YRI ethnic group.

were analyzed in different days or labs. In this latter case we would analyze each batch separately with GADA-JRN (see Figure 5.14 for a simulation result).

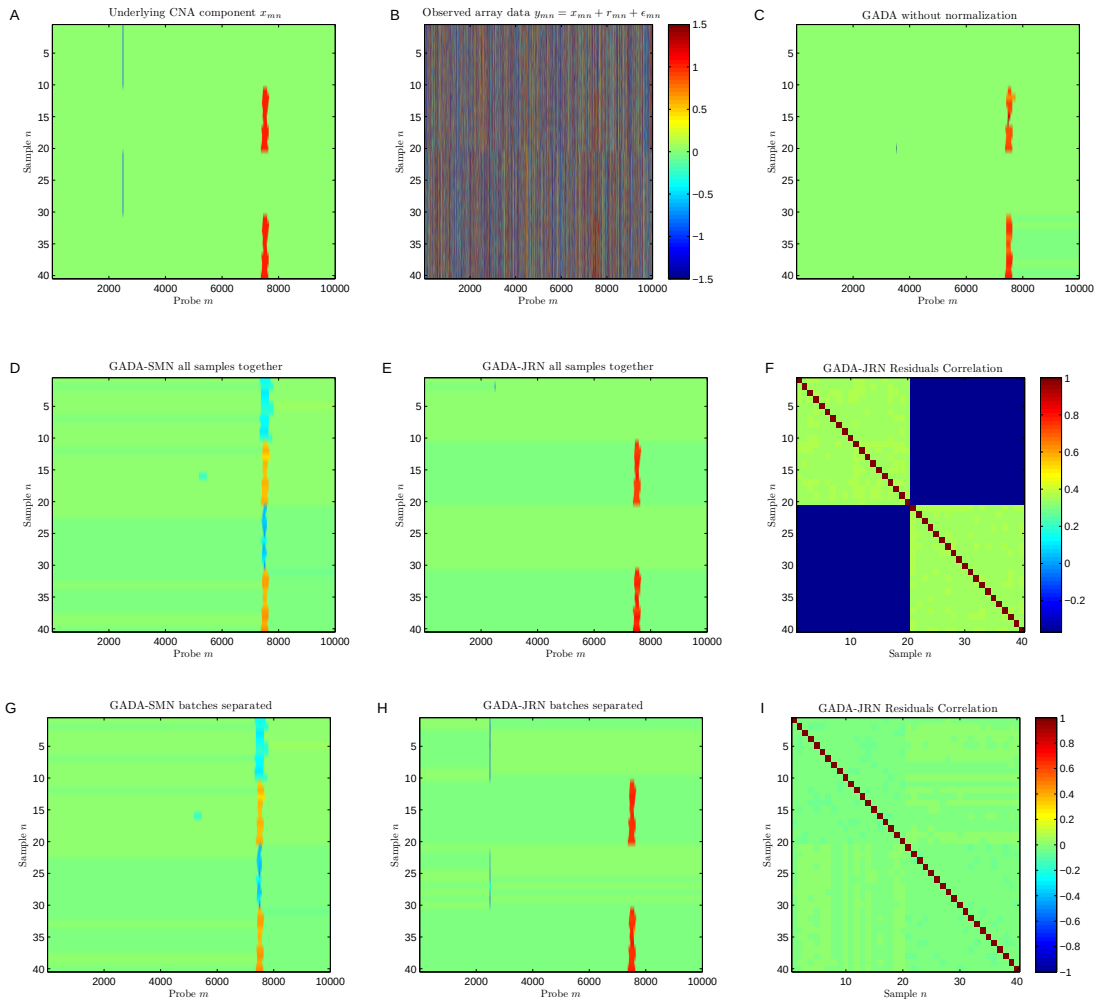


Figure 5.14: Simulation model with a batch effect: for $n = 1..20$ $r_m^n =\sim \mathcal{N}(0, 1.0)$ independent of $n = 21..40$ $r_m^n =\sim \mathcal{N}(0, 1.0)$. The colors represent the observed hybridization intensity and the copy number (blue - loss, red - gain, green - neutral). A) The true underlying CNV component with two CNV regions (CNVR-1 around m=2500 and CNVR-2 around m=7500). B) Simulated array hybridization intensities degraded by noise $\boldsymbol{\epsilon}^n$ and a systematic measurement bias $\boldsymbol{r}$. C) Copy number profile using GADA on non normalized data. D) and E) Copy number profile using GADA-SMN and GADA-JRN on all the samples together. G) and H) Copy number profile using GADA-SMN and GADA-JRN on each batch separately. F) Pairwise correlation between the residuals of GADA-JRN after processing the batches together or I) separately.

5.7 Discussion

The application of the proposed GADA-JRN is not only limited to Affymetrix SNP arrays but it can also be applied to other platforms such as Illumina beadarrays or NimbleGen aCGH. In Illumina BeadStudio the probe hybridization intensities R (obtained after allele crosstalk correction, ACC) can be extracted instead of the LRR values. The extraction of the LRR values uses a cluster approach to compute the expected R values of non-CNV regions [71] which have similar drawbacks as separate median normalization. In aCGH, the reference DNA from a single sample or a pool of samples is used as a reference, but these log-ratio intensities may still contain a remaining "wavy" artifact [63] that our proposed approach could eliminate.

ITALICS [83] is an iterative approach that alternates two separate steps of copy number detection and normalization. The iterative concept is similar to the EM algorithm employed to fit the GADA-JRN model in this chapter but there are two fundamental differences. First, ITALICS operates on a *single* sample and only includes a small set of parameters correcting for global array effects such as the fragment length and GC content; ITALICS assumes that the reference non-CNV probe intensity is fixed and extracted from a separate reference set of samples. Second, the copy number extraction and the normalization model are iterated in practice only twice and not integrated under an unifying probabilistic model as in GADA-JRN. In contrast, GADA-JRN first proposes a *multiple* sample probability model, which includes parameters for the CNV component and the reference non-CNV probe hybridization intensity of every position of the genome;

then an iterative approach to fit the model is derived using the EM algorithm, which guarantees the parameters converge.

GEMCA [54], only available for Affymetrix 500K platform, approaches the problem by finding the reference *after* copy number detection. First, the copy number is estimated on the difference between all the possible pairs. The CNV regions are defined as those identified in certain number of pairs. Then the largest subgroup of samples with no relative variations (found using a maximum clique algorithm) is used to establish the reference set for that particular region of the genome. The Canary algorithm in Birdsuite [55] uses a GMM mixture model to identify a posteriori the copy number variation state of already delimited regions of copy number variation. In both GEMCA and Canary, the underlying assumption that CNVs are tightly aligned across samples and do not overlap with other possible CNVs represents a challenge in dealing with complex polymorphic CNV regions of the genome (e.g., Figure 5.7).

5.8 Conclusions

In this chapter we introduced a new method in which the reference probe hybridization intensity is jointly estimated with the copy number component. This type of methods are essential for the characterization of CNV on normal population using the latest array technologies, in which the underlying genome copy number variations are not known a priori. The currently used approach of separate pre-processing using the median to estimate the hybridization intensities may fail to accurately detect highly polymorphic regions of the genome. The new proposed method extends the previous GADA model by

introducing a new vector of parameters that model a common hybridization bias shared across many arrays. Results demonstrate a significantly better performance with the new extended GADA model while maintaining the attractive linear computational complexity in number of probes and samples.

GADA-JRN may also prove to be a flexible algorithm in a variety of other applications. For example, samples processed by different labs or on different days may not necessarily have the same non-CNV reference hybridization intensity even if the platform is exactly the same. GADA-JRN could be applied to assess if this effect has changed between different experimental batches. Moreover, one or more control samples replicated across experimental batches could be used to better characterize and correct the probe hybridization noise.

In addition to the probe hybridization reference, other parameters can also be introduced to the GADA framework to model other known sources of variation. For example, allowing changes in scale of the bias, using an independent bias correction for each SNP allele, and including probe specific noise variances (heteroscedastic model). The residuals of the model can be used to assess the impact of these other sources of variation on the results In particular, correlation on the model residuals can be used to discover hidden batch effects indicating the need for subgroup analysis. Finally, the GADA-JRN model could be used in combination with the N-GADA model [74] (in Chapter 4) for modeling breakpoints of CNVs across multiple samples.

Chapter 6

Bayesian hierarchical modeling of means and covariances of gene expression data within families

The previous chapters proposed methods for extracting and analyzing only one kind of genomic information. Chapter 2 covered gene expression, and microarrays are used to quantify the transcription activity of each gene in the genome. Chapters 3,4 and 5 covered the analysis of DNA copy number changes using genotyping arrays. In this chapter we will develop a new method that will identify variation in the DNA (SNP genotypes) that influence the gene expression levels.

We propose a novel hierarchical Bayesian model for the influence of constitutional genotypes from a linkage scan on the expression of a large number of genes. This work was presented in Genetic Analysis Workshop GAW-15 and was also selected for publication [76], and can be considered as a first step to find genetic determinants of gene expression at genome-wide scale.

Results on Chr. 11 replicate an already known association between a DNA variation (SNP) and a gene expression level. The approach appears to be a promising way to

address the huge multiple comparisons problem for relating genome-wide genotype-by-expression data.

This chapter is organized in the following way. First we provide the background on the problem, the approaches that exist and we motivate the need for new methods (Section 6.1). Then, on the Methods Section we present our model, the statistical approaches used to fit the model, and the data that has been used. Sections 6.3 and 6.4 present and discuss the results obtained.

6.1 Introduction

The two major genomic informations: i) DNA variation and ii) gene expression have been usually been studied separately. An extensive literature on the analysis of gene expression data has evolved over the last five years, and since the advent of ultra high volume genotyping platforms, genome-wide association and linkage scans using SNPs have also become feasible.

The multiple comparisons problem is central to the analysis of either type of high-volume data. In 2001, Jansen [49] proposed combining the analysis of the two technologies (SNPs and gene expression arrays) in what he called "genetical genomics" to provide insight into the genetic regulation of gene expression.

However, only quite recently have attempts been made to relate the two technologies, first by Morley et al. [67] in a linkage scan for 3,554 expressed genes in relation to 2,756 autosomal SNP markers, and subsequently by the same group [14] in a genome-wide association scan of 27 of the expressed genes with the highest linkage in the first study, in

relation to $> 770,000$ SNPs. (See also Schadt et al. [87] and Stranger et al. [94] for similar analyses.) Independently, Tsalenko [100] proposed a biclustering method to visualize SNPs and the transcripts they regulate, using an approach that is more visual than statistical. The multiple comparisons problem in such analyses (2.7 billion comparisons in the association analysis) dwarfs those from either genome-wide linkage or association analyses of single traits or supervised cluster analyses of expression data in relation to single outcomes.

Therefore, there is a need to develop new statistical methods to analyze all transcripts and genotypes together. Here, we describe a novel hierarchical Bayesian approach to the analysis of all possible pairs of associations and linkages between expressed genes and SNP markers. We demonstrate the results for chromosome 11 and we argue that the method can be extended to cover the entire genome and transcriptome.

The proposed model comprises linear regression models for the means in relation to genotypes and for the covariances between pairs of related individuals in relation to their identity by descent estimates. The matrices of regression coefficients for all possible pairs of SNPs by all possible expressed genes are in turn modeled as a mixture of null values and a normal distribution of non-null values, with probabilities and means given by a third-level model of SNP and trait random effects and a spatial regression on the distance between the SNP and the expressed gene. The latter provides a way of testing for *cis* and *trans* effects, depending if the alteration affects the gene where it is sitting or some other gene.

The method was applied to data on 116 SNPs and 189 genes on chromosome 11, for which Morley et al. [67] had previously reported linkage. We were able to confirm the

association of the expression of *HSD17B12* with a SNP in the same region reported by Morley et al., and also detected a SNP that appeared to affect the expression of many genes on this chromosome. The approach results are promising and could be extended to cover an genome-wide gene expression and genotyping scan.

6.2 Methods

6.2.1 Statistical model

Let Y_{ij}^n denote the expression of gene n in member j of family i and let G_{ij}^m be the corresponding SNP genotype at marker m at location x_m. For the means and covariances of the expression traits, we adopted a Generalized Estimating Equations model of the form used by Thomas et al. [95].

$$E\left(Y_{ij}^n\right) \equiv \mu_{ij}^n = \alpha_0^n + \sum_{m=1}^{M} A^{nm} G_{ij}^m \tag{6.1}$$

$$E\left(C_{ijk}^n\right) \equiv \chi_{ijk}^n = \beta_0^n + B^n Z_{ijk}\left(X^n\right) \tag{6.2}$$

where $C_{ijk}^n = (Y_{ij}^n - \mu_{ij}^n)(Y_{ik}^n - \mu_{ik}^n)$ and $Z_{ijk}(x)$ is the estimated $E(IBD_{ijk}(x)|\boldsymbol{G}_i)$ at chromosomal location x for pairs (j,k) from nuclear family i, based on the complete multilocus marker data. X^n is a latent variable for the location of the unobserved causal locus linked to expression trait n. For $j = k$, $V(Y_{ij}^n) = \chi^n$ models the gene expression variance in (6.2).

In (6.2), the regression coefficients A^{nm} are modeled as mixtures of null values with probabilities $1 - \pi^{nm}$ and a normal distribution of non-null values with means α^{nm} expressed in terms of row and column effects:

$$A^{nm} \sim (1 - \pi^{nm})\, \delta(\mathbf{0}) + \pi^{nm} N\left(\alpha^{nm}, \sigma^2\right) \tag{6.3}$$

where

$$\alpha^{nm} = \gamma_0^A + \gamma_1^A I(x_m \in R_n) + e_m^A + h_n^A \tag{6.4}$$

$$\text{logit}(\pi^{nm}) = \gamma_0^P + \gamma_1^P I(x_m \in R_n) + e_m^P + h_n^P \tag{6.5}$$

The parameter γ_1 distinguishes between *cis* and *trans* effects, a *cis* interaction occurs when the chromosomal location x_m of SNP m is within the interval R_n, the alignment region for the gene expression probe n. The random effects $\boldsymbol{e}$ and $\boldsymbol{h}$ are distributed as

$$\left(e_m^A, e_m^P\right) \sim N_2(\mathbf{0}, \boldsymbol{T}) \tag{6.6}$$

$$\left(h_n^A, e_n^P\right) \sim N_2(\mathbf{0}, \boldsymbol{W}) \tag{6.7}$$

and the γs, $\boldsymbol{T}$, $\boldsymbol{W}$ have uninformative normal and Wishart priors.

The regression coefficients B^n in the covariance model (6.2) are handled similarly, except that we assume each trait has at most one region linked to it. (This is not essential to the method, as Eq.(2) could be extended to a summation over multiple independent linkage regions, but it would not make sense to offer all marker locations simultaneously,

since the IBD variables are highly correlated from one location to the next.) Thus, we assume

$$B^n \sim N\left(\gamma_0^B + \gamma_1^B I\left(X^n \in R_n\right), \tau^2\right) \tag{6.8}$$

and pick a uniform prior for X^n; to simplify the calculations, we restrict X^n to the observed marker locations x_m and compute IBD probabilities only at these locations. X^n thus has a discrete distribution with prior masses inversely proportional to the local marker density, here estimated simply as $|x_{m+1} - x_{m-1}|$. The full model is represented in the directed acyclic graph (DAG) shown in Figure 6.1.

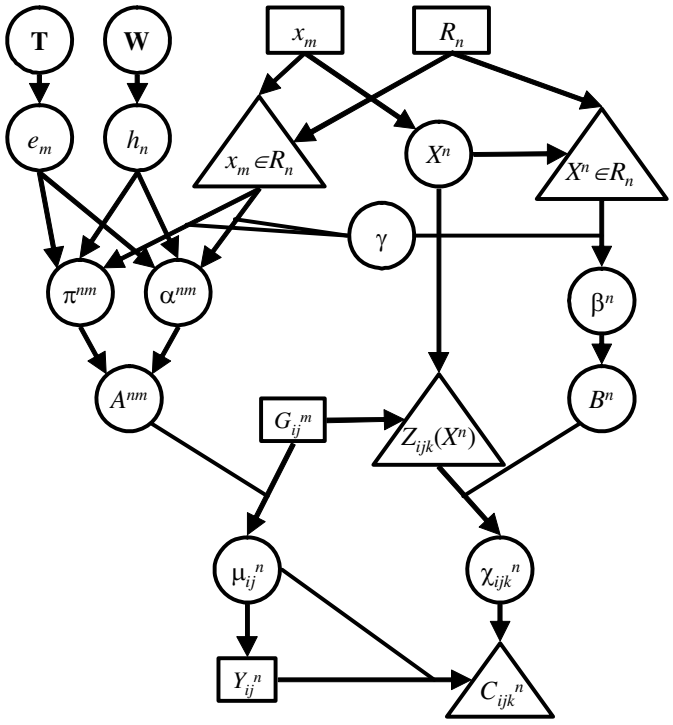


Figure 6.1: Directed acyclic graph for the analysis model. Squares represent observed data, circles represent parameters or latent variables, triangles represent deterministic nodes.

We fitted the model using a Markov chain Monte Carlo (MCMC) approach, implemented in Matlab. Updates of all parameters except the location parameters X^ns involve standard Gibbs sampling from their respective full conditional distributions, e.g., $[\alpha_0^n|\boldsymbol{Y}^n, \boldsymbol{G}, \boldsymbol{A}]$, $[\beta_0^n|\boldsymbol{C}^n, \boldsymbol{Z}, \boldsymbol{B}]$, $[\boldsymbol{A}^{nm}|\boldsymbol{Y}^n, \boldsymbol{G}^m, \alpha_0^n, \pi^{nm}, \alpha^{nm}, \sigma^2]$, etc. The updates of the Xs are based on a Metropolis-Hastings procedure with a random walk proposal. The sequence was started 10 times from several initial points chosen from an overdispersed prior around rough estimates. Half of the initial samples are discarded and the second half is kept. The number of kept samples, L = 4000, is chosen to be large enough so that for all parameters of interest the variance between sequences V_B is comparable to that within sequence V_W, $R < 1.10$:

$$\hat{R} = \sqrt{\frac{L-1}{L} + \frac{1}{L}\frac{V_B}{V_W}} \tag{6.9}$$

The rationale behind this convergence monitoring procedure is described and justified in [32].

6.2.2 Subjects, genotypes, and phenotypes

In order to keep the computation to a manageable level, we restricted this analysis to the SNP genotypes and expressed genes on chromosome 11, as previous analyses by Morley et al. [67] had found evidence of linkages both in *cis* and in *trans* at this chromosome. The final dataset thus had 116 SNPs and 189 expressed genes. IBD status was estimated from the complete two-generation pedigrees (excluding grandparents) by a program written

based on the Lander-Green algorithm [59]. All 378 sibpairs (110 individuals) from the available 14 families were included in the phenotype analysis.

6.3 Results

After convergence has been reached, the number of regression coefficients with nonzero coefficients in (6.2) is very small. This is because in the mixture model employed in (6.3), a large number of the probabilities are close to 0 as shown in Figure 6.2.

Figure 6.2 also shows, as expected, that each gene expression phenotype is explained by relatively few genotypes that have a role in regulating their expression. Table 6.1 lists, for the best predicted phenotypes, the SNPs included most frequently in the model. Significantly, the top ranking phenotype, *HSD17B12* (217869–at), associated with SNP rs1453389, is the same as the one reported by Cheung et al. [14] as associated with another SNP in the same region (not included in the GAW dataset). Figure 6.3 shows that some SNPs in chromosome 11, especially rs916482, are significantly associated with more phenotypes than others. These SNP are possibly within a master regulatory region of gene expression. The list of gene ontology terms that were over-represented in the list of its associated genes involved mostly metabolic functions (Figure 6.4).

The covariance model (2) results are summarized in the right panel of Figure 6.2, and the strongest linkage peaks are listed in Table 6.2. This linkage is for the remaining variation not explained by the association/means model (1), and the peaks would correspond to unseen genotypes that are in linkage disequilibrium (LD) with a marker that was not used in the association model. Thus, this explains in part why linkage results

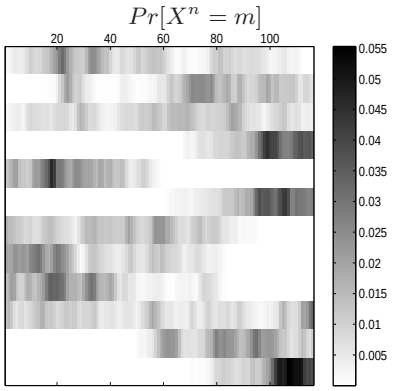
Table 6.1: Top ranking associations

				Top SNPs used in the prediction							
Phenotype	Probe	R^2	$P(R^2 > 0)$	SNP1	π^{nm}	SNP2	π^{nm}	SNP3	π^{nm}	SNP4	π^{nm}
HSD17B12	217869_at	0.25	0.988	*rs1453389	1.00	rs916482	1.00	rs1425151	0.40	rs509628	0.28
C11orf10	218213_s_at	0.12	0.986	rs916482	1.00						
AMPD3	207992_s_at	0.19	0.985	*rs2029463	0.81	rs948215	0.80	rs1157659	0.21	rs1491846	0.17
FEZ1	203562_at	0.12	0.984	rs2029463	1.00	rs2155076	0.20	*rs948215	0.11		
ADM	202912_at	0.11	0.982	rs916482	1.00						
STIP1	213330_s_at	0.11	0.981	rs916482	0.99	rs1319730	0.33				
DDB1	208619_at	0.15	0.978	rs1530966	0.91	rs597345	0.54	rs1499511	0.10		
FADS1	208964_s_at	0.14	0.974	rs1216592	0.85	rs1605026	0.38	rs591804	0.35		
TPP1	200743_s_at	0.13	0.970	rs916482	0.94	rs1157659	0.14	*rs902215	0.14		
RBM14	204178_s_at	0.10	0.966	rs916482	0.98	rs674237	0.10				
HMBS	203040_s_at	0.13	0.963	rs86392	0.49	rs916482	0.47	*rs1319730	0.44	rs1945906	0.20
PPME1	49077_at	0.11	0.958	rs916482	0.82	rs2155001	0.16				
CD44	204490_s_at	0.12	0.957	rs702738	0.34	rs916482	0.28	rs1319730	0.28	*rs1453390	0.17
NRGN	204081_at	0.10	0.946	rs2029463	0.93	rs961746	0.16	rs509628	0.15		
NDUFS8	203190_at	0.11	0.944	rs86392	0.68	rs1319730	0.33	rs1945906	0.32		
PSMD13	201232_s_at	0.09	0.923	rs916482	0.91	rs1319730	0.12				

Phenotypes ranked by most significant coefficient of determination, and some of their top associated SNPs ranked by average π^{nm}. (*) indicates a cis-acting interaction, defined as the SNP being within 10MB of the phenotype probe alignment.

Table 6.2: Linkage of residual gene expression variation after association

Phenotype	R^2	$P(R^2 > 0)$	Samples	Mode$[X^n]$
208964_s_at	0.014	0.912	1749	rs2226844
202223_at	0.009	0.908	1526	rs1453390
220964_s_at	0.007	0.862	1412	rs647837
201432_at	0.005	0.821	1567	rs931811
201477_s_at	0.008	0.802	1492	rs1941817
204178_s_at	0.004	0.800	1548	rs2155076
202076_at	0.004	0.772	1668	rs681267
206067_s_at	0.002	0.749	1535	rs2029463
210364_at	0.003	0.718	1586	rs470719
203675_at	0.006	0.706	1659	rs1216592
205412_at	0.001	0.683	1585	rs470982
202418_at	0.001	0.679	1444	rs470719



Phenotypes ranked by most significant coefficient of determination in the covariance model, the posterior distribution of their locus position X^n, and its mode. The coefficient of determination and its significance are calculated form samples drawn (from 10% of the mass) around the mode.

are less compelling than the association ones. However, for those phenotypes for which significant linkage was found, the expression covariance increased with the IBD status, especially in 208964–s–at.

6.4 Discussion

We have introduced a novel hierarchical Bayes model for genetic control of gene expression. Our approach to dealing with the multiple comparisons problem is to represent the matrices of all possible SNP expressed gene association or linkage coefficients in terms of row and column random effects, along with a spatial regression on the distance between the two. Although this allows inference on specific pairs, we have greater interest in the variances of the row and column effects, which reflect systematic tendencies for SNPs to affect variable numbers of phenotypes and for phenotypes to be differentially expressed. Our mixture model also supports the possibility that the vast majority of such associations or linkages would be truly null, and allows separate estimation of both the probability and magnitude of non-null tests. So far we have not imposed any relationship between the parameters of the association (means) and linkage (covariance) models, but one might contemplate using the broad regions where linkage is seen for a particular phenotype as a prior for testing single-SNP associations with that phenotype.

The strongest gene-expression SNP association reported by Cheung et al. on chromosome 11 also appeared in our results as the most significant association, but with a SNP close to theirs (their reported SNP was not included in the dataset). We also found evidence of at least one SNP that appears to be linked to a large number of expressed genes, suggesting the existence of master regulatory genes in that region.

We chose to restrict these analyses to a subset of genes and SNPs on a single chromosome to test the feasibility of the method. In principle the approach could be extended on a genome-wide scale, since the computation time increases linearly with m, n, sample

size, and number of MCMC samples. Generating 4000 MCMC samples required 6 hours on a 2.2 GHz single-processor machine. However, one outstanding methodological challenge that would have to be addressed before the approach could be applied to dense SNP associations would be how to deal with the multicollinearity problem. This is because very close SNPs will exhibit highly correlated genotypes (strong LD). For this reason, we chose to restrict this analysis to only a subset of SNPs that were not strongly correlated (in strong LD) with each other.

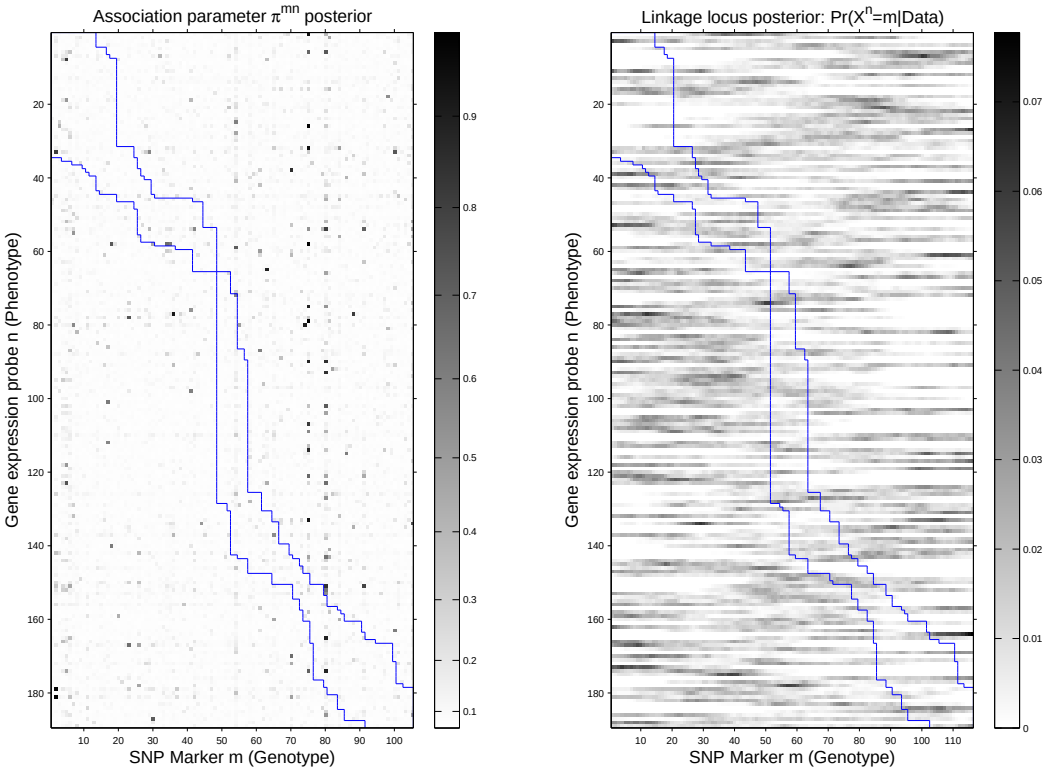


Figure 6.2: Gene expression x Genotype associations and residual linkage summary. (Left) Image describing the mean value of the association parameters π^{nm} between the gene expression phenotypes (rows) and the SNP genotypes (columns). The matrix shows that the interactions are very sparse (dark spots), meaning that phenotypes are controlled by small number of SNPs, with no apparent concentration along the *cis* region delimited by blue lines. However, there exist some SNPs (columns) that seem to be correlated with a large set of phenotypes, potentially indicating a master regulatory region. (Right) Image describing the posterior probability of the linkage locus after removing the association effect from the covariance.

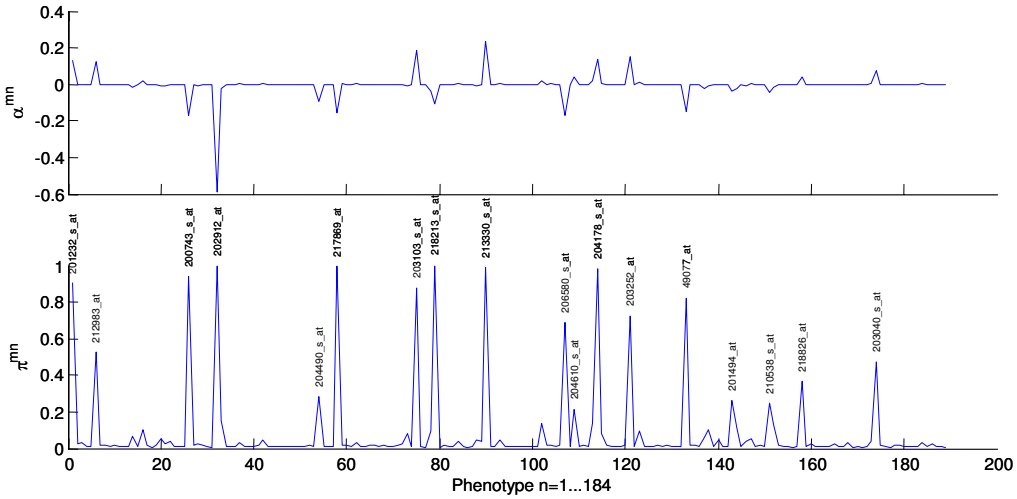


Figure 6.3: Potential Master Regulatory region around rs916482 SNP. Bottom plot is the cross-section of column 84 of Figure 6.2, describing the association between all phenotypes in chromosome 11 and SNP m = rs916482. The top plot shows the magnitude and sign of dependence on the genotype. This SNP has a large number of associated genotypes, providing a strong indication of a Master Regulatory region.

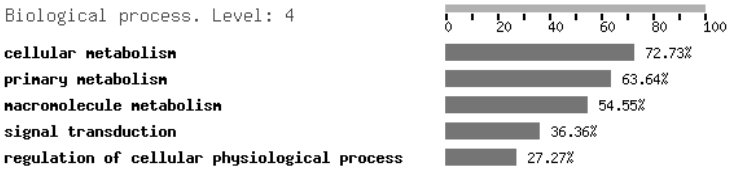


Figure 6.4: Gene Ontology (GO) on potential Master Regulatory region. Overrepresented GO terms by the phenotypes associated to the SNP rs916482 analyzed using FatiGO (http://www.fatigo.org/).

Chapter 7

Conclusions

The research presented in this dissertation has been developed with the primary application of providing new Bioinformatics tools for reliable and efficient processing of very large biological datasets. These datasets can measure millions of biological variables degraded by noise, but the underlying sparseness of these biological systems can be exploited to extract the relevant information.

In Chapter 2, a novel embedded FSS framework was developed that builds a block diagonal linear discriminant model (BDLDA). BDLDA was showed capable of identifying gene interactions and improving the classifying accuracy in simulations. The algorithm is based on a greedy search that recursively seeks to add either a new feature or an interaction term in the model. Feature selection is very important not only for reducing the number of parameters but also for identifying relevant genes or genetic pathways (group of genes that are corregulated) which are consistent with the underlying biology of the disease (e.g., inflammatory genes, DNA repair, cell division).

As higher density microarrays and direct RNA sequencing become available novel analytical techniques will be required for finding differences in gene expression (and exon

splicing) in different kinds of tissues, organisms or tumors. In this context, sparse signal representations can be used to group microarray probes that belong to the same exon and characterize which of the different exon combinations (splicing variants) are active on a given sample. A similar methodology as employed to detect copy number alterations (Chapter 3) can be used to extract these splicing features. These clean features can then be used to build models using the discriminant analysis tools (DLDA and BDLDA) we already developed in Chapter 2. As more knowledge of the underlying biology is becoming publicly available and genetic pathways are better understood, the BDLDA model search strategies can be adapted to look for patterns of coexpression along these pathways.

Copy number alterations can also be represented by a sparse PWC representation. In GADA approach, introduced in Chapter 3, a sparse Bayesian learning (SBL) is used in combination with the PWC representation. When compared to other popularly used copy number detection approaches, GADA achieves one of the highest detection accuracies while improving computational speed by several orders of magnitude, especially on very large arrays. Other segmentation approaches with comparable accuracy to GADA such as CBS have a quadratic cost in terms of computation. The GADA software and source code is publicly available (`http://biron.usc.edu/~piquereg/GADA`) and has been downloaded and used by a large number of institutions including the Sanger Institute, The Center for Applied Genomics (TCAG), and the Center for Genomic Regulation (CRG) among others.

The GADA approach has also been extended to scenarios where multiples samples are available and a joint analysis would help achieve higher detection accuracy (Chapters 4 and 5). The first of these scenarios models breakpoint locations that are shared across

samples (examples include sample replicates or samples with common ancestry), Chapter 4. The second scenario assumes that the breakpoints are not necessarily located at the same position across samples, but there is a systematic perturbation (or measurement bias) on the probes that can be estimated and thereby removed. We demonstrated in Chapter 5 that joint extraction of a reference hybridization intensity and the copy number component of a large cohort of samples has better performance in terms of accuracy and robustness than using the median across the samples.

The shared breakpoint model can be integrated with the joint reference normalization model into a complete multiple sample model. These improved models and methods should be especially indicated to make experimental designs in which a set of well known reference samples are replicated across batches. These methods will help to bring the intensity values together and reduce the experimental variability across different experimental batches or laboratories. The GADA approach could also be extended to find PWC discriminatory regions along the genome to separate groups of samples in a supervised fashion. Copy number alterations associated with a target disease should also exhibit a piecewise constant behavior on the statistical scores used to measure association. GADA can exploit this PWC behavior to accurately detect regions associated with some disease condition.

In the near future we expect that the number of technologies that are available to extract genetic measurements will only continue to increase. If the current trend in microarray technology continues, the number of measurements duplicates every few months. New and cheaper sequencing techniques (Solexa and 454 sequencing) are becoming available enabling a new set of experiments that generate millions of reads that can map to

any position on the genome (3 billion pairs). Chromatin Inmuno-Precipitation (ChIP) studies have been developed to detect the organization of the DNA around nucleosomes in the nucleus and DNA binding proteins. Large scale projects that have been launched in the last three years (e.g., the HapMap project, the cancer genome atlas (TCGA), the 1000 genomes project) are generating large amounts of data that is becoming publicly available. These datasets will shed light on the characteristics of the cancer genome and the natural variation present on human healthy cells, but will require developing new and more efficient analysis techniques. GADA or similar methods could be used to detect signals of interest in data obtained by new generation sequencing techniques that are becoming available.

The knowledge on how to measure gene expression (mRNA) and genetic alterations (DNA) will make it possible to answer more complex biological questions. We will be able to use and extend systems biology precise knowledge on small isolated biological processes to the genomewide scale data obtained by high-throughput experiments on cells in living organisms. Sparse signal processing representations will be useful to interconnect these system biology models in a large sparse network that will span the entire genome. For example, pharmacogenetics studies will measure the impact of a drug on the gene expression for diseases that require better treatments and reduced undesired secondary effects. Other than drugs, new experiments will focus on the DNA alterations that are induced by viruses which may lead to discover new preventable agents of cancer.

We still know very little about which parts of the genome play a role in regulating rates of gene expression, mRNA degradation, or translation. Our ability to interpret the regulatory "language" of DNA sequences is extremely limited. Future research should

aim to provide much better annotation of the regulatory elements in the human genome, and ultimately, to work towards detailed models of the combinatorial nature of gene regulation. These models and corresponding validations would contribute to create a much deeper understanding of the gene regulation mechanisms and the functional impact of genetic variation. The research presented in this thesis provides a starting point to develop analytical methods to jointly analyze the impact of genetic alterations with the gene expression levels to pinpoint important genome regions for gene regulation.

Appendix A

The role of the parameter a in SBL

The parameter a controls the shape of the prior distribution over the weights $p(\boldsymbol{w})$ specified by the hierarchical prior defined by $p(\boldsymbol{w}|\boldsymbol{\alpha})$ (3.14) and $p(\boldsymbol{\alpha})$ (3.15). Following [97], the $\boldsymbol{\alpha}$ hyperparameters can be integrated out to find the marginal "effective" prior $p(\boldsymbol{w})$:

$$\begin{aligned} p(\boldsymbol{w}) &= \int p(\boldsymbol{w}|\boldsymbol{\alpha})\, p(\boldsymbol{\alpha})\, d\boldsymbol{\alpha} \\ &= \prod_{m=1}^{M-1} \int p(w_m|\alpha_m)\, p(\alpha_m)\, d\alpha_m \\ &= \prod_{m=1}^{M-1} p(w_m) \end{aligned} \tag{A.1}$$

where $p(w_m)$ is:

$$\begin{aligned} p(w_m) &= \int p(w_m|\alpha_m)\, p(\alpha_m)\, d\alpha_m \\ &= \frac{\Gamma(a+1/2)}{\Gamma(a)\sqrt{2\pi a}} \sqrt{\frac{a}{b}} \left(1 + \frac{w^2}{2b}\right)^{-\left(\frac{1}{2}+a\right)} \end{aligned} \tag{A.2}$$

a t-distribution with $2a$ degrees of freedom and a scale parameter of $\sqrt{a/b}$. When $b \to 0$ and a is small, this distribution peaks very sharply at 0, and has very thick flat tails, as shown in Figure A.1.

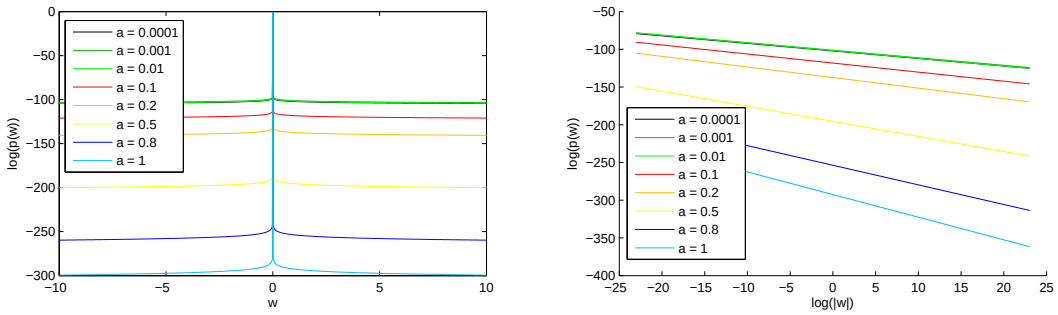


Figure A.1: Plot of the SBL marginal prior distribution on a single weight for different choices of the hyperparameter a. To make the plot we approximated $b \to 0$ by $b = 1E-80$, this is a similar conceptually as approximating a delta distribution by a normal distribution with $\sigma = 1E-80$.

As justified in Section 3.3 and 3.4, the *log* of the prior distribution $p(\boldsymbol{w})$ gives us the sparseness cost measure (i.e., the penalty for not having coefficients different than 0):

$$\log p(\boldsymbol{w}) \quad = \quad C(a,b) + \left(1 + \frac{a}{2}\right) \sum_{m=0}^{M-1} \log\left(1 + \frac{w_m^2}{2b}\right)$$

and we are interested in the case when $b \to 0$, which gives (3.21) which we repeat here for easier reference:

$$\log p(\boldsymbol{w}) \quad \underset{b \to 0}{\to} \quad C(a) + (1+2a) \sum_{m=0}^{M-1} \log |w_m|$$

After providing the details in the derivation of the SBL sparseness cost, the second objective of this appendix section is to provide more discussion on its properties. This sparseness cost is depicted for a single nonzero weight and several a in Figure A.1 and for

multiple nonzero weights and a single a in Figure 3.3. The approximately flat tails makes this sparseness cost a good approximation of the l_0 norm, and much more desirable than the l_1 norm (i.e., Laplacian prior). Considering specifically a in (3.21) and Figure A.1, we can see that the sparseness penalty is proportional to $(1+2a)$. For example, in Figure A.1 (left), for $a = 1$ we get a penalty of around 300 for large coefficients as compared to 100 when $a \sim 0$, i.e., $(1 + 2a)$ times higher. Therefore, we can increase the sparseness by increasing a, this takes mass away from the tails and puts it on the "delta" (point mass at 0) by decreasing the rate on the tail decay. In Figure A.1 (right), the tail decay rate is about $(1 + 2a)$ on the natural logarithmic scale.

The parameter a also has an impact on the convergence rate of the EM algorithm, i.e., the speed of the SBL algorithm. In our experiments, for higher sparseness settings (fewer breakpoints and larger a), the algorithm converges faster than for smaller a. This is also supported with the following argument. The α_m^{-1} parameters, either converge to 0 (breakpoint discarded) or to a finite point (breakpoint accepted). The EM algorithm rate of convergence is governed by the maximum eigenvalue of the Jacobian matrix of the EM mapping defined in (3.20), [66]. In that situation, $1/(1 + 2a)$ would pull out of the derivative of α_m^{-1} in (3.20); thus speeding up convergence since the maximum eigenvalue is divided by $1/(1 + 2a)$.

In conclusion, the a parameter controls the sparseness in the SBL algorithm, and the speed of the algorithm. An increase of a leads to a sparser result, fewer breakpoints, and faster convergence.

Appendix B

Backward Elimination algorithm properties

The backward elimination (BE) procedure used in GADA (Chapters 3, 4 and 5) could be used alone, without the SBL step, for CNA detection. It is based on considering our PWC model (3.8) as a classical variable regression selection problem, $\boldsymbol{y} \sim \boldsymbol{F}\boldsymbol{w}$; where the regressors w_i with less impact on the residual are sequentially removed one by one. To the best our knowledge, this simple procedure has never been proposed as a standalone technique for CNA detection before. This is a greedy approach, which is suboptimal since we may eliminate breakpoints that could be more significant at a later stage. Since errors can be added by each greedy decision, this algorithm tends to be more reliable when the number of regressors (i.e., candidate breakpoints) is smaller. Compared to forward selection (FS), BE has been seen to perform better in situations where, as in our case, the columns of $\boldsymbol{F}$ have high degree of collinearity [53]. Furthermore, the structure of $\boldsymbol{F}$, the design matrix, can be exploited to efficiently find and remove each breakpoint and produce a ranking list as detailed in Algorithm 4.

Using standard linear regression, for a given fixed breakpoint set $\mathcal{I}$, the least squares estimate for the breakpoint weights $\boldsymbol{w}_{\mathcal{I}}$ is found by solving the normal equations:

$$\begin{aligned} \boldsymbol{F}_{\mathcal{I}}^t \boldsymbol{y} &= \boldsymbol{F}_{\mathcal{I}}^t \boldsymbol{F}_{\mathcal{I}} \hat{\boldsymbol{w}}_{\mathcal{I}} \\ \hat{\boldsymbol{w}}_{\mathcal{I}} &= \left(\boldsymbol{F}_{\mathcal{I}}^t \boldsymbol{F}_{\mathcal{I}}\right)^{-1} \boldsymbol{F}_{\mathcal{I}}^t \boldsymbol{y} \end{aligned} \tag{B.1}$$

which gives the orthogonal projection $\hat{\boldsymbol{x}}_{\mathcal{I}}$ of the vector $\boldsymbol{y}$ on $\mathcal{S}_{\mathcal{I}}$ as:

$$\hat{\boldsymbol{x}}_{\mathcal{I}} = \boldsymbol{F}_{\mathcal{I}} \hat{\boldsymbol{w}}_{\mathcal{I}} \tag{B.2}$$

$$\hat{\boldsymbol{x}}_{\mathcal{I}} = \boldsymbol{F}_{\mathcal{I}} \left(\boldsymbol{F}_{\mathcal{I}}^t \boldsymbol{F}_{\mathcal{I}}\right)^{-1} \boldsymbol{F}_{\mathcal{I}}^t \boldsymbol{y} \tag{B.3}$$

and the residual sum of squares $RSS_{\mathcal{I}}$ or norm of the error is:

$$\begin{aligned} RSS_{\mathcal{I}} &= \|\boldsymbol{y} - \hat{\boldsymbol{x}}_{\mathcal{I}}\|^2 \\ &= \|\boldsymbol{y} - \boldsymbol{F}_{\mathcal{I}} \hat{\boldsymbol{w}}_{\mathcal{I}}\|^2 \end{aligned} \tag{B.4}$$

All these operations can be solved efficiently by noticing again that $\boldsymbol{H}_{\mathcal{I}} = \left(\boldsymbol{F}_{\boldsymbol{\mathcal{I}}}' \boldsymbol{F}_{\boldsymbol{\mathcal{I}}}\right)^{-1}$ is a symmetric tridiagonal matrix, with main diagonal $\boldsymbol{h}_0$ (3.22) and first off-diagonals $\boldsymbol{h}_1$ (3.23) (see lines 2 and 3 of Algorithm 4).

The criterion to decide which breakpoint to remove can be seen in three different but equivalent ways.

First, we might consider removing the breakpoint which increases the least the $RSS_{\mathcal{I}}$. If we denote RSS_j to be the residual sum of the squares after removing i_j from $\mathcal{I}$ ($RSS_{\mathcal{I}-j}$), then the increase in RSS is:

$$RSS_j - RSS_{\mathcal{I}} = \frac{\hat{\boldsymbol{w}}_{\mathcal{I}}^2(j)}{\boldsymbol{h}_0(j)} \tag{B.5}$$

Furthermore, when the noise is normal $N(0, \sigma^2\boldsymbol{I})$,

$$F_j = \frac{RSS_j - RSS_{\mathcal{I}}}{RSS_{\mathcal{I}}/(M-K)} \tag{B.6}$$

is distributed as $F_{1,M-K}$ Fisher-Snedecor distribution (M is the number of candidate breakpoints, and $K = |\mathcal{I}|$ the number of breakpoints in the model). If the σ^2 is known, or $M >> K$,then $RSS_{\mathcal{I}}/(M-K) \to \sigma^2$ and $F_{1,\infty} \sim \chi_1^2$; thus

$$t_j^2 = \frac{RSS_j - RSS_{\mathcal{I}}}{\sigma^2} = \frac{\hat{\boldsymbol{w}}_{\mathcal{I}}^2(j)}{\sigma^2\boldsymbol{h}_0(j)} \tag{B.7}$$

is distributed as a χ_1^2 distribution.

Second, if we assume that the noise is normal $N(0, \sigma^2\boldsymbol{I})$, and σ^2 is known. Then the least squares estimate for $\hat{\boldsymbol{w}}_{\mathcal{I}}$ is also normally distributed:

$$\hat{\boldsymbol{w}}_{\mathcal{I}} \sim N\left(\boldsymbol{w}_{\mathcal{I}}, \boldsymbol{H}_{\mathcal{I}}/\sigma^2\right) \tag{B.8}$$

Therefore, under the hypothesis that $w_{\mathcal{I}}(j) = 0$

$$t_j \equiv \frac{\hat{\boldsymbol{w}}_{\mathcal{I}}(j)}{\sqrt{\sigma^2 \boldsymbol{h}_0(j)}} \sim N(0,1) \tag{B.9}$$

Third, developing what t_j represents in terms of $\boldsymbol{y}$ and σ^2 by performing all the operations in (B.1), we can see that:

$$t_j = \frac{\left(\frac{1}{i_{j+1}-i_j} \sum_{m=i_j+1}^{i_{j+1}} y_m\right) - \left(\frac{1}{i_j-i_{j-1}} \sum_{m=i_{j-1}+1}^{i_j} y_m\right)}{\sigma\sqrt{\frac{1}{i_{j+1}-i_j} + \frac{1}{i_j-i_{j-1}}}} \tag{B.10}$$

which can be interpreted as the difference between the sample mean of the right and the left segment of i_j breakpoint divided by the square root of the variance of that difference. Even if the noise is not normal, but has a finite variance σ^2, (B.10) tells us that as the size of the segments increases, under the null hypothesis of no difference, t_j will converge to $N(0,1)$ because of the central limit theorem.

Recalculation of the weights after each removal, can be done efficiently with very few (a constant amount of) operations using the weights already calculated (see lines 9,12 and 16 on Algorithm 4). Thus the overall order of complexity to rank a breakpoint set $\mathcal{I}$ is linear with the size of the set $O(|\mathcal{I}|)$.

Appendix C

Adjustment of the SBL and BE parameters in GADA

Both the SBL or the BE procedure could be used independently to estimate copy number changes. However, the best results and flexibility are obtained with the combination of these two algorithms that was discussed in Section 3.7.

The objectives of this appendix are: 1) show that SBL and BE elimination produce breakpoint sets that are subsets of those obtained from higher sparseness settings, higher T or a, and can produce equivalent breakpoint sets; 2) propose a strategy for efficient parameter adjustment in the most general case; 3) evaluate the effectiveness of this strategy in the simulated dataset [103].

The experiments consist of drawing simulated chromosomes of different lengths M (M=100, 200, 500, 1000 and 2000 probes per chromosome) in the following conditions:

i. Simulation of null hypothesis (no breakpoints) using normal noise with different levels of variance.

ii. Simulation of normal copy number variations (few breakpoints and short segments) with real noise obtained by randomly sampling segments of data of size M from a

pool of a normal (diploid genome) CEPH cell line samples analyzed by Affymetrix 250K Nsp array platform.

iii. Simulation on cancer copy number variations, by sampling random chunks of data of size M from cancer samples analyzed in Section 3.9.3.

iv. Evaluation on the simulated dataset analyzed in Section 3.8.3, (only M=100).

For i. to iii. we simulated $L = 10000$ chromosomes, for the last case iv. all the $L = 500 \times 20 = 2000$ chromosomes of size $M = 100$ were used. Each sample, i.e. chromosome, was analyzed with different options for a and T, and the returned breakpoint sets were evaluated using different metrics. The sparseness of each set was computed as the number of returned breakpoints divided by the size of the chromosome, i.e. $|\mathcal{I}|/M$. λ denotes the average sparseness across all samples. When comparing two breakpoint sets $\mathcal{A}$ and $\mathcal{B}$ obtained for the same sample but with different parameter settings, we denote $\mathcal{A} \cap \mathcal{B}$ the set of common breakpoints, which in our case includes all breakpoints in $\mathcal{A}$ such that there exists a breakpoint in $\mathcal{B}$ less than δ probes away (if there are two breakpoints in $\mathcal{A}$ closer than δ to a breakpoint on $\mathcal{B}$ then only the closest one is assigned to the intersection). We then computed the averages of the following metrics [56] along the L simulated samples:

$$P(\mathcal{A} = \mathcal{B}) = \frac{|\mathcal{A} \cap \mathcal{B}|}{|\mathcal{A} \cup \mathcal{B}|} \qquad \text{(C.1)} \qquad\qquad P(\mathcal{A} \subset \mathcal{B}) = \frac{|\mathcal{A} \cap \mathcal{B}|}{|\mathcal{A}|} \qquad \text{(C.2)}$$

which represent the proportion of breakpoints that are the same on both sets, i.e., concordance (C.1); and the proportion of breakpoints on $\mathcal{A}$ that are also in $\mathcal{B}$, i.e., inclusiveness of $\mathcal{A}$ in $\mathcal{B}$ (C.2).

C.1 Experiments adjusting a and T in GADA

In all four panels of Figure C.1, we can see that in the initial breakpoint sets provided by SBL (at $T = 0$), a higher a setting increases sparseness (lower plots in each panel); but at the same time the breakpoints remain the same since the $P(\mathcal{A} \subset \mathcal{B}) > 99\%$ in all the cases. That means that breakpoint sets obtained with higher a tend to be subsets of those obtained with lower a.

As T increases we can see on the lower plots of each panel that we are monotonically obtaining sparser sets. The breakpoints that we are removing with BE might be different depending on the initial conditions; for example, $a = 0.8$ already has a high degree of sparseness so it will not start removing anything until $T > 2.88$, where the sparseness will start to curb down and eventually will converge to the curves obtained with lower a. On the top plot, we can see that this convergence is not only on the degree of sparseness but also on the breakpoint sets themselves too, since as T increases concordance goes to 1. That means that as we increase T we remove the extra part that it was in the breakpoint set obtained with lower a and we end up with the same breakpoints. Following the example with $a = 0.8$, we can see in Figure C.1 (A), that for $T > 4.15$, the concordance to starting with a lower a is higher than 80%; and for $T > 4.25$ and $T > 4.35$ we obtain concordances that are respectively higher than 90% and 95%.

These results indicate that we can adjust the sparseness of the resulting segmentation equivalently with a and T in a wide margin of settings to give the same breakpoint set. This behavior has been observed in all the experimental settings (i.–iv.). If there is something to be detected, true copy number alterations or outliers (ii.) then the probability of detection is higher and the high concordance is reached for smaller values of T than in the (i.) case (compare A and C , and B and C, on Figure C.1). For (iii.) case (data not shown) the concordance is even higher since cancer samples contain more CNA. The size of the chromosome M also has some impact on the convergence; on chromosomes with larger M high concordance is reached at a higher T, but for $M > 2000$ it does not move further more to the right. Additionally, our results on case (i.) are exactly the same for different noise power σ^2 because both a and T have already been corrected by $\hat{\sigma}^2$.

C.2 Strategy to adjust a and T in GADA

Adjusting sparseness with T can be done at no additional computational cost, while adjusting a requires to run the EM algorithm again. Thus, a good strategy is to select a small a for SBL, i.e., one that provides an initial breakpoint set that reduces most of the unlikely breakpoints but still ensures a high sensitivity. The first step is sufficiently sensitive so as not to miss any breakpoints that would require us to switch back to a lower a. Then, the final degree of sparseness will be adjusted with T.

From the previous experiment in concordance between sets (Figure C.1), we can see that a good sensitivity means that we do not remove anything that would not be removed with a lower a at the same T. The worst case, i.e. requiring a higher T for the same

concordance, is where there is nothing to be detected (Figure C.1 A and B); or when the true copy number alterations are very short and small (the hardest to be detected).

Moreover, dense arrays (higher M) will be more sensitive because CNA will be sampled with more probes and will produce statistically larger t (compare panels A and C to B and D in Figure C.1). Thus, small arrays will be those requiring the smallest T to be highly sensitive. Even very small arrays with 100 probes per chromosome, $T = 4$ provides enough initial sensitivity. Thus, we find that $a = 0.2$ should be small enough in general, and is the value that we have used in all the results on Section 3.8. In Chapter 5 and 6 we increase a up to 0.5 because the size of the arrays M is much larger and we can use a higher T with similar initial sensitivity.

A practical approach to ensure that the parameter a is well chosen for a particular setting of T on real data is the following. Assuming that $a = 0.2$ (or any other choice) we can check that is small enough for a particular T of interest by rerunning the algorithm with a lower a, e.g. $a = a/2$, and checking if the set of breakpoints returned for that particular T and different a's are essentially the same (e.g., $> 95\%$ concordant).

C.3 Sensitivity to the adjustments of a and T

We will use the simulation case (iv.), to evaluate the impact of the parameter setting strategy described in the previous section in terms of accuracy. This is the same dataset as the one used in Section 3.8.3 and by [103], where the underlying breakpoints are known, so we can exactly evaluate the FDR and the sensitivity for different choices of T and a.

In Figure C.2, curves corresponding to different a have different starting point in terms of sensitivity and FDR, but as T increases we decrease the FDR and similar operational points in terms of sensitivity and FDR are reached compared to those obtained from different a. The proposed $a = 0.2$ in Section C.2 offers an initial sensitivity and FDR such that all the remaining points in the curve are reached adjusting only T, providing all the levels of sensitivity or FDR that we might be interested in using without having to switch to another a.

Compared to CBS, we are able to obtain a wider margin of operating points of the PROC curve. Moreover, independently of the initial a we always have a point with similar or better average performance either in terms of FDR or sensitivity.

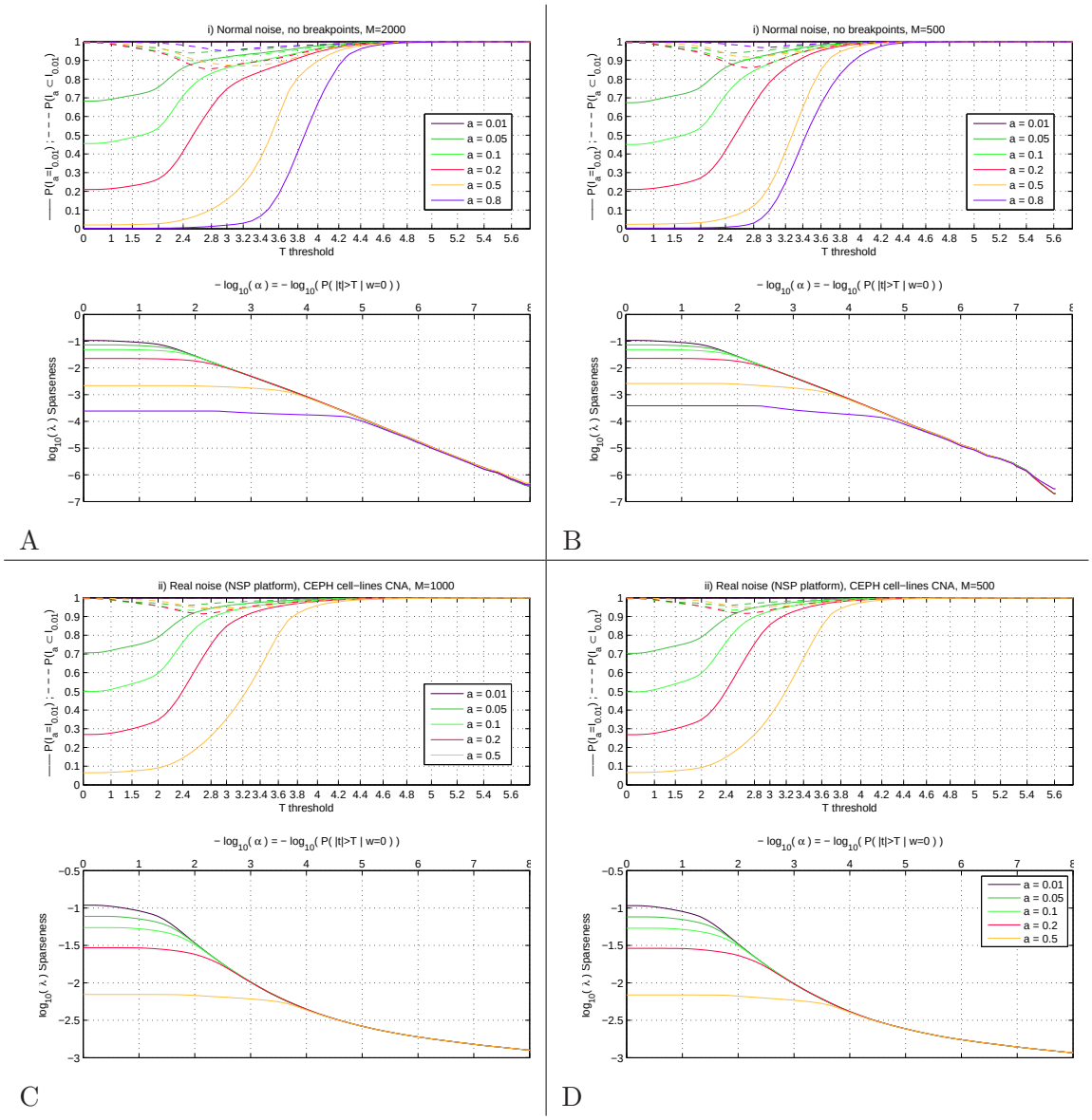


Figure C.1: The four panels (A,B,C and D) represent a different experimental dataset, with the results of applying different settings of a and T parameters. Each color corresponds to different setting of $a = 0.01, 0.05, 0.1, 0.2, 0.5$, and the x-axis increasing values of T or its associated significance level $log_{10}(\alpha)$. On the top plot we have represented the inclusiveness $P(\mathcal{I}_a \subset \mathcal{I}_{a=0.01})$ (dashed line); and the concordance $P(\mathcal{I}_a = \mathcal{I}_{a=0.01})$. The concordant breakpoints are defined within a window of $\delta = 2$ probes. The bottom plot represents the sparseness which on A and B also represent specificity because there are no underlying breakpoints. A and B use the normal noise simulation described in i. with chromosome lengths of $M = 500$ and $M = 2000$ (different noise levels σ^2 generate exactly the same curves); and C and D use the simulation described in ii. with chromosome lengths of $M = 500$ and $M = 1000$.

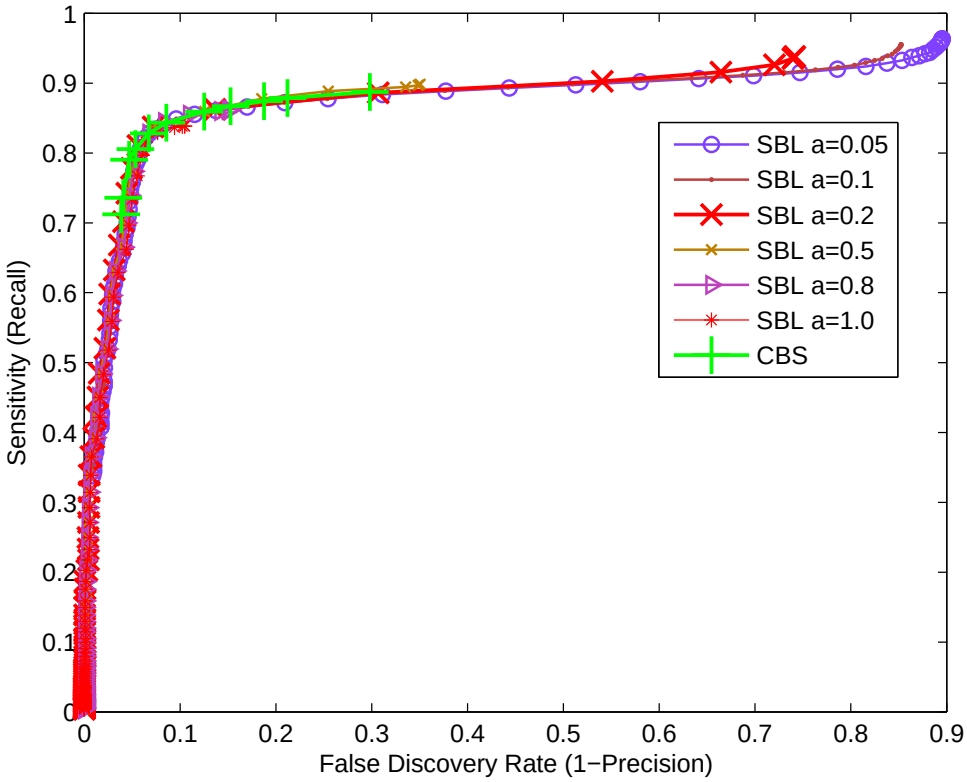


Figure C.2: PR operational curves for sensitivity vs. false discovery rate in detecting copy number changes within $\delta = 2$ probe window. Each line corresponds to SBL+BE with different starting breakpoint sets ($a = 0.05, 0.1, 0.2, 0.5, 0.8, 1.0, 1.5$) and varying T (T increases as we traverse the curve from right to left, i.e. FDR decreases). The light green curve represents the operating points obtained by CBS with different $\alpha = 1E-4, 0.001, 0.002, 0.005, 0.01, 0.05$

Bibliography

[1] Genome-wide human snp array 6.0 sample data set. Affymetrix, 2007. http://www.affymetrix.com/support/technical/sample_data/genomewide_snp6_data.affx.

[2] B. Alberts. *Molecular biology of the cell*. Garland Science, New York, 5th edition, 2008.

[3] D. G. Albertson, C. Collins, F. McCormick, and J. W. Gray. Chromosome aberrations in solid tumors. *Nat Genet*, 34(4):369–76, 2003.

[4] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc Natl Acad Sci U S A*, 96(12):6745–50, 1999.

[5] C. Ambroise and G. J. McLachlan. Selection bias in gene extraction on the basis of microarray gene-expression data. *Proc Natl Acad Sci U S A*, 99(10):6562–6, 2002.

[6] S. Asgharzadeh, R. Pique-Regi, R. Sposto, H. Wang, Y. Yang, H. Shimada, K. Matthay, J. Buckley, A. Ortega, and R. C. Seeger. Prognostic significance of gene expression profiles of metastatic neuroblastomas lacking MYCN gene amplification. *J Natl Cancer Inst*, 98(17):1193–203, 2006.

[7] H. Bengtsson, R. Irizarry, B. Carvalho, and T. P. Speed. Estimation and assessment of raw copy numbers at the single locus level. *Bioinformatics*, 24(6):759–767, 2008.

[8] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society*, 57:289–300, 1995.

[9] Y. Benjamini and D. Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Annals of Statistics*, 29:1165–1188, 2001.

[10] T. Bo and I. Jonassen. New feature subset selection procedures for classification of expression profiles. *Genome Biol*, 3(4):RESEARCH0017, 2002.

[11] P. Broet and S. Richardson. Detection of gene copy number changes in CGH microarrays using a spatially correlated mixture model. *Bioinformatics*, 22(8):911–8, 2006.

[12] E. J. Candès and J. K. Romberg. Robust signal recovery from incomplete observations. In *ICIP*, pages 1281–1284, 2006.

[13] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

[14] V. G. Cheung, R. S. Spielman, K. G. Ewens, T. M. Weber, M. Morley, and J. T. Burdick. Mapping determinants of human gene expression by regional and genome-wide association. *Nature*, 437(7063):1365–9, 2005.

[15] F. S. Collins, E. D. Green, A. E. Guttmacher, and M. S. Guyer. A vision for the future of genomics research. *Nature*, 422(6934):835–847, 2003.

[16] T. I. H. G. M. Consortium. A physical map of the human genome. *Nature*, 409(6822):934–941, 2001.

[17] S. J. Diskin, T. Eck, J. Greshock, Y. P. Mosse, T. Naylor, J. Stoeckert, C. J., B. L. Weber, J. M. Maris, and G. R. Grant. STAC: A method for testing the significance of DNA copy number aberrations across multiple array-CGH experiments. *Genome Res*, 16(9):1149–58, 2006.

[18] S. J. Diskin, M. Li, C. Hou, S. Yang, J. Glessner, H. Hakonarson, M. Bucan, J. M. Maris, and K. Wang. Adjustment of genomic waves in signal intensities from whole-genome snp genotyping platforms. *Nucleic Acids Res*, 36(19):e126, 2008.

[19] M. N. Do and M. Vetterli. The contourlet transform: an efficient directional multiresolution image representation. *Image Processing, IEEE Transactions on*, 14(12):2091–2106, —2005—.

[20] D. Donoho, M. Elad, and V. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Trans. Information Theory*, 52(1):6–18, Jan. 2006.

[21] P. Dragotti and M. Vetterli. Wavelet footprints: Theory, algorithms, and applications. *IEEE-Trans-SP*, 51(5):1306–1323, May 2002.

[22] R. Duda, P. Hart, and D. Stork. *Pattern Classification.* John Wiley and Sons, 2001. 0-471-05669-3.

[23] S. Dudoit, J. Fridlyand, and T. P. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 97(457):77–87, 2002.

[24] B. Efron and R. Tibshirani. *An introduction to the bootstrap.* Monographs on statistics and applied probability. Chapman Hall, New York, —1993—.

[25] D. A. Engler, G. Mohapatra, D. N. Louis, and R. A. Betensky. A pseudolikelihood approach for simultaneous analysis of array comparative genomic hybridizations. *Biostatistics*, 7(3):399–421, 2006.

[26] L. Feuk, A. R. Carson, and S. W. Scherer. Structural variation in the human genome. *Nat Rev Genet*, 7(2):85–97, 2006.

[27] D. Fredman, S. J. White, S. Potter, E. E. Eichler, J. T. Den Dunnen, and A. J. Brookes. Complex snp-related sequence variation in segmental genome duplications. *Nat Genet*, 36(8):861–6, 2004.

[28] J. L. Freeman, G. H. Perry, L. Feuk, R. Redon, S. A. McCarroll, D. M. Altshuler, H. Aburatani, K. W. Jones, C. Tyler-Smith, M. E. Hurles, N. P. Carter, S. W. Scherer, and C. Lee. Copy number variation: new insights in genome diversity. *Genome Res*, 16(8):949–61, 2006.

[29] J. Fridlyand, A. M. Snijders, D. Pinkel, D. G. Albertson, and A. N. A. N. Jain. Hidden markov models approach to the analysis of array cgh data. *Journal of Multivariate Analysis*, 90(1):132–153, 2004.

[30] J. H. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84:165–175, 1989.

[31] L. A. Garraway, H. R. Widlund, M. A. Rubin, G. Getz, A. J. Berger, S. Ramaswamy, R. Beroukhim, D. A. Milner, S. R. Granter, J. Du, C. Lee, S. N. Wagner, C. Li, T. R. Golub, D. L. Rimm, M. L. Meyerson, D. E. Fisher, and W. R. Sellers. Integrative genomic analyses identify MITF as a lineage survival oncogene amplified in malignant melanoma. *Nature*, 436(7047):117–22, 2005.

[32] A. Gelman, J. B. Carlin, H. Stern, and D. B. Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, New York, 2 edition, 2004.

[33] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–7, 1999.

[34] Y. Guo, T. Hastie, and R. Tibshirani. Regularized linear discriminant analysis and its application in microarrays. *Biostatistics*, 8(1):86–100, 2007.

[35] A. E. Guttmacher and F. S. Collins. Genomic medicine–a primer. *N Engl J Med*, 347(19):1512–20, 2002.

[36] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[37] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Mach. Learn.*, 46(1-3):389–422, 2002.

[38] HapMap Consortium. The International HapMap Project. *Nature*, 426(6968):789–96, 2003. `http://www.hapmap.org`.

[39] T. Hastie, A. Buja, and R. Tibshirani. Penalized discriminant analysis. *Annals of Statistics*, 23:73–102, 1995.

[40] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, July 2001.

[41] J. P. Hoffbeck and D. A. Landgrebe. Covariance matrix estimation and classification with limited training data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(7):763–767, 1996.

[42] L. Hsu, S. G. Self, D. Grove, T. Randolph, K. Wang, J. J. Delrow, L. Loo, and P. Porter. Denoising array-based comparative genomic hybridization data using wavelets. *Biostatistics*, 6(2):211–26, 2005.

[43] J. Huang, W. Wei, J. Zhang, G. Liu, G. R. Bignell, M. R. Stratton, P. A. Futreal, R. Wooster, K. W. Jones, and M. H. Shapero. Whole genome DNA copy number changes identified by high density oligonucleotide arrays. *Hum Genomics*, 1(4):287–99, 2004.

[44] T. Huang, B. Wu, P. Lizardi, and H. Zhao. Detection of DNA copy number alterations using penalized least squares regression. *Bioinformatics*, 21(20):3811–7, 2005.

[45] T. J. P. Hubbard, B. L. Aken, K. Beal, B. Ballester, M. Caccamo, Y. Chen, L. Clarke, G. Coates, F. Cunningham, T. Cutts, T. Down, S. C. Dyer, S. Fitzgerald, J. Fernandez-Banet, S. Graf, S. Haider, M. Hammond, J. Herrero, R. Holland, K. Howe, K. Howe, N. Johnson, A. Kahari, D. Keefe, F. Kokocinski, E. Kulesha, D. Lawson, I. Longden, C. Melsopp, K. Megy, P. Meidl, B. Ouverdin, A. Parker, A. Prlic, S. Rice, D. Rios, M. Schuster, I. Sealy, J. Severin, G. Slater, D. Smedley, G. Spudich, S. Trevanion, A. Vilella, J. Vogel, S. White, M. Wood, T. Cox, V. Curwen, R. Durbin, X. M. Fernandez-Suarez, P. Flicek, A. Kasprzyk, G. Proctor, S. Searle, J. Smith, A. Ureta-Vidal, and E. Birney. Ensembl 2007. *Nucl. Acids Res.*, page gkl996, 2006.

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

[47] A. J. Iafrate, L. Feuk, M. N. Rivera, M. L. Listewnik, P. K. Donahoe, Y. Qi, S. W. Scherer, and C. Lee. Detection of large-scale variation in the human genome. *Nat Genet*, 36(9):949–51, 2004.

[48] S. Issue. Sensing, sampling, and compression. *Signal Processing Magazine, IEEE*, 25(2), 2008.

[49] R. C. Jansen and J. P. Nap. Genetical genomics: the added value from segregation. *Trends Genet*, 17(7):388–91, 2001.

[50] S. Jean-Luc, E. J. Candes, and D. L. Donoho. The curvelet transform for image denoising. *Image Processing, IEEE Transactions on*, 11(6):670–684, —2002—.

[51] A. Kallioniemi, O. P. Kallioniemi, D. Sudar, D. Rutovitz, J. W. Gray, F. Waldman, and D. Pinkel. Comparative genomic hybridization for molecular cytogenetic analysis of solid tumors. *Science*, 258(5083):818–21, 1992.

[52] W. J. Kent, C. W. Sugnet, T. S. Furey, K. M. Roskin, T. H. Pringle, A. M. Zahler, and other. The Human Genome Browser at UCSC. *Genome Res.*, 12(6):996–1006, 2002.

[53] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[54] D. Komura, F. Shen, S. Ishikawa, K. R. Fitch, G. Liu, S. Ihara, H. Nakamura, M. E. Hurles, C. Lee, S. W. Scherer, K. W. Jones, M. H. Shapero, J. Huang, and H. Aburatani. Genome-wide detection of human copy number variations using high-density DNA oligonucleotide arrays. *Genome Res*, 16(12):1575–84, 2006.

[55] J. M. Korn, F. G. Kuruvilla, S. A. McCarroll, A. Wysoker, J. Nemesh, S. Cawley, E. Hubbell, J. Veitch, P. J. Collins, K. Darvishi, C. Lee, M. M. Nizzari, et al. Integrated genotype calling and association analysis of snps, common copy number polymorphisms and rare cnvs. *Nat Genet*, 40(10):1253–60, 2008.

[56] B. Kosko. Probable equivalence, superpower sets, and superconditionals. *International Journal of Intelligent Systems*, 19(12):1151–1171, 2004.

[57] W. R. Lai, M. D. Johnson, R. Kucherlapati, and P. J. Park. Comparative analysis of algorithms for identifying amplifications and deletions in array CGH data. *Bioinformatics*, 21(19):3763–70, 2005.

[58] T. Lal, O. Chapelle, J. Weston, and A. Elisseeff. *Embedded methods*, chapter 6. Springer, 2005.

[59] E. S. Lander and P. Green. Construction of multilocus genetic linkage maps in humans. *Proc Natl Acad Sci U S A*, 84(8):2363–7, 1987.

[60] D. Lipson, Y. Aumann, A. Ben-Dor, N. Linial, and Z. Yakhini. Efficient calculation of interval scores for DNA copy number data analysis. *J Comput Biol*, 13(2):215–28, 2006.

[61] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE-Trans-SP*, 41(12):3397–3415, 1993.

[62] S. G. Mallat. *A wavelet tour of signal processing*. Academic, San Diego, Calif. ; London, 2nd edition, —1999—.

[63] J. Marioni, N. Thorne, A. Valsesia, T. Fitzgerald, R. Redon, T. D. Andrews, B. Stranger, E. Dermitzakis, N. Carter, S. Tavare, and M. Hurles. Breaking the waves: improved detection of copy number variation from microarray-based comparative genomic hybridization. *Genome Biology*, 8(10):R228, 2007.

[64] J. C. Marioni, N. P. Thorne, and S. Tavare. BioHMM: a heterogeneous hidden markov model for segmenting array CGH data. *Bioinformatics*, 22(9):1144–6, 2006.

[65] S. A. McCarroll, F. G. Kuruvilla, J. M. Korn, S. Cawley, J. Nemesh, A. Wysoker, M. H. Shapero, P. I. de Bakker, J. B. Maller, A. Kirby, A. L. Elliott, M. Parkin, E. Hubbell, T. Webster, R. Mei, et al. Integrated detection and population-genetic analysis of snps and copy number variation. *Nat Genet*, 40(10):1166–74, 2008.

[66] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley, 1997.

[67] M. Morley, C. M. Molony, T. M. Weber, J. L. Devlin, K. G. Ewens, R. S. Spielman, and V. G. Cheung. Genetic analysis of genome-wide variation in human gene expression. *Nature*, 430(7001):743–7, 2004.

[68] Y. Nannya, M. Sanada, K. Nakazaki, N. Hosoya, L. Wang, A. Hangaishi, M. Kurokawa, S. Chiba, D. K. Bailey, G. C. Kennedy, and S. Ogawa. A robust algorithm for copy number detection using high-density oligonucleotide single nucleotide polymorphism genotyping arrays. *Cancer Res*, 65(14):6071–9, 2005.

[69] A. B. Olshen, E. S. Venkatraman, R. Lucito, and M. Wigler. Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics*, 5(4):557–72, 2004.

[70] Y. Pati, R. Rezaiifar, and P. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of the 27 th Annual Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, Nov 1993.

[71] D. A. Peiffer, J. M. Le, F. J. Steemers, W. Chang, F. Garcia, K. Haden, J. Li, C. A. Shaw, J. Belmont, S. W. Cheung, R. M. Shen, D. L. Barker, and K. L. Gunderson. High-resolution genomic profiling of chromosomal aberrations using infinium whole-genome genotyping. *Genome Res*, 16(9):1136–48, 2006.

[72] G. H. Perry, A. Ben-Dor, A. Tsalenko, N. Sampas, L. Rodriguez-Revenga, C. W. Tran, A. Scheffer, I. Steinfeld, P. Tsang, N. A. Yamada, H. S. Park, J.-I. Kim, J.-S. Seo, Z. Yakhini, S. Laderman, et al. The fine-scale and complex architecture of human copy-number variation. *American journal of human genetics*, 82(3):685–695, 2008.

[73] F. Picard, S. Robin, M. Lavielle, C. Vaisse, and J. J. Daudin. A statistical approach for array CGH data analysis. *BMC Bioinformatics*, 6:27, 2005.

[74] R. Pique-Regi, J. Monso-Varona, A. Ortega, and S. Asgharzadeh. Bayesian detection of recurrent copy number alterations across multiple array samples. In *Genomic Signal Processing and Statistics, 2008. GENSiPS 2008. IEEE International Workshop on*, pages 1–4, 2008.

[75] R. Pique-Regi, J. Monso-Varona, A. Ortega, R. C. Seeger, T. J. Triche, and S. Asgharzadeh. Sparse representation and bayesian detection of genome copy number alterations from microarray data. *Bioinformatics*, 24(3):309–18, 2008.

[76] R. Pique-Regi, J. Morrison, and D. C. Thomas. Bayesian hierarchical modelling of means and covariances of gene expression data within families. In *BMC Genetic Analysis Workshop Proceedings*, Nov. 2006.

[77] R. Pique-Regi and A. Ortega. Block diagonal linear discriminant analysis with sequential embedded feature selection. In *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, volume 5, 2006.

[78] R. Piqué-Regí, A. Ortega, and S. Asgharzadeh. Sequential diagonal linear discriminant analysis (SeqDLDA) for microarray classification and gene identification. In *Computational Systems and Bioinformatics*, Aug. 2005.

[79] R. Pique-Regi, E. S. Tsau, A. Ortega, R. C. Seeger, and S. Asgharzadeh. Wavelet footprints and sparse bayesian learning for DNA copy number change analysis. In *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, April 2007.

[80] J. R. Pollack, C. M. Perou, A. A. Alizadeh, M. B. Eisen, A. Pergamenschikov, C. F. Williams, S. S. Jeffrey, D. Botstein, and P. O. Brown. Genome-wide analysis of DNA copy-number changes using cDNA microarrays. *Nat Genet*, 23(1):41–6, 1999.

[81] J. Quackenbush. Computational analysis of microarray data. *Nat Rev Genet*, 2(6):418–27, 2001.

[82] R. Redon, S. Ishikawa, K. R. Fitch, L. Feuk, G. H. Perry, T. D. Andrews, H. Fiegler, M. H. Shapero, A. R. Carson, W. Chen, E. K. Cho, S. Dallaire, J. L. Freeman, J. R. Gonzalez, M. Gratacos, et al. Global variation in copy number in the human genome. *Nature*, 444(7118):444–54, 2006.

[83] G. Rigaill, P. Hupe, A. Almeida, P. La Rosa, J.-P. Meyniel, C. Decraene, and E. Barillot. ITALICS: an algorithm for normalization and DNA copy number calling for affymetrix snp arrays. *Bioinformatics*, 24(6):768–774, 2008.

[84] C. Rouveirol, N. Stransky, P. Hupe, P. L. Rosa, E. Viara, E. Barillot, and F. Radvanyi. Computation of recurrent minimal genomic alterations from array-CGH data. *Bioinformatics*, 22(7):849–56, 2006.

[85] O. M. Rueda and R. Diaz-Uriarte. Flexible and accurate detection of genomic copy-number changes from aCGH. *PLoS Comput Biol*, 3(6):e122, 2007.

[86] E. E. Schadt, C. Li, B. Ellis, and W. H. Wong. Feature extraction and normalization algorithms for high-density oligonucleotide gene expression array data. *J Cell Biochem Suppl*, Suppl 37:120–5, 2001.

[87] E. E. Schadt, S. A. Monks, T. A. Drake, A. J. Lusis, N. Che, V. Colinayo, T. G. Ruff, S. B. Milligan, J. R. Lamb, G. Cavet, P. S. Linsley, M. Mao, R. B. Stoughton, and S. H. Friend. Genetics of gene expression surveyed in maize, mouse and man. *Nature*, 422(6929):297–302, 2003.

[88] G. A. F. Seber and A. J. Lee. *Linear Regression Analysis*. John Wiley, New York, second edition, 2003.

[89] S. P. Shah, W. L. Lam, R. T. Ng, and K. P. Murphy. Modeling recurrent DNA copy number alterations in array CGH data. *Bioinformatics*, 23(13):i450–8, 2007.

[90] S. P. Shah, X. Xuan, R. J. DeLeeuw, M. Khojasteh, W. L. Lam, R. Ng, and K. P. Murphy. Integrating copy number polymorphisms into array CGH analysis using a robust HMM. *Bioinformatics*, 22(14):e431–9, 2006.

[91] L. Sheng, R. Pique-Regi, A. Asgharzadeh, and A. Ortega. Microarray classification using block diagonal linear discriminant analysis with embeddded feature selection. In *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, April 2009.

[92] R. Simon. Development and Evaluation of Therapeutically Relevant Predictive Classifiers Using Gene Expression Profiling. *J. Natl. Cancer Inst.*, 98(17):1169–1171, 2006.

[93] D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V. D'Amico, J. P. Richie, E. S. Lander, M. Loda, P. W. Kantoff, T. R. Golub, and W. R. Sellers. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1(2):203–9, 2002.

[94] B. E. Stranger, M. S. Forrest, A. G. Clark, M. J. Minichiello, S. Deutsch, R. Lyle, S. Hunt, B. Kahl, S. E. Antonarakis, S. Tavare, P. Deloukas, and E. T. Dermitzakis. Genome-wide associations of gene expression variation in humans. *PLoS Genet*, 1(6):e78, 2005.

[95] D. C. Thomas, D. Qian, W. J. Gauderman, K. Siegmund, and J. L. Morrison. A generalized estimating equations approach to linkage analysis in sibships in relation to multiple markers and exposure factors. *Genet Epidemiol*, 17 Suppl 1:S737–42, 1999.

[96] R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proc Natl Acad Sci U S A*, 99(10):6567–72, 2002.

[97] M. E. Tipping. Sparse Bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.*, 1:211–244, 2001.

[98] J. A. Tropp. Greed is good: algorithmic results for sparse approximation. *IEEE Trans. Information Theory*, 50(10):2231–2242, 2004.

[99] Y. Tsaig and D. L. Donoho. Compressed sensing. *IEEE Trans. Inform. Theory*, 52:1289–1306, 2006.

[100] A. Tsalenko, R. Sharan, H. Edvardsen, V. Kristensen, A. L. Borresen-Dale, A. Ben-Dor, and Z. Yakhini. Analysis of snp-expression association matrices. *Proc IEEE Comput Syst Bioinform Conf*, pages 135–43, 2005.

[101] E. S. Venkatraman and A. B. Olshen. A faster circular binary segmentation algorithm for the analysis of array CGH data. *Bioinformatics*, NIL(NIL):NIL, 2007.

[102] M. Vetterli and J. Kovacevic. *Wavelets and subband coding*. Prentice Hall PTR, Englewood Cliffs, N.J., —1995—.

[103] H. Willenbrock and J. Fridlyand. A comparison study: applying segmentation to array CGH data for downstream analyses. *Bioinformatics*, 21(22):4084–91, 2005.

[104] D. Wipf and B. Rao. Sparse Bayesian learning for basis selection. *IEEE-Trans-SP*, 52(8):2153–2164, Aug. 2004.

[105] M. Xiong, W. Li, J. Zhao, L. Jin, and E. Boerwinkle. Feature (gene) selection in gene expression-based tumor classification. *Mol Genet Metab*, 73(3):239–47, 2001.

[106] J. Ye, T. Li, T. Xiong, and R. Janardan. Using uncorrelated discriminant analysis for tissue classification with gene expression data. *IEEE/ACM Trans Comput Biol Bioinform*, 1(4):181–90, 2004.

[107] X. Zhao, C. Li, J. G. Paez, K. Chin, P. A. Janne, T. H. Chen, L. Girard, J. Minna, D. Christiani, C. Leo, J. W. Gray, W. R. Sellers, and M. Meyerson. An integrated view of copy number and allelic alterations in the cancer genome using single nucleotide polymorphism arrays. *Cancer Res*, 64(9):3060–71, 2004.