Statistical Methods for High-Throughput Genomic Data

Kristof De Beuf

Supervisors:

Prof. dr. Olivier Thas

Prof. dr. Lieven Clement

Prof. dr. Jean-Pierre Ottoy

A thesis submitted in partial fulfillment of the requirements

for the degree of Doctor in Statistical Data Analysis

Academic year 2012-2013

Please refer to this work as follows:

Kristof De Beuf, 2013. Statistical Methods for High-Throughput Genomic Data. Ph.D. thesis, Ghent University, Gent, Belgium.

ISBN number: 978-9-4619710-6-7

Dankwoord

Toegegeven, het gaf een wat vreemd gevoel om zo'n kleine drie jaar na het sluiten van de deur van het Boerekot als bio-ingenieur milieutechnologie, deze opnieuw te openen, gedecideerd om me ten gronde te verdiepen in de statistiek. Het ongestoord kunnen bestuderen van dat door mij nog grotendeels onverkende vakgebied in een vertrouwde omgeving bleek echter al gauw een gesmaakte combinatie. Nu hier na zes jaar het afgewerkte proefschrift naast me ligt, kan ik niet anders dan met tevredenheid en voldoening op de keuze van destijds terugkijken. Uiteraard komt zo'n werk niet zonder slag of stoot tot stand en zijn er heel wat mensen die rechtstreeks of onrechtstreeks, en al dan niet bewust, hun bijdrage hebben geleverd en daarom mijn oprechte dank hebben.

In het bijzonder wens ik vooreerst mijn promotoren Olivier Thas, Lieven Clement en Jean-Pierre Ottoy te bedanken. Olivier, ik vond het enorm prettig om onder jouw supervisie aan mijn doctoraat te kunnen werken, niet in het minst door de grote vrijheid die je me gaf en het vertrouwen dat je in mijn kunnen stelde. Jouw ongeremd enthousiasme en liefde voor de statistiek werkten - en werken nog steeds - heel aanstekelijk. Ik was steeds in bewondering voor de manier waarop je ogenschijnlijk moeiteloos statistische - maar ook biologische - concepten op een bevattelijke manier kan uitleggen, en voor de hoeveelheid witte konijnen die je in je hoge hoed hebt zitten als remedie tegen schijnbaar onoplosbare problemen. Maar wat ik misschien nog het meest apprecieerde is dat dit alles gepaard ging met een groot gevoel voor humor en relativeringsvermogen - ook als het wat moeilijker ging. Lieven, het was een ongelofelijk voorrecht en genoegen om met jou als begeleider aan onderzoek te kunnen doen. Aan de start was statistical genomics voor ons beiden grotendeels onbetreden gebied. Dat maakte de rit echter des te boeiender en uitdagender, waarbij jij je al snel opwierp als uitstekende gids, strooiend met immer sprankelende ideeën en constante aanmoedigingen. Dat ik vandaag een afgewerkt doctoraat kan afleveren met toch een aantal mooie resultaten heb ik dan ook voor een heel groot

deel aan jou te danken. Daarnaast konden we het ook op persoonlijk vlak goed vinden. Zo deed het altijd deugd om voor, na of tussen de discussies over wavelets en base-callers de nieuwste avonturen van de kindjes te delen, of om te dagdromen over de schoonheid van bepaalde muziekstukken. Enorm bedankt voor dit alles! Professor Ottoy, Jean-Pierre - dit mag ik nu wel zeggen, ik wil je graag bedanken voor de kans die je me gaf om als assistent aan de vakgroep te beginnen. Ik kijk ook met voldoening terug op de aangename manier van samenwerken bij het organiseren en geven van de lessen statistiek gedurende de afgelopen zes jaar.

I would like to thank Mark Robinson, Ziv Shkedy and Stijn Vansteelandt for taking the time to carefully read the thesis and for all their useful comments.

De collega-statistici van BioStat en de talrijke andere collega's van de vakgroep verdienen ook een pluim. De toffe en ongedwongen sfeer waar zij dagelijks in de gang, koffiekamer, resto of voetbalzaal voor zorgen, deden me telkens weer met veel goesting naar Gent komen. De ontelbare ongezouten meningen, muziektipuitwisselingen, humoristische uitspattingen en culinaire verwennerijtjes: ze werden allemaal erg geapprecieerd! Ellen en Heidi, bureaugenotes van het eerste uur, bedankt voor het warme welkom en de wegwijs in de begindagen. Peter, ik hoop dat ik niet te veel misbruik heb gemaakt van je legendarische opofferings- en zelfwegcijfervermogen bij weer eens een ICT-probleempje. Je was vooral een erg fijne collega! Yingjie, thank you for being such a lovely office mate. I will deeply miss your shiny "Goedemorgen". Jan, ik kon me geen betere compagnon de route voorstellen. Bedankt voor alle kleine en grote discussies, lachpartijen, en het luisterend oor. Je bent een wijze mens.

Bedankt ook aan alle vrienden, familie, zus en broer voor het meeleven en de interesse in het onderzoek, maar vooral voor alle warme momenten daarnaast die alles de moeite waard maken en telkens opnieuw de nodige energie verschaffen om er weer tegenaan te gaan. Mijn uitzonderlijke dank gaat uit naar mijn ouders. Mama en papa, bedankt voor alle kansen en mogelijkheden die jullie me hebben geboden en voor de onvoorwaardelijke steun in alles wat ik doe. Ik ondervind nu zelf dat opvoeden geen evidente opdracht is, en besef dan ook meer en meer dat jullie het fantastisch hebben gedaan!

Het ultieme woord van dank houd ik achter de hand voor mijn vier grootste schatten die me steeds intens gelukkig maken. Floris, Olivia en Marieke, jullie immense levenslust brengt elke dag opnieuw de zon in huis. Beatrijs, allerliefste schat, dat ik er uiteindelijk ben geraakt is evenzeer jouw verdienste als die van mij. Alsof het de normaalste zaak van de wereld was -

dat was het niet - hield je steeds de boel draaiende terwijl ik weer eens met mijn hoofd in de boeken zat of voor de laptop hing. Altijd was jij daar, stond je klaar om me op te beuren bij een tegenslag, me in mezelf te doen geloven bij twijfels en mee te vieren bij kleine overwinningen. Voor dit alles en nog veel meer: bedankt!

Kristof,

Lochristi, 24 maart 2013

Contents

List of abbreviations

A adenine. 2

ABA abscisic acid. 97

BFDR Bayesian false discovery rate. 51

C cytosine. 2

cDNA complementary DNA. 14

ChIP chromatin immunoprecipitation. 4

CNV copy number variation. 4

DNA deoxyribonucleic acid. 1

DWT discrete wavelet transform. 27

eCDF empirical cumulative distribution function. 132

FDR false discovery rate. 22

G guanine. 2

GAM generalized additive model. 146

GSEA gene set enrichment analysis. 88

HPL homopolymer length. 129

i.i.d. independent and identically distributed. 40

IDWT inverse discrete wavelet transform. 31

IRLS iteratively reweighted least squares. 148

MAD median absolute deviation. 34

MCMC Markov chain Monte Carlo. 50

MLE maximum likelihood estimator. 175

MM mismatch. 14

MML marginal maximum likelihood. 45

mRNA messenger RNA. 4

NGS next-generation sequencing. 3

OTU operational taxonomic unit. 194

PCA principal components analysis. 118

PCR polymerase chain reaction. 126

PGM personal genome machine. 192

PM perfect match. 14

PPV positive predictive value. 58

PTP PicoTiterPlate. 127

qRT-PCR quantitative real-time reverse transcription polymerase chain reaction. 90

RMA robust multi-array average. 19

RNA ribonucleic acid. 1

ROC receiver operating characteristic. 59

SCM structural change model. 22

SNP single nucleotide polymorphism. 4

SPC specificity. 58

T thymine. 2

TAR transcriptionally active region. 20

TPR true positive rate. 58

Chapter 1

General introduction

1.1 Study genomics to explain life

Biological scientists are intrigued by life. They want to understand the things that make life possible. They want to learn about the mechanisms that enable living organisms to reproduce and maintain themselves, but also how they behave, how they become diseased and can be cured again. Much of the information they are looking for can be found in the genome. This is the entirety of hereditary, and by extension, biological information possessed by any living organism. Hence, it comes as no surprise that genomics, which is the branch of molecular biology that studies genomes, has become a very important and widely researched scientific discipline. Broadly, the genetic information contained in an organism's cell or tissue acts at three different levels. These levels correspond with the three main macromolecules that are considered essential for all known forms of life, i.e. (1) deoxyribonucleic acid (DNA), (2) ribonucleic acid (RNA), and (3) proteins. The central dogma of molecular biology, first put forward by Francis Crick (Crick, 1958, 1970), describes the transfer of the genome's sequential information between these macromolecules (see Figure 1.1). For eukaryotic organisms, which are higher organisms such as humans that possess a nucleus in their cells, the general transfers consist of (1) DNA replication, (2) transcription of DNA to RNA, and (3) translation of RNA to proteins. Transcription is also known as DNA expression or gene expression and takes place inside the cell nucleus, while translation occurs outside of the nucleus in the cell cytoplasm.

Genomics research can be divided into several subdisciplines according to these three different

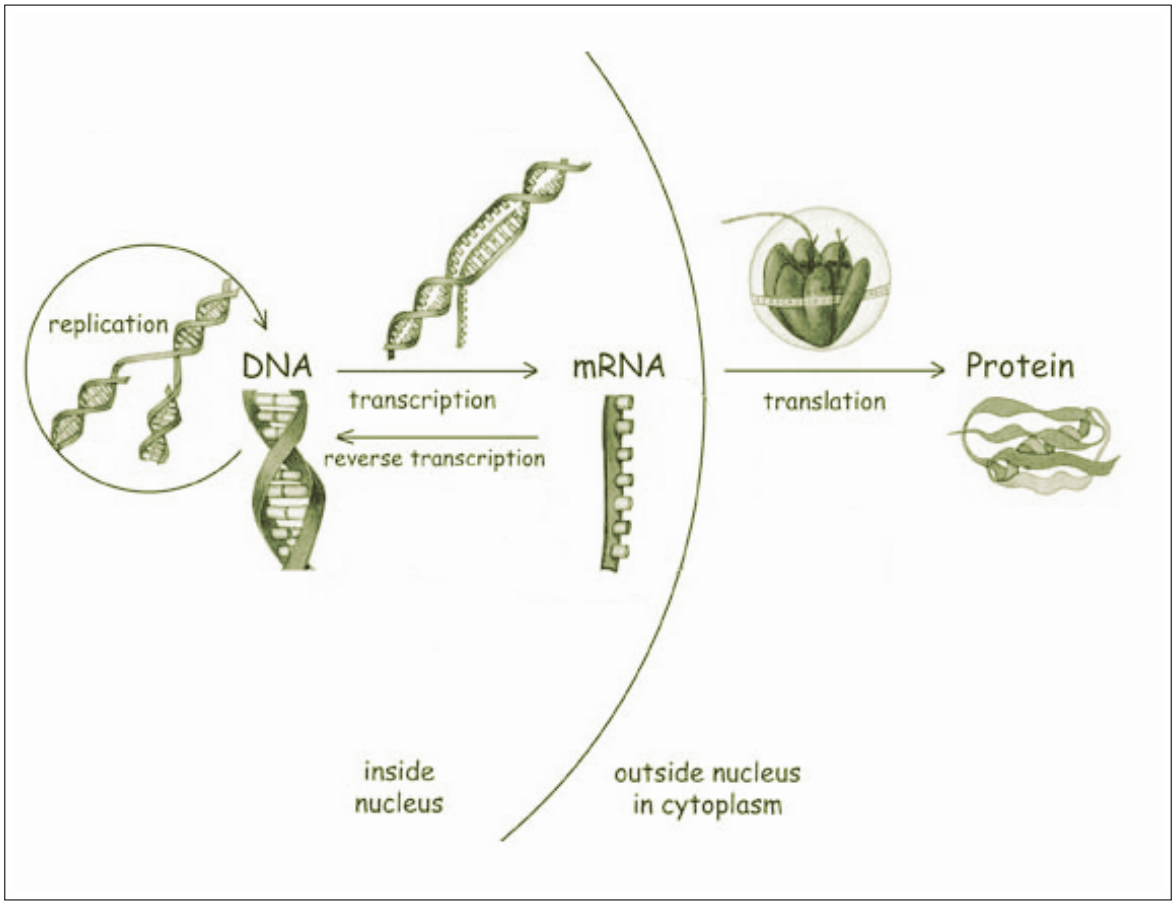


Figure 1.1: Central dogma of molecular biology (adapted from `www.exploringnature.org`)

levels of genetic information. At the level of DNA, one is mainly concerned with elucidating the sequence of the four nucleotides adenine (A), cytosine (C), thymine (T) and guanine (G). These are the building blocks of the DNA molecules giving rise to their primary structure. This task is called DNA sequencing. If detecting sequence variation with respect to some reference sequence or between alleles within an individual is of interest, one also often speaks about genotyping. This branch of genomics is referred to in this dissertation as *DNA genomics*. In literature, however, the term *genomics* is often used in a narrow sense as well when only genomics at the DNA level is involved, as opposed to the RNA and protein level. DNA genomics not only considers the elucidation of the sequence itself. A popular area of research that studies epigenetic modifications of the DNA is called epigenomics. These DNA modifications do not influence the primary DNA sequence itself, but still play a role in the regulation of gene expression. Main topics in epigenomics are the study of DNA methylation and histone modification. The subdisciplines of genomics at the second and third level of the genome's sequential information are transcriptomics, at the level of RNA, and proteomics, at the level of proteins. Both the transcriptome, which is the entire set of RNA transcripts or expressed DNA products,

and the proteome, which is the entire set of expressed proteins, are very dynamic. Within an organism, they can vary considerably between cells, tissues, developmental stages, points in time or environmental conditions. This is in contrast to the DNA genome, which is a more static phenomenon.

1.2 High-throughput genomics

A number of major technological advances in the last 15 years have led to a tremendous revolution in genomics research and the emergence of the high-throughput genomics era. The two most widespread and influential inventions are *DNA microarrays* and *high-throughput DNA sequencing*. The latter is also frequently called next-generation sequencing (NGS). While DNA microarrays are already commonly used from the end of the previous century, the first use of the NGS technology dates back to around 2005. Both new technologies enable the simultaneous interrogation of an increasing amount of cellular products such as genes or transcripts. They primarily play a role at the level of DNA (DNA genomics - epigenomics) and RNA (transcriptomics). At the level of proteins the most important technology is high-throughput mass spectrometry. However, proteomics is not discussed further here.

DNA microarrays and NGS are the two technologies for which statistical methods are developed in this dissertation. Roughly speaking, the common applications of both technologies in genomics research are rather similar. The most obvious exception to this statement is high-throughput DNA sequencing, which is always conducted with NGS technologies, while this can not be addressed with DNA microarrays. Table 1.1 gives an overview of the most common uses for these two technologies.

1.3 Statistical challenges in high-throughput genomics

High-throughput genomics technologies provide the opportunity for biological and biomedical research to make more rapid advancements than was possible before. However, drawing meaningful information from the massive amount of data that are produced often presents a huge bottleneck. When extracting knowledge from high-throughput genomics data, statistical methods are needed in order to quantify the uncertainties inherent to the various sources of vari-

ability contained in the data. Statistical challenges are encountered in every single step of the data-analytic process, often called *pipeline*.

Table 1.1: Overview of the most common uses of DNA microarrays and NGS platforms in genomics research.

Analysis type	*DNA microarrays*	*NGS platforms*
DNA sequence		high-throughput DNA sequencing
DNA sequence variation: single nucleotide polymorphism (SNP). A SNP is a variation in the DNA sequence of one nucleotide by another nucleotide.	SNP arrays for the detection of SNPs	SNP calling after high-throughput DNA sequencing
DNA sequence variation: copy number variation (CNV). A CNV is a form of structural variation where relatively large regions in the genome (10000 to several millions of nucleotides) are deleted or duplicated with respect to a reference genome.	comparative genomic hybridization of CNVs, SNP arrays	CNV-seq
transcription	messenger RNA (mRNA) analysis or gene expression profiling using microarrays	RNA-seq
protein-DNA interaction by means of chromatin immunoprecipitation (ChIP)	ChIP-on-chip	ChIP-seq
DNA methylation	Multiple array-based approaches, e.g. array-based bisulphite methylation profiling	Multiple sequencing-based approaches, e.g. bisulphite sequencing

At the upstream part of the pipeline, close to the technology, statistical methods generally focus on preprocessing the raw measurements. These are meant to act as a proxy for the real quantity of interest, but usually contain a great deal of noise. A typical example is measured light intensity which is a proxy for RNA concentration in gene expression microarrays. Furthermore, there is often also undesired variability at play when experiments are repeated. Hence, low-level statistical methods are needed to efficiently remove all this obscuring variability while retaining as much useful biological information in the data as possible. As a consequence of being situated upstream in the pipeline, these methods are typically more platform-specific, as they depend on the distinctive nature of the raw data generated by the different high-throughput platforms. In contrast, the statistical methods at the downstream end of the pipeline, which make use of these preprocessed data, are often largely driven by the specific application. Therefore, they are usually less platform-dependent.

As there exist a plethora of possible applications in genomics, the spectrum of statistical methods is very diverse. Nevertheless, many of the challenges are recurrent, because they are related to typical properties of genomic data. In each of the applications discussed in this dissertation we have to deal with one or more of these challenges. A first typical problem is the high-dimensionality of the data statistical models have to cope with. High-throughput genomics technologies are developing at a tremendous speed. While genomics platforms continuously become cheaper, the size of the associated data sets only increases. This poses the challenge of developing statistical methods that are sufficiently fast and computationally efficient in order to deal with these large data sets.

The high-dimensional data sets are often combined with small sample sizes, e.g. microarrays that measure the expression of thousands of genes for a small number of patients. Furthermore, genomic data are also regularly characterized by special structures that have to be taken into account, e.g. local dependencies between the expression of different exons of the same gene. Traditional multivariate models therefore appear to be inappropriate in many cases. Another recurring concern is that statistical inference of some hypothesized biological statement, e.g. the mean expression of a gene is not differentially expressed between two treatment groups, is formulated for many genomic features simultaneously. This may pose specific multiple testing problems.

The different steps in the analysis pipeline of high-throughput genomic data are commonly

conducted consecutively. The raw data are first preprocessed and then used as input in the downstream analysis method. In many cases, the errors generated by the preprocessing model are not taken into account later on in the analysis. An essential merit of statistical models is their ability to quantify the uncertainties associated with their fit to the data. This quantification often contains valuable information that may be used in further analysis steps to achieve more accurate results, see e.g. Liu et al. (2006); Rattray et al. (2006) for examples in a microarray context. In the ideal situation this is accomplished by integrating preprocessing and downstream analysis tasks in a unified model that would simultaneously account for all sources of random variation, e.g. Wu and Irizarry (2007). Because this is not always feasible in practice, a more modular approach is often taken. In any case, allowing proper uncertainty quantification when developing statistical methods for high-throughput genomics is an important challenge.

1.4 General objectives and outline of the thesis

In this dissertation we focus on different applications for two important technologies in high-throughput genomics: DNA microarrays and NGS sequencing. Although some of the challenges we face may be rather application-specific, there are some general objectives that are envisaged throughout the whole thesis. A shared objective is that the proposed methods should allow the use of as much raw information as possible. In this way we try to reduce prior preprocessing of raw data to a large extent. Moreover, we also aim to design methods that allow proper error propagation through the whole data analysis pipeline. It is furthermore of paramount importance to develop fast and computationally efficient algorithms to accompany the proposed statistical methods. Finally, researchers in genomics and biomedical sciences can only benefit from novel statistical applications in their fields to the full extent if they are provided with a user-friendly software implementation. A new statistical method can only be properly valorized if it is widely available and easily applied by the scientific community.

The dissertation consists of two parts that focus on applications for the two different technologies. In Part I a statistical methodology is proposed for transcriptome analysis with tiling microarrays, designed to detect regions of RNA expression along the genome. Part II discusses two statistical problems for DNA sequence analysis with the Roche/454 system, which is one of the major NGS platforms. The first problem is situated at the upstream end of the data analysis

pipeline. It concerns the correct elucidation of the DNA sequence, referred to as base-calling, based on the light intensity data measured by the Roche/454 sequencer. The second problem is a downstream application for the same platform. It is related to the detection of DNA sequence variation in homopolymeric regions. For both parts the particular data-generating technology is explained in an introductory chapter. These separate introductions present data sets that motivate the research and they thoroughly describe the background and problem setting. Finally, an overview of the specific objectives and a detailed outline for the respective parts finishes these chapters.

A large part of the contents of this dissertation has been published in the scientific literature. The method described in Chapter 3 has been published in Statistical Applications in Genetics and Molecular Biology (Clement et al., 2012). My contribution to this article was mainly on the simulation study, the data analysis of the case study, and the comparison with existing methods. The contents of Chapters 4 and 5 are published in BMC Bioinformatics (De Beuf et al., 2012b). For this article I developed the method, implemented it as a software package, conducted the case studies and statistical analyses and wrote the manuscript. The base-calling method for 454 data (Chapter 8) was also published in BMC Bioinformatics (De Beuf et al., 2012a). I contributed to the development and implementation of the statistical method, conducted all analyses and wrote the manuscript. A manuscript about the variant detection method, which is the topic of Chapter 9, is currently in preparation. I contributed to the development and implementation of this method, conducted the case study and empirically evaluated the method's performance in a simulation study.

Part I

Statistical methods for transcriptome analysis with tiling array data

A selection of the presented work is published in

Clement, L., De Beuf, K., Thas, O., Vuylsteke, M., Irizarry, R. A., and Crainiceanu, C. (2012). Fast wavelet based functional models for transcriptome analysis with tiling arrays. *Statistical Applications in Genetics and Molecular Biology*, 11:Iss. 1, Article 4.

De Beuf, K., Pipelers, P., Andriankaja, M., Thas, O., Inzé, D., Crainiceanu, C. and Clement, L. (2012). Analysis of tiling array expression studies with flexible designs in Bioconductor (waveTiling). *BMC Bioinformatics*, 13:234.

Chapter 2

Introduction to Part I: Statistical methods for transcriptome analysis with tiling array data

This chapter gives an introduction to the first part of this dissertation, in which we present and discuss a statistical method for transcriptome analysis with tiling microarray data. Section 2.1 briefly introduces the microarray and tiling array technologies. The data set that initialized and motivated the research is presented in Section 2.2, while Section 2.3 describes popular methods in tiling array data analysis. Section 2.4 introduces the reader to wavelets and wavelet-based scatterplot smoothing. We will build upon these concepts in the development of the proposed methods. Finally, the objectives and outline for this part of the dissertation are given in Section 2.5.

2.1 Tiling array technology

In this section, we first describe the classical microarray technology. Next, we focus more specifically on tiling arrays.

Microarrays (Fodor et al., 1991; Schena et al., 1998) allow for the quantitative measurement of thousands of biochemical reactions in parallel. The most common uses of microarrays are the detection of genomic mutations, the elucidation of DNA-protein interactions, and the analysis

of RNA levels or gene expression in the cell. Within microarrays for expression profiling the main types are spotted microarrays and oligonucleotide arrays. We will focus on the latter kind. Detailed information on spotted microarrays is provided in e.g. Eisen and Brown (1999).

The most widespread oligonucleotide microarray platform is the GeneChip® from Affymetrix. Oligonucleotide probes, matching specific locations in the DNA sequence, are synthesized and linked to a particular position on the chip (see Figure 2.1 (A)) using a photolithographic procedure (e.g. Pease et al., 1994). In a microarray experiment, RNA is first extracted from an organism's tissue or cell. It is then reverse transcribed into double-stranded complementary DNA (cDNA). Next, fluorescently labeled cRNA is produced from cDNA and fragmented. This biological sample is then hybridized to the probes on the chip (see Figure 2.1 (B)). In this process a chemical interaction by means of hydrogen bonds is established between complementary nucleotides (A with T and G with C). Finally, the amount of hybridized cRNA (see Figure 2.1 (C)) is measured by imaging the chip in a scanner and recording the amount of excited fluorescence signal for each spatial probe location.

Probes on the GeneChip® are 25 nucleotides in length (see Figure 2.1 (A)) and appear in pairs, also called 25-mer pairs. Each perfect match (PM) probe is perfectly complementary to a particular sequence of the genome of interest, while the corresponding mismatch (MM) probe is identical to the PM probe, except for the middle (13th) nucleotide. MM probes were designed for the quantification of non-specific binding, but are currently omitted from the array. Typically, a collection of 11 to 20 probes interrogate the same gene. This is referred to as a probeset. There are usually between 11000 and 42000 genes analyzed by a single chip (Lipshutz et al., 1999). Since the probes on the chip have to be designed in advance, it is imperative that the DNA sequence of the studied organism is known. Furthermore, probes of classical oligonucleotide microarrays only match to genes that are already annotated. These genes have a known begin and end position along the genome and biological information is often available about them.

In the last decade the genomes of many organisms have been entirely sequenced. Until recently, the traditional view was that only genes encoded for proteins or structural RNAs, and that besides the regulatory promoters upstream these genes, the rest of the genome was considered *junk DNA* or *dark matter* (Mockler and Ecker, 2005; Johnson et al., 2005). Multiple studies (e.g. Saha et al., 2002; Sémon and Duret, 2004), however, indicated that a much larger proportion

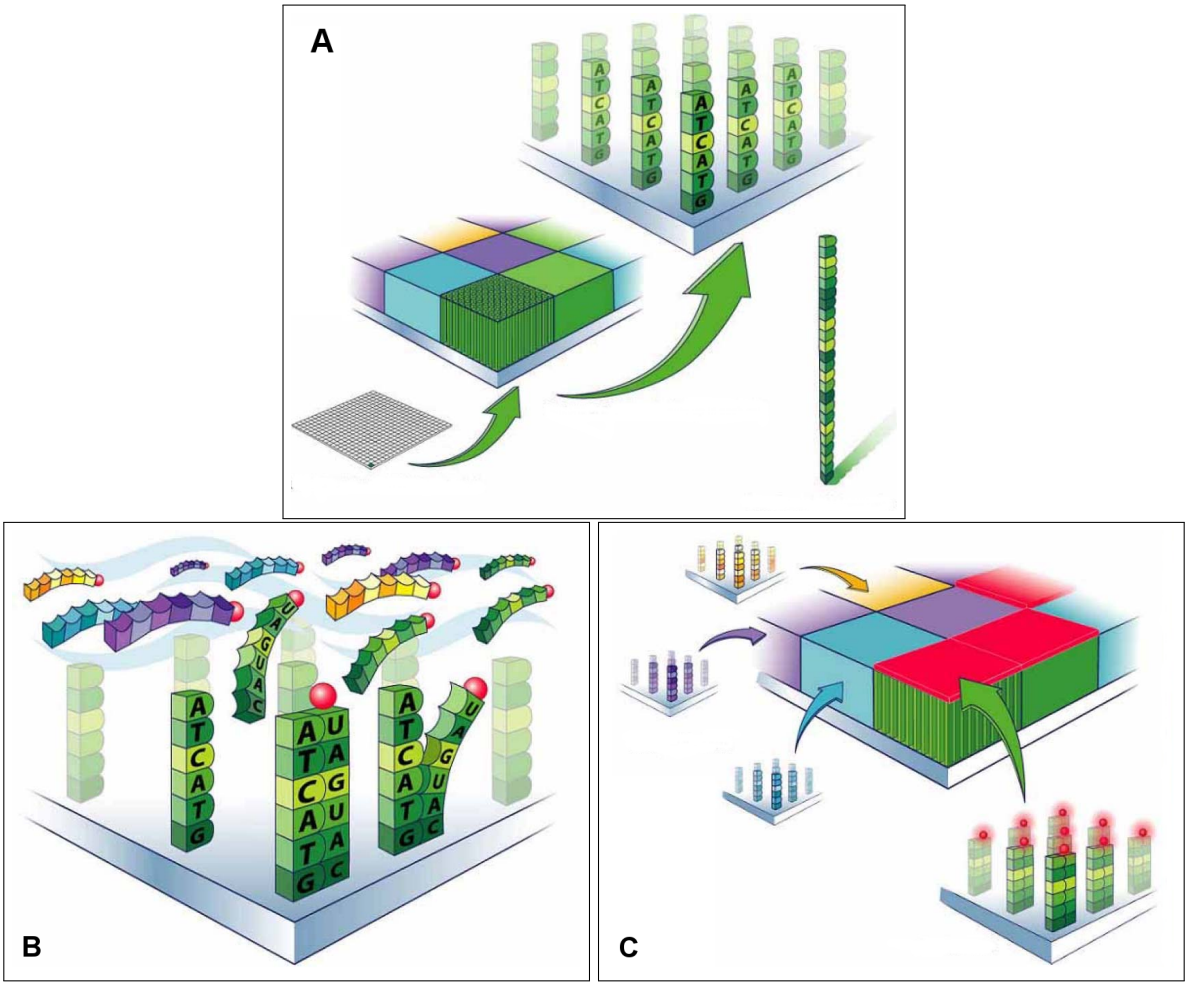


Figure 2.1: Principle of GeneChip® microarray. (A) Synthesis of the array; (B) Hybridization; (C) Excitation. Adapted from `www.affymetrix.com`.

of the genome has the potential to be actively transcribed than was expected based on current annotation. Their evidence was based on more traditional methods like cDNA sequencing (Saha et al., 2002) or a special type of serial analysis of gene expression (SAGE) (Sémon and Duret, 2004). However, a more detailed description of the complete set of RNA transcripts, referred to as the transcriptome, was needed to enhance the knowledge of an organism's functioning and the regulation of its transcriptional networks.

Genome-wide oligonucleotide-based tiling arrays are an extension of classical microarrays in the sense that they interrogate the whole genome including exonic, intronic and intergenic regions. The probes map to genomic regions - or form *tiles* - that either overlap, lay end-to-end, or are spaced at a more or less equal distance along the genomic coordinate (see Figure 2.2). The average number of nucleotides between the centers of two neighboring probes is called the *resolution* of the tiling (Royce et al., 2005). This organization of probes allows for a (largely)

unbiased quantification of transcriptional activity (e.g. Bertone et al., 2004), as tiling arrays are designed without prior consultation of existing gene annotation.

ARTICLE IN PRESS TIGS 344
2 Review TRENDS in Genetics Vol.xx No.xx Monthxxxx

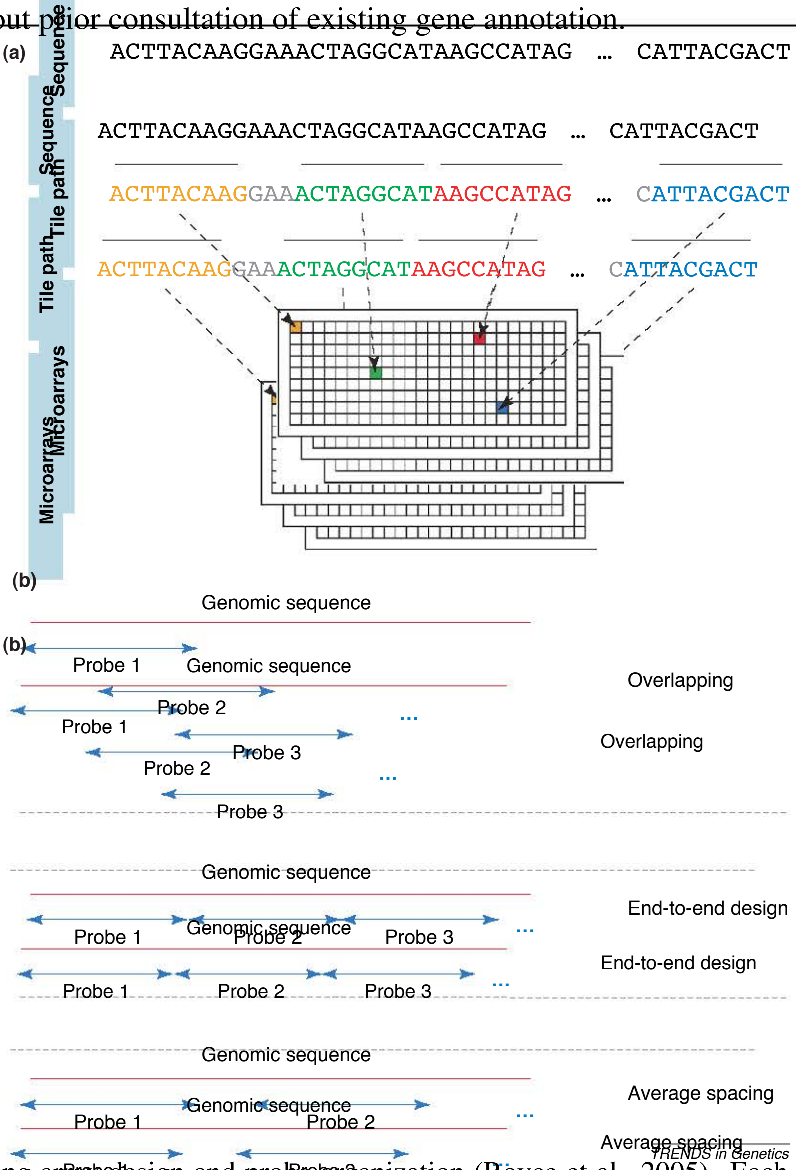


Figure 1. Properties of tiling microarrays. (a) The design of a tiling microarray experiment. Each individual probe in the tiling is indicated by a different color and thick overbar. The probes making up the design constitute a 'tile path'. Nucleotides not incorporated into probes are grayed. Most array designs randomize the position of the adjacent tiles on the array in an attempt to avoid systematic errors. (b) Tiling designs (tile paths) can be overlapping, end-to-end or spaced.

so on. For tiling arrays to reach widespread acceptance, these differences must be identified and resolved. Towards this end, we provide an initial perspective on the tiling microarray experiment from the analytic point of view. In so doing, we provide an introduction to the characteristics of data generated by tiling microarrays, introduce some challenging questions, and give initial views on the analysis of these relatively new types of microarray experiments.

Distribution of signal intensities

For tiling microarrays, a probe representing some genomic sequence is the unit of investigation, and an intensity measurement after hybridization to labeled target is its recorded datum. In theory, this measurement correlates with the number of target nucleic acid molecules that hybridized to that probe during the experiment.

Tiling microarrays built using Affymetrix technology contain a paired 'mismatch' probe for each genomic tile probe (http://www.affymetrix.com/). (For convenience, the genomic tile probe that perfectly matches genomic sequence is typically denoted PM and the mismatch probe is similarly denoted MM.) The MM probe is intended to provide a measurement of nonspecific nucleic acid binding to the PM probe and thus the quantity PM – MM typically serves as the intensity measurement for Affymetrix tiling arrays.

www.sciencedirect.com

Figure 2.2: Tiling array design and probe organization (Royce et al., 2005). Each individual probe in the tiling is indicated by a different color and overbar (upper panel). Tiling designs can be overlapping, end-to-end or spaced (lower panel).

2.2 The E2F study

The statistical methods for transcriptome analysis that are discussed in this part are illustrated on data of a tiling array study on the reference plant *Arabidopsis thaliana*. The study was conducted at the Flemish Institute of Biotechnology (VIB), Department of Plant Systems Biology, Ghent, Belgium. The study fits in the scope of a larger project that aims at increasing the knowledge of the role of E2F transcription factors in the regulation of the plant cell cycle

and plant growth (Naouar et al., 2009). E2Fs are conserved regulators of S phase-specific genes (Blais and Dynlacht, 2007). The genome of the reference plant *Arabidopsis thaliana* encodes three E2Fs, i.e. E2Fa, E2Fb and E2Fc, which are active in association with the dimerization partners DPa or DPb (De Veylder et al., 2007). A complete understanding of the role of the different E2F isoforms requires the comprehensive identification of their target genes. Within this context, Columbia seed (Col-0) plants were used that are ectopically overproducing the heterodimer E2Fa-DPa (Naouar et al., 2009). We will refer to these plants as the E2F plants and to this study as the E2F study.

Expression profiling was performed with Affymetrix GeneChip® Arabidopsis Tiling 1.0R arrays. A single array contains over 3.2 million PM and MM 25-mer probe pairs that are tiled across the complete non-repetitive *Arabidopsis thaliana* genome. The array has a tiling resolution of 35 nucleotides. Hence, the entire genome is tiled with non-overlapping probes with an average gap-width of 10 nucleotides (Naouar et al., 2009). In the study expression data from three biological replicates of both wild type (WT) and E2F plants are used. The wild type refers to the typical form of *Arabidopsis thaliana* as it occurs in nature. The replicates correspond to the target preparation protocol number 3 (TPP3) in Naouar et al. (2009). The aim of the study was to quantify, compare and evaluate the expression of WT and E2F plants. We will call this *transcript discovery*. Furthermore, also the detection of expression changes between WT and E2F plants was of interest. We will refer to this as *differential expression*.

The upper panel of Figure 2.3 shows the raw $\log_2$-transformed intensities from the E2F and WT plant hybridizations as a function of the genomic location for a particular region on chromosome 1. The location of the region can be read from the central horizontal axis on the figure. The numbers on the axis indicate the nucleotide positions along the genome. One point on the plot corresponds to the intensity measured by a specific probe mapping to that genomic location. The biological replicates are indicated by a different symbol. In theory, the measured intensity for a certain probe is proportional to the amount of cRNA target that hybridized to this particular probe. This represents specific binding. In practice, however, the raw data look very noisy, and it is hard to discern the signal of interest, reflecting the true RNA expression, from the obscuring variation (see top panel of Figure 2.3). The latter variation is of no biological interest. It can be introduced by differences in experimental conditions during the sample preparation or the manufacturing and processing of the arrays (Irizarry et al., 2003). Other contributing sources

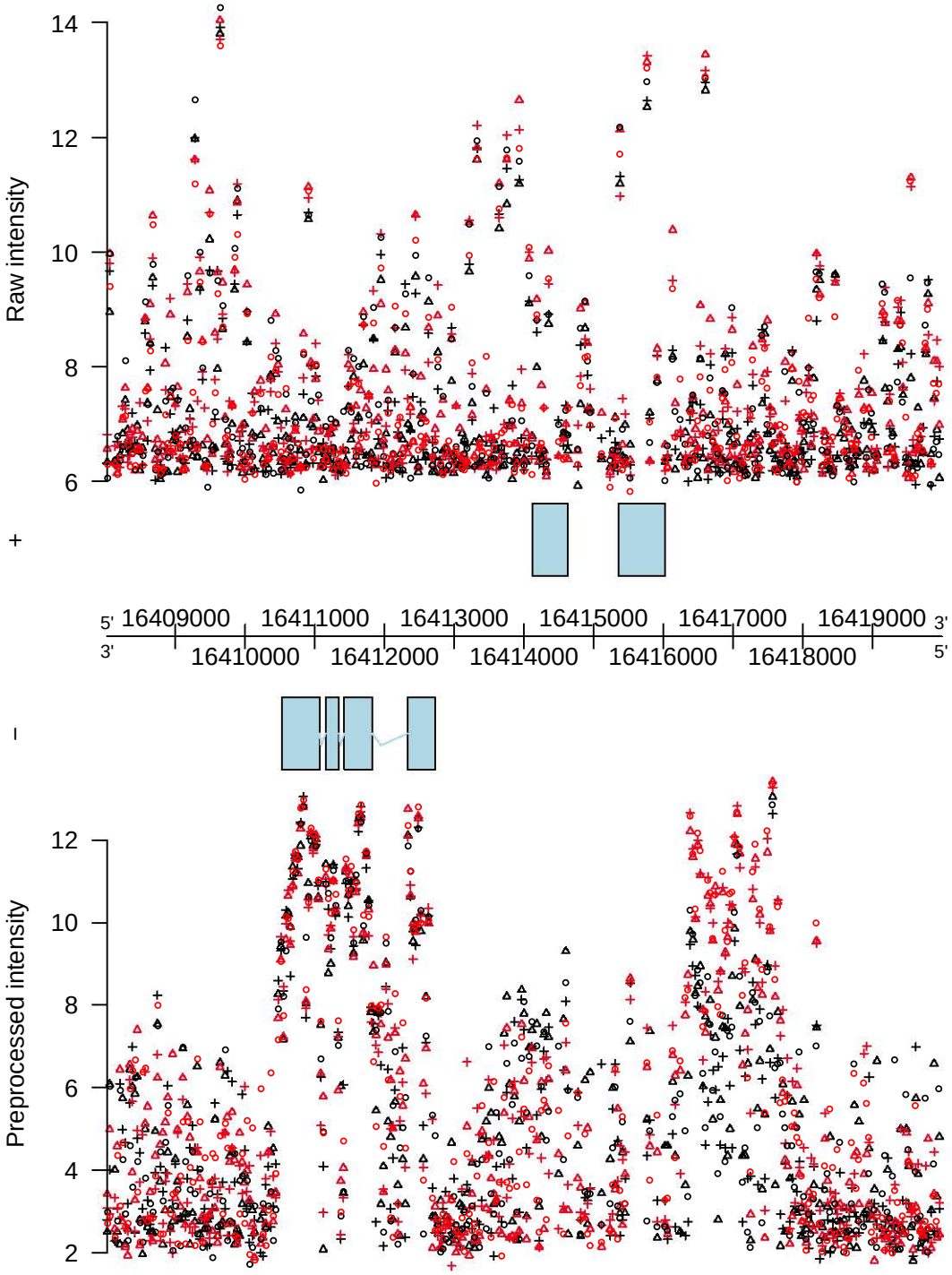


Figure 2.3: Along-chromosome plot of raw (upper panel) and preprocessed (lower panel) $\log_2$-transformed tiling array intensities of WT (black) and E2F (red) plants for a particular region of chromosome 1, denoted by the horizontal axis. The light blue boxes on the forward (+) and reverse (-) strands indicate the position of annotated genes for this region. The preprocessing involved background correction and normalization steps as proposed in the RMA procedure (Irizarry et al., 2003). The 3 different replicates for WT and E2F are indicated by $\circ$, $+$ and $\triangle$. The intensities are measured probe by probe.

include optical or background noise, non-specific binding (with non-complementary molecules) and differences in probe affinity. Due to this complexity of the raw signal composition, some degree of preprocessing is required. For tiling array data this typically involves (1) background correction to account for optical noise and non-specific binding, and (2) normalization to make expression data from different arrays comparable. To this end, we apply the first two steps of the widely used robust multi-array average (RMA) procedure for GeneChip® arrays (Irizarry et al., 2003) on our tiling array data sets. Background correction is conducted by modeling the raw probe-level intensities as a sum of normally distributed background noise and exponentially distributed signal (Irizarry et al., 2003). Quantile normalization (Bolstad et al., 2003) is then applied on the background-corrected signals to force the distribution of probe intensities in all arrays to be identical. The preprocessed $\log_2$-transformed probe-level intensities for the same genomic region on chromosome 1 are presented in the lower panel of Figure 2.3. After proper background correction and normalization, transcriptionally active and differentially expressed regions become more clearly visible. However, probe-to-probe fluctuations within the same transcriptional units are still apparent. Moreover, the sudden jumps in the measured intensities, which are associated with differences in transcriptional activity among exonic regions, and, between exonic, intronic and intergenic regions, make the data very heterogeneous. Well-designed modeling techniques will be needed to cope with these irregularities and expression signal variability in an appropriate way. The second region in the lower panel of Figure 2.3, showing a clear increase in measured intensities, seems to be differentially expressed between WT and E2F plants and does not overlap with previous annotation. This is seen on Figure 2.3 by the lack of a light blue box matching to this genomic region. Nonetheless, we would like to detect these non-annotated regions equally well. Therefore, it is essential to develop methods that do not rely on existing annotation.

2.3 Tiling array data analysis

This section describes some popular methods for analyzing tiling array data from expression studies. While the methods of Kampa et al. (2004) and Huber et al. (2006) focus on transcript discovery within a single biological condition, the RMA method (Irizarry et al., 2003) combined with an empirical Bayes moderated t-test (Smyth, 2004; McCarthy and Smyth, 2009) is widely used when differential expression between two biological conditions is concerned.

2.3.1 Pseudomedian approach for transcript discovery

The approach of Kampa et al. (2004) is based on a summarized expression level of the center probe in a sliding window. Let BW denote the bandwidth of the window, defined as the number of nucleotides from one side of the window to the middle probe. Hence, the window has a predefined size, which is given by $(2 \times BW) + 1$. For each center probe the *pseudomedian* or Hodges-Lehmann estimator (e.g. Hollander and Wolfe, 1999) is calculated based on the neighboring-probe intensities within the window. More specifically, suppose probe k is positioned at genomic coordinate P_k. The pseudomedian expression level for this probe, denoted by E_k, is then given by the median of all $N_k(N_k + 1)/2$ pairwise averages $(Z_l + Z_m)/2$, where Z_l and Z_m are the intensity values at genomic coordinates P_l and P_m, respectively, where P_l and P_m are all genomic coordinates lying within $[P_k - BW, P_k + BW]$, $l \leq m$, and with N_k the number of probes located within this interval $[P_k - BW, P_k + BW]$. The reason for using the pseudomedian is to improve robustness against false-positives due to differences in probe-specific effects and non-specific cross-hybridization events (Kampa et al., 2004). In the initial study the difference between the background-corrected PM and MM intensities was used as intensity values, i.e. $Z_k = PM_k - MM_k$. If the tiling arrays in the experiment do not possess MM probes, as is common for the recent arrays, properly preprocessed PM values may be used instead, e.g. after background-correction and array-to-array normalization by means of the RMA procedure (Irizarry et al., 2003), discussed in Section 2.2. Each probe for which the corresponding pseudomedian expression level exceeds a certain threshold value is called positive. In a postprocessing step adjacent positive probes are merged to form *transcriptionally active regions* (TARs). This involves two additional tuning parameters, (a) *maxgap*, which is the maximum gap between positive probes to be in the same TAR, and (b) *minrun*, which is the minimum length of adjacent positive probes to form a TAR. Kampa et al. (2004) determined the tuning parameter settings based on spiked-in quantitative bacterial RNA transcripts.

Figure 2.4 shows the pseudomedian intensities for the example region of the E2F data (first biological replicate of E2F plants) at bandwidths 50, 150 and 500. The calculation of the pseudomedian within a sliding window has a *smoothing* effect on the intensity signal in function of the genomic position. The smoothness increases with increasing bandwidth, but is constant over the whole genomic region for a fixed bandwidth. This may lead to problems when looking for transcript regions. If the bandwidth is small, the boundaries of the transcript regions can

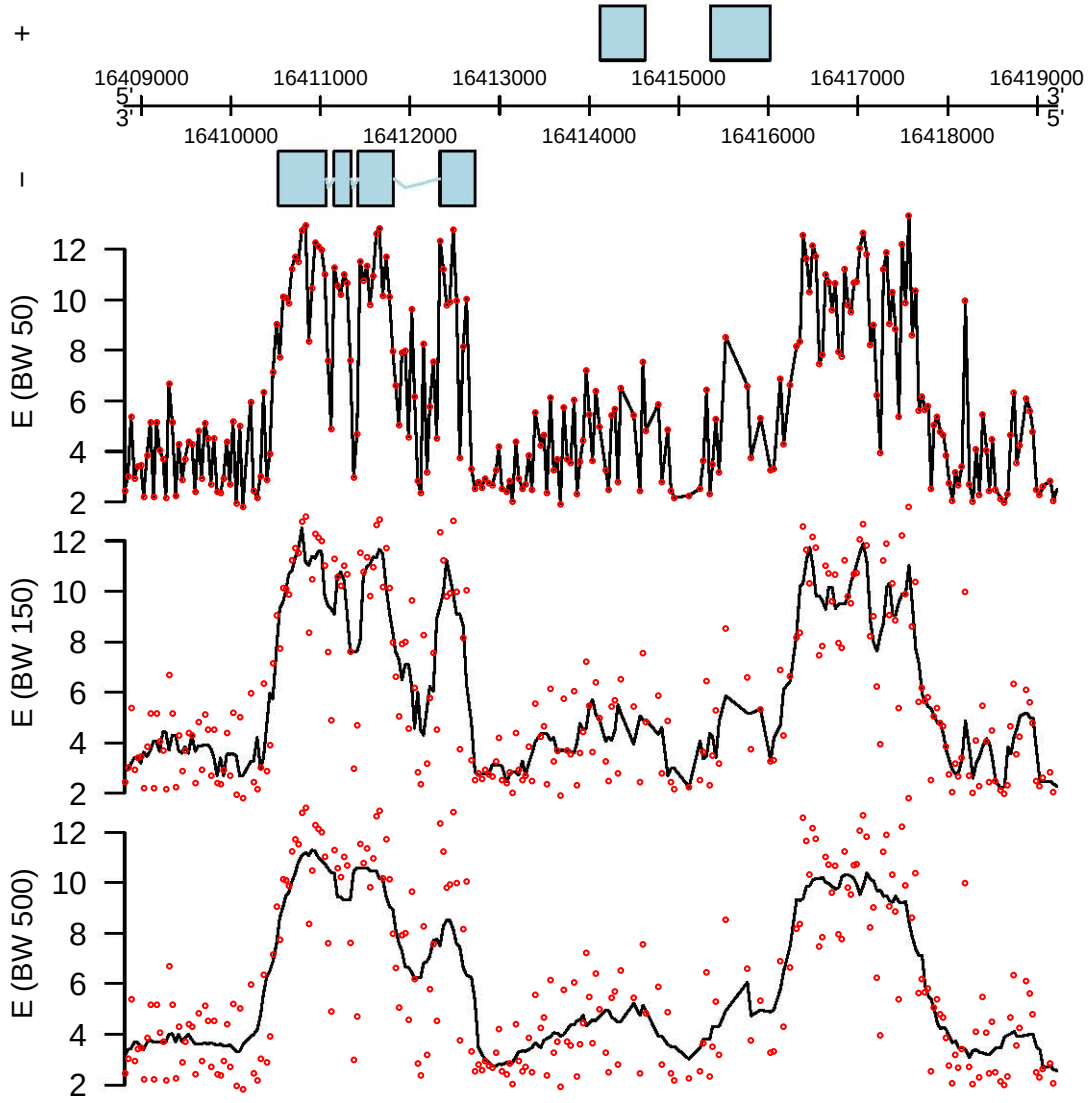


Figure 2.4: Pseudomedian intensities E of the first biological replicate of E2F plants for a genomic region on chromosome 1. The pseudomedian intensities are calculated for a sliding window with bandwidths (BW) 50, 150 and 500.

be determined accurately, but the large variability of the intensities may lead to many false positives. In the case of large bandwidths, the smooth signal may lead to biased estimates of the transcript boundaries, depending on the level of the signal above background (Hastie et al., 2001). This latter problem is clearly illustrated in Huber et al. (2006).

2.3.2 Structural change model for transcript discovery

Huber et al. (2006) recognized the problems arising from sliding window-based approaches in transcript discovery. To obtain less biased estimates of the transcript boundaries, they proposed

an alternative method based on a model that fits a piecewise constant expression profile along genomic coordinates. It is known in literature as the *structural change model* (SCM). The model is motivated by the fact that probe intensities show sudden changes at the boundaries of TARs, while the intensity level remains at a fairly constant level within each TAR. The model can be written as

$$z_{ki} = \mu_s + \epsilon_{ki} \qquad \text{for} \quad t_s \leq k < t_{s+1}, \tag{2.1}$$

where z_{ki} denotes the intensity from the k-th probe $(k = 1, \ldots, n)$ in the i-th array and $t_2, \ldots, t_S$ are the segment boundaries, with $t_1 = 1$, $t_{S+1} = n + 1$, and S the total number of segments. Further, μ_s denotes the mean intensity level of segment s and ϵ_{ki} is a Gaussian error term. The model is fitted by minimizing the sum of squared residuals. The number of segments S is a tuning parameter, which is chosen by a penalized likelihood approach based on the Bayesian Information Criterion (for details, see Huber et al., 2006). Typically, the z_{ki} used to fit the SCM are preprocessed intensity values. Huber et al. (2006) proposed to preprocess the raw intensities by means of a DNA reference normalization. To this end, hybridization intensities from an experiment using genomic DNA have to be available. Details of this normalization method are provided in Huber et al. (2006). In case no such experiment has been conducted, other preprocessing methods may be applied as well, e.g. the background correction and quantile normalization steps in the RMA procedure (Irizarry et al., 2003).

Fitting the SCM results in the segmentation of the transcriptional profile along the genome. Subsequently, a certain segment is called a TAR if the mean intensity of the probes in the segment is above a certain threshold value. The choice of this threshold value is based on the distribution of the mean intensity levels for the segments that do not overlap with any annotated feature (David et al., 2006). This is typically a two-component mixture distribution of which the leftmost component is more or less a normal distribution. The fit to this normal component is considered as the null distribution of mean intensities for the segments that are not transcribed. Hence, a p-value can be assigned to each segment, based on the observed mean intensities. Following David et al. (2006), the threshold value is then selected after controlling the *false discovery rate* (FDR) at 0.1% by the Benjamini-Yekutieli procedure (Benjamini and Yekutieli, 2001).

The relatively simple approach of Huber et al. (2006) has been successfully applied to yeast tiling array data (David et al., 2006). As indicated in Zeller et al. (2008), however, the segmen-

tation problem is considerably more challenging for the genomes of higher eukaryotes that are capable of (alternative) splicing and whose genes typically contain much shorter exon segments interrupted by potentially very long intron sequences.

2.3.3 RMA combined with moderated t-test for differential expression

The RMA algorithm (Irizarry et al., 2003) is originally designed for the normalization and summarization of probe-level data in classical microarray studies. RMA involves three steps: (a) background correction, (b) quantile normalization and (c) summarization using the median polish algorithm. When applying RMA in tiling array data analysis one first has to construct probesets from individual probes. The existing annotation is used for this purpose. RMA results in a summarized intensity value for each probeset on each array. This summarized value is further used as input for the proper differential expression analysis. Differential expression is assessed using an extension of the empirical Bayes moderated t-statistic introduced in Smyth (2004). Such a moderated t-test is conducted for each probeset separately. In principle, it differs from an ordinary two-sample t-test by incorporating prior information on how the estimated model parameters vary across genes in the test statistic (see Smyth (2004) for full details). McCarthy and Smyth (2009) have proposed an extension of this moderated t-testing procedure. This extension allows for testing whether the true differential expression is greater than a given threshold value. In this way reliable p-values can be obtained for finding genomic regions with differential expression that is also biologically meaningful (McCarthy and Smyth, 2009). The FDR is controlled at a predefined level by using the Benjamini-Hochberg method (Benjamini and Hochberg, 1995).

2.4 Scatterplot smoothing and wavelets

2.4.1 Introduction

Since tiling probes are positioned along the genome with a more or less equal resolution and regardless of existing annotation, the probe intensity data can be thought of as realizations of an underlying expression function along the genomic coordinate. These observed intensities are

usually subject to observational error, which we want to remove. In such a situation *smoothing* methods are needed to extract the true function from the observed data. In this section we will focus on smoothing methods in the single-curve setting, e.g. for along-genome expression data from only one tiling array. This is commonly referred to as *scatterplot smoothing*, because one is interested in highlighting the underlying trend in a scatterplot (Ruppert et al., 2003). The problem of scatterplot smoothing essentially corresponds to the standard nonparametric regression problem (e.g. Abramovich et al., 1998; Ruppert et al., 2003)

$$y_i = g(t_i) + \epsilon_i, \quad i = 1, \ldots, T, \tag{2.2}$$

where $t_i = i/T$. The error terms ϵ_i are assumed to be normally distributed random variables with zero mean and variance σ^2. The aim is to estimate some unknown smooth function g from the noisy data (t_i, y_i) without assuming any particular parametric form.

Several smoothing methods have been described in the literature. A popular class of methods performs the function estimation by means of local weights. Well-known examples include kernel smoothing (e.g. Wand and Jones, 1995) and local polynomial smoothing (e.g. Fan and Gijbels, 1996). Sliding window-based approaches such as the calculation of the pseudomedian in Kampa et al. (2004) can also be thought of as belonging to this class of methods.

Another class consists of smoothing methods based on basis function estimators. Following the definition in Ramsay and Silverman (2005), a basis function system is a set of known functions that have the property that many functions can be approximated arbitrarily well by taking a weighted sum or linear combination of a sufficiently large number of these functions. A frequently used basis system for non-periodic data is the spline basis system. The degree of smoothness can be controlled to some extent by the number of basis functions being used. However, it is often preferred to include many basis functions and to impose a penalty on the roughness of the function. This is done for instance in cubic spline smoothing (e.g. Gu, 2002). These smoothing methods are suitable to model a large spectrum of functions. However, one of their major limitations is that they work with global bandwidths or penalties. Therefore, the amount of smoothing is forced to remain the same over the entire support of the function. Furthermore, they are not very suitable for large data sets as spline smoothers with many knots quickly become computationally complex. Consequently, they are not the best choice to model large spatially heterogeneous data with many local features such as the tiling array data of the E2F study. To circumvent these difficulties the use of wavelet basis functions is proposed.

2.4.2 A primer on wavelets

Wavelets are families of orthonormal basis functions that can be used to represent functions in an efficient way. A multitude of different wavelet bases exist. An overview can be found in Ogden (1997) and Vidakovic (1999). The oldest and most simple one is the *Haar wavelet* (Haar, 1910). The Haar *mother wavelet* is a mathematical discontinuous function defined by

$$\psi(t) = \begin{cases} 1 & \text{if } 0 \leq t < 0.5 \\ -1 & \text{if } 0.5 \leq t < 1 \\ 0 & \text{otherwise} \end{cases}, \tag{2.3}$$

while the Haar *father wavelet* or scaling function is given by

$$\phi(t) = \begin{cases} 1 & \text{if } 0 \leq t < 1 \\ 0 & \text{otherwise} \end{cases}. \tag{2.4}$$

A graphical representation of the Haar mother wavelet is given in Figure 2.5.

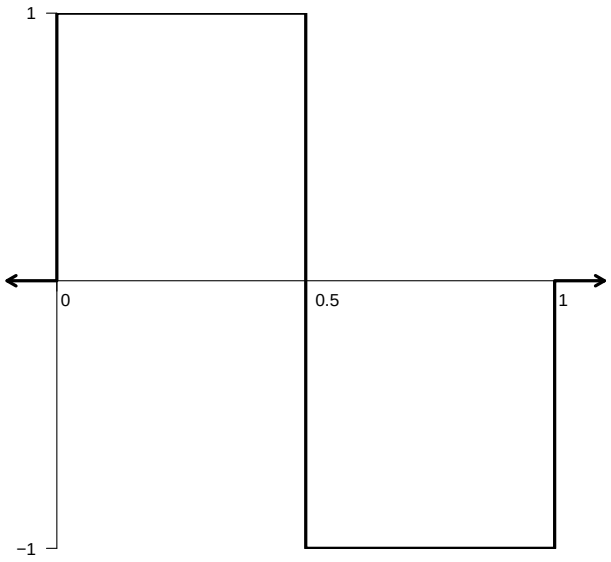


Figure 2.5: Haar mother wavelet

We can construct a wavelet basis by choosing a suitable mother (ψ) and father (ϕ) wavelet

function and considering all dilations and translations

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k), \tag{2.5}$$

$$\phi_{j,k}(t) = 2^{j/2}\phi(2^j t - k), \tag{2.6}$$

for integers j and k.

The mother wavelet is chosen to ensure that an orthonormal set can be formed (e.g. Nason, 2005), i.e.

$$\int_{-\infty}^{\infty} \psi_{j,k}(t)\psi_{j',k'}(t)dt = \delta_{j,j'}\delta_{k,k'}, \tag{2.7}$$

where $\delta_{m,n} = 1$ if $m = n$ and $\delta_{m,n} = 0$ if $m \neq n$.

Typically, the mother wavelet and all derived basis functions have a compact support and the wavelet expansion provides a location and scale decomposition of the underlying function. The Haar mother wavelets for some choices of j and k are shown in Figure 2.6.

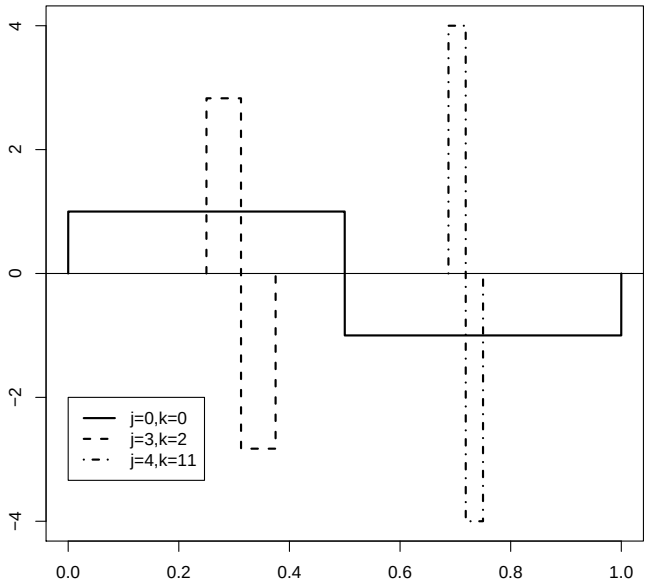


Figure 2.6: Haar mother wavelet for some choices of scale j and location k

In essence, the wavelet decomposition allows a function g to be represented using a wavelet expansion by (e.g. Nason, 2005)

$$g(t) = \sum_{k\in\mathbb{Z}} \mathcal{C}_{j_0,k}\phi_{j_0,k}(t) + \sum_{j=j_0}^{\infty}\sum_{k\in\mathbb{Z}} \mathcal{D}_{j,k}\psi_{j,k}(t), \tag{2.8}$$

where $\mathcal{D}_{j,k}$ are wavelet or detail coefficients, $\mathcal{C}_{j,k}$ are scaling coefficients, and j_0 is the coarsest scale considered in the decomposition. The wavelet and scaling coefficients are given by

$$\mathcal{D}_{j,k} = \int_{-\infty}^{\infty} f(t)\psi_{j,k}(t)dt \tag{2.9}$$

$$\mathcal{C}_{j,k} = \int_{-\infty}^{\infty} f(t)\phi_{j_0,k}(t)dt. \tag{2.10}$$

The first term in Equation (2.8), containing the father wavelet, is the *smooth* part of the function associated with an *average* of the function over the support defined by j_0. The second term, involving the mother wavelets, on the other hand, is related with the *detail* of the function at different scales and locations. The detail of a function can be losely defined as the degree of difference between neighboring function values (Nason, 2005).

Suppose we have an observed data vector $\boldsymbol{y} = (y_1, y_2, ..., y_T)^T$, arisen from the underlying function values $\boldsymbol{g} = (g_1, g_2, ..., g_T)$, where $y_i = y(t_i)$ and $g_i = g(t_i)$. It is assumed that the data points are equally spaced and the number of data points T is a power of two, i.e. $T = 2^J$, for some integer $J \geq 0$. The vector $\boldsymbol{g}$ can be decomposed by means of the *discrete wavelet transform* (DWT), given by

$$\boldsymbol{d} = \boldsymbol{g}\boldsymbol{W}^T, \tag{2.11}$$

where $\boldsymbol{W}$ is a $T \times T$ orthogonal DWT matrix. The vector $\boldsymbol{d}$ contains the discrete scaling coefficient $c_{0,0}$ and $T-1$ discrete wavelet coefficients $\{d_{j,k}\} : j = 0, \ldots, J-1, k = 0, \ldots, 2^j - 1$. These coefficients are analogous to the coefficients in Equation (2.8), with $c_{0,0} \approx \mathcal{C}_{0,0}/\sqrt{T}$ and $d_{j,k} \approx \mathcal{D}_{j,k}/\sqrt{T}$ (e.g. Barber et al., 2002). The factor $\sqrt{T}$ arises from the difference between continuous and discrete orthogonality conditions (Abramovich et al., 1998). Similar to (2.11), the DWT of the observed data vector $\boldsymbol{y}$ gives rise to a vector of empirical discrete scaling and wavelet coefficients $\boldsymbol{d}^*$ by $\boldsymbol{d}^* = \boldsymbol{y}\boldsymbol{W}^T$. Because of the orthogonality of $\boldsymbol{W}$, the DWT of $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, ..., \epsilon_T)$ from Equation (2.2), denoted by $\boldsymbol{\epsilon}^*$, also consists of T normal random variables with zero mean and variance σ^2. This leads to the nonparametric regression model in the wavelet space

$$\boldsymbol{d}^* = \boldsymbol{d} + \boldsymbol{\epsilon}^*. \tag{2.12}$$

To illustrate the idea of the DWT, let us consider an example sequence of observed values $\boldsymbol{y} = (3, 3, 8, 6, 2, 7, 7, 5)$. In this example, we have $T = 8$, hence $J = 3$. For the Haar wavelet with 3 levels the matrix $\boldsymbol{W}$ is defined as (e.g. Nason, 2005)

$$\boldsymbol{W} = \begin{pmatrix} \sqrt{2}/4 & \sqrt{2}/4 & \sqrt{2}/4 & \sqrt{2}/4 & \sqrt{2}/4 & \sqrt{2}/4 & \sqrt{2}/4 & \sqrt{2}/4 \\ 1/\sqrt{2} & -1/\sqrt{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2} \\ 1/2 & 1/2 & -1/2 & -1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 & -1/2 & -1/2 \\ \sqrt{2}/4 & \sqrt{2}/4 & \sqrt{2}/4 & \sqrt{2}/4 & -\sqrt{2}/4 & -\sqrt{2}/4 & -\sqrt{2}/4 & -\sqrt{2}/4 \end{pmatrix}.$$

A very efficient pyramid-based algorithm for conducting this projection from the data space onto the wavelet space has been proposed in Mallat (1989). We illustrate the idea behind this algorithm on $\boldsymbol{y}$ in the example.

On the finest scale the detail coefficients of the Haar wavelets are given by

$$d^*_{J-1,k} = (y_{2k-1} - y_{2k})/\sqrt{2}. \tag{2.13}$$

In order to obtain information at coarser scales we need scaling coefficients, which are scaled local averages. On the finest scale we have

$$c^*_{J-1,k} = (y_{2k-1} + y_{2k})/\sqrt{2}. \tag{2.14}$$

The next coarsest wavelet coefficient is now obtained by differing the local averages at the finer level:

$$d^*_{J-2,l} = (c^*_{J-1,2l-1} - c^*_{J-1,2l})/\sqrt{2}. \tag{2.15}$$

Likewise, the coarser-level scaling coefficients are given by

$$c^*_{J-2,l} = (c^*_{J-1,2l-1} + c^*_{J-1,2l})/\sqrt{2}. \tag{2.16}$$

These calculations can be repeated until the coarsest possible level, $j = 0$, is achieved. This gives rise to a vector of scaling and wavelet coefficients for the data sequence in the example:

$$\begin{aligned} \boldsymbol{d}^* &= (c^*_{0,1}, d^*_{2,1}, d^*_{2,2}, d^*_{2,3}, d^*_{2,4}, d^*_{1,1}, d^*_{1,2}, d^*_{0,1}) \qquad (2.17) \\ &= (41\sqrt{2}/4, 0, \sqrt{2}, -5\sqrt{2}/2, \sqrt{2}, -4, -3/2, -\sqrt{2}/4). \end{aligned}$$

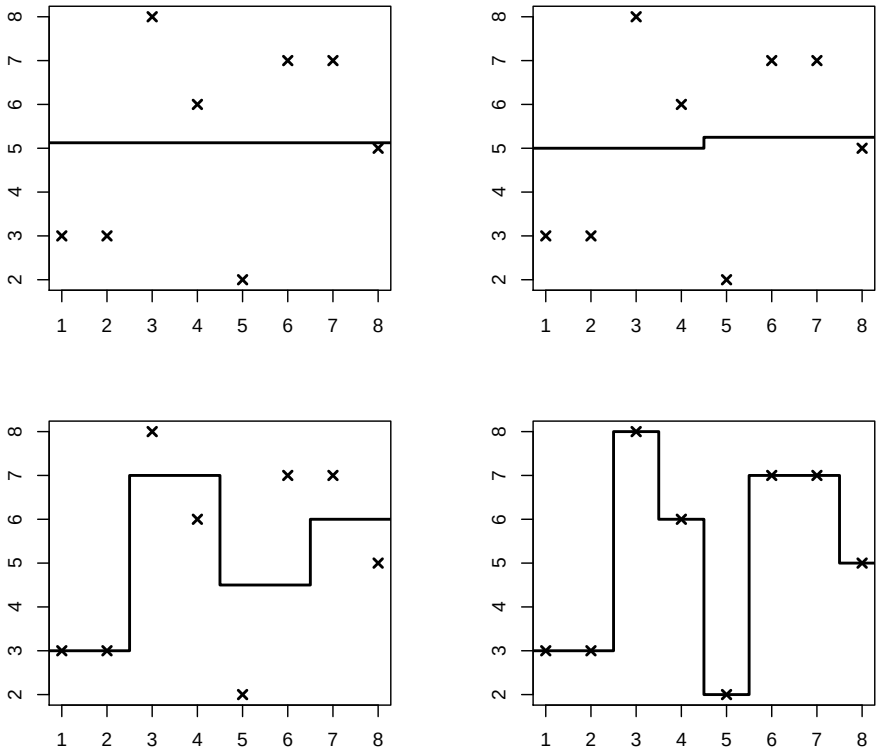


Figure 2.7: Multiresolution analysis of example data with Haar wavelet. Upper left panel: only the father wavelet is used; Upper right panel: father wavelet and coarsest mother wavelet are used; Lower left panel: father wavelet and two coarsest mother wavelet are used; Lower right panel: father wavelet and three mother wavelets are used.

Figure 2.7 shows a *multiresolution analysis* of these example data using the Haar wavelet basis. In the upper left panel only the father wavelet is used. This corresponds with the first row of $\boldsymbol{W}$. The reconstructed function is a constant equal to the average of all data points. When both the father wavelet and the coarsest mother wavelet are used, corresponding with row 1 and row 8 of $\boldsymbol{W}$, a curve is obtained which is piecewise constant and allows for a shift between the averages of the first four and the last four data points. The lower panels give the curves when also the finer levels $j = 1$ (left) and $j = 2$ (right) of wavelet coefficients are taken into account. The lower right panel shows that the data are exactly reconstructed if the coarsest father wavelet and the mother wavelets at all levels are used, corresponding with all rows of $\boldsymbol{W}$.

2.4.3 Wavelet shrinkage or thresholding in scatterplot smoothing

In Section 2.4.2 we have illustrated that observed data points along an equally spaced grid can be perfectly reconstructed by means of a wavelet basis expansion. However, we are more interested in approximating or estimating the true function underlying these observed data. As argued in Section 2.4.1, smoothing methods are needed for this. In what follows, we will often refer to the sequence of observed values $(y_1, y_2, ..., y_T)$ as the *noisy signal*, and to the sequence of estimated values of the underlying function $(\hat{g}_1, \hat{g}_2, ..., \hat{g}_T)$ as the *denoised signal*.

As explained in Section 2.4.2, a multiresolution analysis corresponds with setting all the wavelet coefficients of particular wavelet scales to zero. Hence, conducting a multiresolution analysis is essentially a first way of smoothing the noisy signal by means of wavelets. This smoothing effect is illustrated on the example region of the E2F data (first biological replicate of E2F plants). Figure 2.8 shows the results for $J = 2$, where only the wavelet scales $j = 0, 1, 2$ contain non-zero wavelet coefficients, until $J = 8$, where all wavelet scales contain non-zero wavelet coefficients. It is clear that this procedure leads to a homogeneous smoothing along the genomic coordinate. The aim, however, is not only to obtain a smooth signal at positions where RNA expression is absent or more or less constant. At the same time, it is desired to retain the characteristic features of the underlying function, such as the sudden discontinuities at the boundaries of transcripts or in genes with many short exons and introns.

Unlike most smoothing methods, wavelets enable to perform such an *adaptive regularization* of the noisy signal. One of the properties of the wavelet transform is that it concentrates most of the signal's structure in relatively few wavelet coefficients, while distributing white noise equally over all wavelet coefficients. This is often called the *sparseness* or *heavy-tailedness* property of wavelet coefficients (Figueiredo and Nowak, 2001). Denoising of the signal is thus possible by thresholding or shrinking the smallest wavelet coefficients. *Wavelet shrinkage* or *wavelet thresholding* typically consist of three steps:

1. Compute the empirical wavelet coefficients $\boldsymbol{d}^*$ of the noisy signal.

2. Modify the empirical wavelet coefficients according to a certain shrinkage or thresholding rule. In this way, the true wavelet coefficients $\boldsymbol{d}$ of the underlying function are estimated by $\hat{\boldsymbol{d}}$.

3. Backtransform the modified wavelet coefficients $\hat{\boldsymbol{d}}$ to obtain the denoised signal in the original data space $\hat{\boldsymbol{g}}$ by applying the inverse DWT (IDWT), i.e. $\hat{\boldsymbol{g}} = \hat{\boldsymbol{d}}\boldsymbol{W}$.

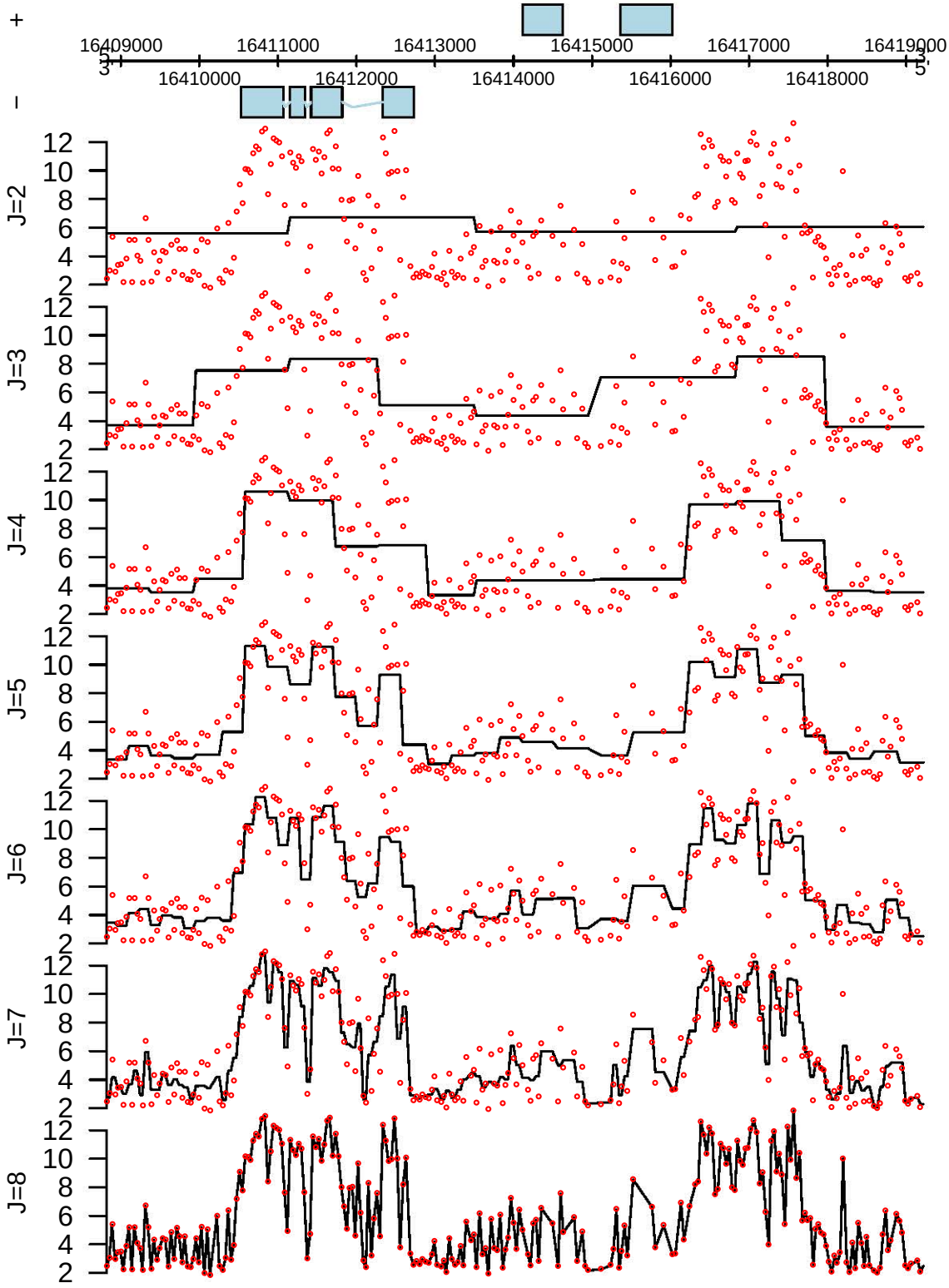


Figure 2.8: Multiresolution analysis for genomic region of the first biological replicate of E2F plants

Technically, thresholding means that some wavelet coefficients are effectively set to zero, while shrinkage involves decreasing their absolute value towards zero, without reaching the value zero exactly. We examine the effect of four different wavelet thresholding procedures.

Two classical thresholding rules are the hard and soft thresholding of the wavelet coefficients (Donoho and Johnstone, 1994, 1995). Let λ denote an overall threshold level. The hard thresholding rule is given by

$$\hat{d}_{j,k}^{(\text{hard})} = \begin{cases} 0 & \text{if } |d_{j,k}^*| \leq \lambda \\ d_{j,k}^* & \text{if } |d_{j,k}^*| > \lambda \end{cases} . \tag{2.18}$$

In other words, hard thresholding sets all those coefficients to zero that have an absolute value below some threshold, while leaving the remaining coefficients unchanged. This results in a discontinuity at the threshold.

The soft thresholding rule, on the other hand, can be written as

$$\hat{d}_{j,k}^{(\text{soft})} = \begin{cases} 0 & \text{if } |d_{j,k}^*| \leq \lambda \\ \text{sgn}(d_{j,k}^*)(|d_{j,k}^*| - \lambda) & \text{if } |d_{j,k}^*| > \lambda \end{cases} , \tag{2.19}$$

where $\text{sgn}(.)$ is the sign function for which holds that $\text{sgn}(x) = 1$ if $x \geq 0$, and $\text{sgn}(x) = -1$ if $x < 0$. Hence, soft thresholding also replaces coefficients with an absolute value below the threshold by zero, but it shrinks the remaining coefficients towards zero by substracting the threshold from the absolute value of the wavelet coefficients. After soft thresholding, the remaining coefficients therefore form a continuous distribution that is centered around zero. Both hard and soft thresholding can be applied either with a common threshold for all levels of the wavelet decomposition or with a more adaptive level-dependent threshold.

Besides these classical thresholding approaches, quite some contributions are available that consider thresholding within a Bayesian framework (e.g. Abramovich et al., 1998; Vidakovic, 1998; Clyde and George, 2000; Figueiredo and Nowak, 2001). Here, a prior distribution is imposed on the true wavelet coefficients of the underlying function in order to capture the sparse nature of the wavelet basis expansion. The thresholded wavelet coefficients are obtained by taking appropriate quantities from the resulting posterior distribution of the true wavelet coefficients, given the empirical wavelet coefficients. We examine two examples of Bayesian thresholding.

Firstly, a *mixture prior* of a Gaussian and a point mass at zero can be imposed on the wavelet

coefficients, i.e.

$$d_{j,k} \sim \pi_j N(0, \tau_j^2) + (1 - \pi_j)\delta(0), \tag{2.20}$$

with $0 \leq \pi_j \leq 1$ the prior probability of having non-zero wavelet coefficients at level j and $\delta(0)$ a point mass at zero. Abramovich et al. (1998) demonstrate that the posterior cumulative distribution $F(d_{j,k}|d^*_{j,k})$ is given by

$$F(d_{j,k}|d^*_{j,k}) = \frac{1}{1+\gamma_{j,k}}\Phi\left\{\frac{d_{j,k} - d^*_{j,k}\tau_j^2/(\sigma^2+\tau_j^2)}{\sigma\tau_j/\sqrt{(\sigma^2+\tau_j^2)}}\right\} + \frac{\gamma_{j,k}}{1+\gamma_{j,k}}I(\nu \geq 0), \tag{2.21}$$

with Φ the standard normal cumulative distribution function, and where the posterior odds ratio for the component at 0 is

$$\gamma_{j,k} = \frac{1-\pi_j}{\pi_j}\frac{\sqrt{\tau_j^2+\sigma^2}}{\sigma}\exp\left\{-\frac{\tau_j^2 d^{*2}_{j,k}}{2\sigma^2(\tau_j^2+\sigma^2)}\right\}. \tag{2.22}$$

Abramovich et al. (1998) suggest to use the posterior median to obtain the Bayesian estimate of the wavelet coefficients. This corresponds to a point estimate of the posterior distribution under a family of loss functions that is equivalent to the use of L^1 norms on the function and its derivatives. The thresholded coefficient is given by

$$\hat{d}^{(\text{mixture})}_{j,k} = \text{med}(d_{j,k}|d^*_{j,k}) = \text{sgn}(d^*_{j,k})\,\max(0, \chi_{j,k}), \tag{2.23}$$

where

$$\chi_{j,k} = \frac{\tau_j^2}{\sigma^2+\tau_j^2}|d^*_{j,k}| - \frac{\tau_j\sigma}{\sqrt{\sigma^2+\tau_j^2}}\Phi^{-1}\left\{\frac{1+\min(\gamma_{j,k},1)}{2}\right\}. \tag{2.24}$$

The hyperparameters π_j and τ_j^2 in the mixture prior are further defined as $\tau_j^2 = 2^{-\alpha j}C_1$ and $\pi_j = \min(1, 2^{-\beta j}C_2)$, where C_1, C_2, α and β are non-negative constants (Abramovich et al., 1998).

In the fourth wavelet thresholding procedure we consider, the second one within the Bayesian framework, a *normal prior* is imposed on the wavelet coefficients combined with a non-informative Jeffrey's hyperprior on the variance parameter, as described in Figueiredo and Nowak (2001), i.e.

$$\begin{aligned} d_{j,k}|\tau^2_{j,k} &\sim N(0, \tau^2_{j,k}) \\ p(\tau^2_{j,k}) &\propto \frac{1}{\tau^2_{j,k}}. \end{aligned} \tag{2.25}$$

It is shown in Figueiredo and Nowak (2001) that this leads to the following thresholded wavelet coefficients:

$$\hat{d}_{j,k}^{(\text{normal})} = \frac{(d_{j,k}^{*2} - 3\sigma^2)_+}{d_{j,k}^*}. \tag{2.26}$$

Whereas in the multiresolution analysis thresholding takes place scale by scale, each of the four thresholding procedures allows the thresholding to work on particular coefficients within each scale. In this way an adaptive smoothing of the noisy signal can be obtained. The denoised signal for the genomic region of the E2F data after applying each of these procedures is depicted in Figure 2.9. In all cases the error standard deviation σ is robustly estimated by the median absolute deviation (MAD) of the empirical wavelet coefficients at the finest level, divided by 0.6745 (e.g. Donoho and Johnstone, 1995; Abramovich et al., 1998). It is important to note that the threshold values used to produce Figure 2.9 are not identical for the four procedures. The hard and soft thresholding procedures make use of the *universal threshold* $\lambda = \sigma\sqrt{2\log(T)}$, proposed by Donoho and Johnstone (1994), while the threshold values for the Bayesian procedures can be obtained from Equations (2.23) and (2.26) for their respective thresholded wavelet coefficients.

At first sight, the hard and soft thresholding rules provide a very smooth thresholded signal. This denoised signal shows not to be very adaptive, however, as the intensity in the regions where much transcriptional activity is present tends be oversmoothed. The smoothness in these regions is only marginally smaller than in the non-expressed regions. Soft thresholding appears to lead to smoother results than hard thresholding. The two Bayesian methods seem to better capture the characteristic features in transcriptionally active regions compared to the classical thresholding rules, while still leading to relatively smooth results in non-expressed regions. In our example, the result for the mixture prior is a little bit smoother than that for the normal prior. In general, the desired behavior of adaptive smoothness of the thresholded signal is better obtained by the Bayesian thresholding methods than by classical hard or soft thresholding. In Chapter 3 we will therefore extend the ideas of thresholding in a Bayesian framework based on both possible priors to a multiple-curve setting.

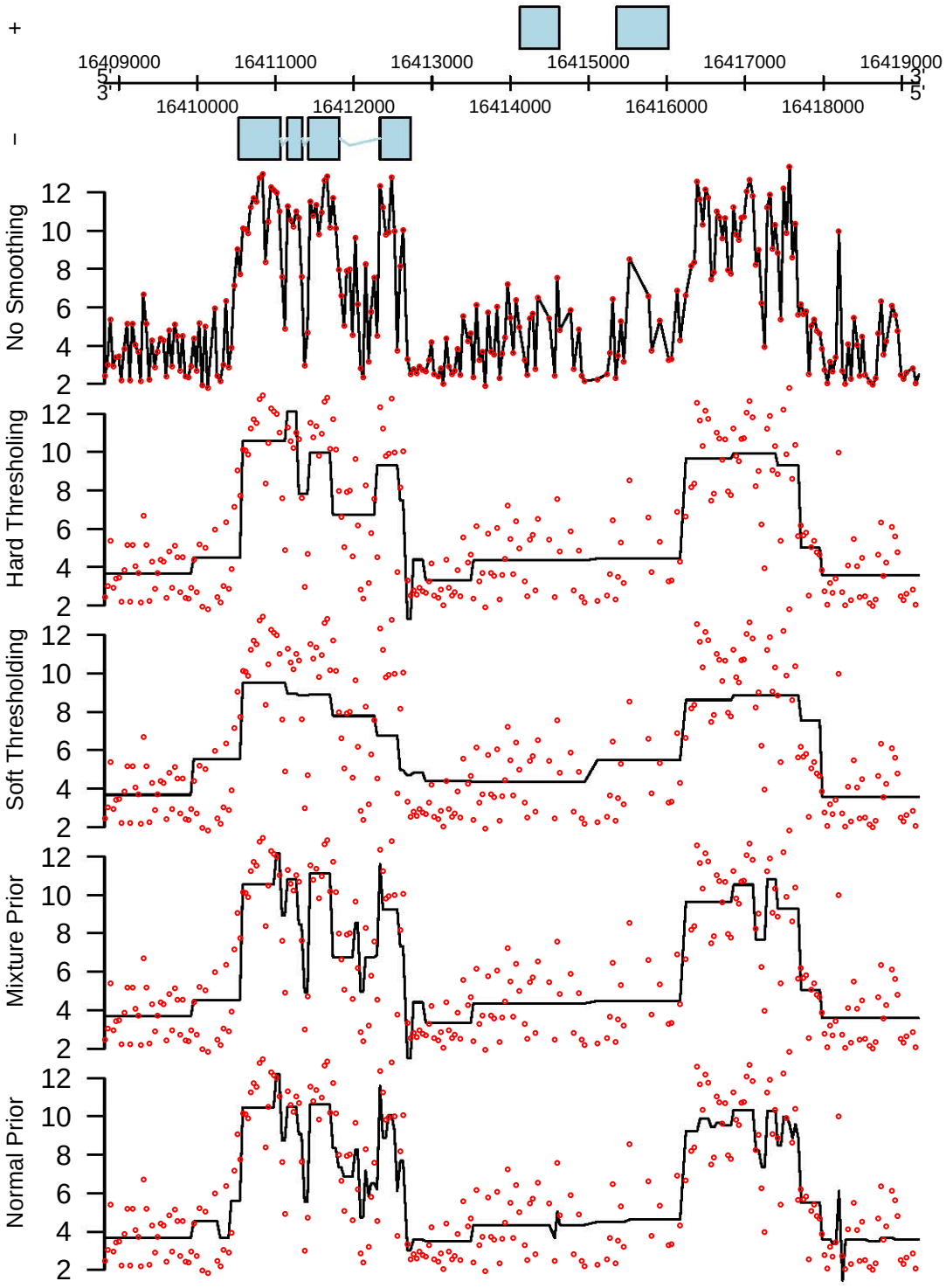


Figure 2.9: Different types of wavelet thresholding for genomic region of the first biological replicate of E2F plants. The hyperparameter values for the mixture prior are $\alpha = 0.5$, $\beta = 1$, $C_1 = 10134$ and $C_2 = 79$. For the normal prior track a non-informative Jeffrey's hyperprior is imposed on the variance parameter.

2.4.4 Inference based on posterior distributions of estimated functions

Within the Bayesian framework inference can be conducted by deriving credible intervals for the estimated function, using its posterior distribution. As indicated in Section 2.4.3, this distribution is obtained by applying the IDWT to the thresholded wavelet coefficients. Hence, it involves a linear combination of these coefficients. If a normal prior is imposed on the wavelet coefficients (Figueiredo and Nowak, 2001), the posterior distribution in the original data space is a linear combination of normal random variables and is thus easily found. However, in case one puts a mixture prior on the wavelet coefficients (e.g. Abramovich et al., 1998), a complicated mixture is obtained, which is impractical to evaluate analytically (Barber et al., 2002).

To avoid the need for time-consuming simulations, Barber et al. (2002) proposed to approximate the posterior distribution with Johnson curves (Johnson, 1949). This collection of distributions consists of three transformations of the standard normal distribution, i.e. (1) the log normal case, $z = \gamma + \delta \log(x - \zeta)$ with $\zeta < x$; (2) the unbounded case, $z = \gamma + \delta \sinh^{-1}\{(x - \zeta)/\eta\}$, and (3) the bounded case, $z = \gamma + \delta \log\{(x - \eta)/(\zeta + \eta - x)\}$, with $\zeta < x < \zeta + \eta$, in which z has a standard normal distribution and x is the Johnson variable. The Johnson curves form a rich family of distributions that provide good approximations of the tails of the distribution. At each location t there exists precisely one Johnson curve with the same first four cumulants, say $\kappa_1(X), \ldots, \kappa_4(X)$, as the posterior distribution of the smoother. These cumulants also have a direct interpretation: $\kappa_1(X)$ and $\kappa_2(X)$ are the mean and variance of X, respectively, $\kappa_3(X)/\kappa_2^{3/2}(X)$ is the skewness and $\kappa_4(X)/\kappa_2^2(X) + 3$ is the kurtosis.

An analytical solution for the first four cumulants of the posterior mixture distribution of the smoother in the data space can be found. The backtransformation from the wavelet space to the data space is a linear transformation and within the wavelet space the coefficients at each (j, k) are assumed to be independent. The cumulants of the distribution in the original data space can therefore be easily acquired by using the following standard properties of cumulants,

$$\kappa_r\left(\sum_i \phi_i Z_i\right) = \sum_i \phi_i^r \kappa_r(Z_i), \tag{2.27}$$

where the ϕ_i represent constants and the Z_i are independently distributed random variables. Once the cumulants are known within the wavelet space, they are readily available in the original data space by applying modified versions of the IDWT using (2.27). Within the wavelet space the posterior distribution of the wavelet coefficients is a mixture of a point mass at zero and

a normal distribution, with density $f(x) = (1-\pi)\delta(0) + \pi N(\mu, \xi^2)$. For such a mixture distribution analytical expressions for the cumulants can be calculated. The first four cumulants are given by

$$\begin{aligned}
\kappa_1 &= \pi\mu \\
\kappa_2 &= \pi\xi^2 + \pi\mu^2 - \pi^2\mu^2 \\
\kappa_3 &= 3\pi\xi^2\mu - 3\pi^2\xi^2\mu^3 + 2\pi^3\mu^3 \\
\kappa_4 &= 3\pi\xi^4 + 6\pi\xi^2\mu^2 - 18\pi^2\xi^2\mu^2 + 12\pi^3\xi^2\mu^2 - \\
&\qquad 3\pi^2\xi^4 + \pi\mu^4 - 7\pi^2\mu^4 + 12\pi^3\mu^4 - 6\pi^4\mu^4.
\end{aligned}$$

Johnson curves with exactly the same cumulants are now fitted using the method of moments and they are used as an approximation of the posterior distribution of the smoother with which inference can be conducted.

2.5 Objectives and outline

In Part I of this dissertation, we aim to develop a statistical method for transcriptome analysis with tiling arrays. The method should enable the simultaneous detection of genomic regions which are (1) transcriptionally active in a particular tissue, cell line or experimental condition and (2) differentially expressed between two such conditions. This feature is not present in existing methods. Since tiling probes are positioned along the genome with a more or less equal resolution and regardless of existing annotation, the probe intensity data can be thought of as realizations of an underlying expression function along the genomic coordinate. Therefore, we aim to use functional models.

The probe-to-probe fluctuations within the same transcriptional unit indicate that a certain degree of smoothing will be needed to obtain stable estimates of the functional effects. However, the spiky and discontinuous nature of the data asks for a more adaptive smoothing approach than is common in most functional data analysis applications. In principle, wavelet-based denoising seems suitable for this task. The use of wavelets allows for an efficient regularization of

the functional effects without losing the ability to model local features. Because of the inherent high-dimensionality of whole-genome tiling array data, it is also key to develop algorithms for fitting and inference that are fast and computationally efficient. One way to do this is to develop methods with closed-form solutions for the model parameter estimators.

Recently, expression studies with tiling arrays have become a common tool for whole-genome transcriptome analysis. As a consequence, more and more studies have more complex experimental set-ups than the simple one-group or two-group designs. The functional model framework that we envision therefore has to be flexible enough to be directly applicable for more complex designs. Finally, we also aim at disseminating our method and algorithms in the scientific community by providing a user-friendly software package.

Part I of this dissertation is organized as follows. In Chapter 3, we introduce fast wavelet-based functional models for transcriptome analysis with tiling arrays. In this chapter, we particularly focus on the two-group design. Tiling array expression studies with flexible designs are discussed in Chapter 4, while in Chapter 5 we present waveTiling, a R/Bioconductor software package for wavelet-based tiling array analysis. Finally, conclusions and future research perspectives for this part of the dissertation are given in Chapter 6.

Chapter 3

Fast wavelet-based functional models for transcriptome analysis with tiling arrays

In this chapter we present a wavelet-based method for transcriptome analysis with tiling arrays. We will focus on the problem of simultaneously detecting transcriptionally active and differentially expressed regions in the two-group design, without relying on existing annotation. In Section 3.1, we first introduce a functional model in the genomic space. Next, we use the DWT to obtain a wavelet-based functional model. The parameter estimation procedure which involves regularization is described in Section 3.2. In Section 3.3 we propose an empirical Bayes FDR procedure for identifying both expressed and differentially expressed regions. Finally, in Section 3.4, the wavelet-based method is compared to existing methods in a simulation study and it is applied to the *Arabidopsis thaliana* E2F study introduced in Chapter 2.

3.1 Wavelet-based functional models for transcriptome analysis

3.1.1 Functional model

Let $N(\mu, \sigma^2)$ denote a univariate normal distribution with mean μ and variance σ^2 and let $MVN(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denote a multivariate normal distribution with mean $\boldsymbol{\mu}$ and variance-covariance matrix $\boldsymbol{\Sigma}$. Suppose that N_1 and N_2 tiling arrays are collected for two distinct experimental

conditions C_1 and C_2, respectively. Let $N = N_1 + N_2$. The expression functions $Y_i(t)$, $i = 1, \ldots, N$, are evaluated on an equally spaced grid, say $\boldsymbol{t} = (1, ..., T)$, corresponding to the locations of the probes within the same chromosome. We consider the functional model

$$Y_i(t) = \beta_1(t) + X_{1,i}\beta_2(t) + E_i(t), \tag{3.1}$$

with $Y_i(t)$ the $\log_2$-transformed probe intensity of probe t on array i, $X_{1,i}$ a dummy variable which is 1 for C_1 and -1 for C_2, and $E_i(t)$ the zero-mean error term for which it is assumed that the $\boldsymbol{E}_i = [E_i(1), \ldots, E_i(T)]^T$ are i.i.d. $MVN(\mathbf{0}, \boldsymbol{\Sigma}_\epsilon)$, where $\boldsymbol{\Sigma}_\epsilon$ is a $T \times T$ covariance matrix defined on the grid $\boldsymbol{t} \times \boldsymbol{t}$. Hence, the intensities are assumed to be correlated within the same sample and are assumed to be independent across samples. The functions $\beta_1(t)$ and $\beta_2(t)$ are referred to as the mean and difference function, respectively. If the design is balanced the use of the $(-1,1)$ coding allows for an orthogonal estimation of both effect functions. After fitting the model, the estimated mean function, say $\hat{\beta}_1(t)$, can be used for transcript discovery. In particular, a segmentation can be performed by assessing in which genomic regions the mean intensity $\beta_1(t)$ exceeds a certain background level. The $(-1,1)$ dummy coding implies that the $\log_2$ fold change, which is used to detect differential expression between two experimental conditions, is given by $2 \times \beta_2(t) = FC(t)$. For Model (3.1) it is assumed that the $\log_2$ probe-level intensities are appropriately background-corrected and normalized.

Model (3.1) can be written in matrix form as

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{B} + \boldsymbol{E}. \tag{3.2}$$

Here, $\boldsymbol{Y}$ is an $N \times T$ matrix whose rows contain the $\log_2$-transformed preprocessed intensities of one array observed on $\boldsymbol{t}$. $\boldsymbol{X}$ is an $N \times q$ design matrix of the covariates. For Model (3.1) two effect functions are defined, so $q = 2$. However, all derivations in this chapter also hold for more complex designs with any arbitrary number of predictors q. These extended models are discussed in more detail in Chapter 4. Each row of the $q \times T$ matrix $\boldsymbol{B}$ contains one of the effect functions evaluated in $\boldsymbol{t}$. The rows of $\boldsymbol{E}$, $\boldsymbol{E}_i$ with $i = 1, ..., N$, consist of the error processes evaluated on $\boldsymbol{t}$, corresponding to each of the N observed tiling arrays.

3.1.2 Wavelet-based functional model

Since wavelets are scale- and location-dependent, they can be used to estimate functions in a very localized manner. Therefore, they are well suited to deal with irregular functional data that are characterized by a high number of local features. Only the coefficients of those basis functions whose support includes the region of the local feature are affected. Hence, wavelets can often provide a very economic representation of a function with relatively few non-zero coefficients. As is clear from Figure 2.7 in Section 2.4.2, a decomposition with Haar wavelets approximates the underlying function by a piecewise constant (e.g. Bruce and Gao, 1996). This is a very convenient feature in the analysis of tiling array expression data because the process of gene transcription can be considered as a piecewise constant function within the genomic domain. This means that probes that cover the same exonic region should, at least theoretically, measure the same expression signal. Therefore, the Haar wavelet will be the wavelet basis of our choice in the methods we describe. We note, however, that the methods are general enough to be applied with other wavelet bases as well.

Let us now return to the $N \times T$ matrix $\boldsymbol{Y}$ from Section 3.1.1 containing the $\log_2$ intensities observed on the probe positions $\boldsymbol{t}$. The projection of the intensities from the data space onto the wavelet space can be written as the matrix product $\boldsymbol{D} = \boldsymbol{Y}\boldsymbol{W}^T$, where $\boldsymbol{W}$ is a $T \times T$ orthogonal DWT matrix. Similar to the toy example of Section 2.4.2, the rows of the matrix $\boldsymbol{D}$ contain the empirical wavelet coefficients for each of the observed curves and they are double-indexed by the location $k = 1, ..., K_j$ within the wavelet scale $j = 0, ..., J$. The wavelet transform allows us to rewrite the model in the wavelet space by post-multiplying both sides of Model (3.2) with the DWT matrix $\boldsymbol{W}^T$, resulting in

$$\boldsymbol{D} = \boldsymbol{X}\boldsymbol{B}^* + \boldsymbol{E}^*. \tag{3.3}$$

Hence, $\boldsymbol{B}^* = \boldsymbol{B}\boldsymbol{W}^T$ and $\boldsymbol{E}^* = \boldsymbol{E}\boldsymbol{W}^T$ are the matrices whose rows contain the wavelet coefficients corresponding to the effect functions and the errors, respectively. Because the DWT is a linear projection, the rows of $\boldsymbol{E}^*$ are i.i.d. multivariate normal with mean zero and covariance matrix $\boldsymbol{S}^* = \boldsymbol{W}\boldsymbol{\Sigma}_\epsilon\boldsymbol{W}^T$. Similar to many contributions in wavelet literature, we will assume that the wavelet coefficients within a given curve are independent across locations k and wavelet scales j, i.e. $\boldsymbol{S}^*$ is a diagonal matrix. This assumption, however, does not imply independence

in the data space unless identical variance components are assumed across all wavelet scales j and locations k. Johnstone and Silverman (1997), for instance, argue that the variance of the empirical wavelet coefficients for stationary correlated noise only depends on the scale j of the wavelet decomposition, but remains constant across locations k within the same scale. Obviously, the assumptions on the correlation structure are further relaxed when the variance components are allowed to vary both within and across wavelet scales.

3.2 Parameter estimation and regularization

For the estimation of the parameters in Model (3.3) we adopt the ideas of Bayesian thresholding that were introduced in Section 2.4.3 in the single-curve setting of scatterplot smoothing. We examine the use of (1) the mixture prior variant, which we refer to as *WavMix*, and of (2) the normal prior variant, which we call *WavNorm*. The wavelet coefficients within a given curve are assumed to be independent (cfr. Section 3.1.2). Due to this property, the likelihood functions in both parameter estimation procedures can be factorized accordingly. This allows to more easily find closed-form solutions for the parameter estimators and their variances and thus develop fast algorithms for parameter estimation and subsequent inference.

3.2.1 Fitting procedure I: mixture prior (WavMix)

Let $\boldsymbol{D}(j,k)$ denote the empirical wavelet coefficients at scale j and location k associated with the background-corrected and quantile-normalized $\log_2$-transformed intensities. Further, let $\beta_m^*(j,k)$, $m = 1, ..., q$, be the wavelet coefficient of the m-th functional effect. The first fitting procedure exists in an adaptive regularization of the fixed effects functions by imposing a mixture prior on $\beta_m^*(j,k)$. This approach was also taken in e.g. Abramovich et al. (1998) and Morris and Carroll (2006). Within the wavelet space, Model (3.1) can be written as

$$\boldsymbol{D}(j,k)|\boldsymbol{\beta}^*(j,k) \sim MVN\left(\boldsymbol{X}\boldsymbol{\beta}^*(j,k), \boldsymbol{I}\sigma_\epsilon^2\right), \tag{3.4}$$

$$\beta_m^*(j,k) \sim \pi_m(j) N\left(0, \tau_m(j)\sigma_\epsilon^2\right) + \{1 - \pi_m(j)\}\, \delta_0(\beta_m^*), \tag{3.5}$$

with $0 \leq \pi_m(j) \leq 1$ and $\delta_0(\beta_m^*)$ the density function of a point mass at zero. The prior probability $\pi_m(j)$ gives the proportion of non-zero wavelet coefficients at level j for the effect function parameters. Note that the error variance σ_ϵ^2 is assumed equal across wavelet scales j and locations k, which is a common approach in the wavelet literature (e.g. Donoho and Johnstone, 1995).

Based on biological grounds, the mixture prior formulation allows to incorporate the assumption that differential expression can only occur for features that are expressed. This implies that $\beta_1^*(j,k)$ and $\beta_2^*(j,k)$ are both non-zero in differentially expressed regions. Therefore, the prior distributions on the effect function parameters $\beta_1^*(j,k)$ and $\beta_2^*(j,k)$ may be written as

$$\begin{aligned} \beta_1^*(j,k) &\sim \{\pi_1(j)+\pi_2(j)\}\, N\left(0,\tau_1(j)\sigma_\epsilon^2\right) + \\ &\qquad \{1-\pi_1(j)-\pi_2(j)\}\,\delta_0(\beta_1^*), \end{aligned} \tag{3.6}$$

$$\beta_2^*(j,k) \sim \pi_2(j)\; N\left(0,\tau_2(j)\sigma_\epsilon^2\right) + \{1-\pi_2(j)\}\,\delta_0(\beta_2^*). \tag{3.7}$$

The marginal density of the data after integrating out the functional effects is given by (see also Appendix A)

$$\begin{aligned} f\left(\boldsymbol{D}(j,k)\right) &= \{1-\pi_1(j)-\pi_2(j)\}\, g_0\left(\boldsymbol{D}(j,k)\right) + \\ &\qquad \pi_1(j) g_1\left(\boldsymbol{D}(j,k)\right) + \pi_2(j) g_2\left(\boldsymbol{D}(j,k)\right), \end{aligned} \tag{3.8}$$

with

$$g_0\left(\boldsymbol{D}(j,k)\right) = MVN\left(\boldsymbol{0}, \boldsymbol{I}\sigma_\epsilon^2\right), \tag{3.9}$$

$$g_1\left(\boldsymbol{D}(j,k)\right) = MVN\left(\boldsymbol{0}, \boldsymbol{V}_1(j)\sigma_\epsilon^2\right), \tag{3.10}$$

$$g_2\left(\boldsymbol{D}(j,k)\right) = MVN\left(\boldsymbol{0}, \boldsymbol{V}_2(j)\sigma_\epsilon^2\right), \tag{3.11}$$

and (e.g. Verbeke and Molenberghs, 2000)

$$\boldsymbol{V}_1(j) = \boldsymbol{I} + \boldsymbol{X}_1^T \boldsymbol{X}_1 \tau_1(j), \tag{3.12}$$

$$\boldsymbol{V}_2(j) = \boldsymbol{I} + \boldsymbol{X}_1^T \boldsymbol{X}_1 \tau_1(j) + \boldsymbol{X}_2^T \boldsymbol{X}_2 \tau_2(j). \tag{3.13}$$

Let the posterior probabilities of non-zero wavelet coefficients at level j for the effect function parameters be denoted by

$$\omega_1(j,k) = \frac{\pi_1(j) g_1\left(\boldsymbol{D}(j,k)\right)}{f\left(\boldsymbol{D}(j,k)\right)}, \tag{3.14}$$

$$\omega_2(j,k) = \frac{\pi_2(j) g_2\left(\boldsymbol{D}(j,k)\right)}{f\left(\boldsymbol{D}(j,k)\right)}. \tag{3.15}$$

The posterior distributions of $\beta_1^*(j,k)$ and $\beta_2^*(j,k)$, given the observed values of $\boldsymbol{D}(j,k)$, are then given by

$$\begin{aligned} \beta_1^*(j,k)|\boldsymbol{D}(j,k) \;\sim\; & \{\omega_1(j,k)+\omega_2(j,k)\}\, N\left(\hat{\beta}_1^*(j,k), \sigma^2_{\hat{\beta}_1}(j)\right) + \\ & \{1-\omega_1(j,k)-\omega_2(j,k)\}\,\delta(0), \end{aligned} \tag{3.16}$$

$$\beta_2^*(j,k)|\boldsymbol{D}(j,k) \;\sim\; \omega_2(j,k)\, N\left(\hat{\beta}_2^*(j,k), \sigma^2_{\hat{\beta}_2}(j)\right) + \{1-\omega_2(j,k)\}\,\delta(0). \tag{3.17}$$

The estimators of the effect functions in Equations (3.16) and (3.17) are the corresponding posterior means. They have the standard form of classical ridge regression estimators, i.e.

$$\hat{\beta}_1^*(j,k) = \left\{\boldsymbol{X}_1^T\boldsymbol{X}_1 + 1/\tau_1(j)\right\}^{-1}\boldsymbol{X}_1^T\boldsymbol{D}(j,k), \tag{3.18}$$

$$\hat{\beta}_2^*(j,k) = \left\{\boldsymbol{X}_2^T\boldsymbol{X}_2 + 1/\tau_2(j)\right\}^{-1}\boldsymbol{X}_2^T\boldsymbol{D}(j,k), \tag{3.19}$$

while their variances have the standard from of the variances of classical ridge regression estimators, i.e.

$$\sigma^2_{\hat{\beta}_1}(j) = \sigma^2_\epsilon\left\{\boldsymbol{X}_1^T\boldsymbol{X}_1 + 1/\tau_1(j)\right\}^{-1}, \tag{3.20}$$

$$\sigma^2_{\hat{\beta}_2}(j) = \sigma^2_\epsilon\left\{\boldsymbol{X}_2^T\boldsymbol{X}_2 + 1/\tau_2(j)\right\}^{-1}. \tag{3.21}$$

A derivation of the posterior densities can be found in Appendix A.

For the model to be fully specified, we still have to define the hyperparameters $\pi_m(j)$ and $\tau_m(j)$. Extending the expressions in Abramovich et al. (1998) to the functional model framework, we assume the hyperparameters of the prior model to be of the form

$$\tau_m(j) \;=\; c_m 2^{-\alpha_m j}, \tag{3.22}$$

$$\pi_1(j) \;=\; \min\left(1-\pi_2(j), q_1 2^{-\phi_1 j}\right), \tag{3.23}$$

$$\pi_2(j) \;=\; \min\left(1, q_2 2^{-\phi_2 j}\right), \tag{3.24}$$

where $m = 1, 2$ and c_m, q_1, q_2, α_m, ϕ_1 and ϕ_2 are non-negative constants. Abramovich et al. (1998) have shown that 0.5 and 1 are robust choices for α_m and ϕ_m, respectively. We have chosen to impose the same degree of shrinkage to all non-zero wavelet coefficients by setting $c_1 = c_2 = c$. The differences in smoothness between the effect functions will thus only be influenced by their corresponding prior probabilities π_m.

When the noise level σ_ϵ is unknown, it can be robustly estimated by the MAD of the empirical wavelet coefficients at the finest level, divided by 0.6745 (e.g. Donoho and Johnstone, 1995; Abramovich et al., 1998). The remaining hyperparameters can be estimated by *empirical Bayes* using direct marginal maximum likelihood (MML), based on the marginal density given in (3.8). The MML estimators are obtained by numerical optimization of the marginal log-likelihood.

3.2.2 Fitting procedure II: normal prior (WavNorm)

The second fitting procedure is described in Clement et al. (2012). The model is defined as

$$\boldsymbol{D}(j,k)|\boldsymbol{\beta}^*(j,k) \sim MVN\left\{\boldsymbol{X}\boldsymbol{\beta}^*(j,k), \boldsymbol{I}\sigma_\epsilon^2(j,k)\right\}, \tag{3.25}$$

where $j = 0, ..., J$ and $k = 1, ..., K_j$.

Regularization can be obtained by imposing a Gaussian prior on the wavelet coefficients $\boldsymbol{\beta}^*(j,k)$:

$$\beta_m^*(j,k)|\tau_m(j,k) \sim N\left\{0, \tau_m(j,k)\sigma_\epsilon^2(j,k)\right\}. \tag{3.26}$$

The hierarchical model (3.25)-(3.26) is a linear mixed effects model within the wavelet space. In this model specification the error variances $\sigma_\epsilon^2(j,k)$ as well as the smoothing parameters $\tau_m(j,k)$ are allowed to vary with scale j and location k. While the model has a Bayesian model interpretation, a fully Bayesian modeling approach would involve the specification of a prior distribution on the variance components $\sigma_\epsilon^2(j,k)$ as well. However, Model (3.25)-(3.26) also accomodates empirical Bayesian methods, i.e. the fully Bayesian analysis chain can be broken by replacing the unknown smoothing parameters $\tau_m(j,k)$ and variances $\sigma_\epsilon^2(j,k)$ by estimates and then performing a Bayesian analysis with the previously unknown parameters regarded as fixed (e.g. Section 16.3 of Ruppert et al., 2003).

The joint likelihood for $\boldsymbol{D}$ and $\boldsymbol{\beta}^*$ within this empirical Bayes setting becomes

$$p\left\{\boldsymbol{D},\boldsymbol{\beta}^*|\boldsymbol{\tau},\boldsymbol{\sigma}_\epsilon^2\right\} \propto \prod_{j=0}^{J}\prod_{k=1}^{K_j}\Bigg(\left[\sigma_\epsilon^2(j,k)\right]^{-N/2} \exp\left\{-\frac{\left[\boldsymbol{D}(j,k)-\boldsymbol{X}\boldsymbol{\beta}^*(j,k)\right]^T\left[\boldsymbol{D}(j,k)-\boldsymbol{X}\boldsymbol{\beta}^*(j,k)\right]}{2\sigma_\epsilon^2(j,k)}\right\} \prod_{m=1}^{q}\left[\tau_m(j,k)\sigma_\epsilon^2(j,k)\right]^{-1/2}\exp\left[-\frac{\beta_m^{*2}(j,k)}{2\tau_m(j,k)\sigma_\epsilon^2(j,k)}\right]\Bigg), \quad (3.27)$$

with $\boldsymbol{\tau}=[\tau_1(1,1),...,\tau_q(J,K_J)]^T$ and $\boldsymbol{\sigma}_\epsilon^2=[\sigma_\epsilon^2(1,1),...,\sigma_\epsilon^2(J,K_J)]^T$.

The marginal likelihood corresponding to Model (3.25)-(3.26) is defined as

$$p\left\{\boldsymbol{D}|\boldsymbol{\tau},\boldsymbol{\sigma}_\epsilon^2\right\} \propto \prod_{j=0}^{J}\prod_{k=1}^{K_j}\left|\boldsymbol{V}(j,k)\sigma_\epsilon^2(j,k)\right|^{-1/2}\times \exp\left[-\frac{\boldsymbol{D}^T(j,k)\boldsymbol{V}^{-1}\boldsymbol{D}(j,k)}{2\sigma_\epsilon^2(j,k)}\right], \quad (3.28)$$

with

$$\boldsymbol{V}(j,k)=\boldsymbol{I}+\sum_{m=1}^{q}\tau_m(j,k)\boldsymbol{X}_m\boldsymbol{X}_m^T. \quad (3.29)$$

Following e.g. the appendix of Seber (1984), in the case of an orthogonal design matrix $\boldsymbol{X}$, the determinant of $\boldsymbol{V}(j,k)$ simplifies to

$$|\boldsymbol{V}(j,k)|=\prod_{m=1}^{q}(\boldsymbol{X}_m^T\boldsymbol{X}_m\tau_m(j,k)+1), \quad (3.30)$$

and therefore

$$\boldsymbol{V}^{-1}(j,k)=\boldsymbol{I}-\sum_{m=1}^{q}\frac{\boldsymbol{X}_m\boldsymbol{X}_m^T}{\boldsymbol{X}_m^T\boldsymbol{X}_m+1/\tau_m(j,k)}. \quad (3.31)$$

Upon using Equations (3.27) and (3.28), the posterior densities of the effect functions can be derived. When the variances $\sigma_\epsilon^2(j,k)$ and smoothing parameters $\tau_m(j,k)$ are assumed to be known, the posterior of $\boldsymbol{\beta}^*(j,k)$ only involves $\boldsymbol{D}(j,k)$, $\sigma_\epsilon^2(j,k)$ and $\tau_m(j,k)$. For an orthogonal design matrix $\boldsymbol{X}$ these posterior densities are given by

$$p\left\{\beta_m^*(j,k)|\boldsymbol{D}(j,k),\tau_m(j,k),\sigma_\epsilon^2(j,k)\right\} \sim N\left\{\frac{\boldsymbol{X}_m^T\boldsymbol{D}(j,k)}{\boldsymbol{X}_m^T\boldsymbol{X}_m+1/\tau_m(j,k)},\frac{\sigma_\epsilon^2(j,k)}{\boldsymbol{X}_m^T\boldsymbol{X}_m+1/\tau_m(j,k)}\right\}. \quad (3.32)$$

A derivation of the posterior distribution is given in Appendix B.

In this second fitting procedure a MML approach is taken. MML is commonly used for deriving empirical Bayes estimators in wavelet-based scatterplot smoothing (e.g. Figueiredo and Nowak, 2001) and is here generalized towards a multiple-curves functional data analysis context. After replacing $|\boldsymbol{V}(j,k)|$ and $\boldsymbol{V}^{-1}(j,k)$ in Equation (3.28), minus twice the marginal log-likelihood becomes

$$-2 \times l(\boldsymbol{D}|\boldsymbol{\tau}, \boldsymbol{\sigma}_\epsilon^2) \propto \sum_{j=0}^{J} \sum_{k=1}^{K_j} \left(\sum_{m=1}^{q} \left\{ \log \left[\boldsymbol{X}_m^T \boldsymbol{X}_m \tau_m(j,k) + 1 \right] \right\} + N \log \left[\sigma_\epsilon^2(j,k) \right] + \frac{1}{\sigma_\epsilon^2(j,k)} \boldsymbol{D}^T(j,k) \left[\boldsymbol{I} - \sum_{m=1}^{q} \frac{\boldsymbol{X}_m \boldsymbol{X}_m^T}{\boldsymbol{X}_m^T \boldsymbol{X}_m + 1/\tau_m(j,k)} \right] \boldsymbol{D}(j,k) \right). \quad (3.33)$$

When the variance of the noise $\sigma_\epsilon^2(j,k)$ is known, the MML-estimator of $\tau_m(j,k)$ can be obtained in closed form:

$$\hat{\tau}_m(j,k) = \left[\frac{\boldsymbol{D}^T(j,k) \boldsymbol{X}_m \boldsymbol{X}_m^T \boldsymbol{D}(j,k)}{\left(\boldsymbol{X}_m^T \boldsymbol{X}_m \right)^2 \sigma_\epsilon^2(j,k)} - \frac{1}{\boldsymbol{X}_m^T \boldsymbol{X}_m} \right]_+ . \quad (3.34)$$

A detailed derivation of this result is given in Appendix C.

Similar to Figueiredo and Nowak (2001), we can also impose a Jeffrey's prior on the smoothing parameters $\tau_m(j,k)$. For our model, the uninformative Jeffrey's prior is defined as

$$p(\boldsymbol{\tau}) \propto |I(\boldsymbol{\tau})|^{1/2}, \quad (3.35)$$

where $|I(\boldsymbol{\tau})|$ is the determinant of the expected Fisher information matrix of the marginal model (e.g. Ibrahim and Laud, 1991). With similar derivations as presented in Appendix C it can be easily verified that the Jeffrey's rule prior for $\boldsymbol{\tau}$ becomes

$$p(\boldsymbol{\tau}) \propto \prod_{j=0}^{J} \prod_{k=1}^{K_j} \prod_{m=1}^{q} (\boldsymbol{X}_m^T \boldsymbol{X}_m \tau_m(j,k) + 1)^{-1}, \quad (3.36)$$

and that the use of this additional prior on $\boldsymbol{\tau}$ simply alters estimator (3.34) by dividing its first term by a factor 3:

$$\hat{\tau}_{m,\text{Jeffrey}}(j,k) = \left[\frac{\boldsymbol{D}^T(j,k) \boldsymbol{X}_m \boldsymbol{X}_m^T \boldsymbol{D}(j,k)}{3 \left(\boldsymbol{X}_m^T \boldsymbol{X}_m \right)^2 \sigma_\epsilon^2(j,k)} - \frac{1}{\boldsymbol{X}_m^T \boldsymbol{X}_m} \right]_+ . \quad (3.37)$$

This imposes additional shrinkage. Note that in the special case of $q = 1$, $N = 1$ and $X = 1$, estimator (3.37) reduces to $\left[\frac{D^2(j,k)}{3\sigma_\epsilon^2(j,k)} - 1\right]_+$, which is equivalent to the result of Figueiredo and Nowak (2001) for wavelet denoising in a single-curve setting.

In the expression for $\hat{\tau}_m(j,k)$ the variance $\sigma_\epsilon^2(j,k)$ is assumed to be known. In real applications, however, it needs to be estimated. Similar to Section 3.2.1, σ_ϵ can be robustly estimated by the MAD of the empirical wavelet coefficients at the finest level, divided by 0.6745 (e.g. Donoho and Johnstone, 1995; Abramovich et al., 1998). Alternatively, the MML can be used for the estimation of the variances of the noise component at the different wavelet scales j and locations k. Given the smoothing parameters $\boldsymbol{\tau}(j,k)$, the MML-estimator of $\sigma_\epsilon^2(j,k)$ becomes

$$\hat{\sigma}_\epsilon^2(j,k) = \frac{1}{N}\boldsymbol{D}^T(j,k)\boldsymbol{V}^{-1}(j,k)\boldsymbol{D}(j,k). \tag{3.38}$$

It is also possible to assume fixed variances within each wavelet scale. The MML-estimator of $\sigma_\epsilon^2(j)$ then becomes

$$\hat{\sigma}_\epsilon^2(j) = \frac{1}{NK_j}\sum_{k=1}^{K_j}\boldsymbol{D}^T(j,k)\boldsymbol{V}^{-1}(j,k)\boldsymbol{D}(j,k). \tag{3.39}$$

Detailed derivations can be found in Appendix C. In the full estimation procedure a Gauss-Seidel algorithm is then used (e.g. Givens and Hoeting, 2005) for the maximization of the marginal likelihood. In particular, we apply the following iterative algorithm:

1. Choose initial parameter values for $\sigma_\epsilon^{2^{(l)}}(j,k)$ ($l = 0$). The MAD-based estimator can be used for this purpose.
2. Calculate $\tau_m^{(l+1)}(j,k)$ by plugging $\sigma_\epsilon^{2^{(l)}}(j,k)$ into (3.34).
3. Calculate $\sigma_\epsilon^{2^{(l+1)}}(j,k)$ by using $\tau_m^{(l+1)}(j,k)$ in the expression for $\boldsymbol{V}(j,k)$ (3.29) in (3.38) or (3.39).
4. Increase l with 1 and repeat steps 2-3 until convergence.

Once we have estimated $\hat{\tau}_m(j,k)$ and $\hat{\sigma}_\epsilon^2(j,k)$, we can plug them into the posterior densities of the effect functions given by Equation (3.32). The result illustrates that Model (3.25)-(3.26) performs adaptive regularization of the wavelet coefficients $\beta_m^*(j,k)$ by specifying location- and scale-dependent regularization parameters $\tau_m(j,k)$. Thresholding takes place whenever $\tau_m(j,k) = 0$.

The empirical Bayes method as described above, however, ignores the extra variability in the posterior distribution caused by estimating the variance components. The posterior variance of the effect functions should be calculated from the joint posterior distribution of $\{\tau_m(j,k), \beta_m^*(j,k)\}$. We use the standard identity (e.g. Ruppert et al., 2003)

$$\begin{aligned} \text{Var}\left\{\beta_m^*(j,k)|\boldsymbol{D}(j,k)\right\} &= \text{E}\left[\,\text{Var}\left\{\beta_m^*(j,k)|\boldsymbol{D}(j,k),\boldsymbol{\theta}\right\}\right] + \\ &\quad \text{Var}\left[\,\text{E}\left\{\beta_m^*(j,k)|\boldsymbol{D}(j,k),\boldsymbol{\theta}\right\}\right], \end{aligned} \tag{3.40}$$

with $\boldsymbol{\theta} = (\boldsymbol{\tau}, \boldsymbol{\sigma}^2)$ the parameter vector of all parameters that need to be estimated in the empirical Bayes procedure. Note that the first term in (3.40) is well approximated by the posterior variance of $\beta_m^*(j,k)$ where $\tau_m(j,k)$ and $\sigma_\epsilon^2(j,k)$ are treated as known and fixed at their posterior mode (Kass and Steffey, 1989). The second term thus corrects for the extra variability in the posterior distribution of $\beta_m^*(j,k)$ that is not accounted for by the approximate posterior variance. We estimate $\text{Var}\left[\,\text{E}\left\{\beta_m^*(j,k)|\boldsymbol{D}(j,k),\boldsymbol{\theta}\right\}\right]$ through the following two steps:

1. Use a parametric bootstrap (e.g. Efron and Tibshirani, 1993) to estimate the covariance matrix of $\boldsymbol{\theta}$: $\hat{\boldsymbol{\Sigma}}_\theta$. First, B bootstrap samples of the data are generated from the wavelet-based model. Next, the model is fitted for each bootstrap sample resulting in B bootstrap estimates $\hat{\boldsymbol{\theta}}^*(b)$, with $b = 1, \ldots, B$. Finally, $\hat{\boldsymbol{\Sigma}}_\theta$ is obtained by taking the sample variance of the B bootstrap estimates.

2. Plug the results from step 1 into the delta-method formula (e.g. Ruppert et al., 2003):

$$\text{Var}\left[\,\text{E}\left\{\beta_m^*(j,k)|\boldsymbol{D}(j,k),\boldsymbol{\theta}\right\}\right] = \left\{\left.\frac{\partial\hat{\beta}_m^*(j,k)}{\partial\boldsymbol{\theta}}\right|_{\hat{\boldsymbol{\theta}}}\right\}^T \hat{\boldsymbol{\Sigma}}_\theta \left\{\left.\frac{\partial\hat{\beta}_m^*(j,k)}{\partial\boldsymbol{\theta}}\right|_{\hat{\boldsymbol{\theta}}}\right\}. \tag{3.41}$$

The partial derivatives of $\hat{\beta}_m^*(j,k)$ are available analytically. Since the correction is a relatively small portion of the corrected posterior variance, it needs not be estimated by the bootstrap with as great a precision as when a variance is estimated entirely by the bootstrap (Ruppert et al., 2003).

3.3 Empirical Bayes inference for tiling array data

In this section an inference procedure for tiling array experiments using an empirical Bayes FDR procedure is described. The FDR procedure relies on the posterior distributions of the effect functions from the wavelet-based functional model. In particular, the mean function $\beta_1(t)$ is used for transcript discovery and the $\log_2$ fold change $FC(t) = 2 \times \beta_2(t)$ for assessing differential expression. The FDR procedures that are presented here are based on the work of Newton et al. (2004). However, we avoid the use of computationally intensive Bayesian Markov chain Monte Carlo (MCMC) methods. This may be seen as an advantage when dealing with a large number of observations.

3.3.1 Empirical Bayes FDR procedure

In tiling microarray experiments differentially expressed regions across treatments can be identified by statistical hypothesis testing. However, they are often found to be only weakly related to the magnitude of the fold change cut-off (e.g. DeRisi et al., 1996; Schena et al., 1996). The use of thresholds is more intuitive for biological or biomedical researchers, but until recently it was lacking statistical rigor. McCarthy and Smyth (2009) developed an empirical Bayes moderated t-statistic for inferring if the fold change is above the threshold, while Morris et al. (2008) provided a procedure for functional models that flags regions significantly exceeding a δ_{FC} fold change between treatment groups while controlling the expected Bayesian FDR at the desired level α. Here, we will use a similar approach. At each probe position t, three different differential expression statuses exist, i.e. overexpression, no differential expression and underexpression.

When there is an overexpression at a particular probe t, the posterior probability is given by

$$p_{DE,1}(t) = \Pr\left\{2 \times \beta_2(t) > \log_2(\delta_{FC}) | \boldsymbol{Y}\right\}. \tag{3.42}$$

When there is no biologically relevant differential expression at a particular probe t, this posterior probability becomes

$$p_{DE,0}(t) = \Pr\left\{-\log_2(\delta_{FC}) \leq 2 \times \beta_2(t) \leq \log_2(\delta_{FC}) | \boldsymbol{Y}\right\}. \tag{3.43}$$

Finally, the posterior probability when there is underexpression at a particular probe t is given

by

$$p_{DE,2}(t) = \Pr\{2 \times \beta_2(t) < -\log_2(\delta_{FC})|\boldsymbol{Y}\}. \tag{3.44}$$

A probe at a certain position can be classified according to the largest among these three posterior probabilities. The local Bayesian FDR (BFDR) (e.g. Newton et al., 2004; Efron, 2003) corresponding to over- or underexpression is then calculated by

$$BFDR_{DE,r}(t) = 1 - p_{DE,r}(t), \tag{3.45}$$

with $r = 1, 2$. Hence, the BFDR is defined as the posterior probability that the fold change does not exceed the threshold value, given the observed data. Setting $\log_2(\delta_{FC}) = 0$ will return all statistically significant results for which the null hypothesis of equal expression can be rejected. Hence, the method can be used for selecting all probes for which the $\log_2$ fold change is statistically significantly different from zero, as well as for returning results that are both of statistical and practical significance.

Model (3.1) can also be used for inferring on transcript discovery by using the mean function $\beta_1(t)$ and a threshold δ_{TD} for the background intensity. The local Bayesian FDR becomes

$$BFDR_{TD}(t) = 1 - \Pr\{\beta_1(t) > \delta_{TD}|\boldsymbol{Y}\}. \tag{3.46}$$

Quantities (3.45) and (3.46) can be used to identify probes that correspond to significantly (differentially) expressed genomic targets, i.e. probes for which $BFDR(t) < \alpha$. They are calculated based on the empirical Bayes posterior distributions obtained from the fitting procedures described in Section 3.2. Morris et al. (2008), on the other hand, would evaluate them using an MCMC approach. The significant probes can be combined in significantly (differentially) expressed regions. These are defined as all sets of probes ϕ_m that can be constructed by joining neighboring locations t_l for which $BFDR(t_l) < \alpha$. Hence, the ϕ_m correspond to consecutive genomic regions with a BFDR below the significance level α.

When the WavNorm procedure, described in Section 3.2.2, is applied, the marginal posterior distributions of $\beta_1(t)$ and $2 \times \beta_2(t)$ are easily calculated as a linear combination of univariate normal distributions. These distributions correspond with the posterior distributions of the $\beta^*_m(j,k)$ in the wavelet space, given by (3.32). However, if the WavMix procedure, described in Section 3.2.1, is used, the posterior distributions of the effect functions in the data space are intractable since they involve linear combinations of mixture distributions. We propose to

approximate these distributions by means of Johnson curves (Johnson, 1949), according to the procedure explained in Section 2.4.4.

3.4 Results and discussion

The two proposed procedures are first evaluated in a simulation study and compared to popular methods for transcript discovery and differential expression. Next, the model is applied to the *Arabidopsis thaliana* E2F case study.

3.4.1 Simulation study

3.4.1.1 Simulating tiling array expression data

Tiling array data are simulated by adapting the model of Purdom et al. (2008):

$$y_i(t) = \log_2 \left[B(t) + I_i(t) \times 2^{c_i(t)+p(t)} \right] + \epsilon_i(t), \tag{3.47}$$

with $y_i(t)$ the $\log_2$-transformed intensity of probe t on array i and where

$$\begin{cases} \log_2 \{B(t)\} & \sim N(\mu_B, \sigma_B^2) \\ c_i(t) & \sim N\{\mu_{c,i}(t), \sigma_c^2\} \\ p(t) & \sim N(0, \sigma_p^2) \\ \epsilon_i(t) & = \text{ARMA}(m,n) + w_i(t) \\ w_i(t) & \sim N(0, \sigma_w^2) \end{cases} . \tag{3.48}$$

In (3.48) $\log_2 \{B(t)\}$ is the $\log_2$-transformed background hybridization signal at probe t, $c_i(t)$ denotes the mean effect of an expressed exon on the same array, $p(t)$ accounts for the differences in probe affinity and $\epsilon_i(t)$ is the noise component of probe t on chip i, which follows an autoregressive moving average process ARMA(m,n) with Gaussian white noise $w_i(t)$. The indicator variable $I_i(t)$ is used to add hybridization signal, i.e. $I_i(t) = 1$ in all exonic regions that are assumed to be expressed and $I_i(t) = 0$ for all probes covering intronic or intergenic regions and exons of genes that are non-expressed in the simulation study. This model features additive background, multiplicative noise, probe-specific affinities and serial correlation in the

data domain. We follow the approach of Purdom et al. (2008) for tuning the simulation parameters by rough estimates of realistic values from real data. An ANOVA model is fitted to the *Arabidopsis thaliana* E2F data for this purpose. The model consists of a fixed probe effect and a fixed group effect, either WT or E2F, nested within probe. The errors of the ANOVA model are serially correlated and can be modeled by an ARMA$(1, 1)$ process. They are used for estimating the AR, MA parameters and the variance parameter of the white noise.

The values for μ_B and σ_B are determined by characterizing the distribution of background noise empirically. This is done by applying the method described in David et al. (2006) on the raw intensities of the E2F experiment. The method works with the intensities for probes that do not overlap with any annotated features. Figure 3.1 shows the nonparametric density estimate of these intensities for the E2F data. The distribution is asymmetric with a sharp peak, corresponding to background probes, and a heavy right tail, corresponding to probes that are targeting non-annotated transcripts. This distribution is considered as a mixture between a normal distribution (the peak) and a second distribution which is further left unspecified (the heavy tail, sometimes called *shoulder*). The mean parameter, μ_B, of the normal component of the mixture is estimated by the mode of the mixture distribution and the standard deviation, σ_B, by the MAD of the distribution that is obtained by mirroring the part of the mixture with values $\leq \mu_B$ about the axis $x = \mu_B$. This results in the following estimates: $\hat{\mu}_B = 6.35$ and $\hat{\sigma}_B = 0.15$. In the simulation study μ_B and σ_B are thus set at those values.

The mean chip effect in exonic regions, $\mu_{c,i}(t) = \mu_c(t) + FC_i(t)$, is arbitrary. It is the mean expression level of the simulated gene in the treatment group corresponding to chip i, with $\mu_c(t)$ the average expression level for a certain gene over all the arrays and $FC_i(t)$ the average $\log_2$ fold change in the group of chip i. The expression standard deviation σ_c is set at 0.2, which approximately equals the standard deviation of the mean expression levels between exonic regions within each group in the E2F study. The standard deviation of the probe effect $p(t)$ is set at $\sigma_p = 1.5$, which corresponds to the 75% quantile of the empirical distribution of the standard deviations of probes that target the same exonic region.

With this model we simulate *Arabidopsis thaliana* data for the entire first chromosome, which is interrogated by 741760 probes. In the simulation study 3 replicates of WT plants and 3 replicates of E2F plants are considered. The average hybridization signal for annotated regions is chosen at random from the set $(0, 3, 4, 5, 6, 7, 8, 9, 10, 11)$ with probabilities ($1/15$, $1/15$, $1/15$,

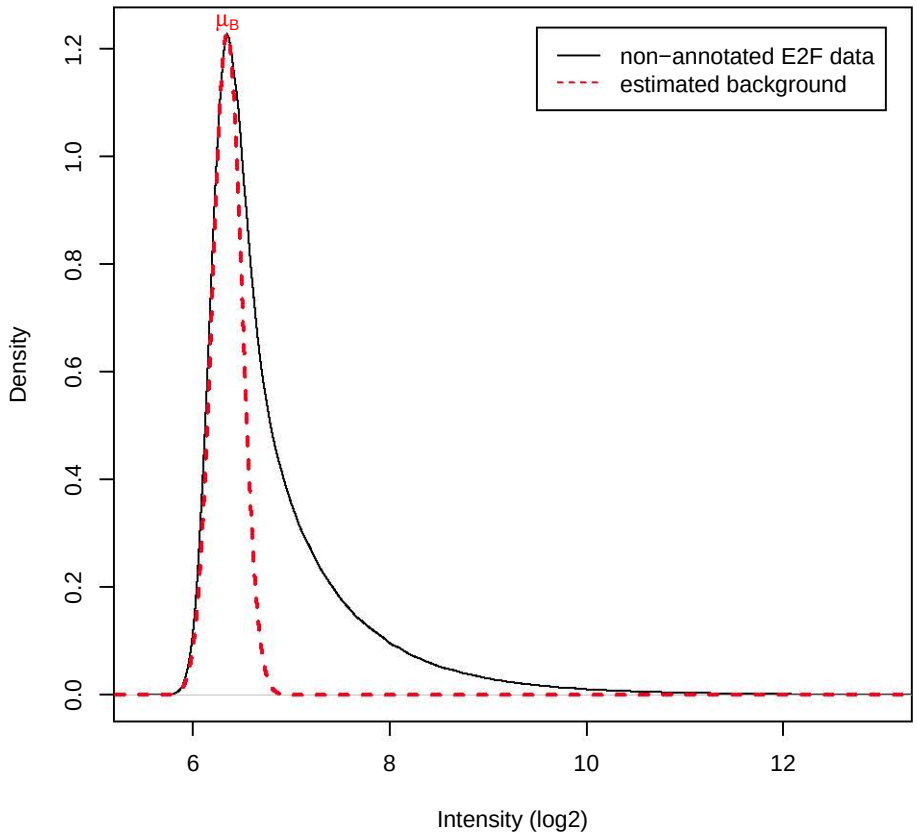


Figure 3.1: Nonparametric density estimate of intensities in the E2F study that do not overlap with any annotated features. The normal distribution is fitted to the left component of this mixture distribution. Based on this fit the values for μ_B and σ_B of the background noise distribution are estimated.

$1/15, 1/15, 2/15, 2/15, 2/15, 2/15, 2/15$), allowing for a realistic distribution of the simulated data. The lower values were set at lower weights so that the distribution of the simulated data corresponds better with the empirical distribution of the raw data in annotated regions. Six different FC levels are used $(0.95, 1.5, 2, 3, 4, 5)$ and each of these are assigned to 100 genes.

The marginal distribution of the simulated and the real data for the annotated regions are displayed in Figure 3.2. The distribution of the simulated data corresponds well with the empirical distribution of the raw data. Figure 3.3 demonstrates that the simulated data also possess a similar serial correlation structure as the E2F data. An example of a genomic region with simulated data is given in Figure 3.4. The top panel shows data from the simulation run, while in the bottom panel real data from the E2F experiment is displayed for comparison. The plot shows that the characteristics of the observed data are realistically preserved in the simulated data.

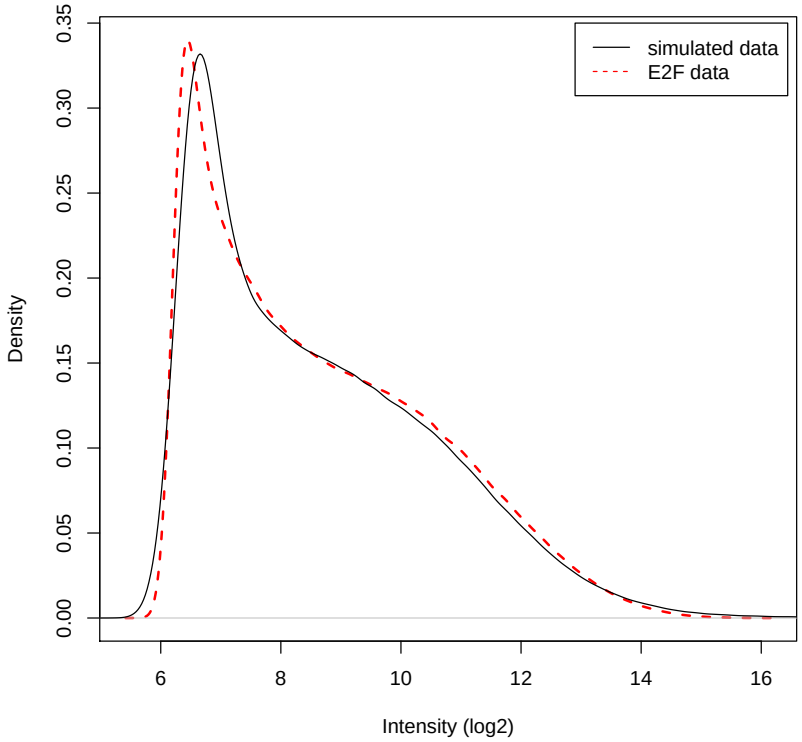


Figure 3.2: Nonparamteric density estimate of $\log_2$-transformed raw probe intensities in annotated regions for E2F data and simulated data

Figure 3.5 provides more insight into the simulation model. In the left panel multiple nonparametric density estimates are displayed for simulated data of one particular exon on chromosome 1: (1) distribution when the exon is not expressed (background + noise): $\log_2 [B(t)] + \epsilon_i(t)$, where $I_i(t) = 0$; (2) hypothetical distribution of an expressed exon when the probe effect is negligible (background + signal + noise): $\log_2 \left[B(t) + 2^{c_i(t)}\right] + \epsilon_i(t)$, with $I_i(t) = 1$, $\mu_{c,i}(t) = 7$ and $\sigma_p = 0$; (3) distribution of a simulated expressed exon with probe effect (background + signal + probe + noise): $\log_2 \left[B(t) + 2^{c_i(t)+p(t)}\right] + \epsilon_i(t)$, using $I_i(t) = 1$, $\mu_{c,i}(t) = 7$ and $\sigma_p = 1.5$. For comparison reasons the same values are used for the background $B(t)$, mean exonic signal $c_i(t)$ and noise $\epsilon_i(t)$. The signal of the expressed exon $c_i(t)$ shifts the background distribution towards higher values and sharpens the distribution. The additional probe effect clearly introduces a huge variability among the probe intensities, which target the same exon. In the lower panel of Figure 3.5, the distributions of the simulated probe intensities for the same exon on different arrays are given ($I_i(t) = 1$, $\sigma_p = 1.5$, $\mu_{c,i}(t) = 7$ for WT and $\mu_{c,i}(t) = 9$ for E2F). The difference in mean exon effect $c_i(t)$ on each array affects both the shape and the location of the distribution.

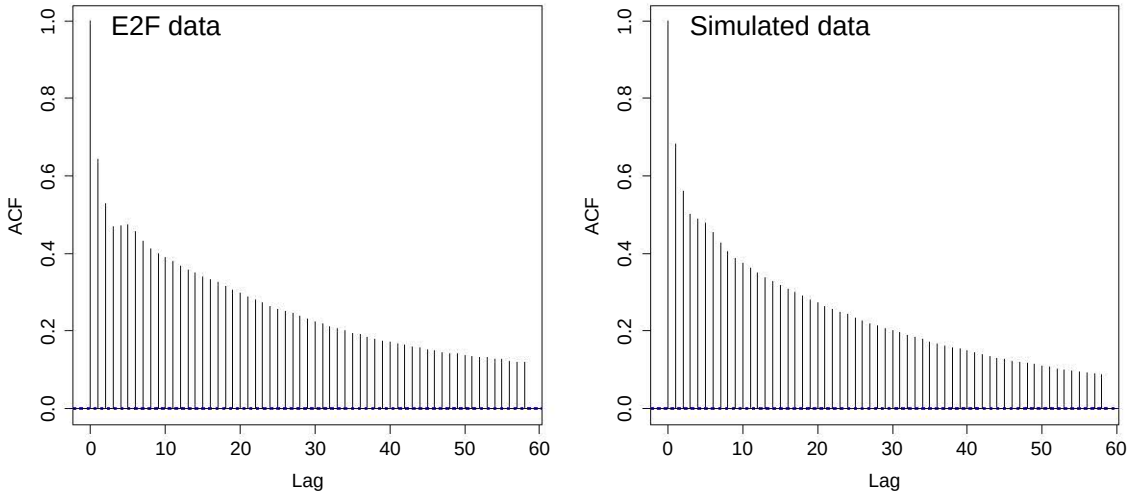


Figure 3.3: Autocorrelation plots of $\log_2$-transformed raw probe intensities for E2F data (left) and simulated data (right)

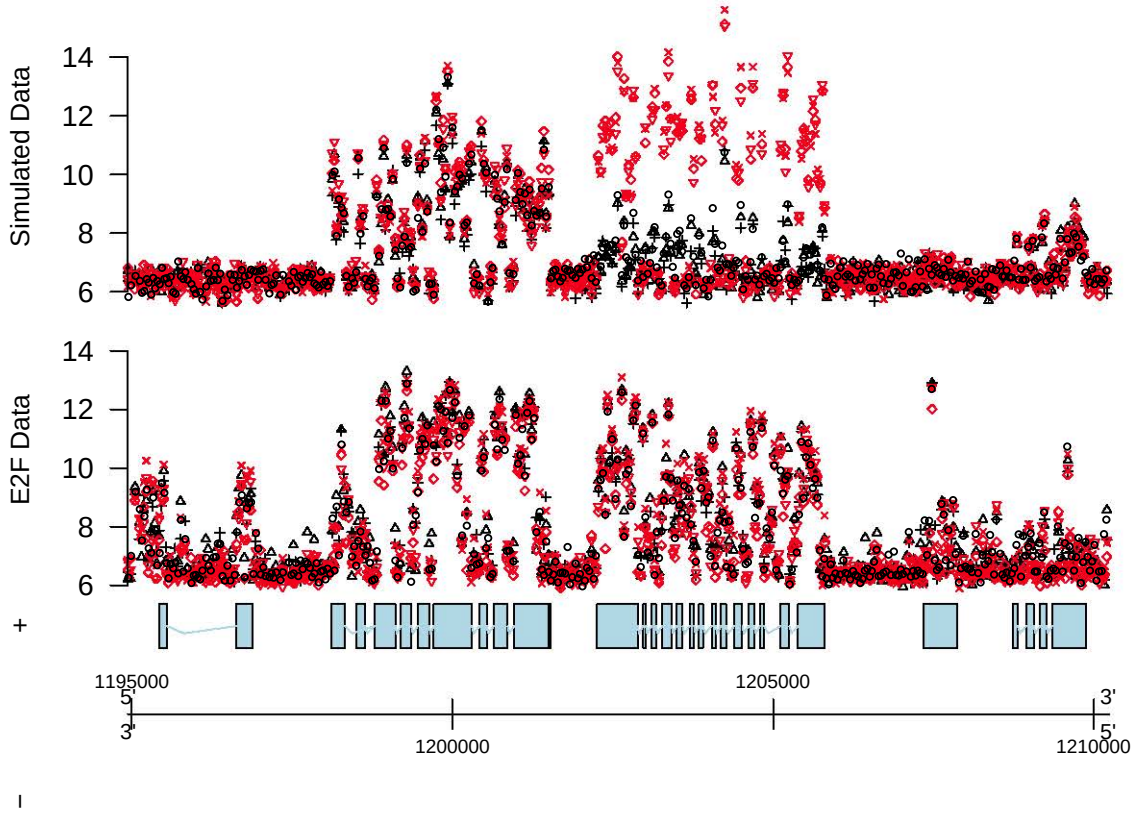


Figure 3.4: Along-chromosome plots of a genomic region with simulated data (upper panel) and real data from the E2F experiment (lower panel). The intensities for the WT plants are indicated in black, those for the E2F plants in red.

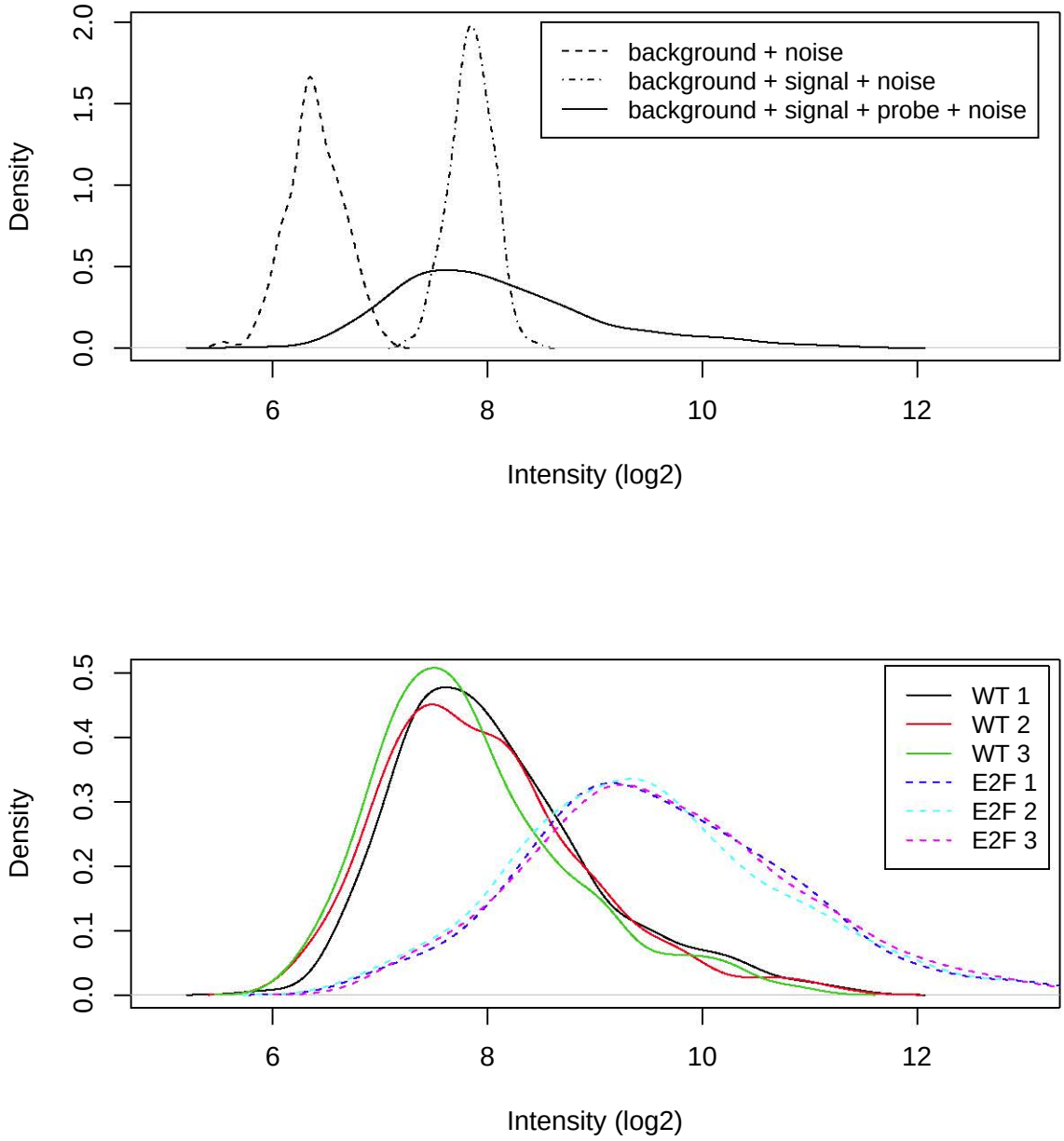


Figure 3.5: Nonparametric density estimates for simulated data of one particular exon on chromosome 1. The upper panel displays the distributions of $\log_2$-transformed raw probe intensities (1) when the exon is not expressed, (2) when the probe effect is negligible, (3) for an expressed exon with probe effect. The lower panel shows the distributions of the simulated probe intensities for the same exon on 6 different arrays for WT and E2F plants.

3.4.1.2 Results simulation study

In the first part of the simulation study the two approaches of the wavelet-based model are assessed, (1) WavMix: method based on the mixture prior fitting criterion and approximate empirical Bayes inference using Johnson curves, and (2) WavNorm: method based on the normal prior fitting criterion and empirical Bayes inference. In both approaches the MAD-based variance estimator is used. In this study the local BFDR is controlled at the 5% level. Differentially expressed transcripts are assumed to become interesting above a fold change $\delta_{FC} = 2$ (or $\log_2 \delta_{FC} = 1$). The threshold for transcript discovery is set at the 90 percentile of $\log_2$-transformed, background-corrected and quantile-normalized simulated intensities of the non-annotated regions in the genome. In a post-processing step, we only maintain regions consisting of at least two consecutive probes. The performance for transcript discovery is compared with two widely used methods: (3) the method of Kampa et al. (2004), and (4) the method of Huber et al. (2006). For differential expression (5) an RMA combined with moderated t-test (henceforth simply referred to as RMA) is used as a benchmark. These methods have been reviewed in Chapter 2.

In Table 3.1 the methods are compared in terms of positive predictive value (PPV), sensitivity or true positive rate (TPR) and specificity (SPC). These quantities are defined as

$$\text{PPV} = \frac{\text{number of true positives}}{\text{number of true positives} + \text{number of false positives}} \tag{3.49}$$

$$\text{TPR} = \frac{\text{number of true positives}}{\text{number of true positives} + \text{number of false negatives}} \tag{3.50}$$

$$\text{SPC} = \frac{\text{number of true negatives}}{\text{number of false positives} + \text{number of true negatives}}. \tag{3.51}$$

Note that SPC corresponds with $1 -$ false positive rate (FPR). In Table 3.1 also the computation time of the different approaches is assessed as measured on a 2×6 Core Intel® Xeon® X7460, 2.66 GHzProcessors GNU/Linux server system with 128 GB RAM. The PPV, TPR and SPC are calculated on probe-level, except for RMA which acts on gene-level. For comparison we also tabulate the TPR for the wavelet-based functional models on gene-level: a gene is called differentially expressed if it contains probes that are flagged as differentially expressed by the wavelet-based methods.

Table 3.1: Comparison of the performance of the wavelet-based methods with the Kampa, Huber and RMA method in terms of positive predictive value (PPV), sensitivity or true positive rate (TPR), specificity (SPC) and computational time per chromosome. PPV, TPR and SPC are given at probe level except for RMA for which they are calculated on gene level. For comparison the TPR of the wavelet-based methods for differential expression are given both on probe level and on gene level. Note that the RMA method heavily relies on the existing annotation while the other methods are unbiased towards existing annotation.

Model	Biased by	Transcript Discovery			Differential Expression				Time
	annotation	PPV	TPR	SPC	PPV	TPR		SPC	(s/chr)
						probe	gene		
WavNorm	no	99.6	80.0	99.8	99.3	75.0	97.1	100	14
WavMix	no	99.8	76.2	99.9	95.8	69.0	90.8	99.9	1049
Kampa	no	85.0	82.4	90.5	-	-	-	-	27
Huber	no	85.3	76.7	91.5	-	-	-	-	6321
RMA	yes	-	-	-	100	-	83.0	100	32

The wavelet-based methods outperform the Kampa method and the Huber method for transcript discovery. They have very large values for PPV and SPC while maintaining a large TPR. The WavMix method suffers from a slight loss in TPR compared to the WavNorm method, although the PPV and SPC is marginally larger at the chosen threshold value. For differential expression, the wavelet-based methods seem to be more sensitive than RMA, while still controlling well for false positives. Among the wavelet-based methods the PPV and TPR for WavMix are clearly smaller than for WavNorm. WavNorm is also very competitive in terms of computation time, while the numerical optimization of the hyperparameters and the calculation of the Johnson curves results in a much larger computation time for WavMix.

Receiver Operating Characteristic (ROC) curves for transcript discovery and differential expression are given in Figure 3.6. The TPR is plotted against the FPR. An optimal test would detect all true positives without any false positives. The two wavelet-based approaches have a very similar performance for transcript discovery and clearly outperform the Huber and Kampa

methods. For differential expression the WavNorm method performs slightly better than the WavMix method.

Figures 3.7 and 3.8 give the segmentation for transcript discovery and differential expression, respectively, of the simulated example region shown in Figure 3.4. For clarity of exposition the genomic region depicted in Figure 3.8 is made a little bit smaller. The top panels display the simulated background-corrected quantile-normalized data along with the regions that are truly expressed or differentially expressed. The bottom panels show the model tracks for the assessed methods. The regions that are discovered by the wavelet-based methods correspond very well with the underlying exonic structure. The Kampa method also mimics the exonic structure, while the method of Huber can not distinguish between intronic and exonic regions. An example of the larger sensitivity of the WavNorm method compared to the WavMix method can be seen from Figure 3.7. The WavMix method does not discover a transcript for the fourth gene in the region, while the WavNorm method does. Furthermore, it finds only 2 of the 4 exons in the fifth transcribed gene, while the WavNorm method discovers 3 transcribed exons. Similar to what was observed in Figure 2.9 the model track of the differential expression in Figure 3.8 is smoother for the WavMix method than for the WavNorm method, i.e. the wavelet coefficients are more frequently thresholded when the mixture prior is imposed upon them.

Since the WavNorm approach provides better results than the WavMix method based on the simulation study and is also clearly faster, we will further focus on the former approach in what follows. In the second part of the simulation study we look more closely at the performance of five different versions of the WavNorm method, as discussed in Section 3.2.2, (1) WavNorm(MAD): based on the MAD estimator for the standard deviation of the errors, which is the same WavNorm method as used in the first part of the simulation study, (2) WavNorm(j): method using variance estimator (3.39), (3) WavNorm(jk): using variance estimator (3.38), (4) WavNorm(b): WavNorm with bootstrap correction and (5) WavNorm(imp): method using an improper prior on the random effect variances $\tau_m(j,k)$ and variance estimator (3.39). The same threshold values for transcript discovery and differential expression as before are used. The results in terms of PPV, TPR, SPC and computation time are given in Table 3.2. They indicate that the performances of the WavNorm methods are very comparable. WavNorm(MAD) and WavNorm(imp) seem to suffer slightly from a decrease in TPR. As expected, WavNorm(b), with bootstrap correction, is computationally more demanding than the other WavNorm methods.

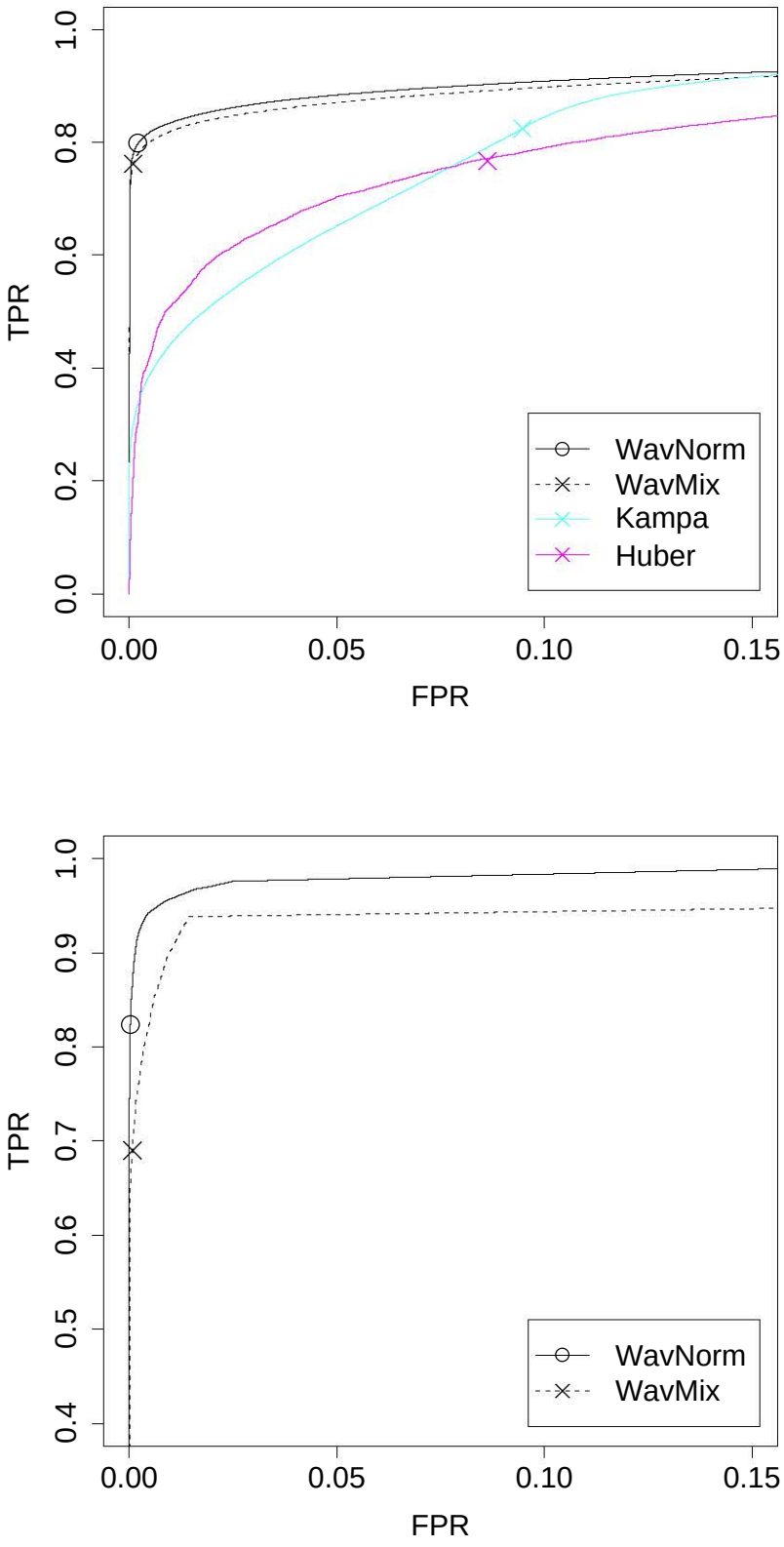


Figure 3.6: ROC curves for transcript discovery (upper) and differential expression (lower). The symbols indicate the results corresponding to Table 3.1

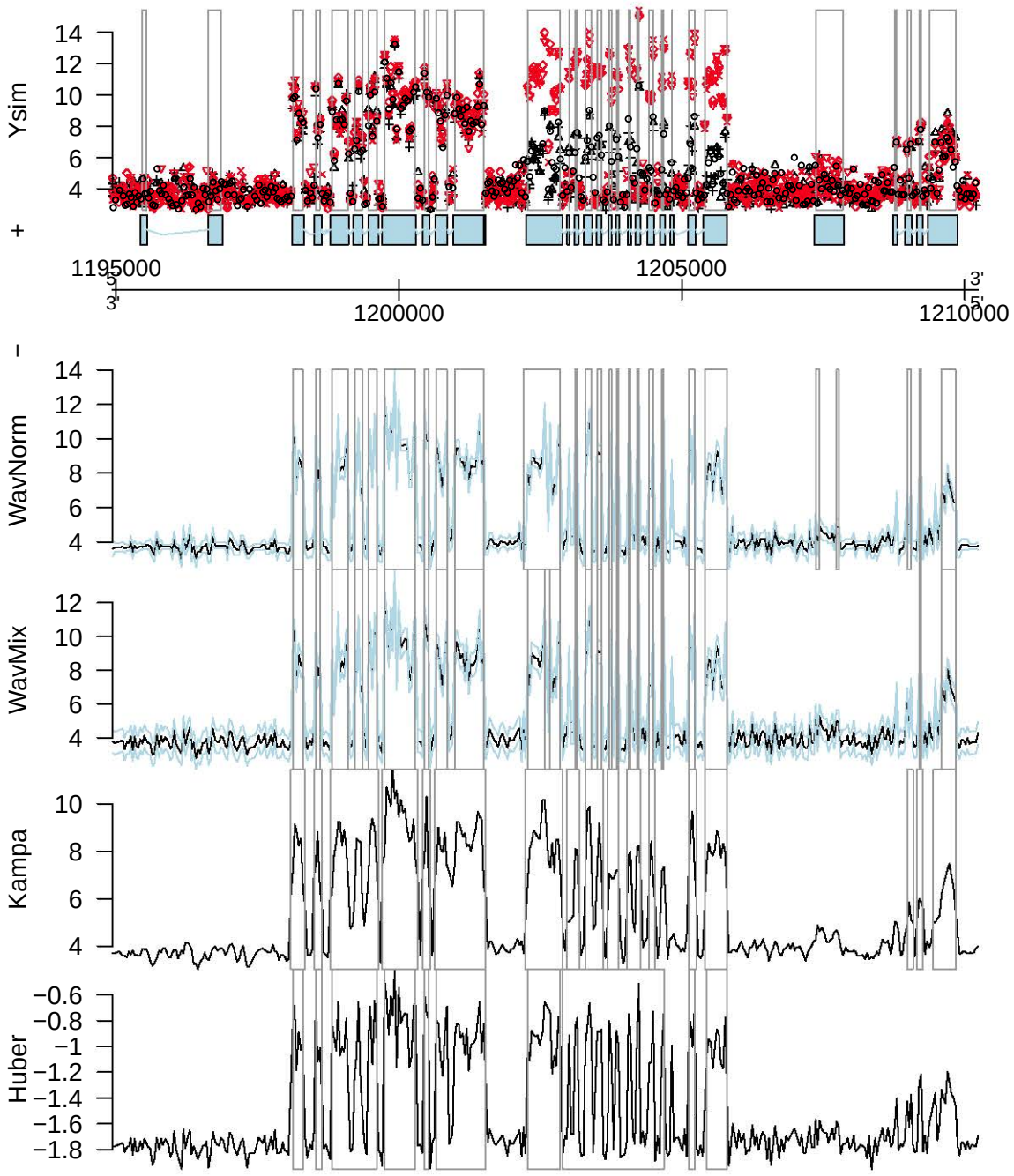


Figure 3.7: Along-chromosome plot of a simulated region with model tracks for transcript discovery. Top panel: background-corrected and quantile-normalized simulated array intensities of WT plants (black) and E2F plants (red); boxes indicate the annotation of the expressed exons. The 3 different replicates for WT and E2F are indicated by $\circ$, $+$ and $\triangle$. Bottom panels: Model tracks with grey boxes indicating the discovered expressed transcripts for the four assessed methods.

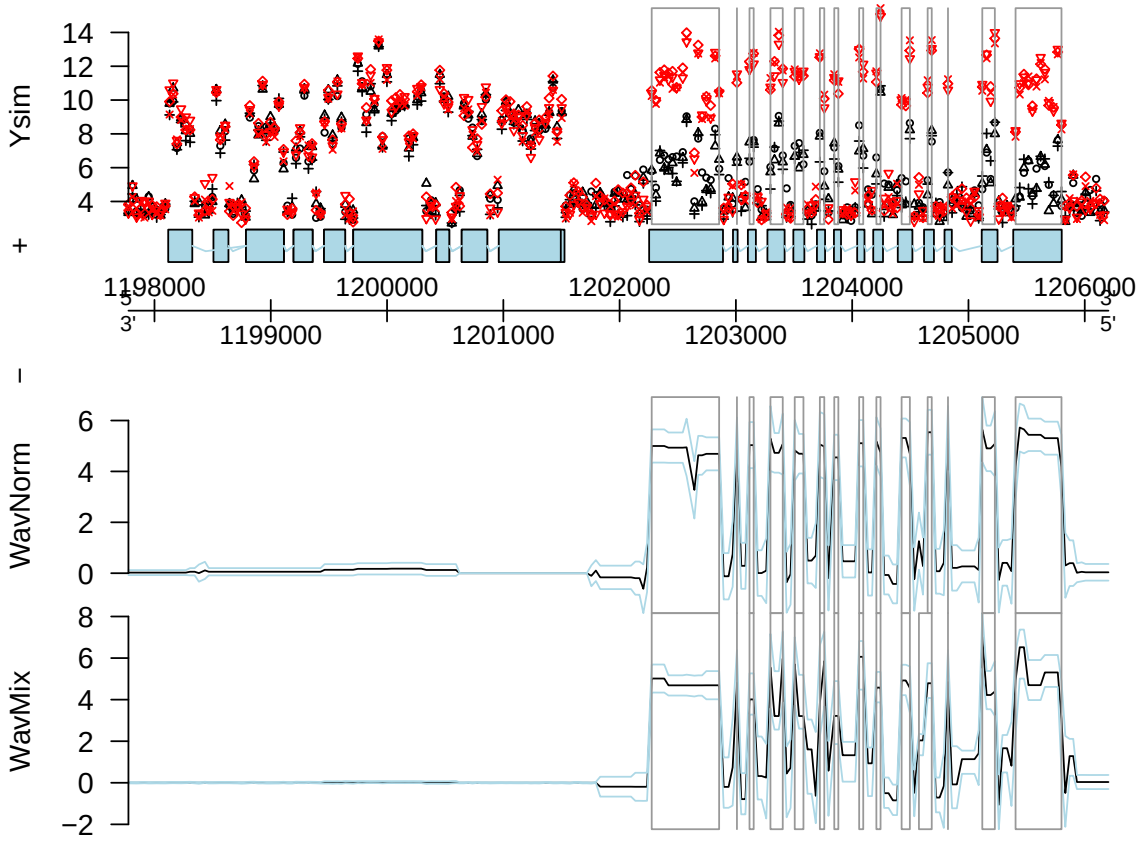


Figure 3.8: Along-chromosome plot of a simulated region with model tracks for differential expression. Top panel: background-corrected and quantile-normalized simulated array intensities of WT plants (black) and E2F plants (red); boxes indicate the annotation of the differentially expressed exons. The 3 different replicates for WT and E2F are indicated by $\circ$, $+$ and $\triangle$. Bottom panels: Model tracks with grey boxes indicating the discovered differentially expressed transcripts for the two wavelet-based methods.

The five approaches are further compared by means of ROC curves for transcript discovery and differential expression, presented in Figure 3.9. This figure confirms the similar performances for transcript discovery. For differential expression WavNorm(MAD) and WavNorm(imp) seem to provide superior results. In practice, however, they are less sensitive than the other WavNorm approaches when using a low FDR, e.g. the FDR of 0.05 used in Table 3.2. Based on Table 3.2 we prefer the WavNorm(j) method for the case study.

An example region of the simulated data for differential expression is displayed in Figure 3.10. The effect of the bootstrap correction on the variance can be seen in the model tracks for the fold change. The credible intervals in non-differentially expressed regions are much wider. The correction has a minor effect in regions with signal. Hence, the variance is mainly underestimated in regions that are not of interest. This explains why the WavNorm methods without

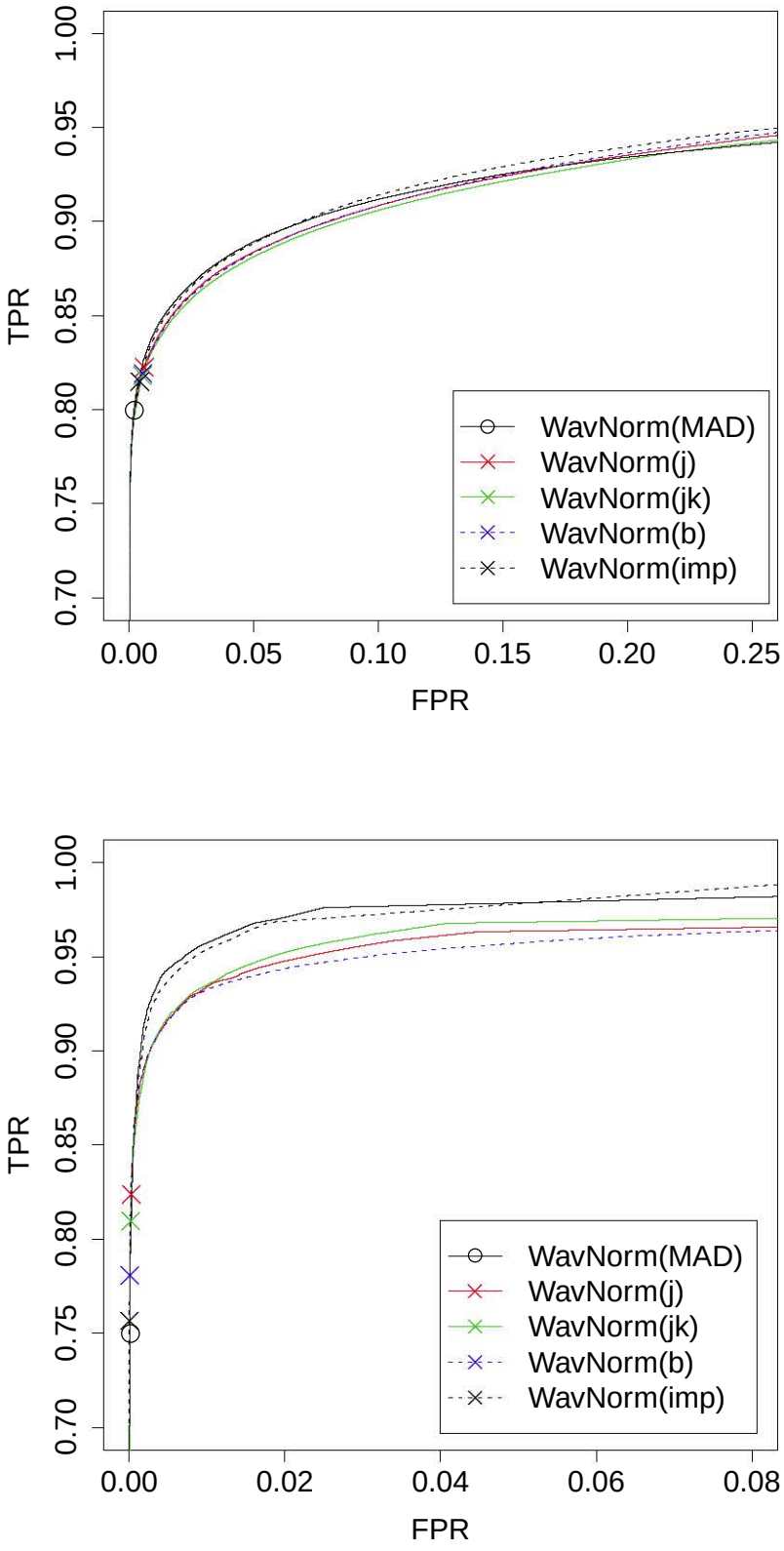


Figure 3.9: ROC curves for transcript discovery (upper) and differential expression (lower) by the different WavNorm methods. The symbols indicate the results corresponding to Table 3.2.

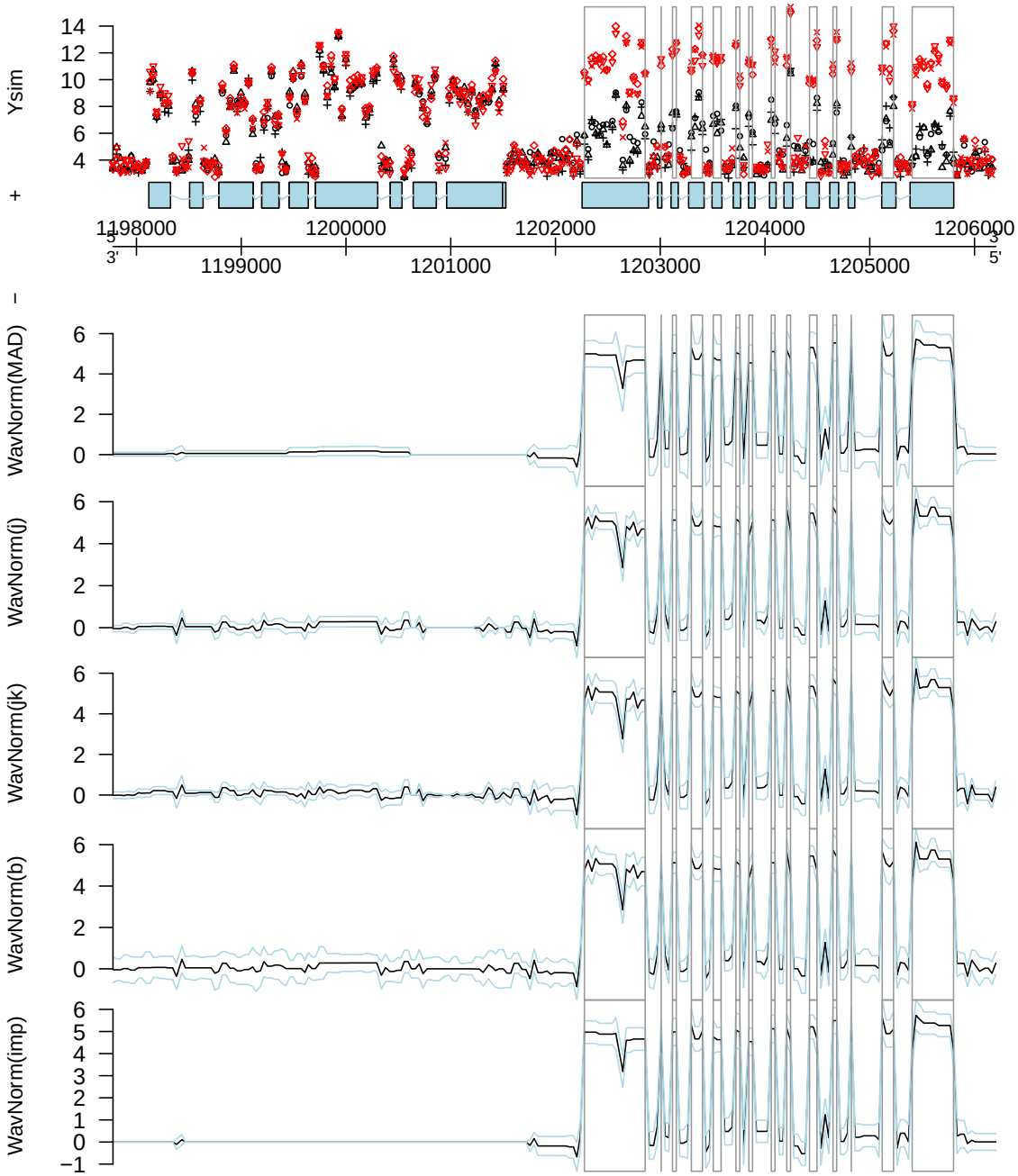


Figure 3.10: Along-chromosome plot of a simulated region with model tracks for differential expression. Top panel: background-corrected and quantile-normalized simulated array intensities of WT plants (black) and E2F plants (red); boxes indicate the annotation of the differentially expressed exons. The 3 different replicates for WT and E2F are indicated by $\circ$, $+$ and $\triangle$. Bottom panels: Model tracks with grey boxes indicating the discovered differentially expressed transcripts for the five different WavNorm methods.

Table 3.2: Comparison of the performance of five different versions of the WavNorm method in terms of positive predictive value (PPV), sensitivity or true positive rate (TPR), specificity (SPC) and computational time per chromosome.

Model	Transcript Discovery			Differential Expression				Time
	PPV	TPR	SPC	PPV	TPR		SPC	(s/chr)
					probe	gene		
WavNorm(MAD)	99.6	80.0	99.8	99.3	75.0	97.1	100	14
WavNorm(j)	98.9	82.3	99.4	98.8	82.4	98.1	100	34
WavNorm(jk)	99.0	81.8	99.5	99.0	81.0	97.8	100	35
WavNorm(b)	99.0	81.9	99.4	99.5	78.1	97.6	100	1023
WavNorm(imp)	99.2	81.5	99.6	99.9	75.7	96.9	100	34

bootstrap correction in regions of interest combined with the drastic increase in computation time indicate that the bootstrap correction is not worth the effort in our application. The introduction of the improper prior on the smoothing parameters $\tau_m(j,k)$ clearly imposes additional regularization. Hence, if more smooth estimates for the fold change are preferred, one can adopt the WavNorm(imp) method.

3.4.2 Case study: the *Arabidopsis thaliana* E2F tiling experiment

The tiling data are obtained by hybridizing $N_1 = 3$ arrays for the E2F plants and $N_2 = 3$ for the WT plants. We remapped the PM probes to the *Arabidopsis thaliana* genome annotation TAIR9 (TAIR resources can be found at `http://www.arabidopsis.org/`) (Swarbreck et al., 2008). The design matrix $\boldsymbol{X}$ in (3.2) equals

$$\boldsymbol{X} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -1 & -1 & -1 \end{pmatrix}^T .$$

For large data sets, the DWT is typically stopped at a certain level J. Similar to Morris et al. (2008), we perform the DWT down to $J = 10$. Hence, the father wavelet spans a region of around 60 kb (or 60000 nucleotides), which is much larger than a typical gene of *Arabidopsis thaliana*. Note that the average gene size in *Arabidopsis thaliana* is about 1.9 kb and large genes

are defined as genes with a length > 3 kb (e.g. Meinke et al., 2003).

Early microarray publications inferred on differential expression by only considering the fold change, with $FC = 2$ typically considered a worthwhile cut-off (e.g. DeRisi et al., 1996; Schena et al., 1996). However, fold change cut-offs do not account for variability nor guarantee reproducibility (McCarthy and Smyth, 2009). Most statistical methods for assessing differential expression, on the other hand, allow for genes with small fold changes to be considered statistically significant. Hence, they report significant genes that are not biologically relevant. In our proposed method, FDR procedures for both transcript discovery and the detection of differential expression can rely on a threshold value that is driven by the biological problem at hand. This eventually leads to results that are both statistically significant and biologically relevant. Similar to the early microarray studies, we consider a fold change between the E2F and WT arrays to be relevant as soon as it exceeds $\delta_{FC} = 2$ or $\log_2(\delta_{FC}) = 1$. If one wants to avoid the use of an arbitrary threshold for the fold change, one can still use $\log_2(\delta_{FC}) = 0$. With this threshold all regions are recovered that exhibit a $\log_2$ fold change that is significantly different from 0. Similar to most existing methods in tiling array literature for transcript discovery (e.g. David et al., 2006; Kampa et al., 2004), we can not avoid the use of a threshold for transcript discovery. This threshold value is obtained by applying the method described in David et al. (2006).

Following David et al. (2006), we consider the distribution of $\log_2$-transformed background-corrected intensities for probes that do not overlap with any annotated features (see Figure 3.11, black dashed curve). This distribution is again a mixture between a normal distribution and some other distribution. If the normal distribution component (solid red line in Figure 3.11) is assumed to be the null distribution of probes that measure background intensities, then we can derive the BFDR for the background probes. We select the background threshold δ_{TD} that corresponds to a BFDR of 0.1%. This leads to a threshold for the mean function $\beta_1(t)$ of $\delta_{TD} = 3.7$. The biologists involved in this study, however, proposed to set the threshold of the mean function $\beta_1(t)$ at $\delta_{TD} = 4.5$, based on former experiences. This approximately corresponds to the median of the $\log_2$-transformed, background-corrected and quantile-normalized intensities, as well as to the location of the minimum between the two modes of the distribution of these intensities in annotated regions (see Figure 3.11). Note that with the latter δ_{TD} slightly more conservative results are obtained. The local BFDR is controlled at 5%. In a postprocessing step,

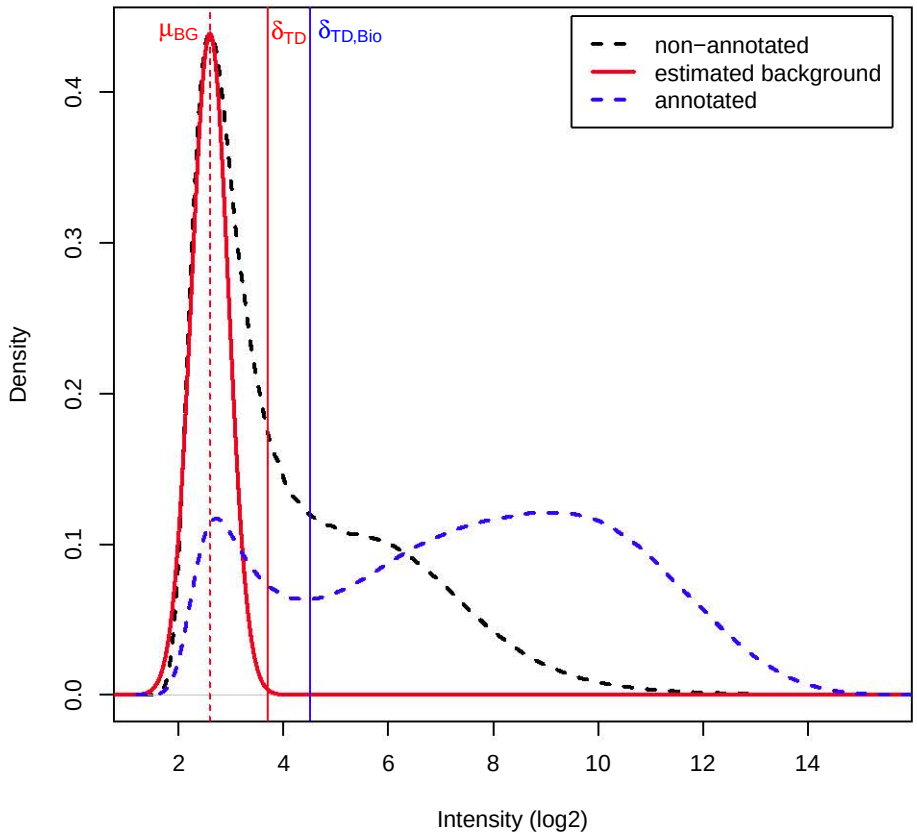


Figure 3.11: Nonparametric density estimates of the $\log_2$-transformed background-corrected and quantile-normalized intensities for non-annotated (black dashed curve) and annotated probes (blue dashed curve), and estimated distribution for background probes (solid red line). The red and blue vertical lines represent the threshold for transcript discovery (δ_{TD}) derived from the background distribution and the one proposed by the biologists that participated in the study ($\delta_{TD,Bio}$), respectively.

we only maintain regions consisting of at least two consecutive probes.

When applying the WavNorm(j) method, we find 77663 transcribed regions and 3885 differentially expressed regions. Of these discovered TARs and differentially expressed TARs, 15149 and 765 do not overlap with existing annotation, respectively. They can be considered as potential discoveries that have to be biologically validated. A more detailed overview of the results for each chromosome is given in Table 3.3.

Figure 3.12 shows the same example genomic region of chromosome 1 as shown in Figure 2.3. The top panel consists of the $\log_2$-transformed background-corrected and quantile-normalized E2F and WT intensities. In the middle panel the genomic coordinate and the annotation are dis-

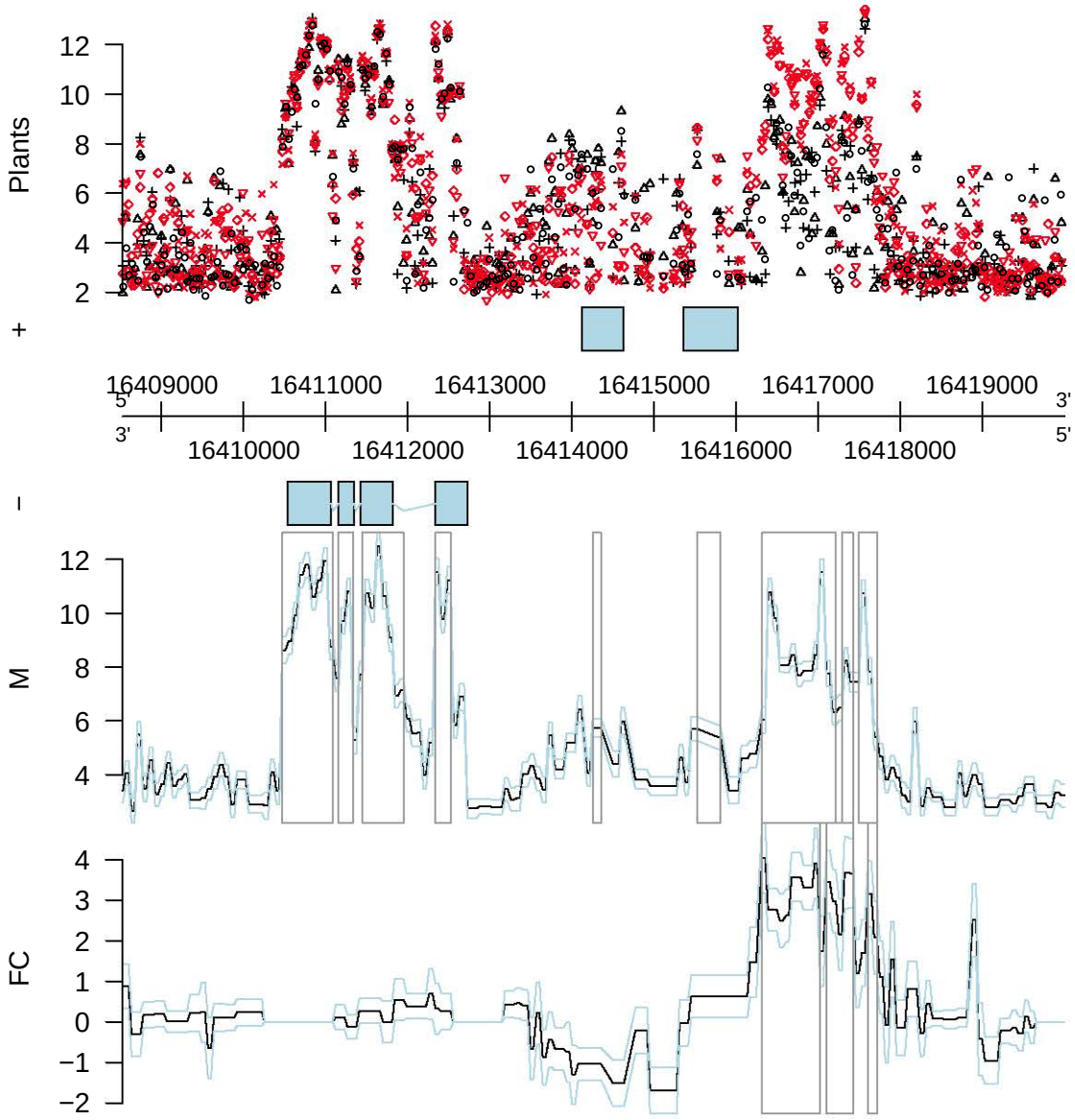


Figure 3.12: Along-chromosome plot of an unannotated region, which seems to be upregulated in E2F plants. Array intensities of E2F plants (red) and wild type plants (black) are given in the top panel and the 3 different replicates for WT and E2F are indicated by $\circ$, $+$ and $\triangle$. The bottom panels depict the mean model track $\beta_1(t)$ (M track) and a track for the $\log_2$ fold change $FC(t) = 2 \times \beta_2(t)$ (FC track). 95% credible intervals are indicated with light blue lines, while discovered transcripts and differentially expressed regions are indicated by grey boxes on the M and FC track, respectively.

Table 3.3: Transcript discovery and differential expression for all chromosomes of *Arabidopsis thaliana* in the E2F experiment. The non-annotated regions represent those regions among the detected regions that do not overlap with existing annotation.

	Transcript Discovery		Differential Expression	
Chromosome	Detected	Non-annotated	Detected	Non-annotated
1	20420	3664	975	204
2	12050	2801	576	86
3	15105	2958	727	178
4	12017	2382	626	109
5	18071	3344	981	188
$1-5$	77663	15149	3885	765

played. The bottom panel shows the posterior means (black lines) of $\beta_1(t)$ and $\log_2$-transformed fold change $FC(t) = 2 \times \beta_2(t)$ along with 95% credible intervals (light blue lines). Discovered transcripts and differentially expressed regions are indicated by grey boxes on the mean (M) and fold change (FC) track, respectively. An annotated gene is transcribed to the same level in both strains and a novel transcript is discovered, which is upregulated in E2F plants. The discovered region spans approximately 1500 nucleotides and seems to have an exonic structure. It is an interesting region for further biological validation.

3.5 Conclusion

In this chapter we have introduced a wavelet-based functional model for transcriptome analysis with tiling arrays. In contrast to other methods for the analysis of tiling arrays, it can assess transcript discovery and identify differentially expressed transcripts simultaneously. It is as powerful in detecting existing as well as novel (differentially) expressed transcripts. The wavelet-based functional model thus exploits tiling array data to their full potential. Hence, it can be seen as a valuable tool to assist biological researchers in further unraveling transcriptional networks. In a simulation study we have shown that the proposed method is superior to competing methods. When the normal prior distribution is imposed on the wavelet coefficients, the method also performs very good in terms of numerical speed. Finally, its use for finding

potential targets or biomarkers without being biased by the existing annotation has been demonstrated on an example data set. While we have focused on the two-group design in this chapter, possible extensions of the model for more complex designs will be discussed in Chapter 4.

Appendix A: Fitting procedure I (WavMix) - Derivation of the posterior density of the functional effects in the wavelet space

In the derivation we will suppress the indices for notational convenience, i.e. $\boldsymbol{D}(j,k) = \boldsymbol{D}$, $\boldsymbol{\beta}^*(j,k) = \boldsymbol{\beta}^*$, $\tau_m(j) = \tau_m$, $\pi_m(j) = \pi_m$. The joint density $f\left\{\boldsymbol{D}, \boldsymbol{\beta}^* | \tau_m, \sigma_\epsilon^2\right\}$ according to the Model (3.4) - (3.5) is given by

$$\begin{aligned} f\left\{\boldsymbol{D}, \boldsymbol{\beta}^* | \tau_m, \sigma_\epsilon^2\right\} &= (1-\pi_1-\pi_2)\, MVN\left(\boldsymbol{X}\boldsymbol{\beta}^*, \boldsymbol{I}\sigma_\epsilon^2\right)\, \delta_0(\beta_1^*, \beta_2^*) \\ &\quad + \pi_1\, MVN\left(\boldsymbol{X}\boldsymbol{\beta}^*, \boldsymbol{I}\sigma_\epsilon^2\right)\, N(0, \tau_1\sigma_\epsilon^2)\, \delta_0(\beta_2^*) \\ &\quad + \pi_2\, MVN\left(\boldsymbol{X}\boldsymbol{\beta}^*, \boldsymbol{I}\sigma_\epsilon^2\right)\, N(0, \tau_1\sigma_\epsilon^2)\, N(0, \tau_2\sigma_\epsilon^2). \end{aligned}$$

The marginal density is obtained by integrating out the functional effects from the joint density, resulting in (e.g. Verbeke and Molenberghs, 2000)

$$\begin{aligned} f\left\{\boldsymbol{D} | \tau_m, \sigma_\epsilon^2\right\} &= \int_{\beta_1^*} \int_{\beta_2^*} f\left\{\boldsymbol{D}, \boldsymbol{\beta}^* | \tau_m, \sigma_\epsilon^2\right\} d\beta_2^* d\beta_1^* \\ &= (1-\pi_1-\pi_2)\, MVN\left(\boldsymbol{0}, \boldsymbol{I}\sigma_\epsilon^2\right) \\ &\quad + \pi_1\, MVN\left(\boldsymbol{0}, \boldsymbol{I}\sigma_\epsilon^2 + \boldsymbol{X}_1\boldsymbol{X}_1^T\tau_1\sigma_\epsilon^2\right) \\ &\quad + \pi_2\, MVN\left(\boldsymbol{0}, \boldsymbol{I}\sigma_\epsilon^2 + \boldsymbol{X}_2\boldsymbol{X}_2^T\tau_2\sigma_\epsilon^2\right). \end{aligned}$$

The posterior density becomes

$$
\begin{aligned}
f\left\{\boldsymbol{\beta}^*|\boldsymbol{D},\tau_m,\sigma_\epsilon^2\right\} &= \frac{f\left\{\boldsymbol{D},\boldsymbol{\beta}^*|\tau_m,\sigma_\epsilon^2\right\}}{f\left\{\boldsymbol{D}|\tau_m,\sigma_\epsilon^2\right\}} \\
&\Updownarrow \\
f\left\{\boldsymbol{\beta}^*|\boldsymbol{D},\tau_m,\sigma_\epsilon^2\right\} &= \frac{(1-\pi_1-\pi_2)\,MVN\left(\boldsymbol{X}\boldsymbol{\beta}^*,\boldsymbol{I}\sigma_\epsilon^2\right)\,\delta_0(\beta_1^*,\beta_2^*)}{f\left\{\boldsymbol{D}|\tau_m,\sigma_\epsilon^2\right\}} \\
&\quad + \frac{\pi_1\,MVN\left(\boldsymbol{X}\boldsymbol{\beta}^*,\boldsymbol{I}\sigma_\epsilon^2\right)\,N(0,\tau_1\sigma_\epsilon^2)\,\delta_0(\beta_2^*)}{f\left\{\boldsymbol{D}|\tau_m,\sigma_\epsilon^2\right\}} \\
&\quad + \frac{\pi_2\,MVN\left(\boldsymbol{X}\boldsymbol{\beta}^*,\boldsymbol{I}\sigma_\epsilon^2\right)\,N(0,\tau_1\sigma_\epsilon^2)\,N(0,\tau_2\sigma_\epsilon^2)}{f\left\{\boldsymbol{D}|\tau_m,\sigma_\epsilon^2\right\}} \\
&\Updownarrow \quad \text{using Bayes' rule} \\
f\left\{\boldsymbol{\beta}^*|\boldsymbol{D},\tau_m,\sigma_\epsilon^2\right\} &= \frac{(1-\pi_1-\pi_2)\,MVN\left(\boldsymbol{0},\boldsymbol{I}\sigma_\epsilon^2\right)\,\delta_0(\beta_1^*,\beta_2^*)}{f\left\{\boldsymbol{D}|\tau_m,\sigma_\epsilon^2\right\}} \\
&\quad + \frac{\pi_1\,MVN\left(\boldsymbol{0},\boldsymbol{V}_1\sigma_\epsilon^2\right)\,N(\hat{\beta}_1,\sigma_{\hat{\beta}_1}^2)\,\delta_0(\beta_2^*)}{f\left\{\boldsymbol{D}|\tau_m,\sigma_\epsilon^2\right\}} \\
&\quad + \frac{\pi_2\,MVN\left(\boldsymbol{0},\boldsymbol{V}_2\sigma_\epsilon^2\right)\,N(\hat{\beta}_1,\sigma_{\hat{\beta}_1}^2)\,N(\hat{\beta}_2,\sigma_{\hat{\beta}_2}^2)}{f\left\{\boldsymbol{D}|\tau_m,\sigma_\epsilon^2\right\}},
\end{aligned}
$$

with

$$
\begin{aligned}
\boldsymbol{V}_1 &= \boldsymbol{I}+\boldsymbol{X}_1^T\boldsymbol{X}_1\tau_1, \\
\boldsymbol{V}_2 &= \boldsymbol{I}+\boldsymbol{X}_1^T\boldsymbol{X}_1\tau_1+\boldsymbol{X}_2^T\boldsymbol{X}_2\tau_2, \\
\hat{\beta}_1 &= \left\{\boldsymbol{X}_1^T\boldsymbol{X}_1+1/\tau_1\right\}^{-1}\boldsymbol{X}_1^T\boldsymbol{D}, \\
\hat{\beta}_2 &= \left\{\boldsymbol{X}_2^T\boldsymbol{X}_2+1/\tau_2\right\}^{-1}\boldsymbol{X}_2^T\boldsymbol{D}, \\
\sigma_{\hat{\beta}_1}^2 &= \sigma_\epsilon^2\left\{\boldsymbol{X}_1^T\boldsymbol{X}_1+1/\tau_1\right\}^{-1}, \\
\sigma_{\hat{\beta}_2}^2 &= \sigma_\epsilon^2\left\{\boldsymbol{X}_2^T\boldsymbol{X}_2+1/\tau_2\right\}^{-1}.
\end{aligned}
$$

Let the posterior probabilities of non-zero wavelet coefficients at level j for the effect function parameters be denoted by

$$
\omega_1 = \frac{\pi_1 g_1\left(\boldsymbol{D}\right)}{f\left(\boldsymbol{D}\right)},
$$

$$
\omega_2 = \frac{\pi_2 g_2\left(\boldsymbol{D}\right)}{f\left(\boldsymbol{D}\right)}.
$$

with

$$g_1(\boldsymbol{D}) = MVN\left(\mathbf{0}, \boldsymbol{V}_1\sigma_\epsilon^2\right),$$

$$g_2(\boldsymbol{D}) = MVN\left(\mathbf{0}, \boldsymbol{V}_2\sigma_\epsilon^2\right).$$

The posterior density is then given by

$$\begin{aligned} f\left\{\boldsymbol{\beta}^*|\boldsymbol{D}, \tau_m, \sigma_\epsilon^2\right\} &= (1-\omega_1-\omega_2)\delta_0(\beta_1^*, \beta_2^*) \\ &\quad + \omega_1\, N(\hat{\beta}_1, \sigma_{\hat{\beta}_1}^2)\delta_0(\beta_2^*) + \omega_2\, N(\hat{\beta}_1, \sigma_{\hat{\beta}_1}^2)\, N(\hat{\beta}_2, \sigma_{\hat{\beta}_2}^2). \end{aligned}$$

We thus find

$$\begin{aligned} \beta_1^*|\boldsymbol{D}, \tau_1, \sigma_\epsilon^2 &\sim \{\omega_1+\omega_2\}\, N\left(\hat{\beta}_1^*, \sigma_{\hat{\beta}_1}^2\right) \\ &\quad + \{1-\omega_1-\omega_2\}\,\delta_0(\beta_1^*), \\ \beta_2^*|\boldsymbol{D}, \tau_2, \sigma_\epsilon^2 &\sim \omega_2\, N\left(\hat{\beta}_2^*, \sigma_{\hat{\beta}_2}^2\right) + \{1-\omega_2\}\,\delta_0(\beta_2^*). \end{aligned}$$

Appendix B: Fitting procedure II (WavNorm) - Derivation of the posterior distribution of the functional effects in the wavelet space

Within the wavelet space, the wavelet coefficients $D_i(j,k)$ are assumed to be independent. When the variances $\sigma_\epsilon^2(j,k)$ and smoothing parameters $\tau_m(j,k)$ are assumed to be known, the posterior of $\boldsymbol{\beta}(j,k)$ only involves $\boldsymbol{D}(j,k)$, $\sigma_\epsilon^2(j,k)$ and $\tau_m(j,k)$. For notational convenience, we will suppress the index (j,k).

$$
\begin{aligned}
p\left\{\boldsymbol{\beta}^*|\boldsymbol{D},\boldsymbol{\tau},\sigma_\epsilon^2\right\} &= \frac{p\left\{\boldsymbol{\beta}^*,\boldsymbol{D}|\boldsymbol{\tau},\sigma_\epsilon^2\right\}}{p\left\{\boldsymbol{D}|\boldsymbol{\tau},\sigma_\epsilon^2\right\}} \\
&\Updownarrow \text{ Equations (3.27) - (3.28)} \\
p\left\{\boldsymbol{\beta}^*|\boldsymbol{D},\boldsymbol{\tau},\sigma_\epsilon^2\right\} &\propto \left\{(\sigma_\epsilon^2)^{N/2}\text{exp}\left[-\frac{(\boldsymbol{D}-\boldsymbol{X}\boldsymbol{\beta}^*)^T(\boldsymbol{D}-\boldsymbol{X}\boldsymbol{\beta}^*)}{2\sigma_\epsilon^2}\right]\times\right. \\
&\quad \left.\prod_{m=1}^{q}(\tau_m\sigma_\epsilon^2)^{-1/2}\text{exp}\left(-\frac{\beta_m^{*2}}{2\tau_m\sigma_\epsilon^2}\right)\right\}\times \\
&\quad |\boldsymbol{V}\sigma_\epsilon^2|^{1/2}\text{exp}\left(\frac{\boldsymbol{D}^T\boldsymbol{V}^{-1}\boldsymbol{D}}{2\sigma_\epsilon^2}\right) \\
&\Updownarrow \text{ Using the orthogonal property of design matrix } \boldsymbol{X} \\
p\left\{\boldsymbol{\beta}^*|\boldsymbol{D},\boldsymbol{\tau},\sigma_\epsilon^2\right\} &\propto \text{exp}\left(-\frac{\boldsymbol{D}^T\boldsymbol{D}}{2\sigma_\epsilon^2}\right)\prod_{m=1}^{q}\left\{(\tau_m\sigma_\epsilon^2)^{-1/2}\times\right. \\
&\quad \left.\text{exp}\left[-\frac{(\boldsymbol{X}_m^T\boldsymbol{X}_m+1/\tau_m)\beta_m^{*2}-2\boldsymbol{X}_m^T\boldsymbol{D}\beta_m^*}{2\sigma_\epsilon^2}\right]\right\}\times \\
&\quad \text{exp}\left(\frac{\boldsymbol{D}^T\boldsymbol{D}}{2\sigma_\epsilon^2}\right)\prod_{m=1}^{q}(\boldsymbol{X}_m^T\boldsymbol{X}_m\tau_m+1)^{1/2}\text{exp}\left(-\frac{\boldsymbol{D}^T\boldsymbol{X}_m\boldsymbol{X}_m^T\boldsymbol{D}}{\boldsymbol{X}_m^T\boldsymbol{X}_m+1/\tau_m}\right) \\
&\Updownarrow \\
p\left\{\boldsymbol{\beta}^*|\boldsymbol{D},\boldsymbol{\tau},\sigma_\epsilon^2\right\} &\propto \prod_{m=1}^{q}\left(\frac{\sigma_\epsilon^2}{\boldsymbol{X}_m^T\boldsymbol{X}_m+1/\tau_m}\right)^{-1/2}\times \\
&\quad \text{exp}\left[-\frac{\beta_m^{*2}-2\frac{\boldsymbol{X}_m^T\boldsymbol{D}}{\boldsymbol{X}_m^T\boldsymbol{X}_m+1/\tau_m}\beta_m^*+\frac{\boldsymbol{D}^T\boldsymbol{X}_m\boldsymbol{X}_m^T\boldsymbol{D}}{(\boldsymbol{X}_m^T\boldsymbol{X}_m+1/\tau_m)^2}}{2\sigma_\epsilon^2(\boldsymbol{X}_m^T\boldsymbol{X}_m+1/\tau_m)^{-1}}\right] \\
&\Updownarrow \\
p\left\{\boldsymbol{\beta}^*|\boldsymbol{D},\boldsymbol{\tau},\sigma_\epsilon^2\right\} &\propto \prod_{m=1}^{q}\left(\frac{\sigma_\epsilon^2}{\boldsymbol{X}_m^T\boldsymbol{X}_m+1/\tau_m}\right)^{-1/2}\text{exp}\left[-\frac{\left(\beta_m^*-\frac{\boldsymbol{X}_m^T\boldsymbol{D}}{\boldsymbol{X}_m^T\boldsymbol{X}_m+1/\tau_m}\right)^2}{2\sigma_\epsilon^2(\boldsymbol{X}_m^T\boldsymbol{X}_m+1/\tau_m)^{-1}}\right] \\
&\Updownarrow \\
p\left\{\boldsymbol{\beta}^*|\boldsymbol{D},\boldsymbol{\tau},\sigma_\epsilon^2\right\} &\sim \prod_{m=1}^{q}N\left(\frac{\boldsymbol{X}_m^T\boldsymbol{D}(j,k)}{\boldsymbol{X}_m^T\boldsymbol{X}_m+1/\tau_m(j,k)},\frac{\sigma_\epsilon^2(j,k)}{\boldsymbol{X}_m^T\boldsymbol{X}_m+1/\tau_m(j,k)}\right)
\end{aligned}
$$

Appendix C: Fitting procedure II (WavNorm) - Estimation of the variance components by empirical Bayes

For notational convenience we will suppress the index (j,k) in the derivation. We start from $2\times$ the marginal log-likelihood given in Equation (3.33):

$$-2\times l(\boldsymbol{D}|\boldsymbol{\tau},\boldsymbol{\sigma}_\epsilon^2) \;\propto\; \sum_{j=0}^{J}\sum_{k=1}^{K_j}\left(\sum_{m=1}^{q}\left\{\log\left[\boldsymbol{X}_m^T\boldsymbol{X}_m\tau_m(j,k)+1\right]\right\}+N\,\log\left[\sigma_\epsilon^2(j,k)\right]+\right.$$
$$\left.\frac{1}{\sigma_\epsilon^2(j,k)}\boldsymbol{D}^T(j,k)\left[\boldsymbol{I}-\sum_{m=1}^{q}\frac{\boldsymbol{X}_m\boldsymbol{X}_m^T}{\boldsymbol{X}_m^T\boldsymbol{X}_m+1/\tau_m(j,k)}\right]\boldsymbol{D}(j,k)\right)$$

The smoothing parameter $\tau_m(j,k)$, given $\sigma_\epsilon^2(j,k)$, can be estimated by solving

$$\frac{\partial l(\boldsymbol{D}|\boldsymbol{\tau},\boldsymbol{\sigma}_\epsilon^2)}{\partial\tau_m}=0,$$

which becomes

$$\begin{aligned}\frac{\boldsymbol{X}_m^T\boldsymbol{X}_m}{\boldsymbol{X}_m^T\boldsymbol{X}_m\tau_m+1}-\frac{\boldsymbol{D}^T\boldsymbol{X}_m\boldsymbol{X}_m^T\boldsymbol{D}}{\left[\boldsymbol{X}_m^T\boldsymbol{X}_m+1/\tau_m\right]^2\tau_m^2\sigma_\epsilon} &= 0\\ \left(\boldsymbol{X}_m^T\boldsymbol{X}_m\right)^2\tau_m\sigma_\epsilon^2+\boldsymbol{X}_m^T\boldsymbol{X}_m\sigma_\epsilon^2-\boldsymbol{D}^T\boldsymbol{X}_m\boldsymbol{X}_m^T\boldsymbol{D} &= 0.\end{aligned}$$

Hence,

$$\tau_m=\frac{\boldsymbol{D}^T\boldsymbol{X}_m\boldsymbol{X}_m^T\boldsymbol{D}}{\left(\boldsymbol{X}_m^T\boldsymbol{X}_m\right)^2\sigma_\epsilon^2}-\frac{1}{\boldsymbol{X}_m^T\boldsymbol{X}_m}.$$

By definition $\tau_m\in[0,\infty]$. Hence, the MML estimator (after reintroducing the index (j,k)) becomes

$$\hat{\tau}_m(j,k)=\left[\frac{\boldsymbol{D}^T(j,k)\boldsymbol{X}_m\boldsymbol{X}_m^T\boldsymbol{D}(j,k)}{\left(\boldsymbol{X}_m^T\boldsymbol{X}_m\right)^2\sigma_\epsilon^2(j,k)}-\frac{1}{\boldsymbol{X}_m^T\boldsymbol{X}_m}\right]_+.$$

The MML-estimator of $\sigma_\epsilon^2(j,k)$, given the smoothing parameters $\tau_m(j,k)$, is obtained by solving

$$\frac{\partial l(\boldsymbol{D}|\boldsymbol{\tau},\boldsymbol{\sigma}_\epsilon^2)}{\partial\sigma_\epsilon^2(j,k)}=0,$$

which becomes

$$\frac{N}{\sigma_\epsilon^2(j,k)}-\frac{\boldsymbol{D}^T(j,k)\boldsymbol{V}^{-1}(j,k)\boldsymbol{D}(j,k)}{\sigma_\epsilon^4(j,k)}=0.$$

We thus find

$$\hat{\sigma}_{\epsilon}^{2}(j,k) = \frac{\boldsymbol{D}^{T}(j,k)\boldsymbol{V}^{-1}(j,k)\boldsymbol{D}(j,k)}{N}.$$

Chapter 4

Tiling array expression studies with flexible designs

In the last few years the use of tiling microarrays for whole-genome transcriptome analysis has become well established. Many studies have shown them to be a convenient tool for exploring and unraveling the complex genome-wide transcriptional landscape of higher organisms, in which not only protein-coding genes, but also non-coding RNAs play an important role (e.g. Yamada et al., 2003; Kampa et al., 2004; Schadt et al., 2004; Stolc et al., 2005). The methods that have been developed for transcriptome analysis with tiling arrays either focus on segmentation and transcript discovery within a single biological condition (Toyoda and Shinozaki, 2005; Zeller et al., 2008; Nicolas et al., 2009; Munch et al., 2006), or on the detection of differential expression between two distinct conditions (Piccolboni, 2008; Otto et al., 2012). The wavelet-based method discussed in Chapter 3 performs these two tasks simultaneously (see also Clement et al., 2012). The focus in tiling array studies has recently shifted towards more complex designs, such as studies with more than two conditions (Andriankaja et al., 2012) and studies with several experimental factors (Okamoto et al., 2010). Furthermore, it is recognized that expression is a dynamic rather than a static phenomenon. Hence, more and more time-course experiments are designed to provide insights into the whole-genome transcript regulation of species during different developmental stages or external periodic changes in the environment (Hazen et al., 2009; Granovskaia et al., 2010).

To our knowledge, no general methodologies for the analysis of tiling array studies with more

complex designs have yet been proposed in literature. Instead, most tiling array analysis pipelines in current studies are very specific for the particular design and research question at hand (e.g. Granovskaia et al., 2010; Hazen et al., 2009; Okamoto et al., 2010; Samanta et al., 2006; Assarsson et al., 2008). Their approach also often consists of applying methods designed for classical microarrays that heavily rely on existing annotation when summarizing probes to probesets (e.g. Naouar et al., 2009; Andriankaja et al., 2012; Okamoto et al., 2010).

The wavelet-based methodology of Chapter 3 was initially developed to simultaneously perform transcript discovery and test for differential expression in a two-group design, while remaining unbiased by existing annotation. However, the modeling framework is flexible and can be extended to cope with more complex designs. This is done by adapting the model design matrix and the probe-wise inference procedure in an appropriate way. The study designs that we will consider are time-course studies, studies with more than two conditions and multiple-factor studies. Section 4.1 describes the methodology for wavelet-based trancriptome analysis for more flexible designs. Firstly, the model extension and associated parameter estimation is discussed. Secondly, we describe the statistical inference method for the detection of transcriptional effect regions. The flexibility of the method is illustrated on three case studies in Section 4.2.

4.1 Wavelet-based transcriptome analysis in more flexible designs

4.1.1 Extending the wavelet-based model towards more flexible designs

First we reconsider Model (3.2), which is the functional model in the genomic data space. In matrix form, with the notation of Chapter 3, it is given by

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{B} + \boldsymbol{E}. \tag{4.1}$$

In the model that we introduced for transcript discovery and testing for differential expression (Chapter 3) the design matrix $\boldsymbol{X}$ has dimensions $N \times 2$ and is constructed as

$$\boldsymbol{X} = \begin{pmatrix} \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{-1} \end{pmatrix},$$

where the upper row represents the dummy coding for the N_1 arrays in the group under condition 1 and the lower row is the dummy coding for the N_2 arrays in the group under condition 2. The $2 \times T$ effect function matrix $\boldsymbol{B}$ relates to the probe-wise effect functions $\beta_1(t)$ and $\beta_2(t)$ on the respective rows. Column 1 of $\boldsymbol{X}$ will be used to find regions with a mean expression level above some threshold, whereas the coding in column 2 allows for assessing differential expression between the two conditions. Note that the coding in $\boldsymbol{X}$ implies that the two effect functions are estimated orthogonally if the study design is balanced, i.e. $N_1 = N_2$. This can be seen from

$$\boldsymbol{X}^T\boldsymbol{X} = \begin{pmatrix} N/2 & 0 \\ 0 & N/2 \end{pmatrix},$$

with $N_1 = N_2 = \frac{N}{2}$, for a balanced design.

As described in Chapter 3, the model in the original data space (4.1) is transformed to the wavelet space to obtain the wavelet-based model (3.3). This wavelet-based model is fitted using the WavNorm(j) method, where a normally distributed prior is imposed on the effect functions in the wavelet space, and where the estimators of the error variances $\sigma^2_\epsilon(j)$ depend on the wavelet scale j, but not on the location k within the same wavelet scale. It was demonstrated in Chapter 3 that the estimated smoothing parameters $\hat{\tau}_m(j,k)$ induce a regularization of the wavelet coefficients of the effect functions. This regularization leads to a denoised expression signal in the original data space, retaining the main features.

More flexible designs can be analyzed with the method by adapting the design matrix $\boldsymbol{X}$ in an appropriate way. Firstly, the parameterization should be tailored for answering the specific research questions. Secondly, it must be compatible with the fast algorithms described in Chapter 3. This second argument comes down to preserving the orthogonality of $\boldsymbol{X}$. In the remainder of this section we first focus on general *time-course designs* and *single-factor designs* for more than 2 groups. Next, specific time-course designs for assessing *circadian rhythms* in the transcriptome are considered. Finally, we describe an adaptation for *non-orthogonal designs*, typically encountered in multi-factor studies.

4.1.1.1 Time-course designs

In tiling array time-course experiments one is often interested in the detection of pairwise differentially expressed regions between any two time points. In addition, one may aim at detecting

effects of transcriptional activity in time, e.g. linearly increasing or decreasing transcriptional expression of certain regions. We deal with these two objectives by modeling orthogonal polynomials of the time points. This approach has also been used in the context of quantitative trait associated expression studies using traditional microarrays (Qu and Xu, 2006). Other coding systems for the design matrix $\boldsymbol{X}$ are possible as well. However, orthogonal polynomials are particularly suited in this setting because the model parameters are directly interpretable as linear, quadratic or higher-order effects in time. Moreover, the orthogonality enables the use of the fast algorithms of Chapter 3.

Consider a time-course experiment with whole-genome expression levels measured at q time points. Let N be the total number of arrays. The numbers of arrays used at each time point are represented by $N_1, \ldots, N_q$, with $N_1 + \ldots + N_q = N$. Suppose the experiment is balanced, i.e. $N_1 = \ldots = N_q$, with equidistant time points. The design matrix $\boldsymbol{X}$ in Model (4.1) has dimension $N \times q$ and can be written as

$$\boldsymbol{X} = \begin{pmatrix} \mathbf{1} & \psi_1(\boldsymbol{X}_1) & \psi_2(\boldsymbol{X}_1) & \cdots & \psi_{q-1}(\boldsymbol{X}_1) \\ \mathbf{1} & \psi_1(\boldsymbol{X}_2) & \psi_2(\boldsymbol{X}_2) & \cdots & \psi_{q-1}(\boldsymbol{X}_2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{1} & \psi_1(\boldsymbol{X}_q) & \psi_2(\boldsymbol{X}_q) & \cdots & \psi_{q-1}(\boldsymbol{X}_q) \end{pmatrix}, \tag{4.2}$$

where $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_q$ are the $N_1, \ldots, N_q$-valued vectors that correspond to the q respective time points in the experiment. In (4.2) each function $\psi_j(\boldsymbol{x})$ is a polynomial of degree j, with $j = 0, \ldots, q-1$, and is orthogonal to $\psi_k(\boldsymbol{x})$ ($k = 0, \ldots, q-1$) if $j \neq k$. Note that each $\mathbf{1}$ in the first column of $\boldsymbol{X}$ corresponds to $\psi_0(\boldsymbol{X}_i) = 1$ ($i = 1, \ldots, q$). The orthogonality of $\boldsymbol{X}$ can be verified by

$$\boldsymbol{X}^T\boldsymbol{X} = \begin{pmatrix} N & 0 & 0 & 0 & \ldots & 0 \\ 0 & \sum_{i=1}^N \psi_1^2(\boldsymbol{X}_i) & 0 & 0 & \ldots & 0 \\ 0 & 0 & \sum_{i=1}^N \psi_2^2(\boldsymbol{X}_i) & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & \sum_{i=1}^N \psi_{q-2}^2(\boldsymbol{X}_i) & 0 \\ 0 & 0 & 0 & \ldots & 0 & \sum_{i=1}^N \psi_{q-1}^2(\boldsymbol{X}_i) \end{pmatrix}. \tag{4.3}$$

With this design matrix a $q \times T$ matrix $\boldsymbol{B}$ for the q effect functions is associated. The first row of $\boldsymbol{B}$ corresponds with the overall mean expression level over all time points, while rows 2 until q are associated with a linear, quadratic, cubic, $\ldots$, $(q-1)$-th order polynomial effects between the time points. The fitted expression levels at each time point are obtained by a linear

combination of the effect functions in accordance with Model (4.1). This allows for a straightforward comparison between any two time points. When dealing with non-balanced and non-equidistance designs, a simple procedure can be applied for obtaining orthogonal polynomials; see Narula (1979).

When inferring on linear combinations of functions, it is desirable to induce the same degree of smoothing for each functional effect. This implies the estimation of one general smoothing parameter $\tau(j,k)$, instead of a separate $\tau_m(j,k)$ for each effect function ($m = 1, ..., q$). To retain the closed-form solutions needed for the fast algorithms of Chapter 3, however, the diagonal elements of $\boldsymbol{X}^T\boldsymbol{X}$ should be identical. This can be obtained by normalizing each column vector of $\boldsymbol{X}$, resulting in the normalized design matrix, say $\boldsymbol{X'}$. Hence,

$$\boldsymbol{X'}^T\boldsymbol{X'} = \begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix} = \boldsymbol{I}_q, \tag{4.4}$$

where $\boldsymbol{I}_q$ is an $q \times q$ identity matrix. For this orthonormal design matrix $\boldsymbol{X'}$ it can be shown that the common smoothing parameter is estimated by

$$\hat{\tau}(j,k) = \left[\frac{\boldsymbol{D}^T(j,k)\boldsymbol{X'}\boldsymbol{X'}^T\boldsymbol{D}(j,k)}{q\sigma_\epsilon^2(j,k)} - 1 \right]_+ . \tag{4.5}$$

The derivation of (4.5) is given in Appendix A.

Although design matrix (4.2) is also suitable for non-ordered single-factor studies, alternative parameterizations can be used, such as the Helmert contrast design matrix. Helmert contrasts are designed to compare the mean expression at a specific time point with the overall mean over all preceding time points. The Helmert contrast design matrix is given by

$$\boldsymbol{X} = \begin{pmatrix} \mathbf{1} & \mathbf{-1} & \mathbf{-1} & \mathbf{-1} & \dots & \mathbf{-1} & \mathbf{-1} \\ \mathbf{1} & \mathbf{1} & \mathbf{-1} & \mathbf{-1} & \dots & \mathbf{-1} & \mathbf{-1} \\ \mathbf{1} & \mathbf{0} & \mathbf{2} & \mathbf{-1} & \dots & \mathbf{-1} & \mathbf{-1} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{3} & \dots & \mathbf{-1} & \mathbf{-1} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \boldsymbol{q-2} & \mathbf{-1} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \boldsymbol{q-1} \end{pmatrix}. \tag{4.6}$$

Helmert contrast design matrices also possess the orthogonality property, i.e.

$$\boldsymbol{X}^T\boldsymbol{X} = \begin{pmatrix} N & 0 & 0 & 0 & \dots & 0 \\ 0 & \sum_{i=1}^{2} N_i & 0 & 0 & \dots & 0 \\ 0 & 0 & 2\sum_{i=1}^{3} N_i & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & (q-2)\sum_{i=1}^{q-1} N_i & 0 \\ 0 & 0 & 0 & \dots & 0 & (q-1)\sum_{i=1}^{q} N_i \end{pmatrix}. \tag{4.7}$$

Similar to the polynomial parameterization, the Helmert contrast design matrix $\boldsymbol{X}$ still needs to be normalized if the same degree of smoothing for all functional effects is desired.

4.1.1.2 Designs for circadian rhythms

We now consider the detection of circadian rhythms in the transcriptome of an organism, based on an equally spaced time-course experiment. It is known that many organisms, e.g. photosynthetic organisms, anticipate changes in the daily environment with an internal oscillator, called the circadian clock (e.g. Hazen et al., 2009). The periodic expression changes that are governed by this oscillator are called the circadian rhythms in the transcriptome. An orthogonal basis system that is well suited to model circular effects is the Fourier basis system. The design matrix

is now given by

$$\boldsymbol{X} = \begin{pmatrix} \mathbf{1} & \sin(\mathbf{0}) & \cos(\mathbf{0}) \\ \mathbf{1} & \sin(\frac{\mathbf{2\pi}}{\boldsymbol{q}}) & \cos(\frac{\mathbf{2\pi}}{\boldsymbol{q}}) \\ \mathbf{1} & \sin(\frac{\mathbf{4\pi}}{\boldsymbol{q}}) & \cos(\frac{\mathbf{4\pi}}{\boldsymbol{q}}) \\ \vdots & \vdots & \vdots \\ \mathbf{1} & \sin(\mathbf{2\pi} - \frac{\boldsymbol{\pi}}{\boldsymbol{q}}) & \cos(\mathbf{2\pi} - \frac{\boldsymbol{\pi}}{\boldsymbol{q}}) \end{pmatrix}. \tag{4.8}$$

Again the separate effect functions can be estimated orthogonally, i.e.

$$\boldsymbol{X}^T\boldsymbol{X} = \begin{pmatrix} N & 0 & 0 \\ 0 & q & 0 \\ 0 & 0 & q \end{pmatrix}. \tag{4.9}$$

To induce the same degree of smoothing for all effect functions, $\boldsymbol{X}$ can again be normalized as described previously.

4.1.1.3 Non-orthogonal designs

Design matrices for two- or multiple-factor designs are typically non-orthogonal and are not compatible with the fast algorithms presented in Chapter 3. This would lead to undesirably increased computation time for parameter estimation. This problem can be overcome by applying the Gram-Schmidt process to orthogonalize $\boldsymbol{X}$ and estimating the model parameters using the orthogonalized design matrix. The Gram-Schmidt orthogonalization comes down to a QR-decomposition (Golub and Van Loan, 1996) of $\boldsymbol{X}$ into an upper-triangular matrix $\boldsymbol{X}_{tri}$ and an orthogonal matrix $\boldsymbol{X}_{orth}$, which is subsequently used to fit the model in the wavelet space. Upon estimation, the parameters are first backtransformed to the original data space using the IDWT. Next, a second backtransformation is needed to obtain the parameter estimates for the original $\boldsymbol{X}$. This is done by premultiplication with $(\boldsymbol{X}_{orth}^T\boldsymbol{X})^{-1}$. In the original space the effect functions are correlated. Their variance-covariance matrix is given by

$$\left\{(\boldsymbol{X}_{orth}^T\boldsymbol{X})^{-1}\right\}^T \text{Var}\left[\hat{\boldsymbol{\beta}}(t)\right] \left\{(\boldsymbol{X}_{orth}^T\boldsymbol{X})^{-1}\right\}, \tag{4.10}$$

where $\hat{\boldsymbol{\beta}}(t)$ is the q-valued column vector $(\hat{\beta}_1(t), \hat{\beta}_2(t), \ldots, \hat{\beta}_q(t))^T$.

Because the Gram-Schmidt process performs a linear transformation of the original predictor variables, the least squares solution after fitting a linear model based on the original predictors

and the orthogonalized design matrix are equivalent. For regularization purposes we introduced a normally distributed prior on the parameters of the effect functions. The resulting shrinkage estimators are influenced by transforming the design matrix. This is clear from the expression of the smoothing parameter

$$\hat{\tau}_m(j,k) = \left[\frac{\boldsymbol{D}^T(j,k)\boldsymbol{X}_{orth,m}\boldsymbol{X}_{orth,m}^T\boldsymbol{D}(j,k)}{\left(\boldsymbol{X}_{orth,m}^T\boldsymbol{X}_{orth,m}\right)^2\sigma_\epsilon^2(j,k)} - \frac{1}{\boldsymbol{X}_{orth,m}^T\boldsymbol{X}_{orth,m}} \right]_+, \tag{4.11}$$

which has a non-linear relationship with $\boldsymbol{X}_{orth,m}$. However, the high-dimensionality of the data justifies this sacrifice as the available closed-form solutions provide a tremendous decrease in the computational complexity of the algorithms for parameter estimation.

4.1.2 Statistical inference: detection of transcriptional effect regions

The goal of the statistical inference procedure is the detection of genomic regions that show changes in their transcriptional activity according to the effect under study, which we call *transcriptional effect regions*. Depending on the study design and the research objective, either the parameters themselves or a function of the parameters are used to detect transcriptional effect regions. Hence, the effect of interest can be represented by $G\{\boldsymbol{\beta}(t)\}$, where G is a linear or non-linear function of the parameters. If the parameters themselves are used for inference, G is equal to the identity function.

For each genomic location t, $G\{\boldsymbol{\beta}(t)\}$ is compared to a threshold value δ, which can be chosen by the researcher. The Bayesian FDR procedure described in Chapter 3 is adopted to evaluate statistical significance. This may be written as

$$BFDR_G(t) = \Pr\{G\{\boldsymbol{\beta}(t)\} < \delta | \boldsymbol{Y}\} \tag{4.12}$$

for positive contrasts, e.g. overexpression in differential expression analysis, and as

$$BFDR_G(t) = \Pr\{G\{\boldsymbol{\beta}(t)\} > -\delta | \boldsymbol{Y}\} \tag{4.13}$$

for negative contrasts, e.g. underexpression in differential expression analysis.

If $G\{\boldsymbol{\beta}(t)\}$ can be written as a linear combination of the effect functions $\boldsymbol{\beta}(t)$, say $\boldsymbol{L}\boldsymbol{\beta}(t)$, Equations (4.12) and (4.13) only involve the calculation of the probability of univariate normally distributed random variables. If $G\{\boldsymbol{\beta}(t)\}$ is a non-linear function of the $\boldsymbol{\beta}(t)$ functions this is not the case. In that situation $BFDR_G(t)$ can be approximated by simulation.

For general time-course designs one may be interested in detecting genomic regions that show a linear or quadratic trend. In this situation $G\{\boldsymbol{\beta}(t)\}$ is the effect function $\boldsymbol{\beta}(t)$ that corresponds with the linear term $\psi_1(\boldsymbol{X})$ or the quadratic term $\psi_2(\boldsymbol{X})$ in (4.2). On the other hand, if interest lies in the detection of differentially expressed regions between different time points, inference is performed on each row of $\boldsymbol{LB} = \boldsymbol{ZXB}$. This is a $\frac{q(q-1)}{2} \times T$ matrix, with $\boldsymbol{Z}$ a $\frac{q(q-1)}{2} \times N$ contrast matrix corresponding to the $\frac{q(q-1)}{2}$ pairwise comparisons between two time points.

In circadian rhythm designs the sine and the cosine effect functions are combined to give the amplitude $A(t)$ of the circadian rhythm per probe position, i.e.

$$G\{\boldsymbol{\beta}(t)\} = A(t) = \sqrt{\beta_2^2(t) + \beta_3^2(t)}. \tag{4.14}$$

Based on the size of $A(t)$ circadian effect regions can be detected. Because of the non-linear dependence of $A(t)$ on the $\boldsymbol{\beta}(t)$'s, $BFDR_G(t)$ is approximated through simulation. In each simulation step we sample from the normally distributed estimated parameters of the sine and cosine effect functions and calculate $A_{\text{sim}}(t)$. $BFDR_G(t)$ is now estimated by the proportion of simulations for which $A_{\text{sim}}(t) < \delta$.

In the case of non-orthogonal designs in multiple-factor studies, there are several choices of $G\{\boldsymbol{\beta}(t)\}$, depending on the study objective. The idea remains the same, however.

4.2 Three case studies

In this section the use and flexibility of the extended wavelet-based modeling approach is illustrated in three case studies for transcriptome analysis with different experimental set-ups.

4.2.1 Case study 1: Time-course experiment

The first data set consists of a tiling array expression study for identifying the molecular events associated with early leaf development of the plant species *Arabidopsis thaliana* (Andriankaja et al., 2012). This study had two main goals. The researchers wanted to unravel the underlying mechanisms of the transition from cell division to cell expansion, while they also focused on the study of the transition from non-photosynthetic to photosynthetic leaves. Trancriptome analysis for six developmental time points (day 8 to day 13) was conducted with AGRONOMICS1 tiling

arrays (Rehrauer et al., 2010), with three biological replicates per time point. Primarily, the detection of differentially expressed regions between any two pairs of developmental time points was studied. This specific study design, however, also allows for the detection of expression regions that change linearly over time.

4.2.1.1 Pairwise comparison

Figure 4.1 shows an example of a genomic region on chromosome 1 of *Arabidopsis thaliana* found to be differentially expressed between different time points with the wavelet-based method. The threshold value used here is $\delta_{FC} = 1.2$. This choice allows to detect small differences in mean expression between two days. The expression of gene *AT1G04350* clearly increases from day 8 to day 13. For the most significant contrasts, especially between days 13 and 8, the detected regions clearly match with the exons of this gene. Between two consecutive days, however, the difference between the fitted expression functions is not always large enough to completely mimic this exonic structure, e.g. between day 9 and day 8.

The differentially expressed regions between any two pairs of days, detected by the wavelet-based analysis, are evaluated against the differentially expressed genes reported by the RMA method (Irizarry et al., 2003). This is done by comparing the results of a gene set enrichment analysis (GSEA) based on both methods. GSEA is an analytical method that extracts biological insight from expression data by focusing on gene sets or groups of genes that share a common biological function, chromosomal location, or regulation (Subramanian et al., 2005). By mapping the genomic regions found by the wavelet-based method to the *Arabidopsis thaliana* TAIR9 annotation (Swarbreck et al., 2008), a list of genes is created for this method. The enrichment analysis is conducted with the Plaza tool (Proost et al., 2009). It reveals a strong overlap in the biological processes for which the genes detected by both methods encode. A total of 483 enrichments are identified using both gene sets of which 360 common enrichments are shared. The RMA gene list has 75 specific enrichments, while the wavelet-based gene list has 48.

The genes for which similar enrichments are found between the two methods depend on the existing genome annotation. However, with the wavelet-based method also non-annotated differentially expressed regions can be discovered, which is not possible with the RMA method. Selected regions were validated with quantitative real-time reverse transcription polymerase chain

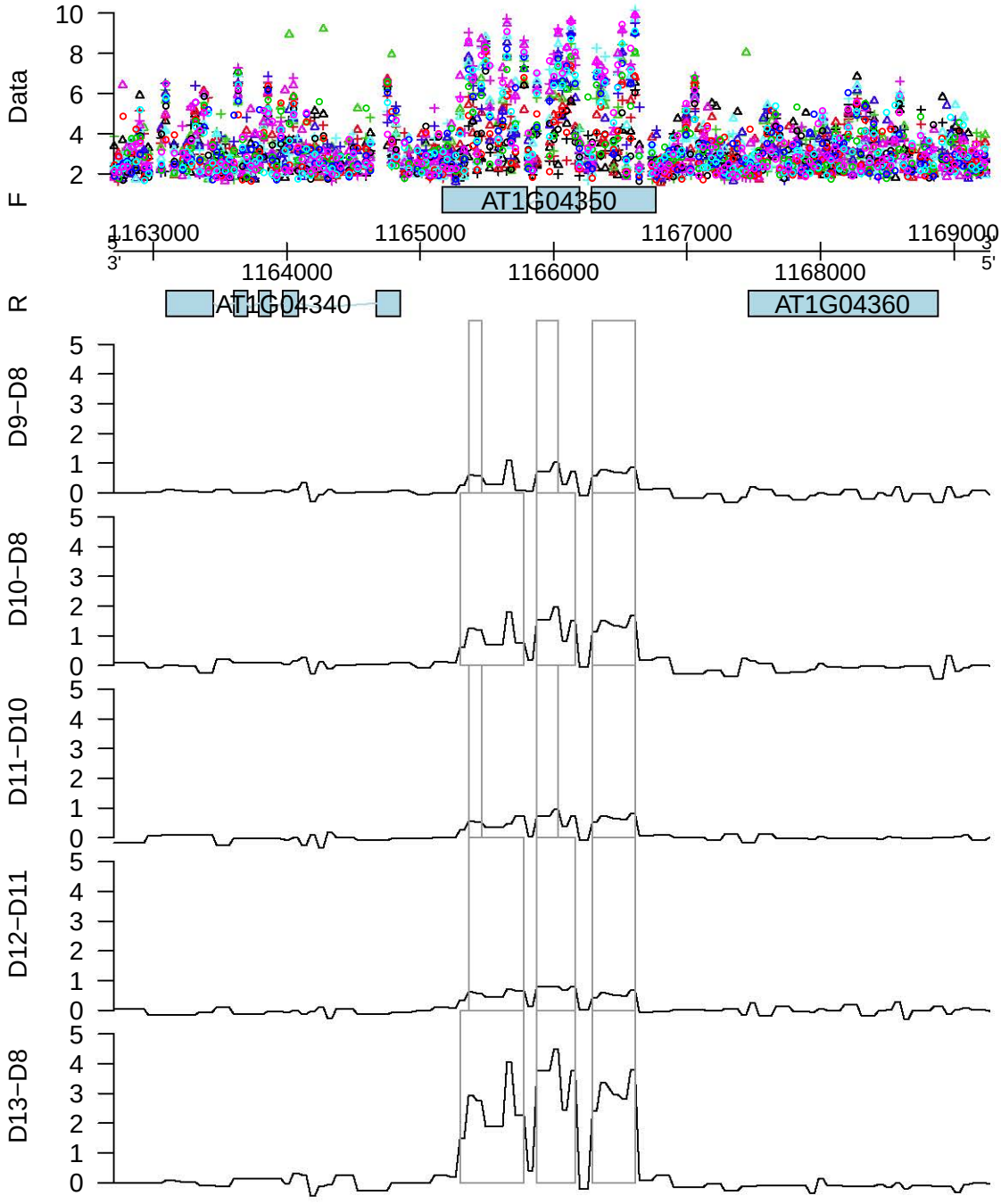


Figure 4.1: Fitted differential expression effect for the genomic region of gene *AT1G04350* on the forward strand of chromosome 1, between selected pairs of developmental time points varying from day 8 (D8) to day 13 (D13). The grey rectangles indicate the regions showing a significant differential expression effect ($FDR = 0.05$). The three replicates are indicated by $\circ$, $+$ and $\triangle$, while the different days are represented by different colors: black (D8), red (D9), green (D10), blue (D11), cyan (D12) and magenta (D13).

reaction (qRT-PCR) by Megan Andriankaja, Department of Plant Systems Biology, Flemish Institute of Biotechnology. This is a molecular technique that is often used in biological validation studies to quantify messenger RNA or non-coding RNA in cells or tissues. The regions for validation are chosen based on the following criteria:

1. Region is not in or near an exon or promoter from an annotated gene.
2. Longer regions containing more differentially expressed probes are preferentially selected.

Table 4.1: Overview of non-annotated regions selected for qRT-PCR validation. Coordinates on the *Arabidopsis thaliana* genome are presented in the first four columns. The *Contrast* column gives the pair of days for which the comparison is made; *WavNorm FC* gives the FC estimated by the wavelet-based model average over the probes along the whole transcript; *qRT-PCR* indicates whether the differential expression is confirmed by qRT-PCR, as well as its directionality.

Chromosome	Strand	Start	End	Contrast	WavNorm FC	qRT-PCR
5	forward	1042594	1042866	D13-D8	-2.83	< 0
4	reverse	7642124	7643140	D13-D8	-1.83	no DE
1	forward	2407636	2408052	D10-D9	2.02	> 0
1	forward	17392521	17392937	D10-D9	1.58	> 0
1	reverse	17391545	17391961	D10-D9	0.99	> 0
1	reverse	16115545	16115929	D13-D8	-2.68	< 0
5	forward	1042610	1042866	D10-D9	-0.87	< 0
2	forward	17224523	17224779	D10-D9	-0.87	> 0
1	reverse	29649701	29649956	D13-D8	-1.84	< 0
2	reverse	3326499	3326724	D13-D8	-0.74	< 0
4	forward	10180294	10180518	D10-D9	-0.62	< 0
4	forward	13716555	13716779	D10-D9	-0.60	> 0

Using these criteria 12 regions are selected and qRT-PCR analysis is performed. Note that the RNA material used for the qRT-PCR analysis is extracted from different plants of the same experiment as the RNA extracted for the tiling array analysis. Of the 12 regions, 11 are confirmed to contain differentially expressed transcripts during the time-course analysis and 1 region has

no detectable transcriptional products. Of these 11 regions, 9 regions show fold changes in the same direction as previously identified from the tiling arrays, while 2 regions show fold changes in the opposite direction. These 2 regions, however, also have the lowest estimated fold changes in the wavelet-based analysis. The results are summarized in Table 4.1.

4.2.1.2 Linear and quadratic time effects

In addition to a pairwise comparison analysis, the wavelet-based functional model with the orthogonal polynomial design matrix is also useful for detecting genes with linear and quadratic expression patterns over time. In fact, the estimated parameters now give direct interpretations in terms of the different order time effects, such as linear or quadratic effects. Some example plots of genes from the forward strand of chromosome 1 with a clear linear effect are shown in the upper panels of Figure 4.2. These genes overlap with two of the top detected regions with the largest linear time effect for chromosome 1. The means of the probe-wise linear time effect parameter estimates in these regions are 1.08 and -1.16, respectively.

The lower panels of Figure 4.2 show two genes with a strong quadratic effect. Note that the fitted probe-wise $\log_2$ intensities at the different time points (orange lines) are clearly closer to the mean fitted $\log_2$ intensities over all the probes in the whole detected region at these time points (green line) in comparison to the corresponding observed probe-wise $\log_2$ intensities. This can be explained from the fact that in the wavelet domain strength is borrowed from neighboring probes providing a more reliable estimate for each probe-wise effect.

The upper panel of Figure 4.3 shows an along-chromosome plot with the estimated linear time effects close to gene *AT1G62500*. The negative sign of the estimated parameters implies a decreasing effect over time, which was also seen in the upper right panel of Figure 4.2. More specifically, the effect at probe t is $\hat{\beta}_1(t) \times time$. The lower panel of Figure 4.3 shows regions with a significant quadratic time effect overlapping with gene *AT1G16410*. The quadratic effect seen in the lower left panel of Figure 4.2 translates into a positive value for $\hat{\beta}_2(t)$.

By constructing a linear combination of the fitted polynomial effect functions $\hat{\beta}_m(t)$ according to the design matrix $\boldsymbol{X}$, the fitted $\log_2$ intensities at the different time points can be assessed. Hence, transcript discovery at each time point can be assessed within the same analysis. Figure 4.4 gives the corresponding along-chromosome plots for the same linearly decreasing gene

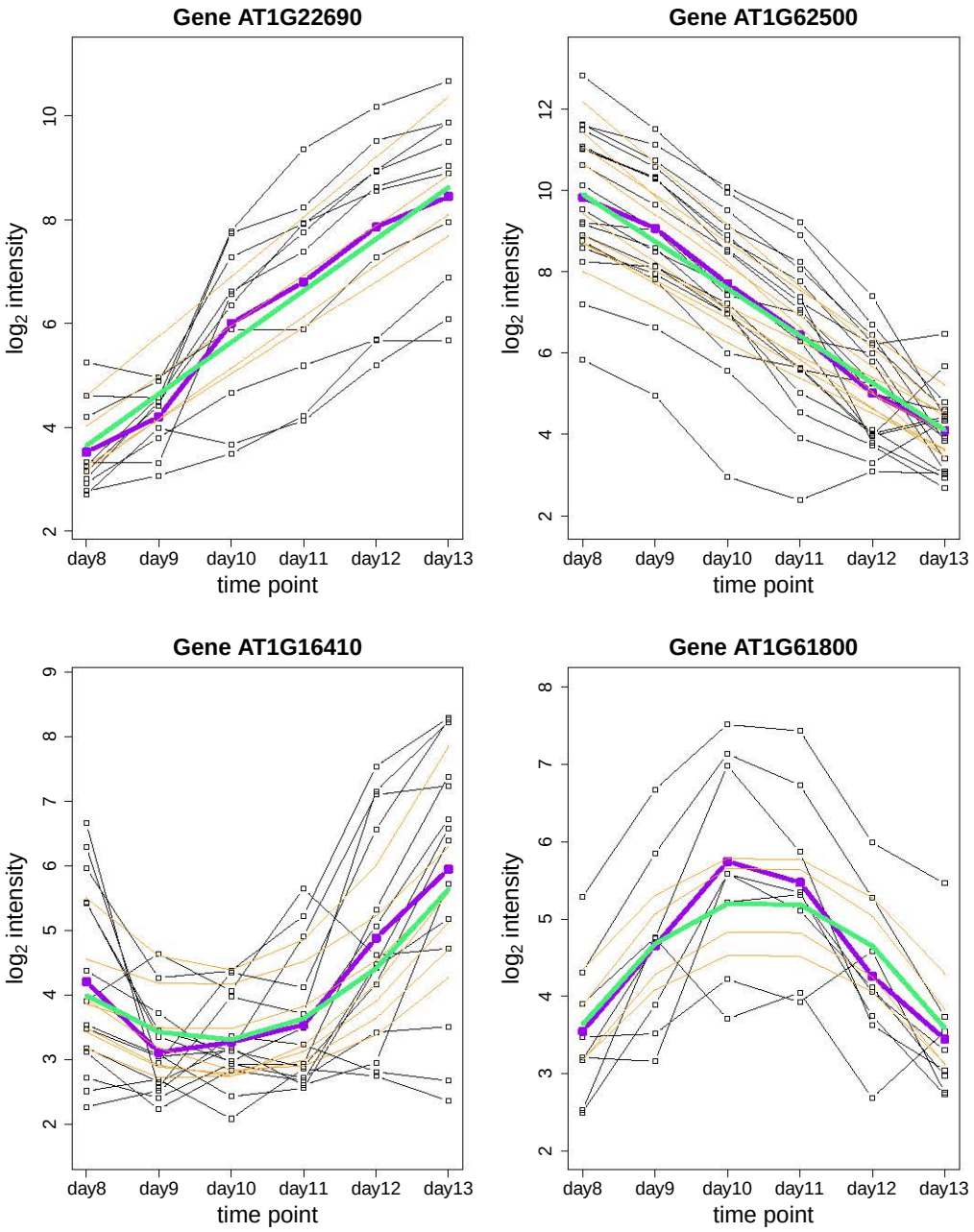


Figure 4.2: Example plot for two genes showing a linearly increasing (upper left) and decreasing (upper right) mean $\log_2$ intensity level and two genes showing a strong quadratic effect over the 6 days in the time course (lower panels). The dotted black lines represent the mean observed $\log_2$ expression for the probes over the three biological replicates at the different time points. The dotted purple line is the mean observed $\log_2$ expression over all the probes in the region. The orange lines are the probe-wise fitted $\log_2$ expression values when only considering the intercept and the linear time effect in the model (upper panels) or considering the intercept, the linear and the quadratic time effect in the model (lower panels). The green line gives the corresponding mean fitted $\log_2$ expression values at the different time points over all the probes in the region.

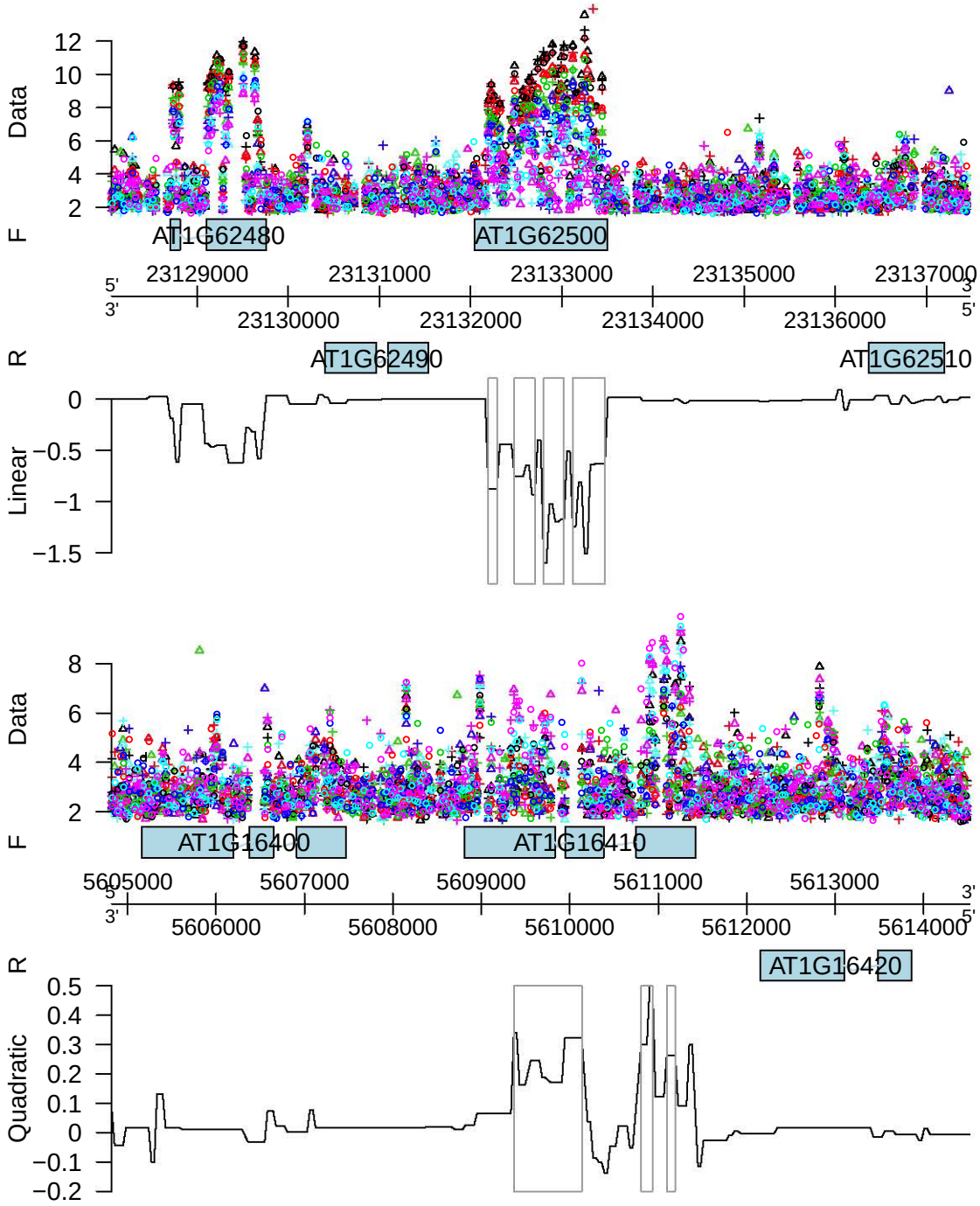


Figure 4.3: Fitted linear time effect for the genomic region of gene *AT1G62500* (upper panel) and fitted quadratic time effect for the genomic regions of gene *AT1G16410* (lower panel) on the forward strand of chromosome 1. The three replicates are indicated by $\circ$, $+$ and $\triangle$, while the days are represented by colors: black (D8), red (D9), green (D10), blue (D11), cyan (D12) and magenta (D13). The grey rectangles indicate the regions showing a significant linear (upper panel) or quadratic (lower panel) time effect ($FDR = 0.05$), while the black line corresponds with the estimated linear or quadratic effect function.

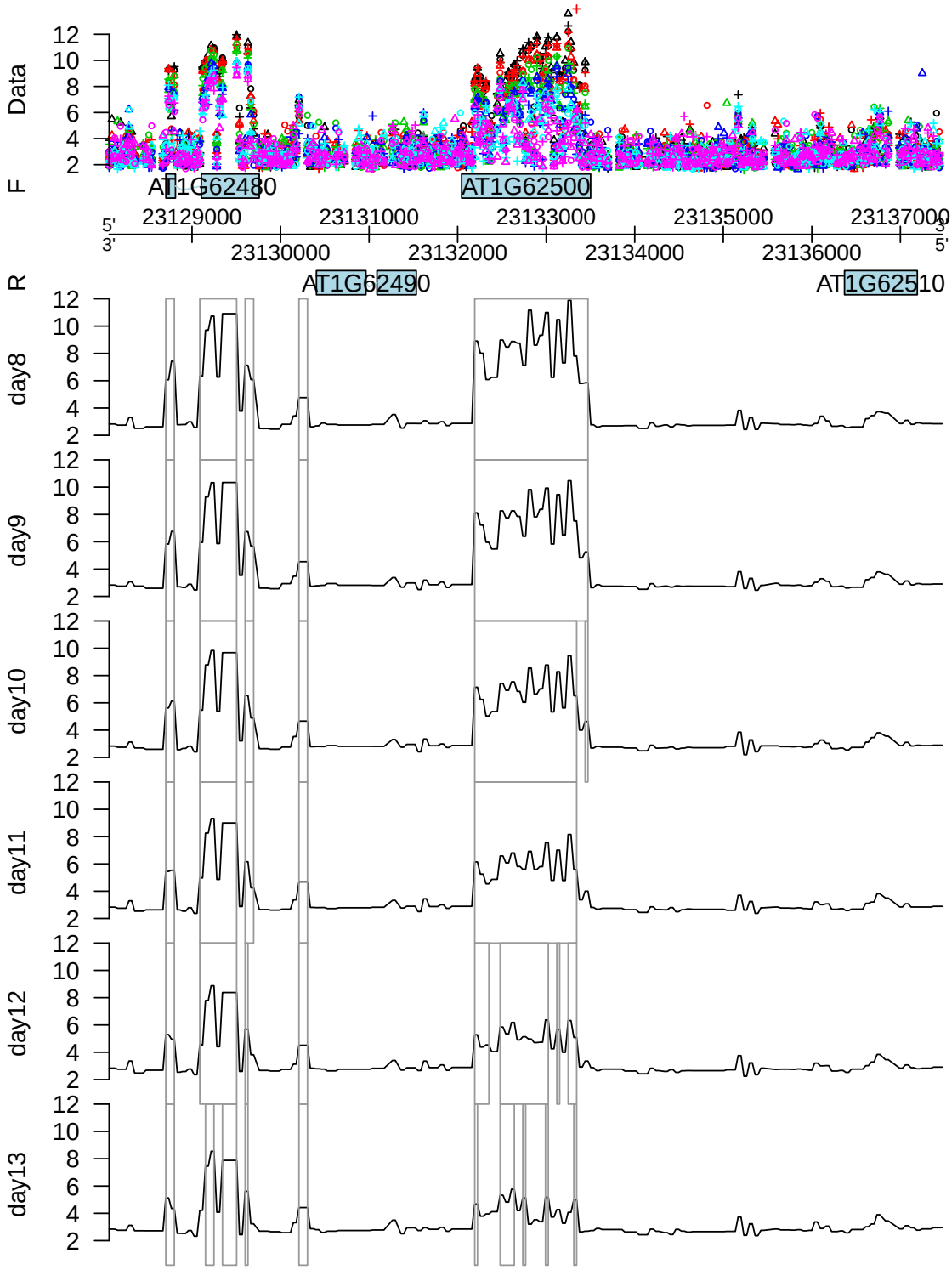


Figure 4.4: Fitted $\log_2$ intensities per time point of the genomic region of gene *AT1G62500* on the forward strand of chromosome 1. The three replicates are indicated by $\circ$, $+$ and $\triangle$, while the days are represented by colors: black (D8), red (D9), green (D10), blue (D11), cyan (D12) and magenta (D13). The grey rectangles indicate the regions showing a significant mean expression ($FDR = 0.05$).

AT1G62500 as depicted in the upper panel of Figure 4.3. The grey rectangles in the plots indicate the discovered regions with mean $\log_2$ intensities significantly above a certain threshold, which was chosen according to the procedure described in Chapter 3. The same decreasing trend is also obvious from this figure.

4.2.2 Case study 2: Circadian rhythms

The second case study concerns an expression analysis to examine circadian rhythms in *Arabidopsis thaliana*. The aim of the study was to explore the genome-wide extent of the rhythmic expression patterns governed by the internal oscillator present in these plants. In this experiment, 12 samples were collected from *Arabidopsis thaliana* seedlings that were placed under a 12 h light / 12 h dark cycles regime. Every 4 hours 2 samples were taken and hybridized to the Affymetrix AtTile 1.0F and 1.0R tiling arrays. The experiment is described in more detail in Hazen et al. (2009).

Figure 4.5 shows an example of the model fit for gene *AT2G46830* with a strong circadian effect. This gene has been previously described and is known under the name *CIRCADIAN CLOCK ASSOCIATED1 (CCA1)*. Besides the circadian effects, no other time-dependent effects are considered in the model. Therefore, the fitted $\log_2$ intensities for time points at identical moments in the 24h day/night cycle coincide. This strong circadian effect is confirmed by Figure 4.6, which shows the fitted effect close to gene *CCA1*. This effect corresponds with the amplitude of the circadian rhythm as estimated by the model, i.e. $\hat{A}(t) = \sqrt{\hat{\beta}_2^2(t) + \hat{\beta}_3^2(t)}$.

The performance of the wavelet-based method for circadian rhythms is further tested by examining previously annotated and circadian clock associated genes on the forward strand of the *Arabidopsis thaliana* genome (see Gardner et al., 2006; Hazen et al., 2009). The results are shown in Table 4.2. Except for *TIME FOR COFFEE* (*AT3G22380*), a considerable overlap is found between all these clock-associated genes and the genomic regions for which a circadian effect is detected significantly above the threshold value $\log_2(1.1)$. They also have a quite large maximum estimated effect or amplitude size, except *TIME FOR COFFEE* and *ZEITLUPE* (*AT5G57360*). These latter two genes are the only genes from the list that are not among the top 20 genes with the strongest estimated circadian effect for the chromosome on which they are located. The gene *TIME FOR COFFEE* is known as a clock gene that does not cycle at

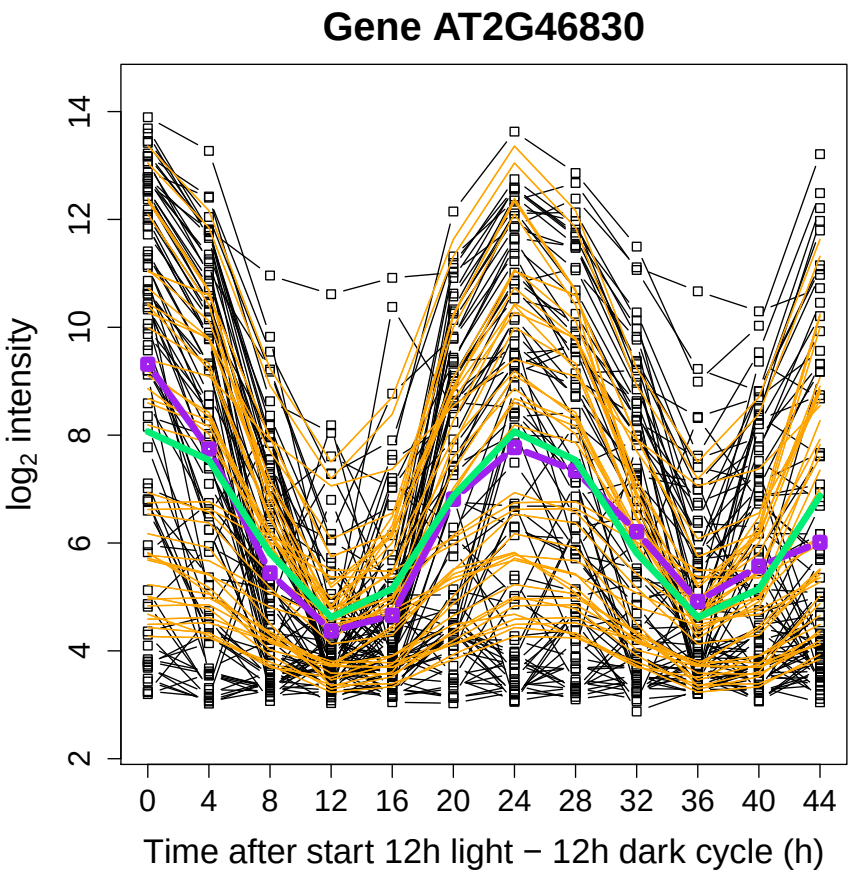


Figure 4.5: Example plot for gene *AT2G46830*, better known as *CIRCADIAN CLOCK ASSOCIATED1*, showing a clear circadian rhythm effect of the mean $\log_2$ intensity level over the 48h time course. The dotted black lines represent the observed $\log_2$ expression for the probes at the different time points. The dotted purple line is the mean observed $\log_2$ expression over all the probes in the region. The orange lines are the probe-wise fitted $\log_2$ expression values, while the green line gives the corresponding mean fitted $\log_2$ expression values at the different time points over all the probes in the region.

the transcriptional level (Ding et al., 2007). Hence, it is as expected that the overlap between detected region and gene annotation is very small, as is the effect size. The gene *ZEITLUPE* is reported as having weak rhythms at the transcriptional level (Gardner et al., 2006). This is confirmed by the small maximum effect size, while still showing a considerable overlap of the significant region with the existing annotation. Hence, the results of Table 4.2 are in line with existing literature.

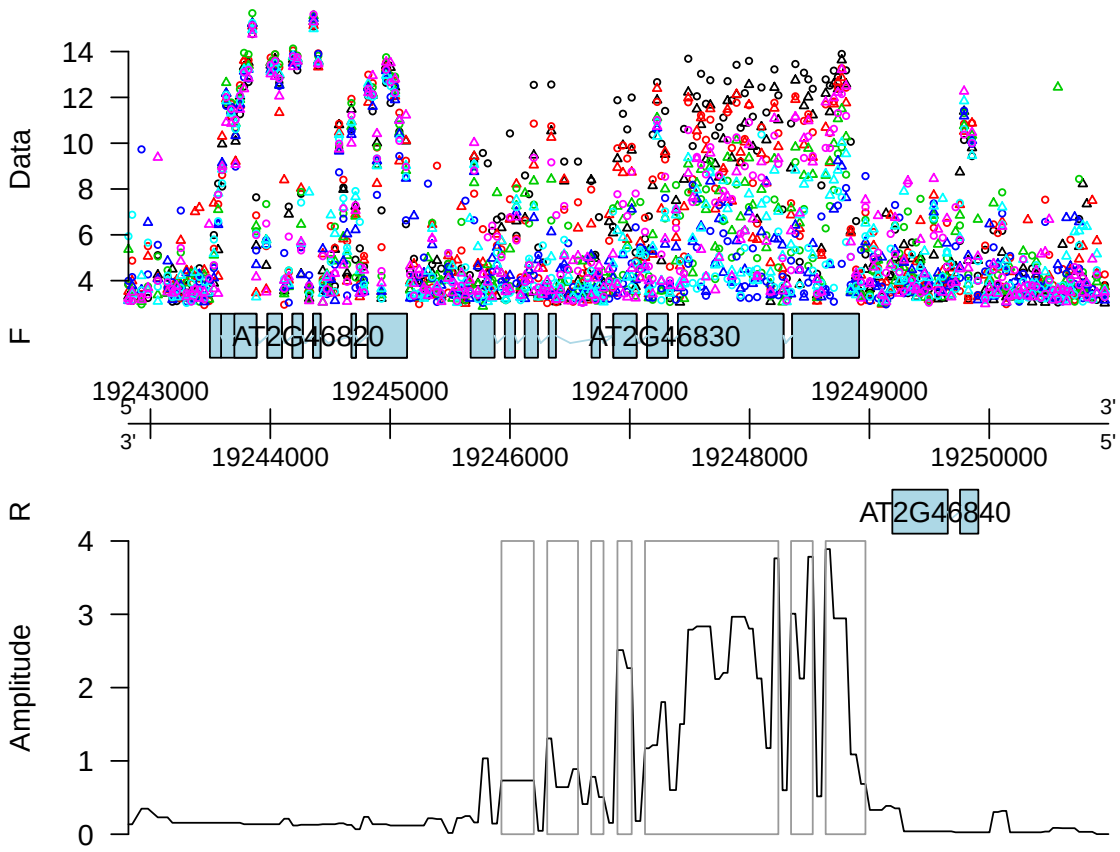


Figure 4.6: Fitted circadian effect for the genomic region of gene *AT2G46830* on the forward strand of chromosome 2. On the vertical axis the amplitude of the circadian rhythm $\hat{A}(t) = \sqrt{\hat{\beta}_2^2(t) + \hat{\beta}_3^2(t)}$ is given. The grey rectangles indicate the regions showing a significant circadian effect ($FDR = 0.05$). The two replicates are indicated by $\circ$ and $\triangle$, while the samples in the 12 h light / 12 h dark cycles regime are represented by different colors.

4.2.3 Case study 3: Non-orthogonal two-factor design

The third data set is used to illustrate the analysis of a tiling array experiment with a two-factor design. The data are taken from a study of the genome-wide analysis of endogenous abscisic acid (ABA)-mediated transcription in dry and imbibed seeds of *Arabidopsis thaliana* (Okamoto et al., 2010). ABA is a phytohormone that is important for the induction and maintenance of seed dormancy. To understand how endogenous ABA regulates the transcriptome in seeds, whole-genome expression analyses were conducted in two ABA metabolism mutants, an ABA-deficient mutant (*aba2*) and an ABA over-accumulation mutant (*cyp707a1a2a3* triple mutant), and compared to a wild type. This is the first factor in the design. Since endogenous levels of ABA often change drastically during seed imbibition (Okamoto et al., 2010), these experiments were done both for dry and for 24-h imbibed seeds. This is the second factor in the design. For

Table 4.2: Analysis results for 8 circadian clock associated genes and for *TIME FOR COFFEE*, a clock gene that does not cycle at the transcriptional level. *Overlap* indicates the proportion of overlap between the regions detected by the wavelet-based method and the gene annotation; *Max. Eff.* gives the maximum estimated effect or amplitude size for this gene; *Top 20* indicates whether the gene is within the top 20 genes with the strongest circadian effect for the chromosome on which the gene is located, as estimated by the wavelet-based model.

Gene ID	Name	Overlap	Max. Eff.	Top 20
AT1G22770	*GIGANTEA*	0.529	2.28	yes
AT1G68050	*FLAVIN-BINDING KELCH DFB PROTEIN1*	0.867	2.90	yes
AT2G25930	*EARLY FLOWERING3*	0.562	1.46	yes
AT2G46790	*PSEUDO RESPONSE REGULATOR9*	0.473	1.38	yes
AT2G46830	*CIRCADIAN CLOCK ASSOCIATED1*	0.867	3.89	yes
AT3G22380	*TIME FOR COFFEE*	0.040	0.06	no
AT3G46640	*LUX ARRHYTHMO*	0.717	1.69	yes
AT5G57360	*ZEITLUPE*	0.350	0.41	no
AT5G61380	*TIMING OF CAB2 EXPRESSION1*	0.797	1.74	yes

each design point, three biological replicates were hybridized using the Affymetrix AtTile 1.0F and 1.0R tiling arrays, resulting in 18 samples.

For this example, Model (4.1) becomes

$$\begin{aligned} Y_i(t) &= \beta_0(t) + \beta_1(t)\, imbibed + \beta_2(t)\, mutant1 + \beta_3(t)\, mutant2 + \\ &\quad \beta_4(t)\, imbibed * mutant1 + \beta_5(t)\, imbibed * mutant2 + E_i(t), \end{aligned} \tag{4.15}$$

where $imbibed = 1$ if the seed was imbibed and $imbibed = 0$ if the seed was dry, $mutant1 = 1$ for the *aba2*-mutant and $mutant1 = 0$ otherwise, and $mutant2 = 1$ for the *cyp707a1a2a3* triple mutant and $mutant2 = 0$ otherwise.

	$\hat{\beta}_{0,gene}$	$\hat{\beta}_{1,gene}$	$\hat{\beta}_{2,gene}$	$\hat{\beta}_{3,gene}$	$\hat{\beta}_{4,gene}$	$\hat{\beta}_{5,gene}$
AT1G69530	4.76	8.70	3.98	-0.82	-4.34	-7.09
AT1G61520	4.27	0.13	0.72	0.13	5.11	-0.44

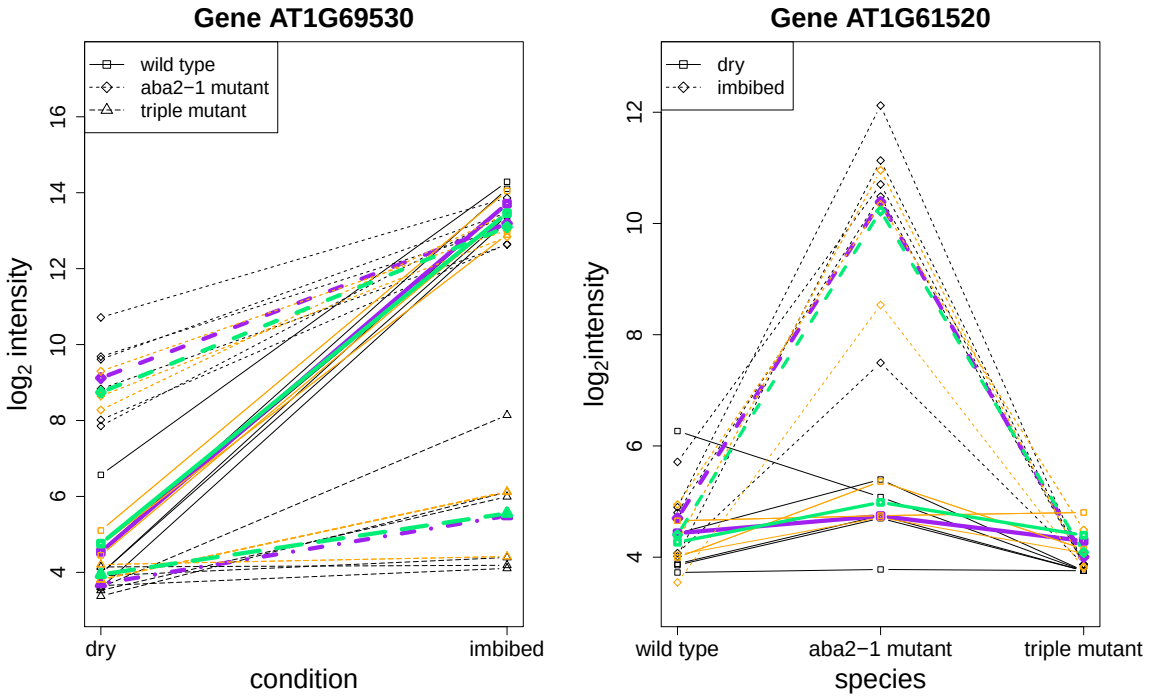
Table 4.3: Gene-wise mean parameter estimates in the two-factor model for genes *AT1G69530* and *AT1G61520*.

This model specification implies that the design matrix $\boldsymbol{X}$ used for this model is

$$\boldsymbol{X} = \begin{pmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} \end{pmatrix}.$$

Column 1 of $\boldsymbol{X}$ corresponds with an overall mean expression level over all samples. The main imbibition effect is coded in column 2. Note that this corresponds with the imbibition effect for wild types, which is the reference species. Columns 3 and 4 are associated with the main ABA mutation effects, whereas columns 5 and 6 allow to examine an interaction effect between imbibition and ABA mutation statuses. Table 4.3 gives the associated gene-wise mean parameter estimates for these genes. Figure 4.7 shows two examples of the model fits for the genes *AT1G69530* and *AT1G61520* on the forward strand of chromosome 1. The left panel plot of Figure 4.7 suggests a larger mean expression level of gene *AT1G69530* for imbibed seeds as compared to dry seeds. The estimated increase in mean expression level, however, is larger for wild types than for ABA-related mutants. The increase in mean expression level between imbibed seeds compared to dry seeds is estimated as $\hat{\beta}_{1,gene} = 8.70$ for wild types, while for *aba2* mutants this increase is estimated as $\hat{\beta}_{1,gene} + \hat{\beta}_{4,gene} = 4.36$ and for *cyp707a1a2a3* triple mutants as $\hat{\beta}_{1,gene} + \hat{\beta}_{5,gene} = 1.61$. In the right panel of Figure 4.7 we see an increased estimated mean expression level of gene *AT1G61520* for *aba2* mutants as compared to wild types and *cyp707a1a2a3* triple mutants. In addition, this increase is much stronger for imbibed seeds.

Figure 4.7: Interaction plots for genes *AT1G69530* and *AT1G61520*. The black lines represent the observed $\log_2$ expression for the probes at the different combinations of the two factor levels. The dotted purple line represents the mean observed $\log_2$ expression over all the probes in the region. The orange lines are the probe-wise fitted $\log_2$ expression values, while the green line gives the corresponding mean fitted $\log_2$ expression values for all the probes in the region.



4.3 Conclusion

In this chapter we have described the extension of the wavelet-based functional model for transcriptome analysis towards more complex experimental set-ups. By appropriate adaptations of the basic model design matrix it becomes possible to easily analyze the transcriptome for single-factor experiments with more than two biological conditions, to detect linear and quadratic time effects or a circadian rhythm effect in time-course experiments, and to handle two- or multiple-factor studies. The use of the model has been illustrated on three case studies on the reference plant *Arabidopsis thaliana*. These cases have shown the potential of the method to cope with a multitude of study designs and associated specific research questions, while still providing reliable results.

Appendix A: The common smoothing parameter estimator $\hat{\tau}(j,k)$

Reconsider the marginal likelihood (3.28) corresponding to the WavNorm model described in Chapter 3,

$$p\left\{\boldsymbol{D}|\boldsymbol{\tau},\boldsymbol{\sigma}_\epsilon^2\right\} \propto \prod_{j=0}^{J}\prod_{k=1}^{K_j}\left|\boldsymbol{V}(j,k)\sigma_\epsilon^2(j,k)\right|^{-1/2} \times \exp\left[-\frac{\boldsymbol{D}^T(j,k)\boldsymbol{V}^{-1}\boldsymbol{D}(j,k)}{2\sigma_\epsilon^2(j,k)}\right].$$

Suppose that $\boldsymbol{X'}$ is an $N \times q$ orthonormal design matrix: $\boldsymbol{X'}^T\boldsymbol{X'} = \boldsymbol{I}_q$, and let $\tau(j,k) = \tau_1(j,k) = \tau_2(j,k) = \ldots = \tau_q(j,k)$ be the common smoothing parameter.

The expressions for $\boldsymbol{V}(j,k)$, $|\boldsymbol{V}(j,k)|$ and $\boldsymbol{V}^{-1}(j,k)$, given in (3.29), (3.30) and (3.31), respectively, now become

$$\begin{aligned}
\boldsymbol{V}(j,k) &= \boldsymbol{I}_N + \tau(j,k)\boldsymbol{X'}\boldsymbol{X'}^T \\
|\boldsymbol{V}(j,k)| &= (1+\tau(j,k))^q \\
\boldsymbol{V}^{-1}(j,k) &= \boldsymbol{I}_N - \frac{\boldsymbol{X'}\boldsymbol{X'}^T}{1+1/\tau(j,k)}.
\end{aligned}$$

The $-2\times$ log-likelihood can now be written as

$$\begin{aligned}
-2 \times l(\boldsymbol{D}|\boldsymbol{\tau},\boldsymbol{\sigma}_\epsilon^2) \propto \sum_{j=0}^{J}\sum_{k=1}^{K_j}\Bigg(& q\log\left[1+\tau(j,k)\right] + N\log\left[\sigma_\epsilon^2(j,k)\right] + \\
& \frac{1}{\sigma_\epsilon^2(j,k)}\boldsymbol{D}^T(j,k)\left[\boldsymbol{I}_N - \frac{\boldsymbol{X'}\boldsymbol{X'}^T}{1+1/\tau(j,k)}\right]\boldsymbol{D}(j,k)\Bigg).
\end{aligned}$$

The smoothing parameter $\tau(j,k)$, given $\sigma_\epsilon^2(j,k)$, can be estimated as the solution of

$$\frac{\partial l(\boldsymbol{D}|\boldsymbol{\tau},\boldsymbol{\sigma}_\epsilon^2)}{\partial\tau(j,k)} = 0.$$

Hence,

$$\begin{aligned}
\frac{q}{\tau(j,k)+1} - \frac{\boldsymbol{D}^T(j,k)\boldsymbol{X'}\boldsymbol{X'}^T\boldsymbol{D}(j,k)}{\left(1+1/\tau(j,k)\right)^2\tau(j,k)^2\sigma_\epsilon(j,k)} &= 0 \\
q\tau(j,k)\sigma_\epsilon^2(j,k) + q\sigma_\epsilon^2(j,k) - \boldsymbol{D}^T(j,k)\boldsymbol{X'}\boldsymbol{X'}\boldsymbol{D}(j,k) &= 0
\end{aligned}$$

$$\tau(j,k) = \frac{\boldsymbol{D}^T(j,k)\boldsymbol{X'}\boldsymbol{X'}^T\boldsymbol{D}(j,k)}{q\sigma_\epsilon^2(j,k)} - 1.$$

By definition $\tau(j,k) \in [0,\infty]$. Hence, the MML estimator becomes

$$\hat{\tau}(j,k) = \left[\frac{\boldsymbol{D}^T(j,k)\boldsymbol{X'}\boldsymbol{X'}^T\boldsymbol{D}(j,k)}{q\sigma_\epsilon^2(j,k)} - 1\right]_+.$$

Chapter 5

waveTiling: a Bioconductor package for wavelet-based tiling array transcriptome analysis

In this chapter we present a user-friendly software implementation of the wavelet-based transcriptome analysis for tiling arrays that was described in Chapters 3 and 4. The software is provided as a Bioconductor add-on package, called waveTiling. Bioconductor (Gentleman et al., 2004) is an open source, open development software project to provide tools for the analysis and comprehension of high-throughput genomic data. It is based primarily on the R programming language (R Development Core Team, 2012). Currently, waveTiling provides a standard analysis flow for transcriptome analysis on single-factor experiments with two or more biological conditions, the detection of linear and quadratic effects and circadian rhythms in time-course experiments, and the analysis of two-factor experiments. Furthermore, more experienced users can also specify customized designs.The package also generates along-genome plots and contains functions to easily extract the detected genes and unannotated regions. Where possible, waveTiling uses the standard Bioconductor S4-class data structures, making it fully compatible with existing Bioconductor packages. These S4-classes are an essential part of the object-oriented programming system in R. A good introduction on this topic can be found in Gentleman (2008). Figure 5.1 gives a general overview of the waveTiling package. In the following sections the structure and the main functionalities of the package are explained in more detail. Section 5.1 provides a description of how to import and preprocess the raw in-

tensity data. In Section 5.2 the implementation of the main fitting and inference functions is explained. Finally, Section 5.3 shows the main analysis output options of the package. Note that it is not our intention to provide an exhaustive list of all the package's functions with their individual arguments in this chapter. To this end, the package's manual and help files may be consulted. Currently, the package is freely available from `http://bioconductor.org/packages/release/bioc/html/waveTiling.html`. Throughout the chapter all steps in the standard analysis flow are illustrated based on the leaf development case study that was discussed in Section 4.2.1.

5.1 Importing and preprocessing raw intensity data

A typical data analysis of Affymetrix microarrays or tiling arrays starts from the CEL-files which contain the raw intensities. One CEL-file corresponds to the intensities from one array. In the waveTiling package the CEL-files are imported with the aid of the oligo package (Carvalho and Irizarry, 2010). When reading in the CEL-files an array design information file has to be provided. The design information is needed to map the XY locations of the probes onto the array to the exact genomic positions of the organism under study, in our case *Arabidopsis thaliana*. By using the pdInfoBuilder (Falcon and Carvalho, 2012) package a custom array design package can be created based on the array design information file. This package is called pd.atdschip.tiling in our example. Importing the CEL-files with the oligo package results in a *TilingFeatureSet*-class object. The *TilingFeatureSet*-class is specifically designed for representing tiling array data and in turn extends the *ExpressionSet*-class which is commonly used in Bioconductor to store general microarray expression data. Existing instance methods from oligo and other Bioconductor packages that support this structure are therefore applicable as well. The *TilingFeatureSet*-class is extended in waveTiling to the *WaveTilingFeatureSet*-class, which is used as input for the wavelet-based transcriptome analysis. Phenotypic data like the number of different treatment groups, the group names and the number of replicates within in each group can be added to an object of *WaveTilingFeatureSet* using the addPheno() function. This information is used later on in the analysis. Code 5.1 illustrates the import of the data for our case study.

Before starting the transcriptome analysis, the probes that map to multiple genomic locations are

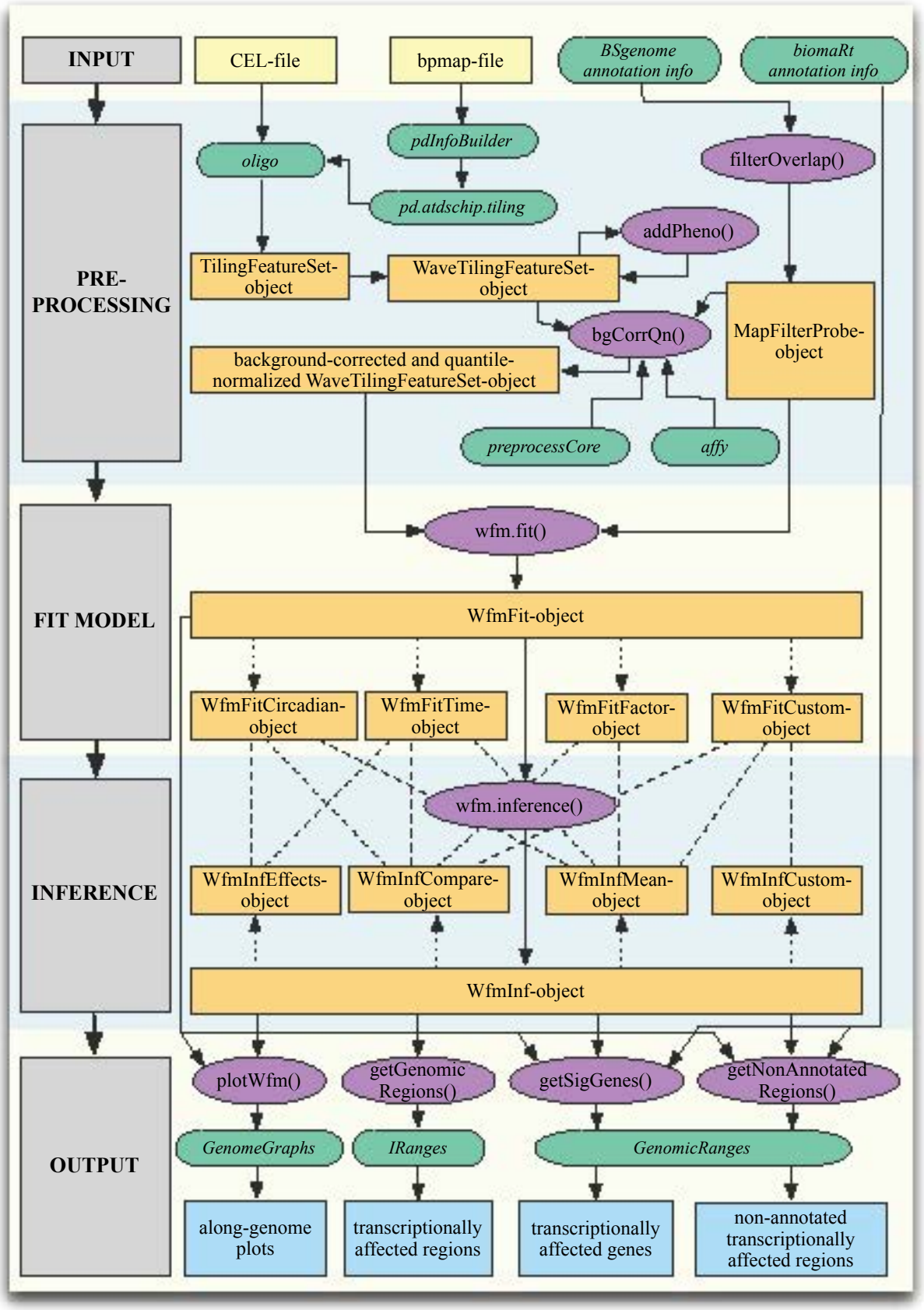


Figure 5.1: Flowchart of the waveTiling package. The main steps in the data analysis flow are indicated by the grey blocks. External input files are in yellow; external Bioconductor help packages are in green; waveTiling S4-classes are in orange; the main functions are in purple; the output is in blue.

```
> library(pd.atdschip.tiling) # load matrix design package
> library(oligo) # load oligo package
> leafdev <- cel2TilingFeatureSet(dataPath="/data/tiling/leafdevelopment",
    annotationPackage="pd.atdschip.tiling") # read in CEL files
```

Code 5.1: Importing raw intensity data for the leaf development study

filtered using filterOverlap(). This is needed to reduce cross-hybridization effects. For instance, probes (PM or MM) corresponding to a particular genomic location (forward or reverse strand) may have the same sequence as probes (PM or MM) corresponding to a different genomic location (forward or reverse strand). This filterOverlap() function can also be used if the probes have to be remapped to another version of the genome sequence as the version used for the array design. In the case study, the probes on the AGRONOMICS1 array are build based on the TAIR8 genome sequence and they are remapped onto the more recent TAIR9 genome sequence. The function needs an argument BSgenomeObject available from loading the appropriate BSgenome package (Pagès, 2012). The output is an object of class *MapFilterProbe*. Code 5.2 shows the implementation.

```
> library(BSgenome.Athaliana.TAIR.TAIR9) # load BSgenome package
> leafdevMapAndFilterTAIR9 <- filterOverlap(leafdev,remap=TRUE,
    BSgenomeObject=Athaliana,chrId=1:5,
    strand="both",MM=FALSE) # filter and remap probes
> leafdevMapAndFilterTAIR9 # show MapFilterProbe-class object
Remapped and filtered probe information
No. of filtered probes: 5894070
```

Code 5.2: Filtering redundant probes and remapping probes onto recent annotation for the leaf development study

After filtering and/or remapping, the expression data are background-corrected and quantile-normalized by the bgCorrQn() function. This function makes use of bg.adjust() from the affy package (Gautier et al., 2004) and normalize.quantiles() from the preprocessCore package (Bolstad, 2012). The bgCorrQn() function also uses a *MapFilterProbe*-class object as input to make sure only the filtered probes contribute to the background correction and normalization steps. This preprocessing step is illustrated in Code 5.3.

```
> leafdev <- as(leafdev,"WaveTilingFeatureSet") # change to WaveTilingFeatureSet
> leafdev <- addPheno(leafdev,noGroups=6,
    groupNames=c("day8","day9","day10","day11","day12","day13"),
    replics=rep(3,6)) # add phenotypic info
> leafdevBQ <- bgCorrQn(leafdev,useMapFilter=leafdevMapAndFilterTAIR9)
# preprocess raw intensities
> leafdevBQ # show WaveTilingFeatureSet-class object
WaveTilingFeatureSet (storageMode: lockedEnvironment)
assayData: 5894070 features, 18 samples
 element names: exprs
protocolData
 rowNames: caquinof_20091023_S100_v4.CEL caquinof_20091023_S101_v4.CEL
  ... caquinof_20091023_S117_v4.CEL (18 total)
 varLabels: exprs dates
 varMetadata: labelDescription channel
phenoData
 rowNames: day8.1 day8.2 ... day13.3 (18 total)
 varLabels: group replicate
 varMetadata: labelDescription
featureData: none
experimentData: use 'experimentData(object)'
Annotation: pd.atdschip.tiling
```

Code 5.3: Preprocessing raw intensity data for the leaf development study

5.2 Wavelet-based transcriptome analysis

The background-corrected and quantile-normalized *WaveTilingFeatureSet*-class object is used as input for the wavelet-based transcriptome analysis. The analysis can be conducted for a complete chromosome or for a specific genomic region on the chromosome. The wavelet-based model is fitted to the expression data with the wfm.fit() function, leading to a *WfmFit*-class object. Depending on the design of the study a *WfmFitFactor* (factorial design), *WfmFitTime* (time-course design), *WfmFitCircadian* (circadian rhythm design) or *WfmCustom* (custom

design) subclass is used. A normal prior distribution can be imposed on the wavelet coefficients associated with the effect functions like in Equation (3.26) with the function argument prior="normal" . If one wants to obtain additional smoothing, a Jeffrey's hyperprior can also be put on the smoothing parameters by prior="improper". For the error variance either the marginal maximum likelihood estimator (3.39) (var.eps="margLik") or the MAD estimator (var.eps="mad") can be used. Part of the code for fitting the model is implemented in C to speed up computation. Fitting the wavelet-based model in waveTiling for genomic position 22000000 to 24000000 on the forward strand of chromosome 1 in our case study is illustrated in Code 5.4. The leaf development study is an example of a study with a time-course design, hence we choose design="time". The wavelet decomposition goes down to wavelet level n.levels=10.

```
> leafdevFit <- wfm.fit(leafdevBQ,filter.overlap=leafdevMapAndFilterTAIR9,
    design="time",n.levels=10,chromosome=1,strand="forward",
    var.eps="margLik",prior="improper",minPos=22000000,
      maxPos=24000000,skiplevels=1,save.obs="plot",trace=TRUE)
> leafdevFit
Fitted object from wavelet based functional model - Time Design
Wavelet filter used: haar
Number of wavelet decomposition levels: 10
Number of probes used for estimation: 52224

Genome Info :
    Chromosome: 1
    Strand: forward
    Minimum probe position: 22000000
    Maximum probe position: 23988867
```

Code 5.4: Fitting the wavelet-based model for the leaf development study

In the second step of the transcriptome analysis different inference procedures can be conducted corresponding to the particular research question. This is done with the wfm.inference() function. The type of inference procedure depends on the *WfmFit*-subclass. The results are stored as a *WfmInf*-class object. Also for this class there are four subclasses. The *WfmInfCompare*-class contains the results of a pairwise comparison between two groups or time points. The

results of a trancript discovery analysis for each individual group or time point are stored in a *WfmInfMeans*-class object. These two types of inference can be obtained from all four *WfmFit*-subclasses. The *WfmInfEffects*-class contains results with linear or quadratic time effects, obtained from a *WfmFitTime*-class object, or with circadian rhythm effects, obtained from a *WfmFitCircadian*-class object. Finally, custom inference results are stored in a *WfmInfCustom*-class object and are obtained from a *WfmFitCustom*-class object. The type of inference is indicated in the wfm.inference() function by the contrasts argument. In the case study a pairwise comparison between any combination of recorded days in the leaf development is performed. Hence, we set contrasts="compare". With the delta argument the threshold value to calculate the empirical Bayesian FDRs is chosen. The inference procedure for pairwise comparisons is illustrated in Code 5.5.

```
> delta <- log(1.2,2) # set threshold for differential expression
> leafdevInfCompare <- wfm.inference(leafdevFit,contrasts="compare",
    delta=c("median",delta))
> leafdevInfCompare
Inference object from wavelet based functional model - Pairwise Comparison
Genome Info :
    Chromosome: 1
    Strand: forward
    Minimum probe position: 22000000
    Maximum probe position: 23988867
```

Code 5.5: Pairwise comparison for the leaf development study

5.3 Results output

All transcriptionally affected regions can be extracted from the *WfmInf*-class objects using the getGenomicRegions() accessor. They are stored as a list of *IRanges*-class objects (Pagès et al., 2012). This class is designed to efficiently represent and handle sequences and ranges from indices along those sequences. It gives the start and end position and the nucleotide length of each significant region. For the leaf development case study the list contains 16 of those *IRanges*-class elements. The first element always gives the significant regions for transcript

discovery based on the mean expression over all arrays. Elements 2 to 16 give the differentially expressed regions for any pair of contrasts between different time points. The order is always $2-1, 3-1, 3-2, 4-1, \ldots$. The output for the differentially expressed regions between day 9 (time point 2) and day 8 (time point 1) in the leaf development study is displayed in Code 5.6.

```
> sigGenomeRegionsCompare <- getGenomicRegions(leafdevInfCompare)
> sigGenomeRegionsCompare[[2]]  # time point 2 − time point 1
IRanges of length 23
          start       end   width
[1]   22448608  22448704    97
[2]   22700160  22700256    97
[3]   22804928  22805024    97
...         ...       ...   ...
[22] 23619042  23619138    97
[23] 23953826  23953986   161
```

Code 5.6: Differentially expressed regions between day 9 and day 8 in leaf development study

The waveTiling package also provides two additional accessor functions: (1) getSigGenes() to extract significantly affected genes and (2) getNonAnnotatedRegions() to extract significantly affected non-annotated regions. These accessor functions can be applied if the appropriate annotation info containing gene identifiers is available for the organism under study. For this purpose, we make use of the biomaRt package (Durinck et al., 2005, 2009b). This package offers a convenient interface to many publicly available biological data repositories. Both functions output a list of *GRanges*-class objects created by the GenomicRanges package (Aboyoun et al., 2012). Basically, the *GRanges*-class is an extension of the *IRanges*-class that can handle the additional storage of genomic info accompanying the ranges of sequences. An example of the use of these functions is shown in Codes 5.7 and 5.8.

A visual representation of the significant regions can be made with the plotWfm() function. This function needs both the *WfmFit*- and *WfmInf*-class objects of the analysis as input as well as the annotation info obtained with the biomaRt package. The plot function makes use of the implementations in the GenomeGraphs package (Durinck et al., 2009a). As an example, the code used to create Figure 4.1 is given in Code 5.9.

```
> library(biomaRt)
> library(TxDb.Athaliana.BioMart.plantsmart12)  # load annotation package
> sigGenesCompare <- getSigGenes(fit=leafdevFit,inf=leafdevInfCompare,
    biomartObj=TxDb.Athaliana.BioMart.plantsmart12)
> sigGenesCompare[[2]]
GRanges with 31 ranges and 6 elementMetadata cols:
     seqnames        ranges        strand |     tx_id      tx_name       regNo
       <Rle>     <IRanges>     <Rle> | <integer> <character> <integer>
  [1]        1 [22447848, 22449526]      - |     14091 AT1G60970.1         1
  [2]        1 [22699715, 22701169]      + |     27751 AT1G61520.3         2
...
 [31]        1 [23953233, 23954492]      + |     36016 AT1G64500.1        23
     percOverGene percOverReg totPercOverGene
        <numeric>   <numeric>      <numeric>
  [1]     5.777248         100       5.777248
  [2]     6.666667         100       6.666667
...
 [31]    12.777778         100      12.777778
```

Code 5.7: Extracting significant genes in waveTiling package for the leaf development study

```
> nonAnnoCompare <- getNonAnnotatedRegions(fit=leafdevFit,inf=leafdevInfCompare,
    biomartObj=TxDb.Athaliana.BioMart.plantsmart12)
> nonAnnoCompare
GRanges with 834 ranges and 0 elementMetadata cols:
     seqnames            ranges        strand
       <Rle>         <IRanges>     <Rle>
  [1]        1 [22001344, 22001440]      +
  [2]        1 [22004577, 22004801]      +
...
 [834]       1 [23978594, 23978754]      +
```

Code 5.8: Extracting significant non-annotated regions in waveTiling package for the leaf development study

```
> trs <- transcripts(TxDb.Athaliana.BioMart.plantsmart12)
> sel <- trs[elementMetadata(trs)$tx_name %in% "AT1G04350",]
> start <- start(ranges(sel))-2500
> end <- end(ranges(sel))+2500
> plotWfm(fit=leafdevFit,inf=leafdevInfCompare,
    biomartObj=TxDb.Athaliana.BioMart.plantsmart12,
    minPos=start,maxPos=end,two.strand=TRUE,
    plotData=TRUE,plotMean=FALSE,tracks=c(1,2,6,10,11))
```

Code 5.9: Plotting significant regions with waveTiling for the leaf development study

5.4 Conclusion

The wavelet-based methods for transcriptome analysis with tiling arrays described in Chapters 3 and 4 have been implemented as a user-friendly and freely available R/Bioconductor software package, called waveTiling. The main structure and functionalities of the package have been shown. Based on a case study of leaf development in *Arabidopsis thaliana*, the full standard analysis flow of waveTiling has been illustrated.

Chapter 6

Discussion, conclusions and future research perspectives for Part I

6.1 Discussion and conclusions

Tiling array technology is a commonly used tool for whole-genome transcriptome analysis. Unlike classical microarrays, tiling arrays measure transcriptional activity regardless of existing annotation. This occurs by means of more or less equally spaced probes along the genome. Tiling array probe intensities can thus be viewed as realizations of an underlying function for RNA expression. Therefore, a logical choice is to use the functional modeling framework for tiling array data analysis. Even after applying suitable background-correction and array-wise normalization procedures, probe-to-probe fluctuations within the same transcriptional units are still apparent. Hence, modeling the expression signal asks for the use of proper smoothing techniques in order to control the bias-variance trade-off. The sudden jumps at the boundaries of these trancriptional units, however, give the data a discontinuous and spatially heterogeneous nature. For this reason, a wavelet-based denoising approach is taken. The use of wavelets allows an efficient regularization of the expression signal without losing the ability to model local features.

The functional model that we have presented can assess transcript discovery and identify differentially expressed transcripts simultaneously. This is in contrast to existing methods for the analysis of tiling arrays. The model is transformed from the genomic space to the wavelet

space, where the wavelet coefficients of the effect functions undergo thresholding. As a result, these effect functions are adaptively smoothed when transforming the modified coefficients back to the genomic domain. To this end, a Bayesian thresholding framework is adopted in which a normally distributed prior is imposed on the wavelet coefficients of the effect functions. The smoothing and error variance parameters are estimated by a marginal maximum likelihood approach. Because of the decorrelating property of the wavelet transform, computationally efficient analytical solutions of the resulting estimators and their posterior distributions are obtained. An empirical Bayes inference procedure has been proposed, which makes use of these posterior distributions. Both for transcript discovery and differential expression a probe-wise local Bayesian FDR is calculated. This result is associated with a predefined threshold value which enables obtaining transcriptionally affected regions that are statistically significant as well as biologically relevant.

The wavelet-based method with a normal prior imposed on the parameters (WavNorm) was compared in a simulation study to an alternative wavelet-based approach using a multiple shrinkage mixture prior, containing a normal component and a point mass at zero (WavMix). For transcript discovery, both wavelet-based methods were additionally compared to two popular methods described in Kampa et al. (2004) and Huber et al. (2006), while for differential expression also the RMA procedure was included. Tiling array data were simulated based on an adapted model from Purdom et al. (2008) that features additive background, multiplicative noise, probe-specific affinities and serial correlation in the genomic domain. The simulation parameters were tuned by rough estimates of realistic values from real data. For this purpose, expression data from the *Arabidopsis thaliana* E2F case study were used. Using our approach, we have shown that the characteristics of the observed real data could be realistically preserved in the simulated data.

The simulation results have indicated that the wavelet-based approaches outperform the existing methods for transcript discovery in terms of positive predictive value and specificity, while maintaining a high true positive rate. Moreover, the wavelet-based methods are more sensitive than RMA, while keeping the number of false positives small. Both for transcript discovery and differential expression, the discovered regions by the wavelet-based methods correspond well with the underlying exonic structure. The WavNorm-method was found to be very competitive in terms of computation time, while WavMix is computationally less efficient. Moreover,

WavMix seemed to oversmooth too much for our purposes, resulting in a decreased sensitivity, especially for differential expression. In a next step, five different versions of the WavNorm have been compared. The error variance can be allowed to vary with different wavelet scales only, or with both different wavelet scales and different wavelet locations. Alternatively, the MAD estimator can be used. It was shown that, in general, this does not have a great impact on the performance of the method, although the sensitivity decreased slightly when the MAD estimator was used. A bootstrap correction can be applied to account for the extra variability in the posterior distribution of the effect functions, induced by estimating the variance components. However, the variance is mainly underestimated only in regions with small signal, if no correction is applied. Therefore, the bootstrap correction was found not to improve performance in the regions that are of interest. Given the enormously increased computation time, applying a bootstrap correction seems not worthwhile for our purposes. Finally, one might want to impose an additional Jeffrey's hyperprior on the smoothing parameters. This does not have a large influence on the method's performance. It basically leads to slightly more smooth results, particularly for differential expression analysis.

By applying the WavNorm method on the *Arabidopsis thaliana* E2F case study, we have shown the method's use for finding potential targets in whole-genome transcription studies. The probe-wise and functional approach makes the method completely unbiased of existing annotation. Therefore, it exploits tiling array data to their full potential.

As the focus in tiling array studies has recently shifted towards more complex designs than the two-group design, we have extended the applicability of the wavelet-based model accordingly. This basically implies the adaptation of the model design matrix in a way that allows to answer the specific research questions that involve more complex experimental designs. Moreover, the orthogonality of the design matrix must be preserved to ensure analytical solutions with fast computation. We have considered time-course studies, studies with more than two conditions and multiple-factor studies. In regular time-course studies one can be interested in the detection of differentially expressed regions between two different time points, or more directly in significant effects of transcriptional activity over time, such as linear time effects. We have shown that for both cases the design matrix can be adapted by considering a functional relationship of the designed time points described by orthogonal polynomials. If one wants to detect circadian rhythms in the transcriptome of an organism, the circular effect can be modeled by

constructing the design matrix by means of Fourier basis functions. Design matrices for two- or multiple-factor studies are typically non-orthogonal. However, this problem has been tackled by applying a Gram-Schmidt orthogonalization of the design matrix and backtransforming the results to the original predictor space after estimation. A similar empirical Bayes procedure as for the two-group design has been used for inference. This procedure either occurs on the parameters themselves or on a function of the parameters, depending on the study design. The use and flexibility of the extended wavelet-based modeling approach has been illustrated on three case studies with the reference plant *Arabidopsis thaliana*. With these examples we have demonstrated the potential of the method to cope with a multitude of study designs and associated specific research questions, while still providing reliable results.

We have implemented the wavelet-based methods as a user-friendly R/Bioconductor package, called waveTiling. The package provides a standard analysis flow for wavelet-based transcriptome analysis on single-factor experiments with two or more biological conditions, the detection of linear and quadratic effects and circadian rhythms in time-course experiments, and the analysis of two-factor experiments or customized designs. Furthermore, it generates along-genome plots and contains functions to easily extract the transcriptionally affected genes and unannotated regions. Where possible the package uses the standard Bioconductor S4-class data structures making it fully compatible with existing Bioconductor packages. The package also contains help functions and a manual in which the package's functions are explained and illustrated.

6.2 Future research perspectives

The wavelet-based functional model presented in the previous chapters has the potential to be extended or adapted in several different directions. In Section 6.2.1 the integration of preprocessing into the model is discussed. Section 6.2.2 explores the possibilities and challenges of an unsupervised version of the wavelet-based model involving functional principal components analysis. In Section 6.2.3 the extension of the model towards other high-throughput genomics platforms and profiles is considered.

6.2.1 Integration of preprocessing into the model

In Chapter 2 we have argued that we required a certain degree of preprocessing of the raw intensity signal, due to its complexity and the presence of substantial obscuring variability. Conceptually, it could be possible to integrate at least part of the preprocessing of the raw data into the model itself. This would imply that the uncertainties associated with preprocessing would be accounted for by the wavelet-based model itself. Consider the basic functional model for the two-group design in the genomic domain (3.1). Suppose now that one has available another N_0 arrays that are hybridized to a DNA reference, besides the N_1 and N_2 arrays with RNA expression intensities for the two experimental conditions. In theory, all features within each array should exhibit the same intensity on this DNA reference array because the same copy number of genomic DNA is hybridized throughout the genomic coordinate. In practice, however, large differences in the measured intensities are observed. Although some of the variation can be explained by stochastic noise, the major part of the variation is due to differences in probe affinity (e.g. Wu et al., 2004). The information of the DNA reference hybridizations can be easily incorporated in the model, which is now given by

$$Y_i(t) = \beta_0(t) + X_{1,i}\beta_1(t) + X_{2,i}\beta_2(t) + E_i(t), \tag{6.1}$$

with $i = 1, ..., N$, $N = N_0 + N_1 + N_2$, $\beta_0(t)$ a function that is related to the probe affinities derived from the DNA reference hybridizations, $\beta_1(t)$ the mean function that is used for transcript discovery, $X_{1,i}$ a dummy variable which is 1 for the C_1 and C_2 arrays, and 0 for the reference DNA arrays, $\beta_2(t)$ the difference function, and $X_{2,i}$ a dummy variable which is 1 for C_1, -1 for C_2 and 0 for the reference DNA array. By including $\beta_0(t)$ in the model, the mean function $\beta_1(t)$ gets the interpretation of a $\log_2$ fold change with respect to the average intensities of the DNA reference hybridizations. Hence, DNA reference normalization is done automatically during the parameter estimation for Model (6.1). For balanced designs of the C_1 and C_2 arrays the use of the ($-$1,1,0) coding for $X_{2,i}$ implies an estimation orthogonality between $\beta_2(t)$ and the other two functions.

6.2.2 Functional principal components analysis

Suppose that genomic profiles of expression data from different independent samples are available, and that one is primarily interested in exploring the main sources of variability in this

data set, without prior knowledge of any possible groups or classes the samples might belong to. As the data are functional in nature, an obvious approach would be to conduct a functional principal components analysis (PCA) on the data set.

Consider the following model in the genomic data space:

$$\boldsymbol{Y} = \boldsymbol{Z} + \boldsymbol{E}, \tag{6.2}$$

where $\boldsymbol{Y}$ denotes a $N \times T$ matrix of mean-centered expression values observed on an equally spaced grid, $\boldsymbol{t} = (1, ..., T)$, i.e. the mean expression value over the N samples at each genomic location t is equal to zero. Further, $\boldsymbol{Z}$ is a $N \times T$ matrix of functional effects, which can be written as a linear transformation of a $N \times q$ latent matrix $\boldsymbol{X}$, i.e. $\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{L}^T$, where $\boldsymbol{L}$ is a $T \times q$ transformation matrix. The $N \times T$ matrix $\boldsymbol{E}$ contains the error processes as defined in Equation (3.2). By applying the DWT the model can be written in the wavelet space:

$$\boldsymbol{D} = \boldsymbol{Z}^* + \boldsymbol{E}^*. \tag{6.3}$$

PCA can be formulated as a maximum likelihood solution to a latent variable model, an interpretation better known as probabilistic PCA (Tipping and Bishop, 1999). Therefore, Model (6.3) can be applied to perform PCA. The functional effects in the wavelet space, $\boldsymbol{Z}^*$, can also be written as a linear transformation of a $N \times q$ latent variable matrix $\boldsymbol{X}$, i.e. $\boldsymbol{Z}^* = \boldsymbol{X}\boldsymbol{L}^{*T}$. For each individual profile $\boldsymbol{D}_i$ $(i = 1, \ldots, N)$ we have

$$\boldsymbol{D}_i = \boldsymbol{X}_i\boldsymbol{L}^{*T} + \boldsymbol{E}_i^*. \tag{6.4}$$

Similar to the model in Chapter 3, a multivariate normal distribution is assumed for the error terms $\boldsymbol{E}_i^* \sim MVN(\boldsymbol{0}, \sigma_\epsilon^2(j,k)\boldsymbol{I}_T)$. To obtain sparse results a normally distributed prior is put on the loadings $\boldsymbol{L}_m^{*T} \sim MVN(\boldsymbol{0}, c_m^2(j,k)\boldsymbol{I}_T)$, $m = 1, \ldots, q$. Furthermore, the score vectors are also assumed multivariate normally distributed, $\boldsymbol{X}_i \sim MVN(\boldsymbol{0}, \boldsymbol{I}_q)$. Due to the latter assumption the posterior distribution $p\{\boldsymbol{X}, \boldsymbol{L}^*|\boldsymbol{D}\}$ is not tractable. This problem can be solved by applying variational Bayes methods to approximate the posterior distribution by a computationally tractable trial distribution (e.g. Guan and Dy, 2009). This can be done by means of the iterated conditional modes algorithm (Bishop, 2006). Since this algorithm is computationally intensive, Nakajima et al. (2010, 2011) have proposed an efficient analytical solution that can possibly be used here.

6.2.3 Extensions to other platforms and 'omics profiles

The advent of next-generation sequencing technologies enables researchers to assess genome-wide profiles of the transcriptome at an unprecedented resolution. Therefore, a logical next research step is to examine whether the wavelet-based modeling framework designed for transcriptome data from tiling arrays can be adapted to work for transcriptome analysis with RNA-seq as well. In RNA-seq a count of nucleotide sequence reads acts as a proxy for the underlying concentration of gene expression products. This would imply the extension of the Gaussian wavelet-based functional model presented in Chapter 3 towards count regression. Due to biological variation RNA-seq data is usually overdispersed with respect to the Poisson distribution. However, this variation could be captured by introducing profile-specific random effect functions for each biological replicate. This functional mixed model can be represented within the generalized linear mixed model framework, which would allow the model parameters to be estimated by penalized quasi-likelihood. More details about this estimation approach can be found, for instance, in Ruppert et al. (2003).

Besides transcriptome profiles many other fine-resolution 'omics profiles can be measured on high-throughput platforms, such as copy number variation, epigenomes and proteomes. It would be interesting to explore whether the ideas from wavelet-based adaptive regularization within the functional modeling framework could be ported towards these other profiles, each with their own specific error structure and challenges.

Taking it it yet another step further, methods for unraveling the interaction between the genome, transcriptome, epigenome and proteome will bring biomedical research to the next level. As a result, data integration of multiple high-dimensional profiles has become one of the major themes in this area. When different profiles are available for the same subject, methods from functional data analysis can be used for jointly modeling them. Potentially, the combination of complex dependence structures and regularization can be seamlessly integrated within the hierarchical mixed model framework. A possible direction would then be to adopt pseudo-likelihood methods developed for large and complex longitudinal data sets within statistical genomics.

Part II

Statistical methods for 454 high-throughput sequencing data

A selection of the presented work is published in

De Beuf, K., De Schrijver, J., Thas, O., Van Criekinge, W., Irizarry, R.A. and Clement, L. (2012). Improved base-calling and quality scores for 454 sequencing based on a Hurdle Poisson model. *BMC Bioinformatics*, 13:303.

Chapter 7

Introduction to Part II: Statistical methods for 454 high-throughput sequencing data

In this chapter, an introduction to part II of the dissertation is given. This part discusses two statistical problems in the analysis of 454 DNA sequencing data. As these problems are driven to a large extent by the specific nature of the 454 data, a good understanding of the sequencing and data-generating process is very important. Therefore, we start with describing the 454 technology in Section 7.1. Section 7.2 introduces a typical 454 data set used to motivate the research. Finally, the objectives and outline for this part of the dissertation are given in Section 7.3.

7.1 Roche/454 technology

In the last eight years the advent and emergence of next-generation sequencing (NGS) technologies have revolutionalized biological and biomedical research. In the past, many efforts have been made to optimize DNA sequencing by Sanger chemistry (Sanger et al., 1977). However, this traditional method only allows a limited level of parallellization. Therefore, it still appears to be too costly and time-consuming for many research aims to be accomplished in practice (Shendure and Ji, 2008). NGS technologies, on the other hand, do have the ability to produce a large volume of data quite cheaply by sequencing in parallel. In the last couple of years, multiple sequencing platforms have become commercially available. One of the prominent players

and the first NGS technology to reach the market in 2004 is the *Roche/454 sequencing* platform (Margulies et al., 2005; Rothberg and Leamon, 2008). This will be the platform of study in this part of the dissertation. Similar to other NGS technologies, sequencing on the 454 platform consists of three main steps: (1) *DNA library or template preparation*, (2) *amplification* of DNA templates with polymerase chain reaction (PCR), and (3) *sequencing-by-synthesis* with the 454-specific *pyrosequencing* reaction. In the following paragraphs the different steps of the 454 sequencing process are discussed. Additional details can be found in e.g. Mardis (2008); Shendure and Ji (2008); Metzker (2010); Ledergerber and Dessimoz (2011).

The first step in 454 sequencing is the library or template preparation (see Figure 7.1). DNA is extracted from the biological sample and is broken into small fragments, also called templates. Subsequently, the DNA fragments are denatured to make them single-stranded. A collection of DNA fragments constitutes the library. The fragments in the library are prepared for sequencing by ligating specific adaptor sequences to both ends of each fragment (panel B of Figure 7.1). They are bound to *beads* that have millions of short sequences attached to them, each of which is complementary to the adaptor sequences on one end of the DNA fragments (panel C of Figure 7.1). The adaptor sequence on the other end of the fragments contains universal primer sites that allow genomes to be amplified with common PCR primers. This occurs on the surfaces of the in total hundreds of thousands of beads by means of emulsion PCR (panel D of Figure 7.1). Hereby, the proper circumstances are created to favor the binding of only one single fragment to each bead, leading to uniform clusters of the same sequence on each bead. This amplification step is needed to increase the intensity signal that will be eventually produced during the sequencing process (see below).

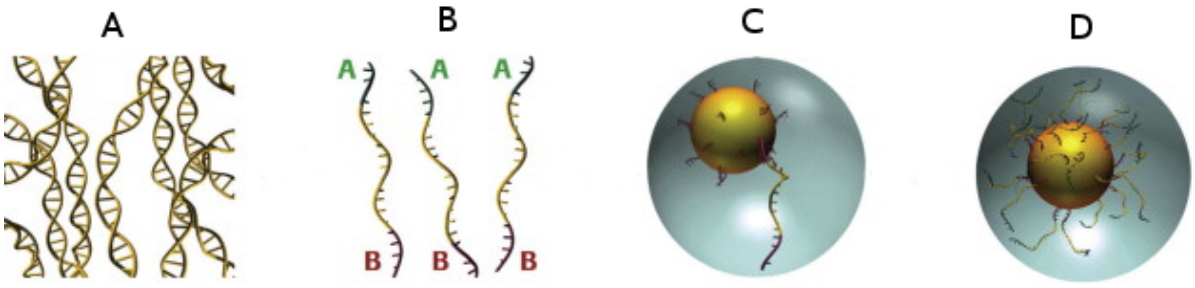


Figure 7.1: DNA template preparation and emulsion PCR in 454 sequencing (source: `www.454.com`). A. Genomic DNA is fragmented; B. Adaptors are ligated to each end of the denatured fragments; C. Each template is attached to a single bead; D. The templates are clonally amplified by emulsion PCR.

After amplification, the beads, each of which contains a unique amplified fragment, are loaded

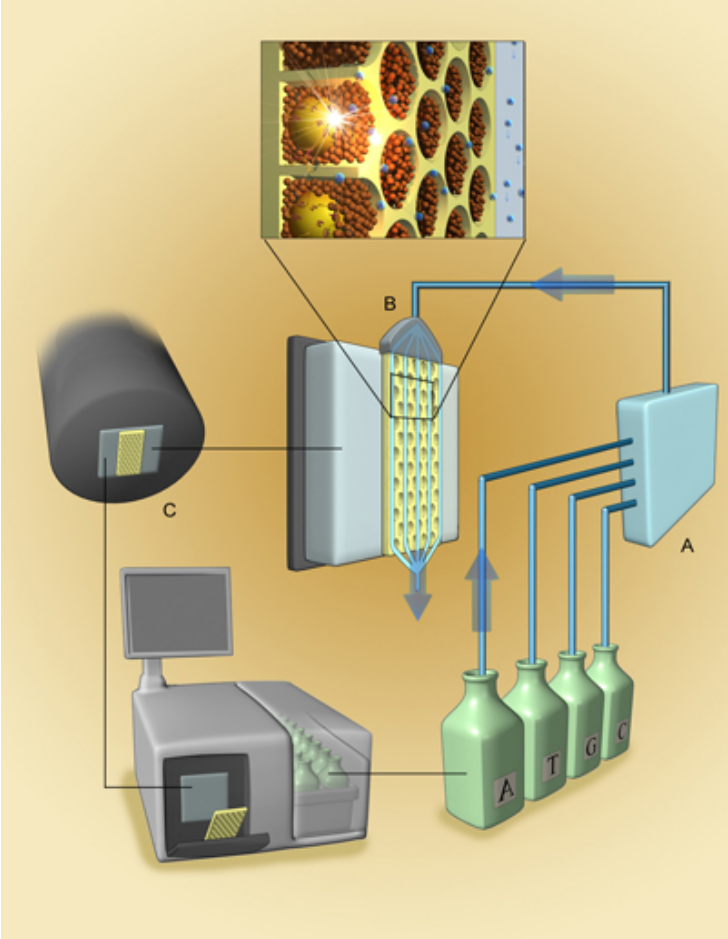


Figure 7.2: Overview of the 454 sequencing process (source: `www.454.com`). A. In a predetermined order a solution of one nucleotide type is mixed with reagents to be added to the PTP; B. The solution flows over the PTP potentially inducing a pyrosequencing reaction; C. The emitted light is recorded with a CCD camera.

to a 454 PicoTiterPlate (PTP). This is an array with hundreds of thousands of picoliter-scale wells. The addition of beads to the PTP occurs in such a way that each well contains a single bead. This spatial separation enables a major parallellization of the sequencing reactions to be performed. The actual sequencing is based on the pyrosequencing reaction (Ronaghi, 2001). For this reaction to take place, much smaller beads of about 1 μm are added to surround the DNA-containing beads in the PTP (top picture of Figure 7.2). On these small beads the active enzymes needed for pyrosequencing are attached. These enzymes are sulphurylase and luciferase and are used to facilitate light production.

After all the wells are properly filled, the PTP is placed in the sequencer. Subsequently, a solution of reagents and nucleotides is flowed over the PTP (pictures A and B of Figure 7.2). The addition of each of the 4 possible nucleotide solutions A, C, G or T occurs in a fixed and prede-

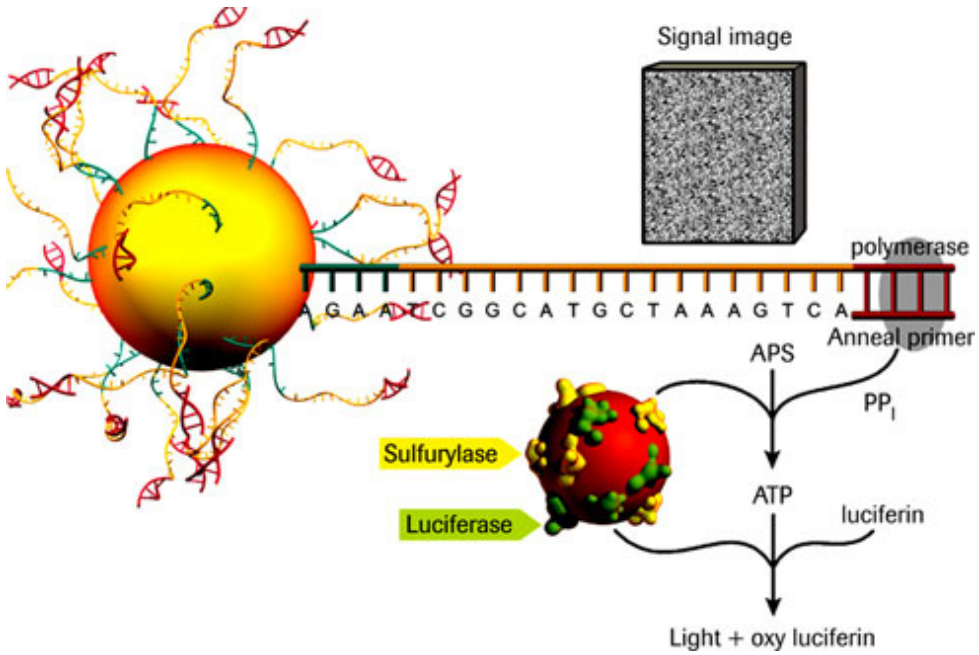


Figure 7.3: Detailed representation of the pyrosequencing reaction (source: `www.454.com`). Incorporation of the added nucleotide initiates a cascade of enzymatic reactions and eventually produces a burst of light.

termined order. A single addition of a nucleotide solution is called a nucleotide *flow*, whereas one round of 4 flows with each of the 4 nucleotides is referred to as a *cycle*. If the nucleotide is complementary to the nucleotide at the free end of the DNA template, it is *incorporated*. Incorporation of a nucleotide leads to the release of pyrophosphate, which is converted in a burst of light by the active enzymes attached to the smaller beads and an added luciferin reagent (see Figure 7.3). The emitted light is then detected with a charge-coupled device (CCD) camera placed at the other end of the wells (picture C of Figure 7.2). If multiple complementary nucleotides of the same type are available at the free end of the DNA template, multiple nucleotides are incorporated, leading to a larger light intensity signal detected by the CCD. A run of identical nucleotides is called a *homopolymer* run. If the nucleotide at the free end of the DNA template is not complementary to the nucleotide added to the PTP, no incorporation event takes place and no light is emitted.

The raw images are processed by the 454 software. This eventually results in a *flowgram* for each well of the PTP. A flowgram consists of a series of processed signal intensities for successive flows of the sequencing process. An example of a typical flowgram is given in Figure 7.4. The signal intensity for a flow is rounded to an integer to give the number of monomers of the corresponding nucleotide that were incorporated (Brockman et al., 2008). Hence, the order of

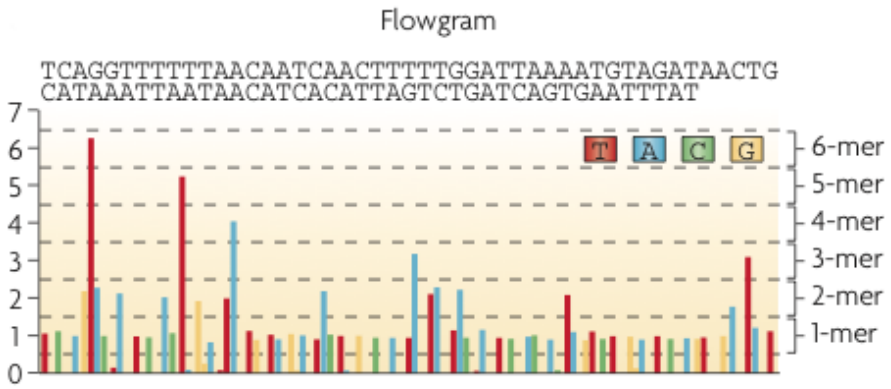


Figure 7.4: Example of a typical 454 flowgram; adapted from Metzker (2010).

the flows and the light intensities recorded for each flow reveals the underlying DNA sequence in each well. The full process of transforming the measured intensity signals to a sequence of nucleotides is referred to as *base-calling*.

7.2 Motivating data set

To motivate the research we consider data from a sequencing experiment conducted on the reference K-12 strain MG1655 of the bacterium *Escherichia coli*. This is a strain of *E. coli* which is often constructed in the laboratory for bacteriologic research purposes. DNA from *E. coli* was sequenced at the NXTGNT sequencing center, Ghent, Belgium, using shotgun sequencing. This implies that the DNA was broken into smaller fragments at random locations. The raw images were processed with the native 454 software, which is the software delivered with the sequencer, version 2.3. This resulted in a total number of 635979 produced reads.

As already mentioned in Section 7.1, the sequencing process does not allow the individual nucleotides in the DNA sequence to be read directly. In each flow, however, the number of successively incorporated nucleotides or the *homopolymer length* (HPL) has to be inferred from the measured light intensity. Each incorrect prediction of the HPL thus results in an insertion or deletion error. As a consequence, insertions and deletions are much more frequent in 454 sequencing than substitution errors, where one nucleotide type is replaced by another. Moreover, substitution errors are typically caused by a combination of successive insertion and deletion errors (e.g. Brockman et al., 2008). In the context of 454 sequencing, insertions are also referred

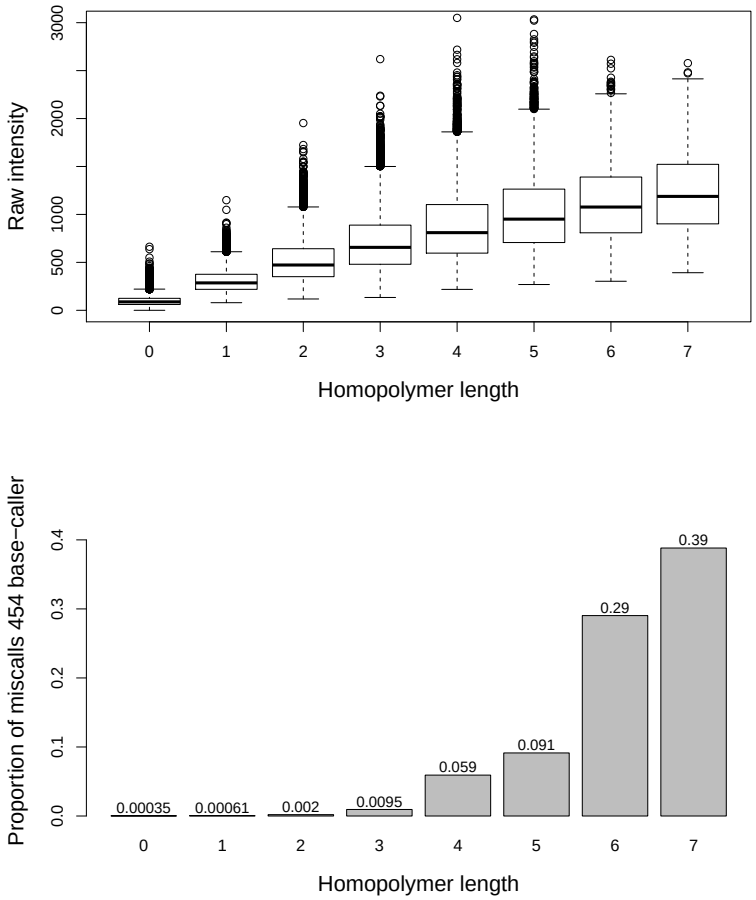


Figure 7.5: Effect of homopolymer length for a typical 454 sequencing experiment. Upper panel: distribution of measured raw intensities for different HPLs. Lower panel: Proportion of miscalls made by the native 454 base-caller for different HPLs in the reference sequence.

to as *overcalls*, while deletions are often called *undercalls*.

In general, the increase of intensity signal when more nucleotides are incorporated, attenuates at larger HPLs. Moreover, the variability of the raw intensities increases by increasing HPL. This is illustrated on the *E. coli* data set in the upper panel of Figure 7.5. As a consequence, it becomes harder to discriminate between subsequent HPLs at larger HPLs, resulting in an inflation of undercalls or overcalls as the HPL increases (e.g. Holt and Jones, 2008). Hence, more insertions and deletions are present when a DNA sequence contains many long homopolymers (e.g. Huse et al., 2007; Shendure and Ji, 2008). The lower panel of Figure 7.5 shows the sample proportion of miscalls in the *E. coli* data set, made by the 454 base-caller at different HPLs in the *E. coli* reference sequence. The base-calling error rate clearly increases by increasing HPL and becomes quite substantial from HPL 4.

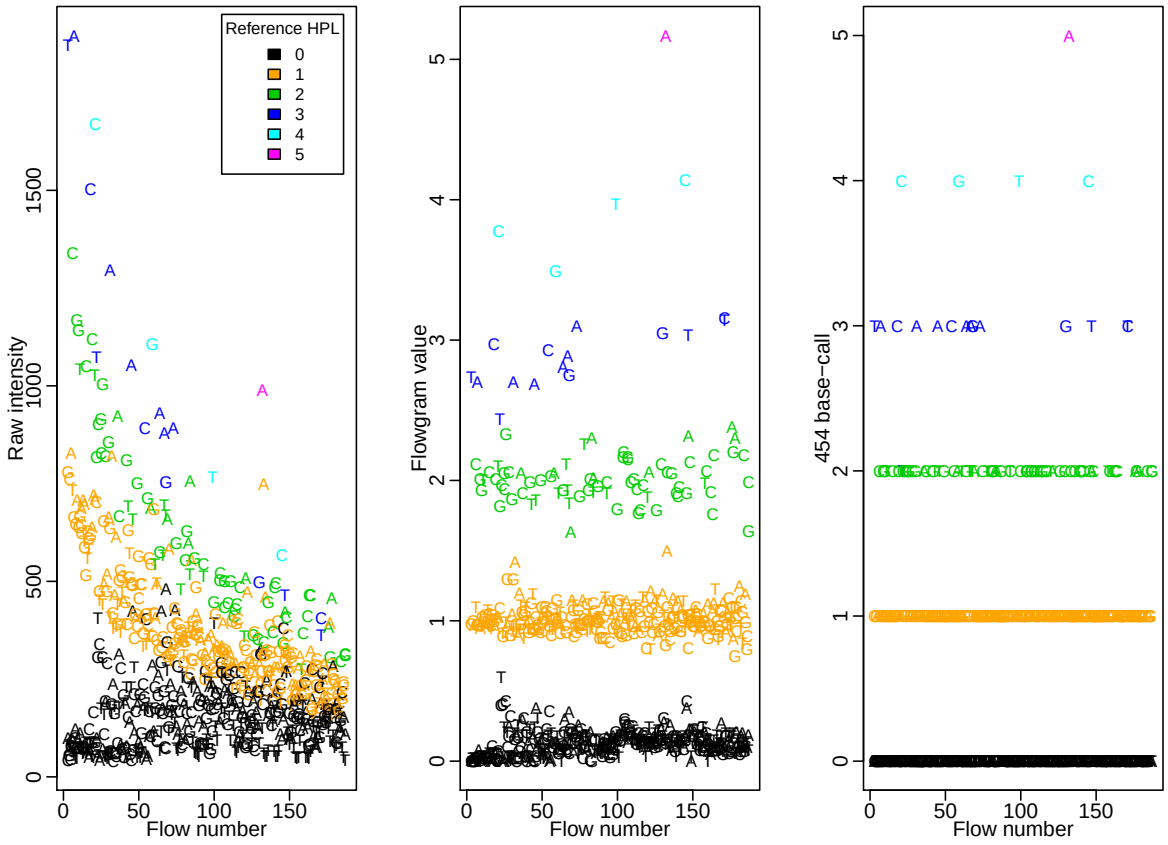


Figure 7.6: Raw intensities, flowgram values and 454 base-calls versus flow number for a typical read of the *E. coli* data set. The colors represent the reference HPLs.

In the first step of the 454 base-calling pipeline the raw signal intensities are first background-corrected and further preprocessed to flowgram values by correcting for the major error sources. These include spatial and read-specific effects such as the abundance of long homopolymers in a read (Margulies et al., 2005; Brockman et al., 2008). They are discussed in more detail in Chapter 8. This preprocessing eliminates much obscuring noise, but may remove some useful information as well. The left and middle panels of Figure 7.6 show this preprocessing effect for a typical well of the PTP in the *E. coli* data. The right panel of Figure 7.6 displays the 454 base-calls for this well. These base-calls are integer values corresponding to the predicted HPLs in the DNA sequence.

After base-calling, the 454 software assigns a *quality score* to each called base. The closer the flowgram value is to the called HPL, the larger the assigned quality of the base-call. Hence, for base-calls with a large quality score one is more certain that the base-call is correct. Figure 7.7 shows the distribution of flowgram values corresponding to a base-called HPL of 0, 1, 2 and 3

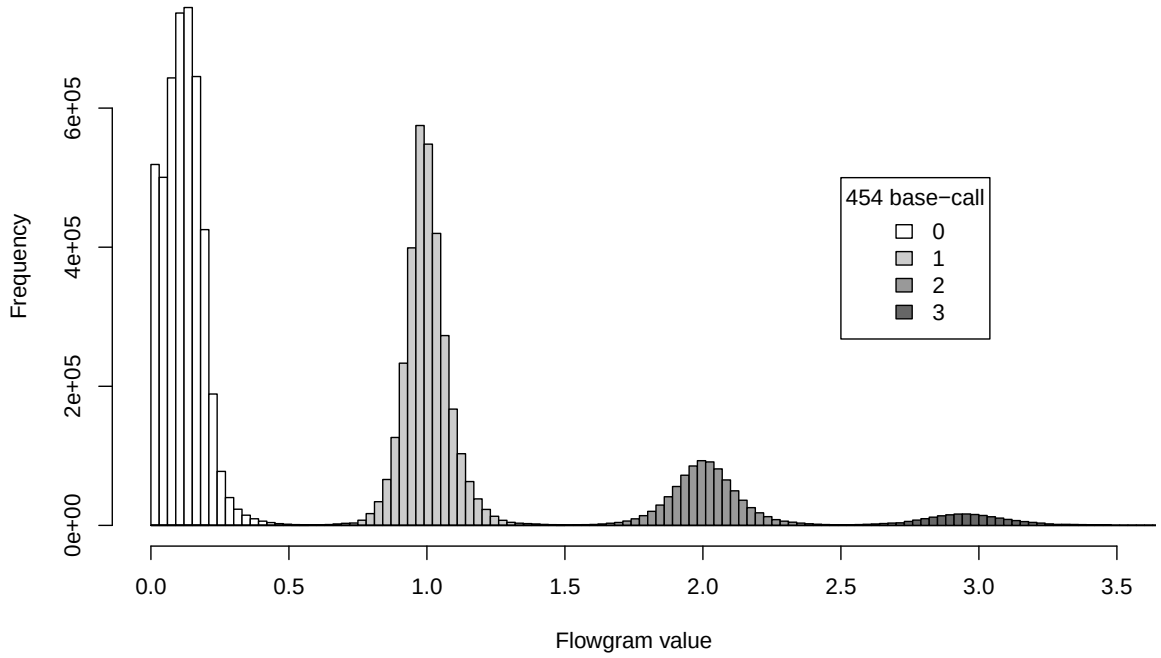


Figure 7.7: Distribution of flowgram values for base-calls 0 until 3 in the *E. coli* data

for the *E. coli* data. Flowgram values in the tails of the respective distributions are associated with low-quality base-calls.

Quality score calculation in current 454 base-calling is based on a multidimensional binning algorithm of different so-called *noise predictors* of the read (Brockman et al., 2008). These predictors are only based on the read's flowgram values and HPLs. This means that information from the preprocessing steps is not considered (Brockman et al., 2008). Therefore, the base-calling uncertainties inherent to the base-calling model or algorithm are not directly utilized in the construction of the quality scores. Another concern is that, although 454 quality scores are well-known, widely used and provide a measure for the quality of the base-call, they lack additional information on whether there might be an undercall or an overcall. Moreover, they can not be used to deduce how likely other HPLs are the correct call instead of the chosen base-call. Nevertheless, this feature is particularly essential in 454 sequencing, because of its high insertion and deletion error rate. This is illustrated in the example shown in Figure 7.8, based on the *E. coli* data. This figure shows the empirical cumulative distribution functions (eCDFs) of 454 quality scores for sequences with reference HPL 3 in case of an overcall, i.e. the called HPL is at least 4. As the 454 software provides a quality score for each nucleotide

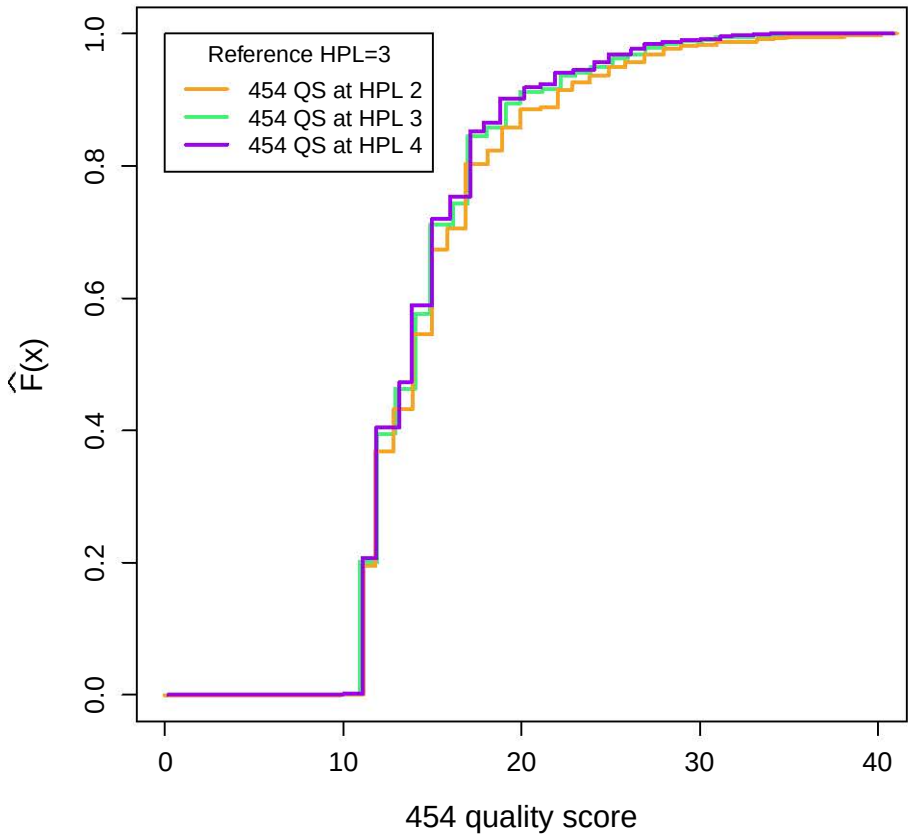


Figure 7.8: Empirical cumulative distribution functions of 454 quality scores assigned to nucleotides associated with called HPL 2, 3 and 4, in case of an overcall and for sequences with reference HPLref 3 in the *E. coli* data.

of the called HPL, the eCDFs can be separated by position in the homopolymer. Curves for the quality scores at position 2, 3 and 4 of the called HPL are shown. These eCDFs appear to be nearly identical. Hence, the quality scores do not give any insight into whether it is more likely to have an undercall or an overcall, given that a base-calling error was made.

7.3 Objectives and outline

In this second part of the dissertation we focus on problems in 454 sequencing data analysis that are mainly caused by inaccurate homopolymer calling. In line with the general philosophy adopted in this dissertation, as much raw information as possible is used and methods are proposed that allow errors to be propagated in further steps of the data analysis pipeline. Firstly,

it is in our interest to develop an improved base-calling method that tackles the main shortcomings of the native 454 base-caller, as illustrated in Section 7.2. This can be accomplished by means of a general probabilistic framework that seamlessly integrates the base-calling with more informative quality score assignment. As the outcome of interest is the HPL, a statistical model for count data seems a logical choice. The HPL can be modeled as function of several explanatory variables to correct for the main sources of obscuring variability. A particular challenge is to model the large number of zeros characteristic for 454 sequencing data. A HPL of 0 occurs each time the added nucleotide flow is not complementary to the interrogated position on the DNA template. Furthermore, the probabilistic nature of the model should allow to construct more informative quality scores that directly reflect the base-calling's uncertainties and provide information about potential undercall or overcall errors.

Secondly, we also focus on a downstream application of 454 sequencing data. In particular, we aim at detecting DNA sequence variants in homopolymers. Diploid organisms, like human beings, have their genetic information organized in pairs of homologous chromosomes. The nucleotide sequences of specific genomic locations on these two chromosomes are often identical. In that case these locations are said to be homozygous. However, it frequently occurs that their sequences are slightly different, in which case they are called heterozygous. An important task is to detect these sequence variants because they often contribute to an increased susceptibility for developing diseases, such as cancer (Stratton et al., 2009). More specifically, we aim at improving the detection of heterozygosity in homopolymeric regions caused by insertions or deletions. To our knowledge, no proper statistical methods have been developed yet for sequence variant detection in this specific setting. Currently, ad-hoc approaches based on the integer values of the base-calls are usually taken (e.g. De Leeneer et al., 2011; Coppieters et al., 2012). In these procedures the uncertainties of the base-calling are not taken into account in the variant detection pipeline. As Figure 7.7 already indicated, flowgram values provide this information to a certain extent. Hence, it may be worthwhile to use flowgram values rather than discrete base-calls to obtain an increased performance for the detection of these sequence variants.

Part II of this dissertation is organized as follows. In Chapter 8 we present a method for improved base-calling and quality score construction of 454 sequencing data based on a Hurdle Poisson model. Chapter 9 introduces and discusses a statistical method for the detection of DNA

sequence variants in 454 data, in which we focus on detection of homozygosity and heterozygosity in homopolymeric DNA regions of diploid organisms. Finally, discussion, conclusions and future research perspectives for this part of the dissertation are given in Chapter 10.

Chapter 8

Improved base-calling and quality scores for 454 sequencing based on a Hurdle Poisson model

In this chapter we present an improved method for base-calling and quality score construction of 454 sequencing data. The *E. coli* data set presented in Section 7.2 already provided a flavor of 454 sequencing data. In Section 8.1 the properties and main error sources of the 454 data are explored in more detail. Section 8.2 introduces a *weighted Hurdle Poisson model* for 454 base-calling and quality score construction. Finally, in Section 8.3, the performance of the method is assessed and compared to the base-calling of the 454 software, which we refer to as the native base-caller, and to Pyrobayes, which is another competing method.

8.1 Exploration of 454 sequencing data

Prior to introducing a statistical methodology for improved base-calling, typical 454 sequencing data as produced by the current base-caller are explored in more depth. The aim is to reveal some aspects of the nature of these data that may help in developing the method. For this purpose, we use the data set on *E. coli* that was presented in Section 7.2.

Figure 8.1 shows nonparametric density estimates of the raw signal intensities in the *E. coli* data. The left panel gives the untransformed raw intensities. The distribution of untransformed

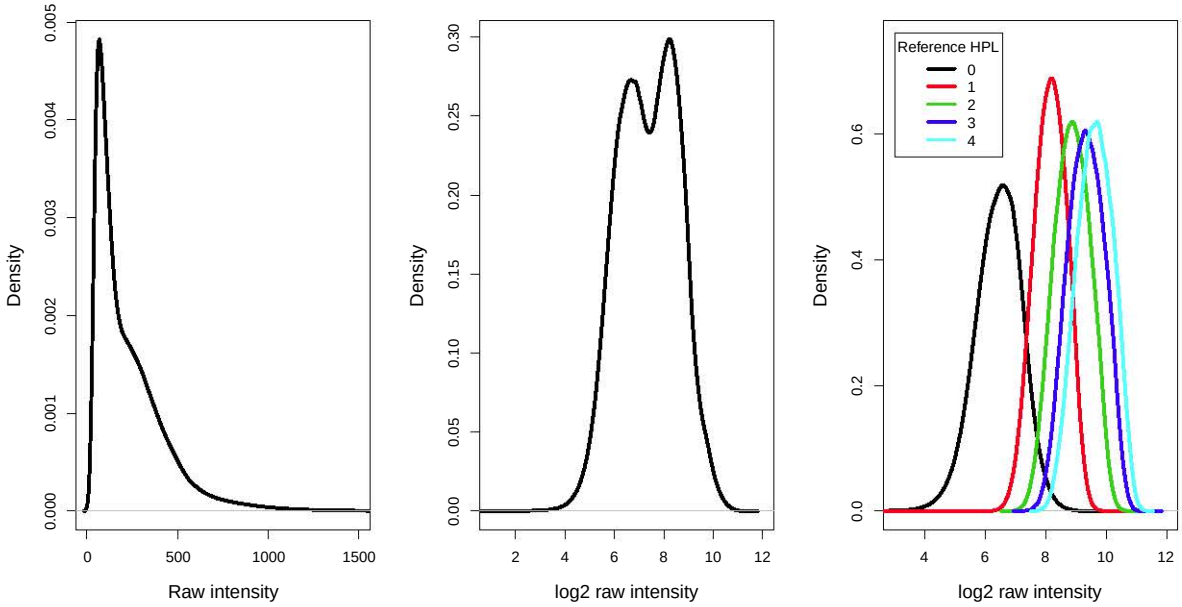


Figure 8.1: Nonparametric density estimates of raw signal intensities in the *E. coli* data. Left: untransformed raw intensities; Middle: $\log_2$-transformed raw intensities; Right: $\log_2$-transformed raw intensities separated by reference HPL.

raw intensities is unimodal, but it has a shoulder in its heavy right tail. It represents a mixture distribution composed of two underlying processes. The left part of the distribution, mainly to the left of the shoulder, is associated with background intensities. This is the intensity measured when the added nucleotide is not complementary to the free end of the interrogated DNA template and is therefore not incorporated. We refer to this component as the *background signal* (HPL $= 0$). The second component, on the right side of the mixture distribution, consists of raw intensities corresponding to one or multiple nucleotide incorporation events. This component is henceforth called the *incorporation signal* (HPL > 0). In the middle panel of Figure 8.1 the raw intensities are $\log_2$-transformed. The mixture distribution is now bimodal, which makes the separation between the background and incorporation signal more clear. The right panel of Figure 8.1 displays the same $\log_2$-transformed raw intensities, but now separated according to the reference HPLs. The measured intensities generally increase by increasing HPL, but the increase attenuates for larger HPLs (see also Chapter 7). This figure particularly shows that the separation of the background signal (HPL $= 0$) and incorporation signal (HPL > 0) distributions is much better than the separation between the distributions corresponding to one or more incorporation events (HPL $= 1$, HPL $= 2$, HPL $= 3$,...).

Nonparametric density estimates of the $\log_2$-transformed raw intensities for different cycle num-

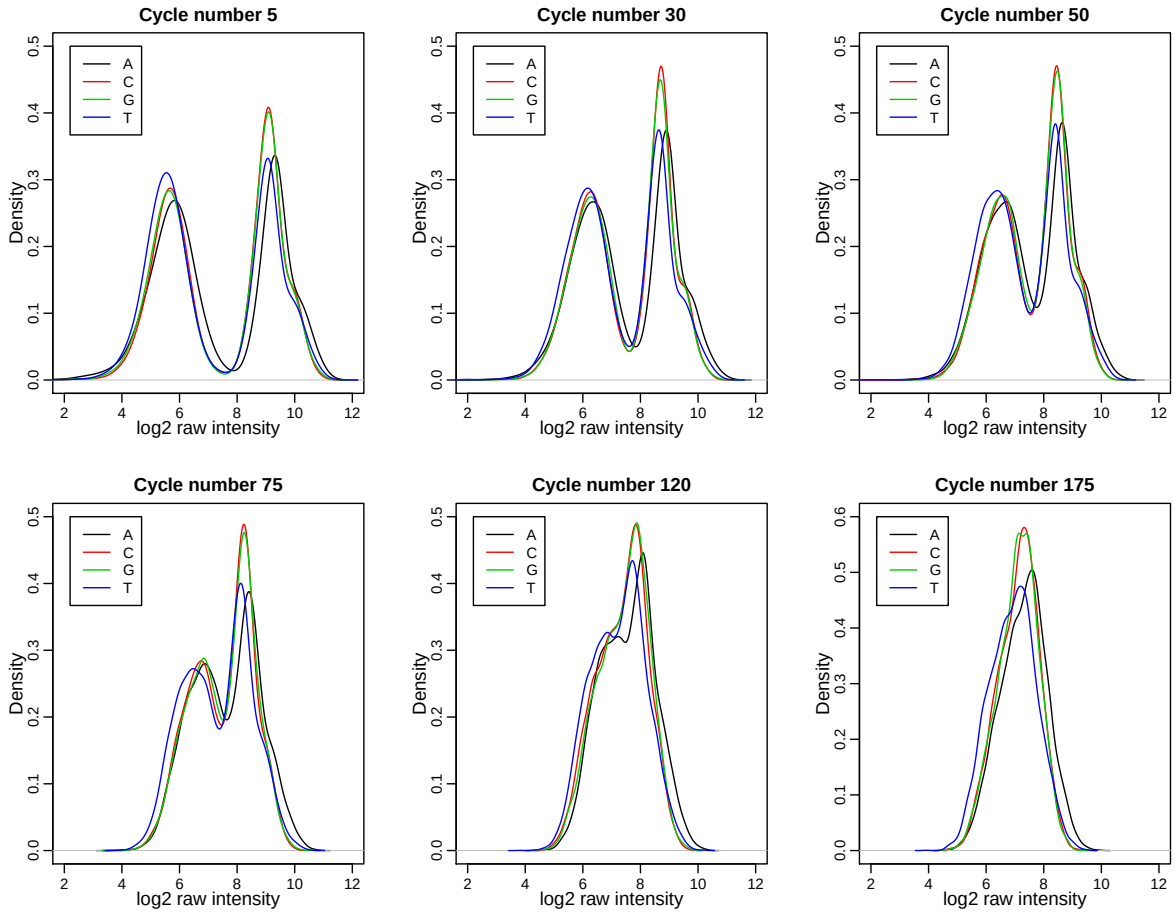


Figure 8.2: Nonparametric density estimates of $\log_2$-transformed raw signal intensities in the *E. coli* data for different cycle numbers and separated by nucleotide type.

bers and nucleotide types are depicted in Figure 8.2. Cycle number 5, for instance, corresponds with the fifth time that the solutions of T, A, C and G were consecutively added. The bimodal structure observed in the middle panel of Figure 8.1 appears to be much more pronounced at smaller cycle numbers, corresponding to the beginning of the sequencing process. As the cycle number increases, the background and incorporation signal distributions move more and more towards each other. At the end of the sequencing process the two components of the mixture distribution can even no longer be distinguished: see, e.g., the lower right panel of Figure 8.2 which shows the densities of $\log_2$-transformed raw intensities at cycle number 175.

To a large extent this phenomenon can be explained by the *dephasing* effect that occurs during 454 sequencing, which is also referred to as *loss of synchrony* (e.g. Brockman et al., 2008). In the beginning of the sequencing process all of the millions of identical DNA templates on a certain bead are *in sync* and incorporate the same nucleotide at the same position in the fragment in each flow. Hence, there is a clear separation between background signal, i.e. no nucleotide is

incorporated, and the incorporation signal, i.e. at least one nucleotide is incorporated, in these first cycles. As the sequencing progresses, more and more templates fall out of sync. This is known to occur more frequently in the neighborhood of homopolymers and in reads containing many homopolymers (e.g. Huse et al., 2007). One of the reasons is the presence of insufficient nucleotides within a flow. This may lead to incomplete synthesis of the complementary DNA strand within homopolymers, sometimes called *incomplete extension*. Another source of dephasing errors is due to insufficient flushing of reagents between flows. This may cause incorporation of nucleotides of a different type in a single flow, resulting in a *carry forward* effect (Huse et al., 2007).

The shape and location of the densities of $\log_2$-transformed raw intensities in Figure 8.2 are very similar for different nucleotide types. The resemblance is especially large between densities of nucleotide C and G on the one hand, and between A and T on the other hand. The densities for nucleotide A are slightly shifted towards larger intensities compared to the densities for nucleotide T.

We further explore to which extent these dephasing properties affect the raw intensities and flowgram values in neighboring flows and cycles. These effects are assessed for the nucleotide C flow, but similar effects are observed for the other nucleotide types. First, the effect of the reference HPL in the preceding flow of the sequencing process is examined. Usually, the nucleotides are added in a predetermined order: T, A, C, G. Figure 8.3 gives the $\log_2$-transformed raw intensities for flows of nucleotide C corresponding with a reference HPL of 0 (left panel) or 1 (right panel). They are separated by reference HPL for nucleotide A in the preceding flow. It appears that the distributions of the $\log_2$-transformed raw intensities are nearly identical in shape and location. Hence, the reference HPL for the nucleotide in the preceding flow does not influence the intensity in the current flow.

Figure 8.4 again shows the $\log_2$-transformed raw intensities for flows of nucleotide C corresponding with different reference HPLs. In this case the density estimates are separated by reference HPL for nucleotide C in the preceding cycle. Recall that this is equivalent with 4 flows earlier in the sequencing process. In general, the raw intensity clearly increases by increasing reference HPL in the preceding cycle. This effect is especially pronounced for reference HPLs of 0 in the current cycle (upper left panel of Figure 8.4), and gradually decreases by increasing reference HPL in the current cycle (upper right for reference HPL 1, lower left for reference

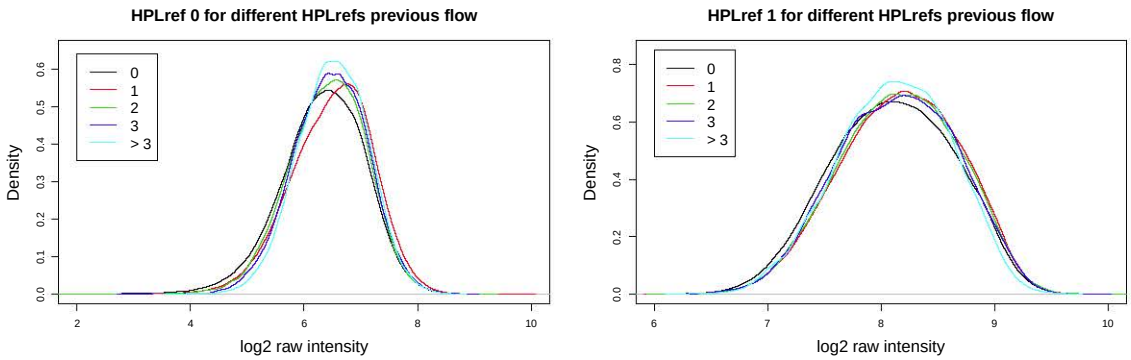


Figure 8.3: Nonparametric density estimates of $\log_2$-transformed raw intensities for flows of nucleotide C corresponding with a reference HPL of 0 (left panel) or 1 (right panel). Different curves are given for different numbers of HPL for the preceding flow of nucleotide A.

HPL 2 and lower right for reference HPL 3). These findings can be explained by the dephasing effect. Due to incomplete extension, a proportion of DNA templates on the bead are behind most templates with respect to their read position. For many of these dephased templates the incorporation of a homopolymer occurs one cycle later compared to the templates that are still in sync. Recall that the measured signal in each flow is a consensus of the emitted light of all templates on the bead. Even if there is no incorporation event on the templates in sync, the average amount of emitted light is increased by the homopolymer incorporation on the dephased templates. If there occurs an incorporation event of one or more nucleotides in the current cycle, the size of this effect decreases. This is because the additional amount of light emitted from the dephased templates is only small compared to the emitted light from the templates that are in sync at the homopolymer position.

Nonparametric density estimates of the flowgram values for nucleotide C flows corresponding with a reference HPL of 2 (left panel) or 3 (right panel) are depicted in Figure 8.5. The result is quite interesting, because the observed dephasing effect seems to be overcorrected for by the 454 software in the preprocessing from raw intensities to flowgram values. For long homopolymers in the preceding cycle the distribution of flowgram values is not centered around the ideal values of 2 or 3, but is shifted towards smaller values.

The left panel of Figure 8.6 shows $\log_2$-transformed raw intensities for flows of nucleotide C corresponding with a reference HPL of 0 in both the current and the preceding cycle. The density estimates are separated by reference HPL for nucleotide C two cycles earlier. The effect on the distributions of the $\log_2$-transformed raw intensities observed in Figure 8.4 is still present,

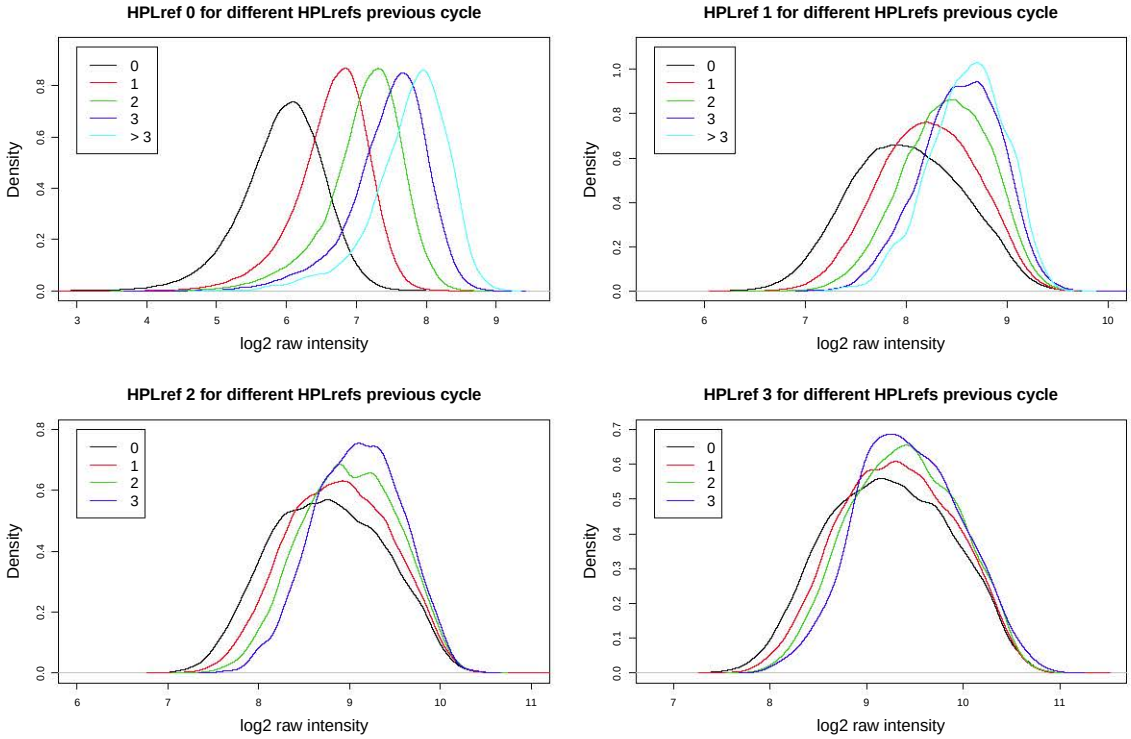


Figure 8.4: Nonparametric density estimates of $\log_2$-transformed raw intensities for flows of nucleotide C corresponding with a reference HPL of 0 (upper left panel), 1 (upper right panel), 2 (lower left panel) or 3 (lower right panel). Different curves are given for different numbers of HPL for the flow of nucleotide C in the preceding cycle.

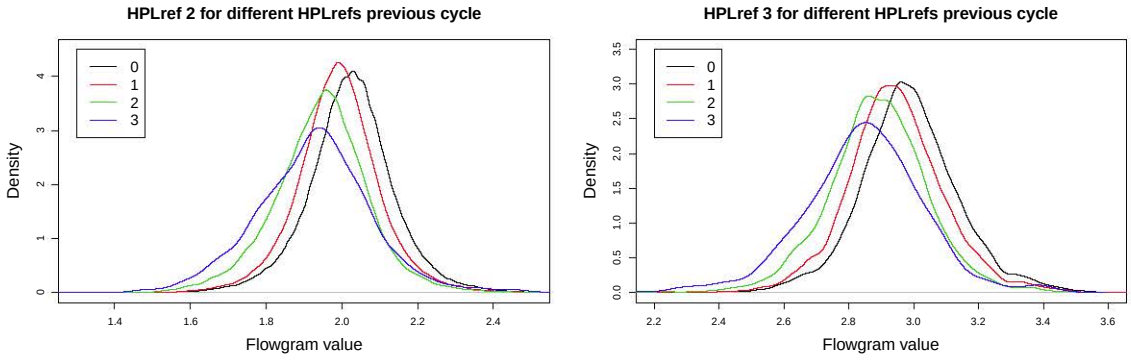


Figure 8.5: Nonparametric density estimates of flowgram values for flows of nucleotide C corresponding with a reference HPL of 2 (left panel) or 3 (right panel). Different curves are given for different numbers of HPL for the flow of nucleotide C in the preceding cycle.

albeit to a smaller extent. When reference HPLs in the current or previous cycle of more than 0 are considered, the effect becomes negligible (results not shown). The right panel of Figure 8.6 shows $\log_2$-transformed raw intensities for flows of nucleotide C corresponding with a reference HPL of 0, separated by reference HPL for nucleotide C in the following cycle, instead of the previous. This plot illustrates that the dephasing effect also works in the other direction because

of the carry forward effect explained previously.

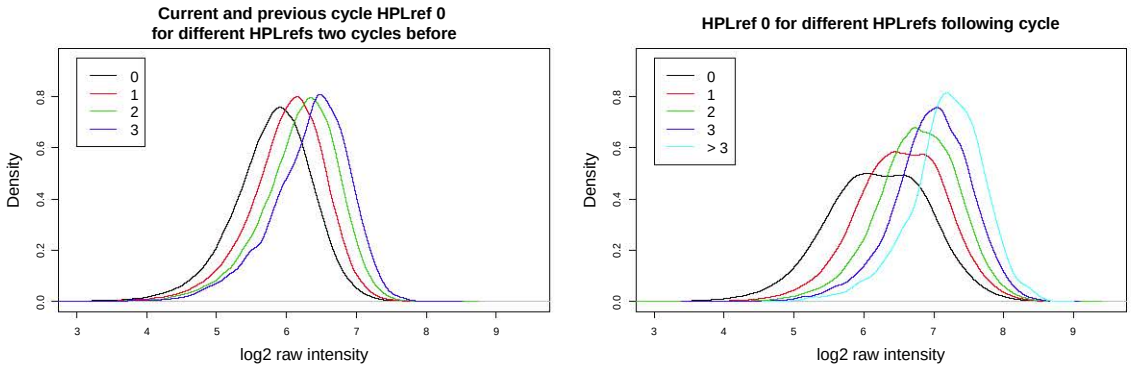


Figure 8.6: Nonparametric density estimates of $\log_2$-transformed raw intensities for flows of nucleotide C corresponding with a reference HPL of 0. In the left panel different curves are given for different numbers of HPL for the flow of nucleotide C two cycles earlier in the sequencing process, and given a reference HPL of 0 for nucleotide C in the preceding cycle. In the right panel the different curves correspond to different numbers of HPL for the flow of nucleotide C in the following cycle.

8.2 Weighted Hurdle Poisson model for 454 base-calling and quality scores

In this section a new base-caller is presented that builds upon a weighted Hurdle Poisson model. We first introduce the model structure. Next, the parameter estimation procedure for this model is discussed. Finally, it is explained how the fitted model can be used for base-calling and constructing more informative quality scores as compared to the native 454 base-caller.

8.2.1 Model specification

Let N_{bc} be the number of nucleotides b that are incorporated in cycle c, with $b \in B = \{A, C, T, G\}$ and $c = 1, \ldots, L$, where L represents the total number of cycles in the sequencing experiment. Note again that one cycle consists of 4 flows of nucleotide solutions added to the sequencer in fixed order (T, A, C, G). The base-calling problem is treated as a classification problem, where each possible value for N_{bc} corresponds to a different class. Based on the observed input information on the raw intensities and flowgram values, the flows are assigned to one of these classes. If there are only two possible classes, this is often done by logistic

regression. Here, we use Poisson regression, because multiple HPLs have to be classified. Furthermore, these models also allow for extrapolation to larger HPLs, in contrast to a multinomial modeling approach.

It often occurs that during the sequencing process no nucleotide is incorporated in a certain flow. Hence, many HPLs of 0 are recorded ($N_{bc} = 0$). Figure 8.7 shows the marginal distribution of the reference HPLs in the data set. Apparently, there are not more zeros in the data set than expected for a Poisson distribution and the data appears to fit well to a Poisson distribution with estimated mean $\hat{\lambda} = 0.647$.

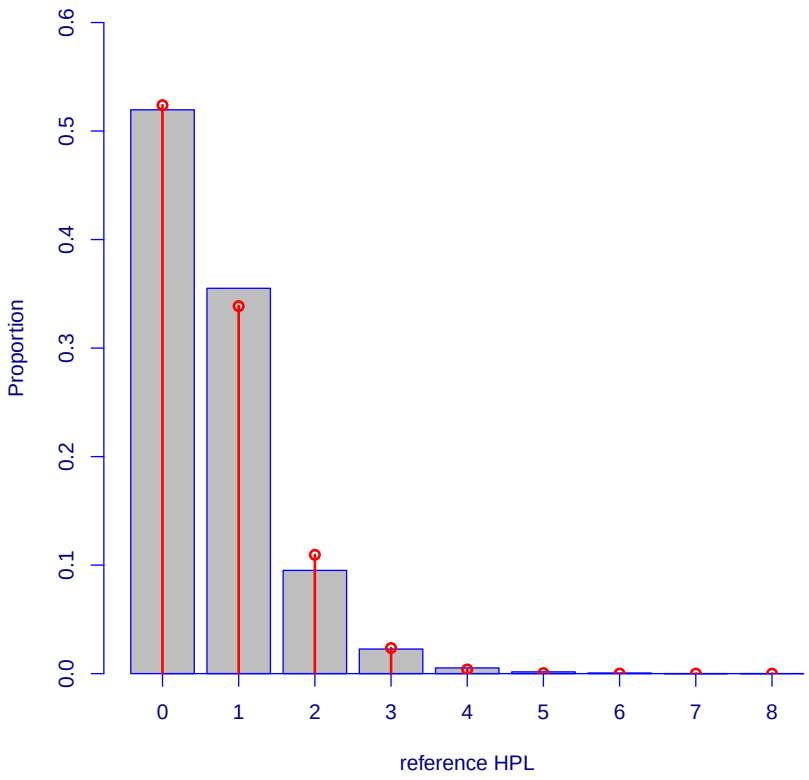


Figure 8.7: Histogram with observed proportions of reference HPLs in *E. coli* data set. The height of the red vertical lines indicates the estimated probabilities of the reference HPLs according to the Poisson distribution fitted to the data. The estimated mean is $\hat{\lambda} = 0.647$.

In a regression context, however, the assumption of a Poisson distribution might be too restrictive. From a conceptual point of view there is a clear difference between having a nucleotide incorporation event or not. In the latter case only a background intensity signal is measured, while in the former case the intensity is a sum of background intensity and intensity of the light emitted from the pyrosequencing reaction after nucleotide incorporation. This implies the ex-

istence of two distinct data-generating mechanisms: one for the background signal (HPL $= 0$) and one for the incoporation signal (HPL > 0). Hurdle models constitute a class of models designed to allow for such a distinction (e.g. Ridout et al., 1998). They are mixture models with a binomial component that distinguishes between zero counts and positive counts, and a zero-truncated Poisson component which models the positive counts, conditional on having a non-zero count or having "crossed the hurdle".

After zero-truncation the Poisson distribution no longer provides a good fit to the data (see Figure 8.8). The data show considerable underdispersion after truncation, which means that the variance is smaller than the mean. The sample variance for the *E. coli* data is 0.456, while the fitted mean to the Poisson distribution is now $\hat{\lambda} = 1.346$.

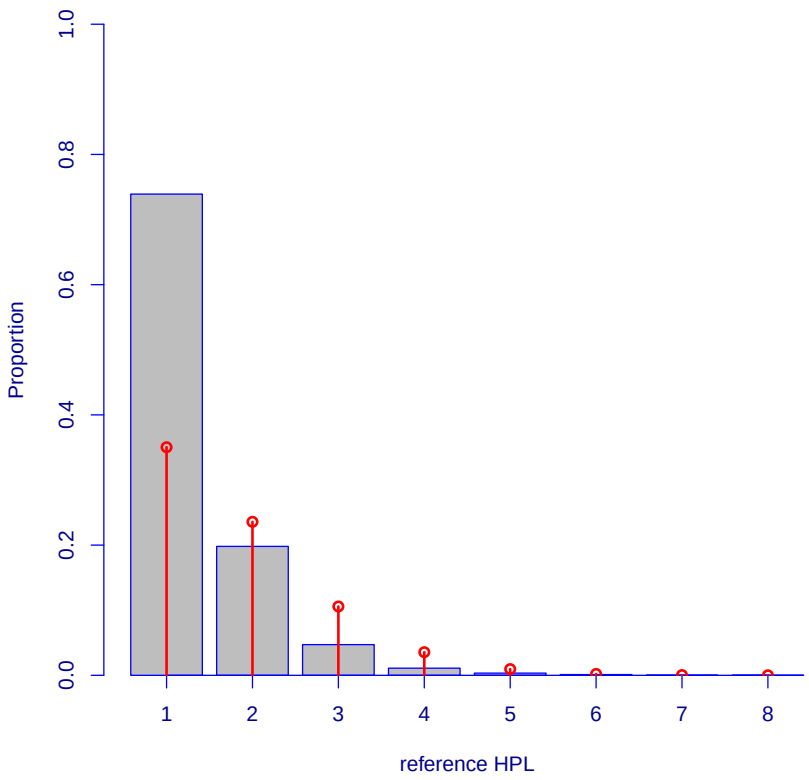


Figure 8.8: Histogram with observed proportions of reference HPLs in *E. coli* data set after zero-truncation. The height of the red vertical lines indicates the estimated probabilities of the reference HPLs according to the Poisson distribution fitted to the zero-truncated data. The estimated mean is $\hat{\lambda} = 1.346$.

To cope with underdispersion a weighted Poisson component is adopted (Ridout and Besbeas, 2004). The following Hurdle Poisson model is considered:

$$\Pr\{N_{bc} = n_{bc} | \boldsymbol{x}_{bc}, \boldsymbol{y}_{bc}\} = \begin{cases} 1 - \pi_{bc} & \text{if } n_{bc} = 0, \\ \pi_{bc}\, f_{\text{ZTWP}}(n_{bc}; \lambda_{bc}, \theta) & \text{if } n_{bc} = 1, 2, 3, \ldots, \end{cases} \tag{8.1}$$

where $\boldsymbol{x}_{bc}$ and $\boldsymbol{y}_{bc}$ are the vectors of predictor variables for the two components in the mixture model. These are discussed in more detail further down in the text. Further, f_{ZTWP} is the density of a zero-truncated weighted Poisson distribution, given by

$$f_{\text{ZTWP}}(n_{bc}; \lambda_{bc}, \theta) = \frac{f_{\text{WP}}(n_{bc}; \lambda_{bc}, \theta)}{1 - f_{\text{WP}}(0; \lambda_{bc}, \theta)} \quad \text{for } n_{bc} = 1, 2, 3, \ldots, \tag{8.2}$$

with f_{WP} denoting the density of the weighted Poisson distribution,

$$f_{\text{WP}}(n_{bc}; \lambda_{bc}, \theta) = \frac{w_{n_{bc}} e^{-\lambda_{bc}} \lambda_{bc}^{n_{bc}}}{W_{bc} n_{bc}!}. \tag{8.3}$$

In (8.3), $\{w_{n_{bc}}\}$ denotes a set of weights, $\lambda_{bc} > 0$ is a nucleotide- and cycle-specific Poisson rate parameter, and W_{bc} is a normalizing constant to ensure that the probabilities sum to one. It is given by

$$W_{bc} = \sum_{n_{bc}=0}^{\infty} \frac{e^{-\lambda_{bc}} \lambda_{bc}^{n_{bc}} w_{n_{bc}}}{n_{bc}!}. \tag{8.4}$$

We use exponential weights similar to those of Ridout and Besbeas (2004),

$$w_{n_{bc}} = e^{-\theta(\lambda_{bc} - n_{bc})^2} \quad \text{with } \theta > 0. \tag{8.5}$$

In the weighted model formulation an additional dispersion parameter θ is considered. For $\theta > 0$ the underdispersion in the count data can be modeled properly.

The nucleotide- and cycle-specific parameters π_{bc} in the binomial component and λ_{bc} in the Poisson component are modeled with predictors $\boldsymbol{x}_{bc}$ and $\boldsymbol{y}_{bc}$, respectively. We allow the predictor effects to be nonlinearly associated with the HPL by considering generalized additive models (GAMs) (Hastie and Tibshirani, 1990). In particular,

$$\text{logit}(\pi_{bc}) = \beta_{0,bc} + \sum_{j=1}^{k} f_j(x_{j,bc}), \tag{8.6}$$

$$\log(\lambda_{bc}) = \gamma_{0,bc} + \sum_{j=1}^{l} g_j(y_{j,bc}), \tag{8.7}$$

with the f_j and g_j being smooth functions of the corresponding predictor variables $x_{j,bc}$ and $y_{j,bc}$, respectively. Cubic smoothing splines are chosen for this purpose (e.g. Hastie et al., 2001). The predictor variables in (8.6) and (8.7) can be specified separately, which is one of the main reasons the hurdle model approach is taken.

As shown in Section 8.1, the distributions of raw intensities for the four nucleotide types are not exactly the same. For this reason, and also because of computational efficiency, a different model is fitted for each nucleotide type. The following covariates are used in either or both of the 2 submodels: (1) intensities in the current flow; (2) cumulative sum of intensities up to the current flow; and (3) intensities 4 and/or 8 flows before and 4 and/or 8 flows after the current flow.

Both the flowgram values and the $\log_2$-transformed raw intensities are used in the submodels (8.6) and (8.7). The flowgram values are processed by the default 454 software for obtaining as much information on the HPL as possible, while reducing obscuring noise to a great extent. However, some valuable information may have been lost in this process. Moreover, Figure 8.5 has shown that some of the idiosyncrasies of 454 raw intensity data are somewhat overcorrected when computing the flowgram values. Therefore, also the $\log_2$-transformed raw intensities are used in the model. The cumulative sum of intensities allows for correcting for the cycle-specific effect displayed in Figure 8.2. Figures 8.4, 8.5 and 8.6, on the other hand, have shown the need to use the information of intensity values in the preceding and following cycles of the sequencing process.

8.2.2 Parameter estimation

The parameters of the Hurdle Poisson model are estimated by maximizing the likelihood. Because of the special two-component structure of the model, the likelihood function can be factorized according to the two components of the mixture. The log-likelihood can be written as

$$\ln(L) = \sum_{n_{bc}=0} \ln(1-\pi_{bc}) + \sum_{n_{bc}>0} \ln(\pi_{bc}) + \sum_{n_{bc}>0} f_{\text{ZTWP}}(n_{bc}; \lambda_{bc}, \theta). \tag{8.8}$$

Due to this decomposition the parameters of (8.6) and (8.7) can be estimated orthogonally,

which means that each part can be maximized separately. The parameters in both parts of the model are estimated by means of an iteratively reweighted least squares (IRLS) procedure (e.g. McCullagh and Nelder, 1989). For the binomial component standard software for logistic regression can be used. For the weighted Poisson component, on the other hand, the expected Fisher Information matrix $\mathrm{E}\{I\}$ is derived to be used in the IRLS, a procedure referred to as Fisher's scoring (e.g. McCullagh and Nelder, 1989). The derivation is given in Appendix A, resulting in

$$\mathrm{E}\{I\} = \begin{bmatrix} (1+2\theta\lambda_{bc})^2 \ \mathrm{Var}\{N_{bc}\} & -(1+2\theta\lambda_{bc}) \ \mathrm{Covar}\{(N_{bc}-\lambda_{bc})^2, N_{bc}\} \\ (1+2\theta\lambda_{bc}) \ \mathrm{Covar}\{(N_{bc}-\lambda_{bc})^2, N_{bc}\} & \mathrm{Var}\{N_{bc}-\lambda_{bc}\} \end{bmatrix}. \tag{8.9}$$

8.2.3 Base-calling and quality score production

After fitting the model with the parameter estimation procedure described in Section 8.2.2, the estimated parameters $\hat{\pi}_{bc}$, $\hat{\lambda}_{bc}$ and $\hat{\theta}$ are plugged into Model (8.1) to obtain estimated probabilities for all possible HPLs. Subsequently, base-calling for each flow bc occurs by determining the HPL n_{bc} for which $\hat{\Pr}\{N_{bc} = n_{bc}|\boldsymbol{x}_{bc}, \boldsymbol{y}_{bc}\}$ is maximal. The base-calling with the weighted Hurdle Poisson model is referred to as *HPCall*. The probabilities obtained from HPCall are very useful because they provide a direct probabilistic interpretation to the base-calling uncertainties. In this way they give insight into potential undercall or overcall errors. This will be more thoroughly discussed in Section 8.3.1. Moreover, they can also be used for the construction of quality scores in a similar fashion as the traditional quality scores produced by the standard 454 software, which are provided in the *Phred* format (Ewing and Green, 1998). These quality scores reflect the probability that the called nucleotide is not an overcall. In particular, the Phred-like quality score of the k-th called nucleotide in a homopolymer stretch ($k > 0$) is thus given by: $QS_{k,\text{over}} = -10 \log_{10}(1 - \sum_{n_{bc}=k}^{\infty} \hat{\Pr}\{N_{bc} = n_{bc}|\boldsymbol{x}_{bc}, \boldsymbol{y}_{bc}\})$.

Since the model-based base-calling allows to obtain the probabilities for all possible HPLs, we can also calculate an alternative quality score that reflects the probability that the called base is not an undercall. This is given by $QS_{k,\text{under}} = -10 \log_{10}(1 - \sum_{n_{bc}=0}^{k} \hat{\Pr}\{N_{bc} = n_{bc}|\boldsymbol{x}_{bc}, \boldsymbol{y}_{bc}\})$. Furthermore, using $QS_{k,\text{over}}$ and $QS_{k,\text{under}}$, we also propose to calculate a new quality score:

$QS_{k,\text{HPCall}} = I_{\text{dir}} \times \min(QS_{k,\text{under}}, QS_{k,\text{over}})$ with $I_{\text{dir}} = -1$ if $QS_{k,\text{under}} < QS_{k,\text{over}}$ and $I_{\text{dir}} = 1$ if $QS_{k,\text{under}} > QS_{k,\text{over}}$. Hence, the sign of $QS_{k,\text{HPCall}}$ indicates whether an undercall or an overcall is more likely (see also Section 8.3.1).

8.3 Results

In this section the performance of HPCall is compared with the native 454 base-caller and Pyrobayes (Quinlan et al., 2008), using the *E. coli* data set. Pyrobayes applies Bayes' rule to obtain a posterior probability for the HPL, given the distribution of flowgram values. However, no additional error sources are taken into account. Just as in HPCall, the called nucleotide sequence in Pyrobayes is produced by concatenating the most likely number of nucleotides in each consecutive flow. Somewhat arbitrarily, if the presence of a nucleotide is above some minimum probability, one extra nucleotide is called for that flow in order to minimize the undercall error rate (Quinlan et al., 2008). In the first part of this section the base-calling accuracy of the three methods is evaluated. This is followed by a discussion on the performance and properties of the different quality scores. Finally, an overview is given of the different steps in the HPCall software pipeline.

8.3.1 Base-calling results

8.3.1.1 Prediction accuracy

The estimation of the Hurdle Poisson model parameters is based on the use of a representative training data set generated from a reference DNA sequencing experiment. For this purpose the *E. coli* data set is used. Since variation of this *E. coli* K-12 strain with respect to the reference sequence is extremely rare, the data set can be treated as a *known-truth* data set. For computational convenience the performance evaluation is conducted on a random subset of 15000 wells of the data set. The HPCall Hurdle Poisson model is fitted using 1000 random wells out of these 15000. The other 14000 wells are used for evaluation. The percentages and absolute numbers of base-calling errors for this data set, separated by HPL, are depicted in Figure 8.9. An overall decrease of base-calling errors with 35% is observed for HPCall as compared to the native 454 base-caller, while Pyrobayes leads to even larger numbers of base-calling errors. The

smaller number of errors is consistent throughout the whole range of HPLs, with peaks at HPL 4 (55% decrease compared to native 454) and HPL 6 (50% decrease compared to native 454).

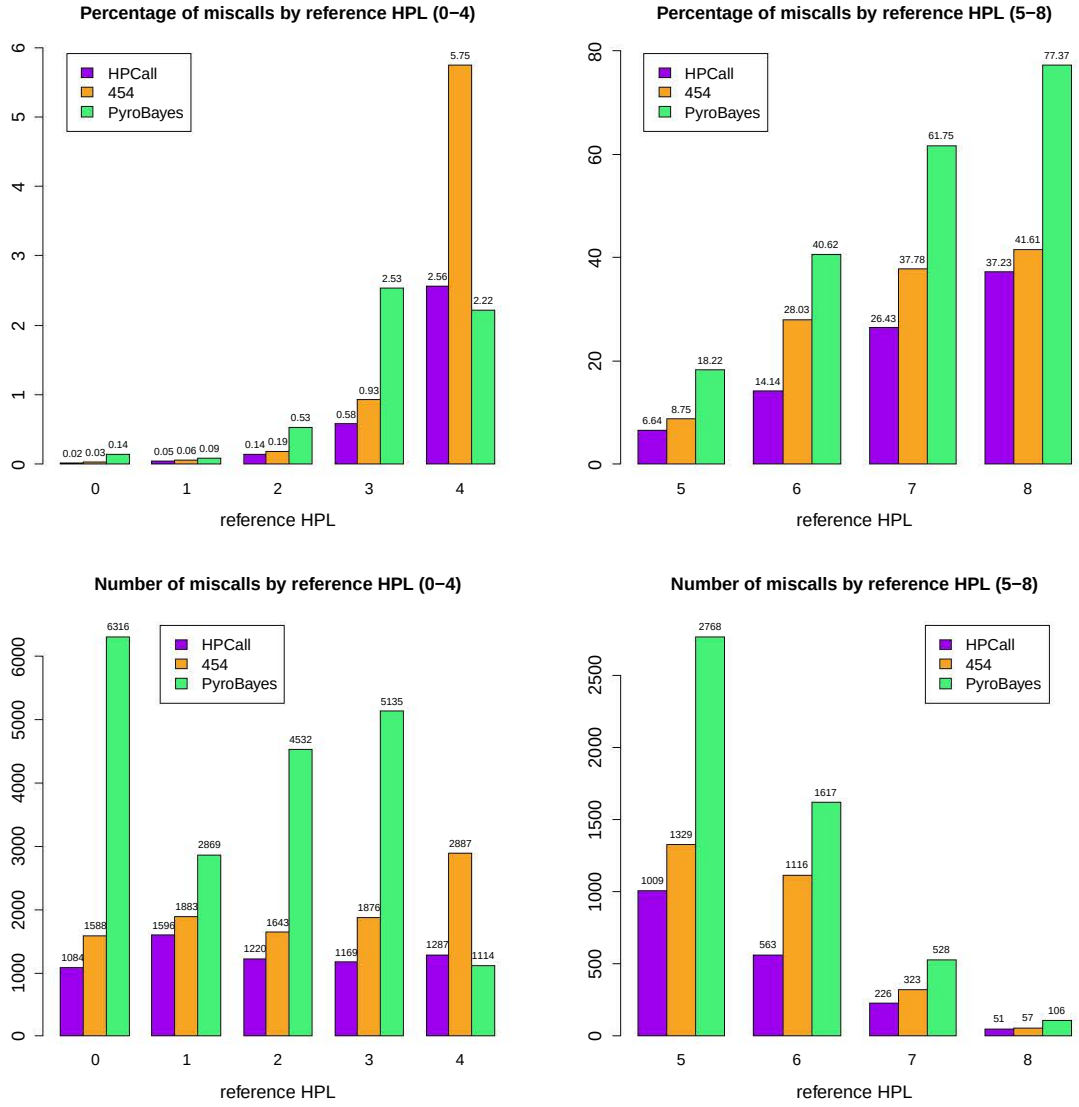


Figure 8.9: Comparison of the percentages (upper panel) and absolute numbers (lower panel) of base-calling errors by HPL for the three base-calling methods on 14000 reads of the *E. coli* data set.

The results in Figure 8.9 are based on using information from both raw intensities and flowgram values. If only flowgram values are used, the prediction accuracy is slightly smaller, but still larger as compared to the competing base-callers. This is seen from Table 8.1.

The sensitivity of the base-calling accuracies obtained with HPCall is examined by considering 10 different training data sets. Each training data set is selected by randomly sampling 1000 wells from the 15000 wells available in the *E. coli* data set. The model is fitted on the training data and the other 14000 wells are used to obtain the prediction accuracies. Figure 8.10 shows the distributions of the prediction accuracies for the different nucleotide types. The variances are

Table 8.1: Prediction accuracy (in percentage correctly called HPLs) for the different base-calling methods separated by nucleotide type (fg = flowgram value).

	correct HPLs (%)				
	A	C	G	T	Overall
HPCall	99.86	99.95	99.94	99.88	99.91
HPCall (only fg)	99.80	99.94	99.94	99.86	99.89
native 454	99.72	99.94	99.93	99.84	99.86
Pyrobayes	99.50	99.85	99.84	99.67	99.72

small which means that the prediction accuracies are very stable across different training data sets. The standard deviations of the prediction accuracies range from 0.000024 (for nucleotide C) to 0.000047 (for nucleotide T).

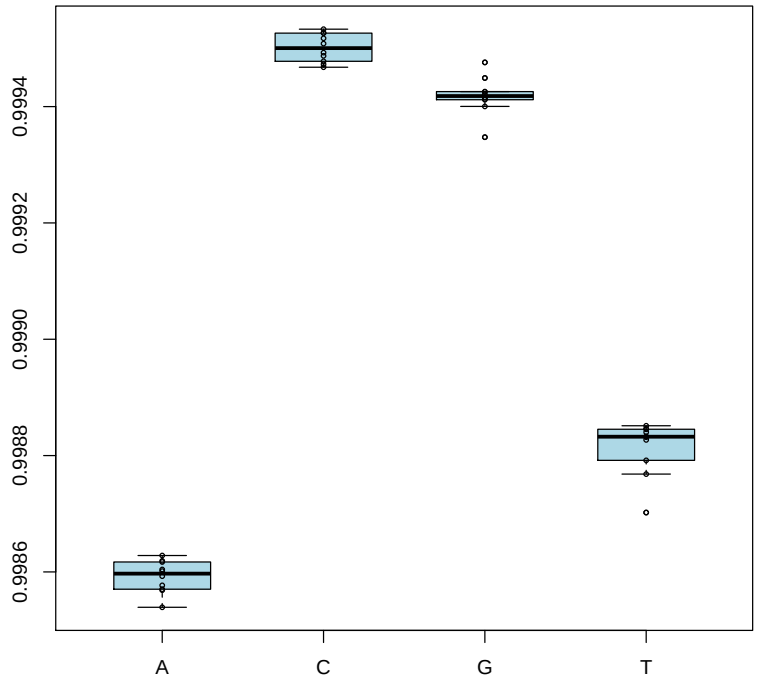


Figure 8.10: Boxplots of prediction accuracies for HPCall across multiple different training sets, separated by nucleotide type.

8.3.1.2 Read-wise assessment and sequence variant analysis

The base-called reads are mapped to the reference sequence using the specialized alignment programs ssaha2 (Ning et al., 2001) and subread (`http://sourceforge.net/projects/subread/`). In the mapping of the HPCall reads the traditional Phred-like quality scores produced by HPCall, without sign information, are used. The read-wise error rate of HPCall is compared to that of the native 454 base-caller. The results are given in Table 8.2. For this data set mapping percentages of 99.47% (ssaha2) and 99.43% (subread) are obtained. HPCall appears to lead to more perfect-matching reads than the native 454 base-caller. This evidently leads to a higher percentage of 454 reads with at least one mismatch to the reference genome as compared to reads generated by HPCall.

Table 8.2: Percentage of reads with different numbers of mismatches in the mapping between the reads produced by either HPCall or the native 454 base-caller and the *E. coli* K-12 reference sequence. For mapping ssaha2 or subread is used.

Mapping	ssaha2		subread	
	HPCall	native 454	HPCall	native 454
Number of errors per read (%)				
0	66.03	56.37	69.42	60.42
1	22.97	26.78	22.25	26.08
2	6.81	10.20	5.53	8.53
3	2.45	3.79	1.81	3.11
4	0.90	1.54	0.60	1.12
5	0.84	1.32	0.40	0.74

The variant calling program ssahaSNP (Ning et al., 2005) is used to compute the number of sequence variants, both for SNPs and indels, of the mapped reads. False positive calls are determined by comparing the base-calls to the *E. coli* K-12 strain reference genome. A reduction of the number of sequence variants with 40% is obtained when using HPCall as compared to the native 454 base-caller. The decrease is observed both for indels and for SNPs (see Table 8.3).

Table 8.3: Detected number of sequence variants for the *E. coli* data set using ssahaSNP for HPCall and the native 454 base-caller.

SNP calling	ssahaSNP	
	HPCall	native 454
Indels	4954	8388
SNPs	528	756
Total sequence variants	5482	9144

8.3.2 Quality scores and base-calling probabilities

HPCall provides conditional probabilities for each HPL in each flow, given all covariates in the model. These probabilities are thus the most direct way to quantify the base-calling uncertainty. In addition, they can also be used to compute Phred-like quality scores as generated by the native 454 base-caller and by Pyrobayes. These Phred scores are designed to have a maximum value of 40. Therefore, HPCall quality scores larger than 40 are trimmed to 40 too so as to make the methods more comparable.

The Phred quality scores calculated by the different base-callers are compared to *observed* quality scores, which are computed following a procedure that is also applied in Brockman et al. (2008). Based on the *E. coli* reference data set all bases with an equal quality score are grouped together. Subsequently, the proportion of overcalls is computed for each group. An observed quality score is calculated as $QS_{\text{observed}} = -10 \log_{10}(\text{observed overcall error rate})$. Figure 8.11 shows the results of this comparison. Both for HPCall and for the native 454 base-caller the predicted quality scores seem to reflect the observed quality quite well, and the quality score assignment seems equally good (upper panel of Figure 8.11). For Pyrobayes the performance is clearly worse, as predicted high quality scores overestimate the true quality of the base-calls. We also observe that HPCall generates more high quality scores than the other two base-callers (lower panel of Figure 8.11). As an illustration, HPCall assigns to 95% of the called bases a quality score of 30 or more, whereas this cumulative base fraction is only 82% for the native 454 base-caller and 54% for Pyrobayes.

Figure 7.8 in the introductory Chapter 7 illustrated that the native 454 quality scores do not give any insight on the nature of potential errors, i.e. whether it is more likely that a potential under-

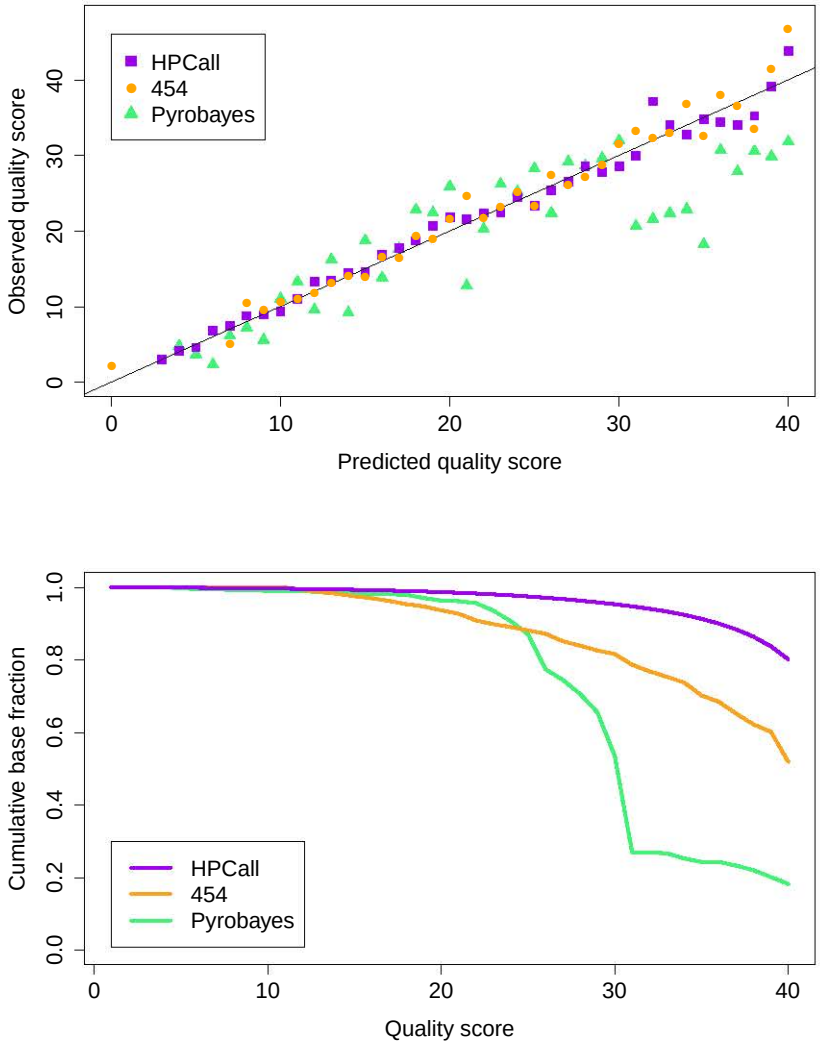


Figure 8.11: Comparison of quality score assignment. Top: Observed versus predicted quality score. Bottom: Cumulative proportion of called bases versus the assigned quality score.

call or overcall is made. Figure 8.12 shows the same plot of overcalls for reference sequences with reference HPL 3 for the HPCall quality scores QS_{over}. This figure clearly indicates that overcalls are more likely in this situation. This can be seen from the large quality scores associated with HPL 2 and HPL 3, whereas HPL 4 gives smaller quality scores. This insight is not provided by the native 454 quality scores. A similar picture is seen for the undercalls of reference sequences with reference HPL 3, based on QS_{under} (right panel of Figure 8.13). Such a plot can not be made for 454 quality scores, since quality scores for HPL 3 and HPL 4 are not available in case of an undercall (left panel of Figure 8.13). Hence, information with respect to undercalls is not provided in the native quality scores.

Empirical cumulative distribution functions for the combined HPCall quality score QS_{HPCall} for

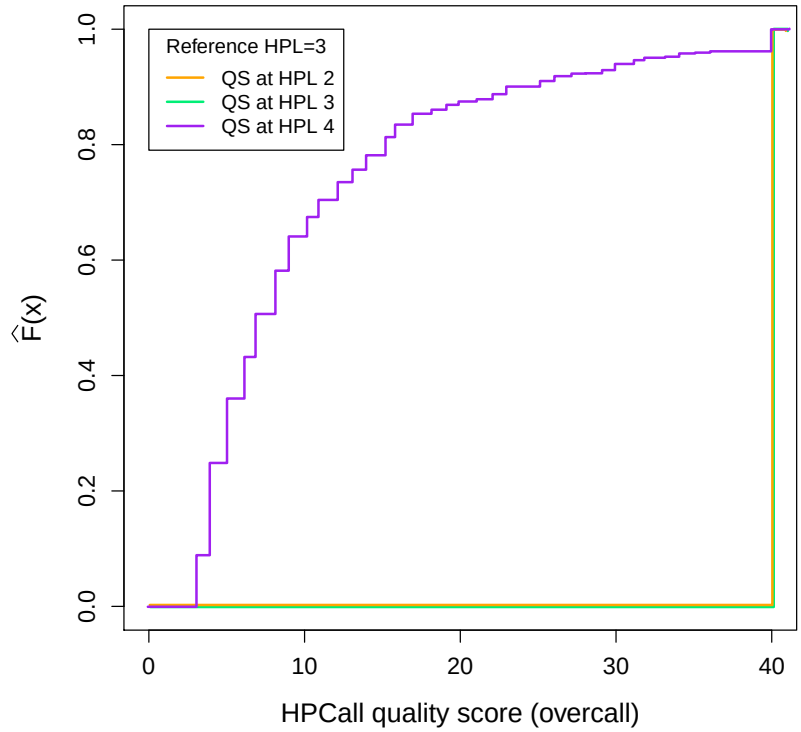


Figure 8.12: Empirical cumulative distribution functions of HPCall quality scores QS_{over} assigned to bases associated with HPL 2, 3 and 4 and for sequences with reference HPL 3, in case of an overcall.

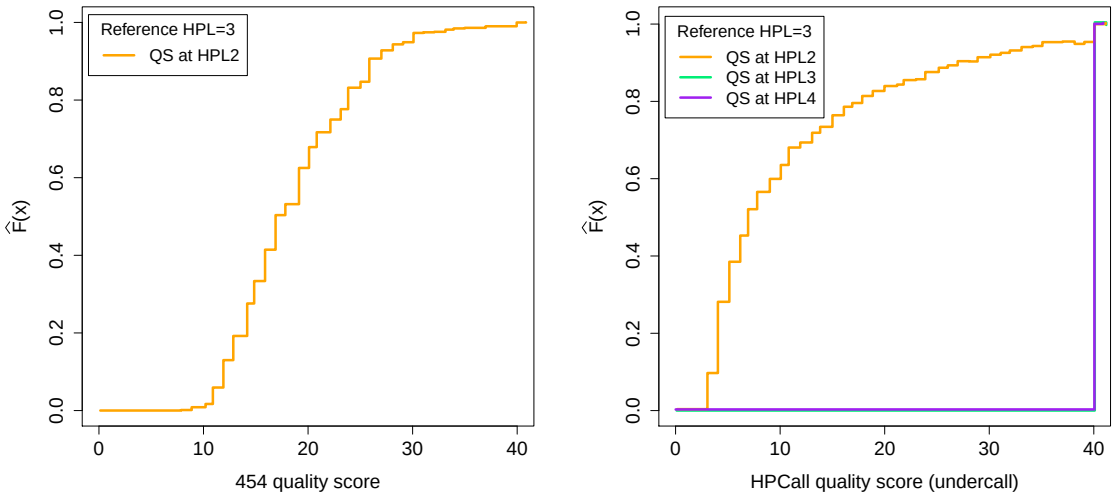


Figure 8.13: Empirical cumulative distribution functions of quality scores for sequences with reference HPL 3, in case of an undercall. Left: 454 quality scores (only those for HPL 2 are available). Right: HPCall quality scores QS_{under} assigned to bases associated with HPL 2, 3 and 4.

sequences with reference HPL 3 in case of an undercall, a correct call and an overcall are displayed in Figure 8.14. Clearly, the sign of these quality scores provides additional information

about whether an undercall or an overcall is more likely. Further down in the text this will be explored in some more detail.

As mentioned before, the HPCall quality scores are based on estimated probabilities of being the correct call. Hence, these probabilities are also very useful to assess the base-calling quality. Their empirical cumulative distribution functions for sequences with reference HPL 3 shows that undercalls and overcalls are associated with larger base-calling uncertainties than correct calls (Figure 8.15). Figure 8.16 shows histograms of HPCall estimated probabilities. In case of a correct call, almost all probabilities at HPL 3 are very close to 1 (upper left panel of Figure 8.16). On the other hand, the cumulative sum of probabilities below HPL 3 in case of an undercall and above HPL 3 in case of an overcall are more evenly distributed between 0.5 and 1 (upper right panel of Figure 8.16). In case of a miscall, the estimated probability at the reference HPL is very often second largest (lower left panel of Figure 8.16). Moreover, the miscalled maximal probability and the probability at the reference HPL nearly always sum to a value close to 1 (lower right panel of Figure 8.16).

The merit of having the base-calling probabilities at our disposal is further demonstrated by examining some examples of indels that are flagged in the sequence variant detection discussed in Section 8.3.1.2. In the first example an undercall with respect to the reference sequence *AAAAA* is called by both HPCall and the native 454 base-caller (see Table 8.4). The native 454 base-caller assigns a quality score of 22 to the fourth *A* in the homopolymer sequence. This score of 22 does not indicate whether it is more likely that the fourth called *A* is a potential under- or overcall. Either way, there is no fifth quality score available to provide more information about a possible fifth *A* to be called. For HPCall we have the additional information that the estimated probability that there should be five *A*'s called is 0.17. This indicates that a miscall for this flow would almost certainly be an undercall. This is confirmed by the negative sign of $QS_{\text{HPCall}} = -8$ for this example. The absolute value of QS_{HPCall} at HPL 4 is also clearly smaller than the 454 quality score at HPL 4. Hence, the HPCall quality score provides a good indication that this call could be problematic. It is obvious that mapping algorithms that take this additional information into account will be able to more reliably map the base-called reads to the reference sequence.

A very similar situation is observed in the case of an overcall (see Table 8.5). A homopolymer stretch *AA* is considered in the reference sequence, but is called as *AAA* by both base-callers.

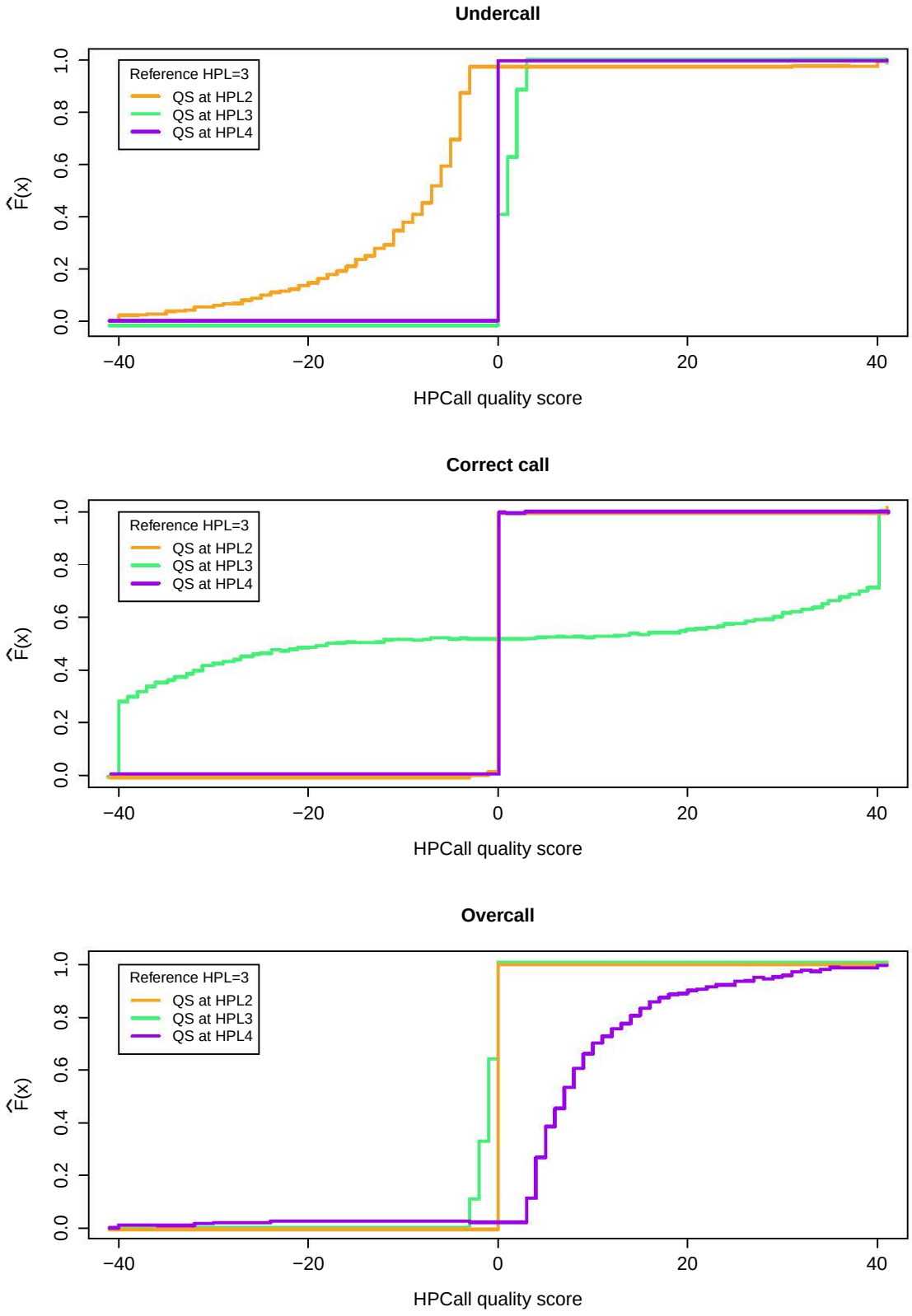


Figure 8.14: Empirical cumulative distribution functions of HPCall quality scores QS_{HPCall} for sequences with reference HPL 3 assigned to bases associated with HPL 2, 3 and 4, in the case of an undercall (upper panel), correct call (middle panel) or overcall (lower panel).

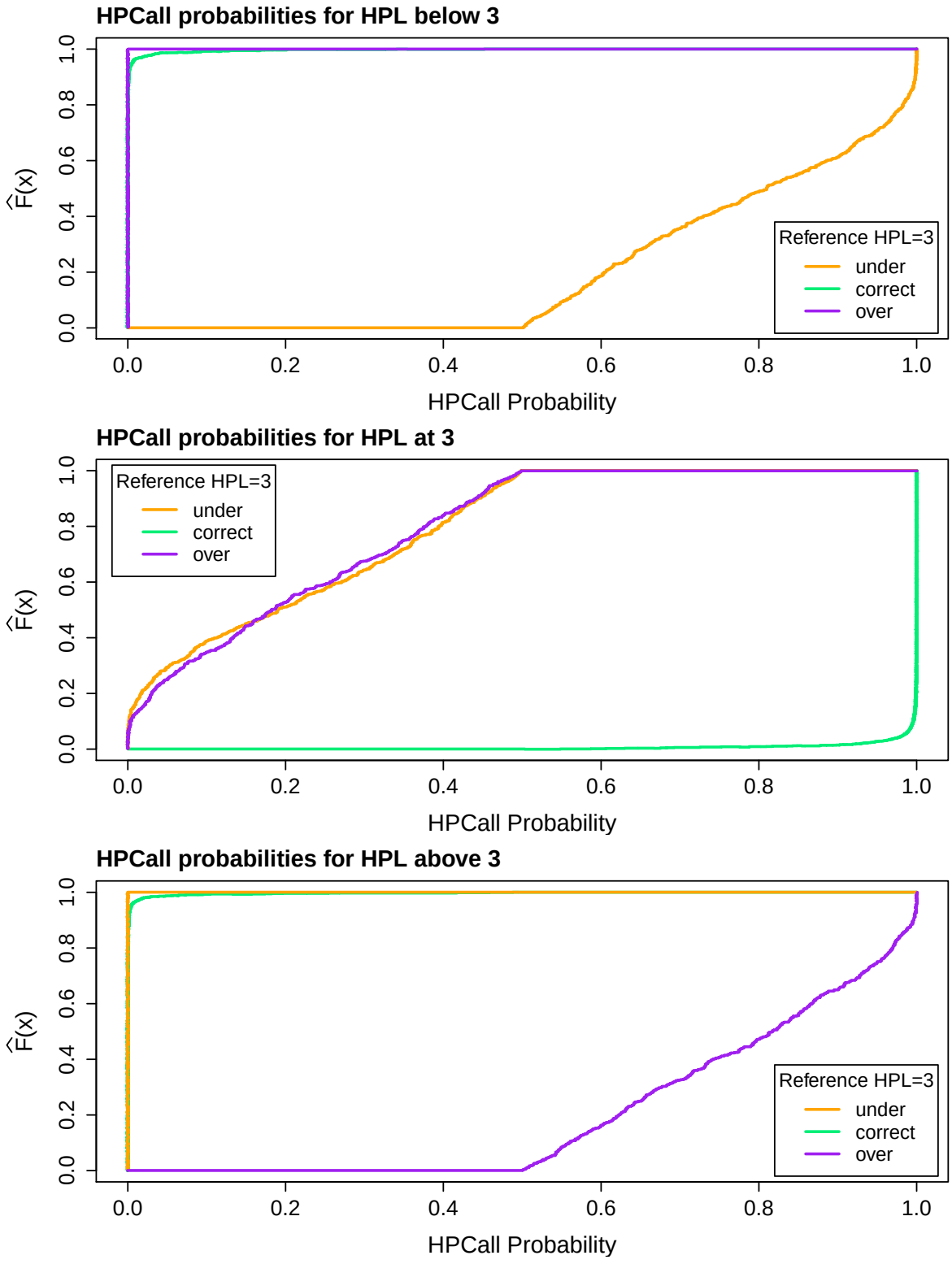


Figure 8.15: Empirical cumulative distribution functions of probabilities estimated by HPCall for sequences with reference HPL 3. The depicted probabilities are the cumulative sum of probabilities below HPL 3 (left panel), the probabilities at HPL 3 (middle panel), and the cumulative sum of probabilities above HPL 3 (right panel). In each panel the empirical cumulative distribution functions are plotted separately in case of an undercall, correct call or overcall.

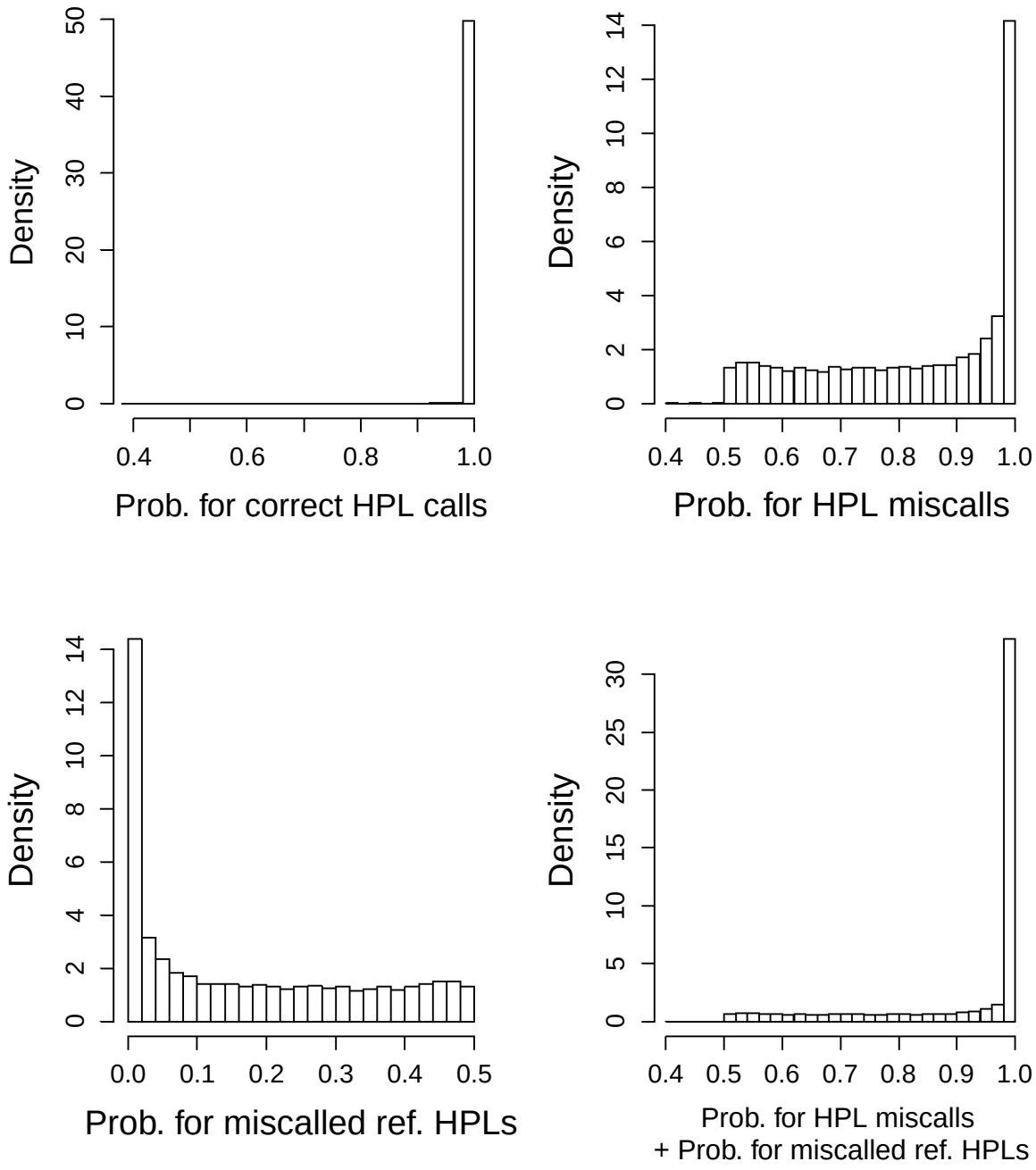


Figure 8.16: Histograms of HPCall estimated probabilities. Upper left panel: maximal estimated probabilities in the case of a correct call. Upper right panel: maximal estimated probabilities in the case of a miscall. Lower left panel: estimated probabilities for the reference HPLs in the case of a miscall. Lower right panel: the sum of the probabilities given in the upper right and lower left panels.

Table 8.4: Base-calling probabilities example 1: undercall (qs x = quality score at HPL x)

reference sequence: AAAAA					
native 454: AAAA					
	qs 2	qs 3	**qs 4**	qs 5	qs 6
QS_{454}	28	22	**22**	-	-
HPCall: AAAA					
	HPL 2	HPL 3	**HPL 4**	**HPL 5**	HPL 6
$\hat{\Pr}\{N_{nc} = n_{bc} \mid \boldsymbol{x}_{bc}, \boldsymbol{y}_{bc}\}$	<1E-15	7.4E-9	**0.83**	**0.17**	6.3E-11
QS_{HPCall}	0	0	**-8**	1	0

Again, the quality score of 23 given by the native 454 base-caller for the third *A* does not give an indication of the probability of having an undercall or an overcall, given that there is a miscall. HPCall on the other hand does provide this information. Since the estimated probability of HPL 2 is 0.29, an overcall seems much more likely than an undercall. Also here this is confirmed by the positive sign of $QS_{\text{HPCall}} = 5$.

Table 8.5: Base-calling probabilities example 2: overcall (qs x = quality score at HPL x)

reference sequence: AA					
native 454: AAA					
	qs 1	qs 2	**qs 3**	qs 4	qs 5
QS_{454}	22	22	**23**	-	-
HPCall: AAA					
	HPL 1	**HPL 2**	**HPL 3**	HPL 4	HPL 5
$\hat{\Pr}\{N_{nc} = n_{bc} \mid \boldsymbol{x}_{bc}, \boldsymbol{y}_{bc}\}$	1.7E-10	**0.29**	**0.71**	3E-9	<1E-15
QS_{HPCall}	0	-1	**5**	0	0

Finally, an example is considered of the special situation where no base is called while there is one in the reference sequence (see Table 8.6). Because the native 454 base-caller only produces

a quality score for every called base, there is no quality score provided in this situation. Hence, there is no indication of the uncertainty of not having a call in the current flow. HPCall estimates the probability of having HPL 0 at 0.75, and of having HPL 1 at 0.25, with an associated QS_{HPCall} of -6, indicating that it is not unlikely that there should be one base called instead of none.

Table 8.6: Base-calling probabilities example 3: 0-1 undercall (qs x = quality score at HPL x)

reference sequence: T					
native 454: -					
	qs 0	qs 1	qs 2	qs 3	qs 4
QS_{454}	**-**	-	-	-	-
HPCall: -					
	HPL 0	**HPL 1**	HPL 2	HPL 3	HPL 4
$\hat{\Pr}\{N_{nc} = n_{bc} \mid \boldsymbol{x}_{bc}, \boldsymbol{y}_{bc}\}$	**0.75**	**0.25**	2.9E-8	<1E-15	<1E-15
QS_{HPCall}	**-6**	1	0	0	0

8.3.3 HPCall software pipeline

A preliminary data preparation step is performed in HPCall before running the Hurdle Poisson base-calling model. In this step several raw data files are merged to create a data set in flow space that can be used as input for the base-calling method. Both the flowgram values (*.sff*) and the raw intensities measured prior to signal processing (*.cwf*) are used. It is also possible to only include information on the flowgram values if the raw intensities are no longer available. For calibration or training of the model a reference sequence is first transformed from nucleotide space to flow space according to the flow order used in 454 sequencing (T, A, C, G). Next, the flowgram values and raw intensities are mapped onto these reference HPLs in the corresponding sequencing cycle, together with other relevant information such as added nucleotide type and cycle number. Furthermore, flowgram values and raw intensities in previous and next flows and cycles are calculated to be used as covariates in the base-calling model. Based on the probabilities obtained by the base-calling model four output files are created: (a) a file with the base-called reads in nucleotide space, (b) a file with the associated Phred-like quality scores, (c)

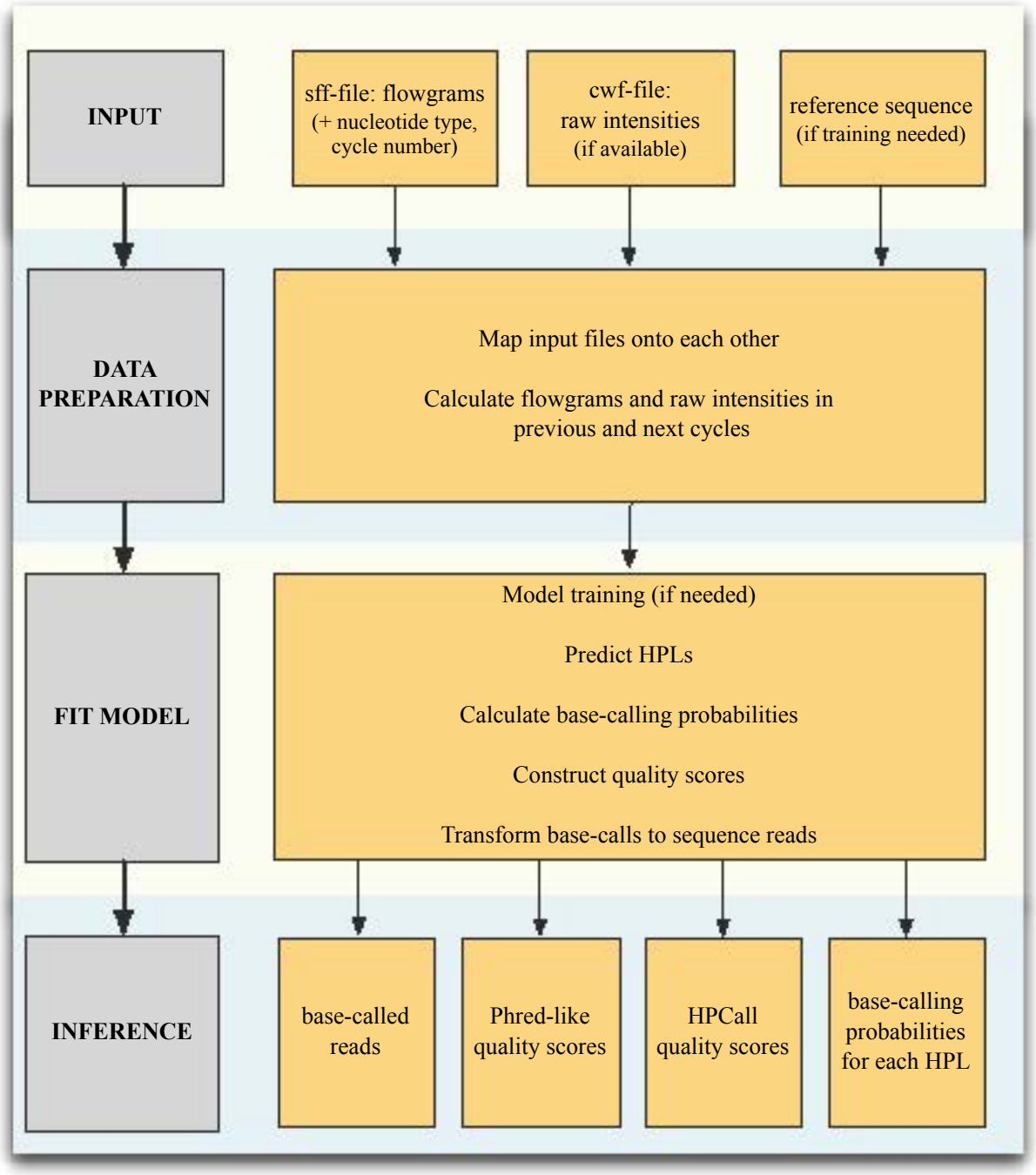


Figure 8.17: Overview of the HPCall base-calling pipeline.

a file with new HPCall quality scores QS_{HPCall} and (d) a file with the base-calling probabilities for each HPL. The pipeline is visualized in Figure 8.17.

The data preparation step is implemented in Perl and stores all required data in a SQL database. Next, these data are imported in R for the actual base-calling. To fit the model the R package VGAM (Yee, 2008) is used. For the weighted Poisson component a new family function was written in VGAM that allows to efficiently conduct the IRLS parameter estimation. The

HPCall software and manual are available at `https://sourceforge.net/projects/hpcall/`.

8.4 Conclusion

In this chapter, we have presented an alternative method for the base-calling of 454 sequencing data based on a weighted Hurdle Poisson model. The method is referred to as HPCall. HPCall uses a probabilistic framework to call the homopolymer lengths in the sequence by modeling 454 noise predictors. Base-calling is assessed based on estimated probabilities for each homopolymer length, which are easily transformed to useful quality scores. Using a reference data set of *Escherichia coli* K-12 strain, we have shown that HPCall produces improved quality scores that are very informative with respect to the occurence of possible insertion and deletion errors, while maintaining a base-calling accuracy that is better than the current one.

Appendix A: Derivation of expected Fisher information for weighted Poisson component

Based on the density of the weighted Poisson distribution the log-likehood for observation n_{bc} can be written as

$$\log L_{bc}(\lambda_{bc}, \theta; n_{bc}) = n_{bc} \log \lambda_{bc} - \log n_{bc}! - \theta(n_{bc} - \lambda_{bc})^2 - \log W_{bc}. \tag{8.10}$$

For clarity of exposition, the indices bc are dropped in the expressions that follow. To obtain the maximum likelihood estimator for λ and θ, the scores are computed. Because of the log-link in Model (8.7), it is convenient to solve for $\log \lambda$.

$$\frac{\partial \log L}{\partial \log \lambda} = n + 2\theta(n-\lambda)\lambda - \frac{\partial \log W}{\partial \lambda}\frac{\partial \lambda}{\partial \log \lambda}. \tag{8.11}$$

First an expression for $\frac{\partial \log W}{\partial \lambda}$ is derived:

$$\begin{aligned}
\frac{\partial \log W}{\partial \lambda} &= \frac{\partial \left(\sum_{n=0}^{\infty} \frac{\lambda^n e^{-\theta(n-\lambda)^2}}{n!} \right)}{\partial \lambda} \frac{1}{W} \\
&= \sum_{n=0}^{\infty} n\lambda^{n-1} e^{-\theta(n-\lambda)^2} \frac{1}{W} + \sum_{n=0}^{\infty} \lambda^n 2\theta(n-\lambda) e^{-\theta(n-\lambda)^2} \frac{1}{W} \\
&= \mathrm{E}\left\{ \frac{N}{\lambda} \right\} + \mathrm{E}\left\{ 2\theta(N-\lambda) \right\} \\
&= (\frac{1}{\lambda} + 2\theta) \ \mathrm{E}\left\{ N \right\} - 2\theta\lambda.
\end{aligned} \tag{8.12}$$

We plug the result of (8.12) into (8.11), resulting in

$$\begin{aligned}
\frac{\partial \log L}{\partial \log \lambda} &= n + 2\theta(n-\lambda)\lambda - (\frac{1}{\lambda} + 2\theta) \ \mathrm{E}\left\{ N \right\} \lambda + 2\theta\lambda\lambda \\
&= (1 + 2\theta\lambda)n - (1 + 2\theta\lambda) \ \mathrm{E}\left\{ N \right\} \\
&= (1 + 2\theta\lambda)(n - \mathrm{E}\left\{ N \right\}).
\end{aligned} \tag{8.13}$$

We also need an expression for $\frac{\partial \log W}{\partial \theta}$:

$$\begin{aligned}\frac{\partial \log W}{\partial \theta} &= \frac{\partial \left(\sum_{n=0}^{\infty} \frac{\lambda^n e^{-\theta(n-\lambda)^2}}{n!}\right)}{\partial \theta} \\ &= -\sum_{n=0}^{\infty} \frac{\lambda^n e^{-\theta(n-\lambda)^2}}{n!}(n-\lambda)^2 \\ &= -\operatorname{E}\left\{(N-\lambda)^2\right\}. \end{aligned} \tag{8.14}$$

Upon using (8.14) and (8.13), we have

$$\frac{\partial \log L}{\partial \theta} = -(n-\lambda)^2 + \operatorname{E}\left\{(N-\lambda)^2\right\}, \tag{8.15}$$

and

$$\frac{\partial^2 \log L}{\partial \left(\log \lambda\right)^2} = 2\theta\lambda(n - \operatorname{E}\{N\}) + (1+2\theta\lambda)\frac{\partial(n - \operatorname{E}\{N\})}{\partial \lambda}\frac{\partial \lambda}{\partial \log \lambda}. \tag{8.16}$$

An expression for $\frac{\partial(n-E\{N\})}{\partial \lambda}$ is first derived:

$$\begin{aligned}\frac{\partial(n - E\{N\})}{\partial \lambda} &= -\frac{\partial \operatorname{E}\{N\}}{\partial \lambda} \\ &= -\frac{\partial \sum_{n=0}^{\infty} n \frac{\lambda^n e^{-\theta(n-\lambda)^2}}{W}}{\partial \lambda} \\ &= -\operatorname{E}\left\{\frac{N^2}{\lambda}\right\} - \operatorname{E}\{2\theta N(N-\lambda)\} + \sum_{n=0}^{\infty} \frac{n\lambda^n e^{-\theta(n-\lambda)^2}}{W^2}\frac{\partial W}{\partial \lambda} \\ &= -\frac{1}{\lambda}\operatorname{E}\left\{N^2\right\} - 2\theta \operatorname{E}\left\{N^2\right\} + 2\theta\lambda \operatorname{E}\{N\} + \operatorname{E}\{N\}\frac{\partial W}{W \partial \lambda} \\ &= -\frac{1}{\lambda}\operatorname{E}\left\{N^2\right\} - 2\theta \operatorname{E}\left\{N^2\right\} + 2\theta\lambda \operatorname{E}\{N\} + \operatorname{E}\{N\}\frac{\partial \log W}{\partial \lambda} \\ &= -(\frac{1}{\lambda} + 2\theta)\operatorname{E}\left\{N^2\right\} + 2\theta\lambda \operatorname{E}\{N\} + \operatorname{E}\{N\}(\frac{1}{\lambda} + 2\theta)\operatorname{E}\{N\} - 2\theta\lambda \operatorname{E}\{N\} \\ &= -(\frac{1}{\lambda} + 2\theta)\operatorname{Var}\{N\}. \end{aligned} \tag{8.17}$$

We plug the result of (8.17) into (8.16), resulting in

$$\frac{\partial^2 \log L}{\partial \, (\log \lambda)^2} = 2\theta\lambda(n - \mathrm{E}\{N\}) - (1 + 2\theta\lambda)(\frac{1}{\lambda} + 2\theta) \, \mathrm{Var}\{N\} \, \lambda$$

$$= 2\theta\lambda(n - \mathrm{E}\{N\}) - (1 + 2\theta\lambda)^2 \, \mathrm{Var}\{N\} \, . \tag{8.18}$$

From this result it follows that

$$\mathrm{E}\left\{ \frac{\partial^2 \log L}{\partial \, (\log \lambda)^2} \right\} = -(1 + 2\theta\lambda)^2 \, \mathrm{Var}\{N\} \, . \tag{8.19}$$

Next, $\frac{\partial^2 \log L}{\partial \theta^2}$ is derived:

$$\begin{aligned}
\frac{\partial^2 \log L}{\partial \theta^2} &= \frac{\partial \, \mathrm{E}\{(N-\lambda)^2\}}{\partial \theta} \\
&= \frac{\sum_{n=0}^{\infty} (n-\lambda)^2 \frac{\lambda^n}{n!} \frac{\partial e^{-\theta(n-\lambda)^2}}{\partial \theta}}{W} - \frac{\sum_{n=0}^{\infty} (n-\lambda)^2 \frac{\lambda^n}{n!} e^{-\theta(n-\lambda)^2}}{W^2} \frac{\partial W}{\partial \theta} \\
&= -\mathrm{E}\left\{(N-\lambda)^4\right\} - \mathrm{E}\left\{(N-\lambda)^2\right\} \frac{\partial \log W}{\partial \theta} \\
&= -\mathrm{E}\left\{(N-\lambda)^4\right\} + \mathrm{E}\left\{(N-\lambda)^2\right\} \, \mathrm{E}\left\{(N-\lambda)^2\right\} \\
&= -\mathrm{Var}\left\{(N-\lambda)^2\right\} .
\end{aligned} \tag{8.20}$$

To obtain an expression for $\frac{\partial^2 \log L}{\partial \theta \, \partial \log \lambda}$ we start from (8.15),

$$\frac{\partial^2 \log L}{\partial \theta \, \partial \log \lambda} = 2(n-\lambda)\lambda + \lambda \frac{\partial \, \mathrm{E}\{(N-\lambda)^2\}}{\partial \lambda} . \tag{8.21}$$

To this end $\frac{\partial \, \mathrm{E}\left\{(N-\lambda)^2\right\}}{\partial \lambda}$ is first computed:

$$
\begin{aligned}
\frac{\partial \,\mathrm{E}\left\{(N-\lambda)^2\right\}}{\partial \lambda} &= \frac{\partial \left(\sum_{n=0}^{\infty}(n-\lambda)^2 \frac{\lambda^n e^{-\theta(n-\lambda)^2}}{n!W}\right)}{\partial \lambda} \\
&= -2\,\mathrm{E}\left\{N-\lambda\right\} + \sum_{n=0}^{\infty}(n-\lambda)^2 \frac{\partial \left(\frac{\lambda^n e^{-\theta(n-\lambda)^2}}{n!W}\right)}{\partial \lambda} \\
&= -2\,\mathrm{E}\left\{N-\lambda\right\} + \sum_{n=0}^{\infty}(n-\lambda)^2 \frac{e^{-\theta(n-\lambda)^2}}{n!W} n\lambda^{n-1} \\
&\quad + \sum_{n=0}^{\infty}(n-\lambda)^2 \frac{\lambda^n e^{-\theta(n-\lambda)^2}}{n!W} 2\theta(n-\lambda) - \sum_{n=0}^{\infty}(n-\lambda)^2 \frac{\frac{\lambda^n e^{-\theta(n-\lambda)^2}}{n!}}{W^2} \frac{\partial W}{\partial \lambda} \\
&= -2\,\mathrm{E}\left\{N-\lambda\right\} + \mathrm{E}\left\{(N-\lambda)^2 \frac{N}{\lambda}\right\} + 2\theta\,\mathrm{E}\left\{(N-\lambda)^3\right\} \\
&\quad - \mathrm{E}\left\{(N-\lambda)^2\right\} \frac{\partial \log W}{\partial \lambda} \\
&= -2\,\mathrm{E}\left\{N-\lambda\right\} + \frac{1}{\lambda}\,\mathrm{E}\left\{(N-\lambda)^2 N\right\} + 2\theta \mathrm{E}\left\{(N-\lambda)^2 N - (N-\lambda)^2\lambda\right\} \\
&\quad - \mathrm{E}\left\{(N-\lambda)^2\right\} \left((\frac{1}{\lambda}+2\theta)\mathrm{E}\left\{N\right\} - 2\theta\lambda\right) \\
&= -2\,\mathrm{E}\left\{N-\lambda\right\} + (\frac{1}{\lambda}+2\theta)\,\mathrm{E}\left\{(N-\lambda)^2 N\right\} - 2\theta\lambda \mathrm{E}\left\{(N-\lambda)^2\right\} \\
&\quad - (\frac{1}{\lambda}+2\theta)\,\mathrm{E}\left\{(N-\lambda)^2\right\}\,\mathrm{E}\left\{N\right\} + 2\theta\,\mathrm{E}\left\{(N-\lambda)^2\right\} \\
&= -2\,\mathrm{E}\left\{N-\lambda\right\} + (\frac{1}{\lambda}+2\theta)\,\mathrm{Covar}\left\{(N-\lambda)^2, N\right\}. \qquad (8.22)
\end{aligned}
$$

We now plug (8.22) into (8.21). This gives

$$
\frac{\partial^2 \log L}{\partial\theta\;\partial \log \lambda} = 2(n-\lambda)\lambda - 2\lambda\,\mathrm{E}\left\{N-\lambda\right\} + (1+2\theta\lambda)\,\mathrm{Covar}\left\{(N-\lambda)^2, N\right\}. \qquad (8.23)
$$

The expected value then equals

$$
\mathrm{E}\left\{\frac{\partial^2 \log L}{\partial\theta\;\partial \log \lambda}\right\} = (1+2\theta\lambda)\,\mathrm{Covar}\left\{(N-\lambda)^2, N\right\}. \qquad (8.24)
$$

Thus, the expected Fisher Information matrix can be written as

$$
\mathrm{E}\left\{I\right\} = \begin{bmatrix} (1+2\theta\lambda)^2\,\mathrm{Var}\left\{N\right\} & -(1+2\theta\lambda)\,\mathrm{Covar}\left\{(N-\lambda)^2, N\right\} \\ (1+2\theta\lambda)\,\mathrm{Covar}\left\{(N-\lambda)^2, N\right\} & \mathrm{Var}\left\{N-\lambda\right\} \end{bmatrix}. \qquad (8.25)
$$

Chapter 9

A statistical method for the detection of DNA sequence variants from 454 sequencing data

9.1 Introduction

In this chapter we focus on an application of 454 sequencing data that is situated more downstream in the data analysis pipeline compared to the base-calling that was discussed in Chapter 8. More specifically, the detection of DNA sequence variants in homopolymers is considered.

Humans are diploid organisms having all their hereditary information organized in 23 pairs of homologous chromosomes. For each chromosome pair, one chromosome is inherited from the mother and the other from the father. Hence, each individual typically possesses two copies of each gene or genetic *locus*, which is the location of a gene or DNA sequence on the chromosome. One such a copy is referred to as an allele. From the Human Genome Project, which succeeded in assembling the first human reference genome in 2003, it was clear that all humans, except identical twins, have a unique DNA sequence (International Human Genome Sequencing Consortium, 2004). All differences in the DNA sequence account for 0.1% of the human genome, which corresponds to around 3 million nucleotides. Due to this human genetic variation, it often occurs that the two alleles of a locus in an individual are not identical, but exhibit such DNA sequence variants.

Depending on the point of reference, two levels of DNA sequence variation may be distinguished. The first level refers to sequence variants between the two alleles of an individual, which implies heterozygosity. The individual is homozygous for a locus if the DNA sequence of the two alleles are identical. Secondly, the variation of the alleles with respect to the human reference sequence may also be considered. In most cases individuals who are heterozygous for a certain locus will have one allele that is identical to the reference sequence, and one allele that is different. Homozygosity will often indicate that both identical alleles are also identical to the reference sequence. However, it may also occur that heterozygous or homozygous alleles both differ from the reference sequence, though this event is rather rare.

Another categorization of DNA sequence variants can be made based on the type of molecular variation. Three of the most important classes are single nucleotide polymorphisms (SNPs), copy number variations (CNVs), and short insertion and deletion variants (indels). While SNPs contain a variation with just one nucleotide in the DNA sequence substituted by a nucleotide of another type, CNVs consist of varying numbers of repeated DNA sequences, which amount from 1kb to several Mb (Redon et al., 2006). The type of variation observed in short indels is the insertion or deletion of 1 or more nucleotides in the DNA sequence. These different types of variants lead to phenotypical variability and, together with environmental factors, contribute to an increased susceptibility to the development of many diseases, such as cancer (Stratton et al., 2009). An accurate detection of DNA sequence variation is a critical first step in attaining an improved understanding of the development of these diseases, which may eventually result in creating appropriate medicine. In this chapter, we are particularly interested in the detection of heterozygosity in homopolymeric regions, associated with the presence of indels.

For a long time Sanger sequencing has been the most commonly used technology in DNA sequence variant detection studies. Nowadays, however, next-generation sequencing (NGS) platforms are more frequently applied. Since they are fast and relatively cheap, the implementation of these NGS systems for sequence variant detection in a diagnostic setting promises to play an important role in personalized medicine (Mills et al., 2011). Just like for other NGS technologies, detection of DNA sequence variants of specific genetic loci with 454 sequencing requires a customized library preparation. In this case the DNA fragments are collected for the genomic region of interest using target-specific primers. This type of sequencing is often called amplicon sequencing and the DNA fragments are referred to as the amplicons.

To date, no statistical methods have been developed for the analysis of 454 amplicon sequencing data used in the context of diagnostic testing. Instead, a database-oriented analysis pipeline has been described to conduct variant calling in this setting (De Schrijver et al., 2010). The method is based on the determination of nucleotide-level differences between the sequenced read and the reference sequence. False positives and negatives are subsequently reduced by applying a collection of filters. These filters may include (De Leeneer et al., 2011; Coppieters et al., 2012): (1) the read coverage or the number of reads sequenced for a specific amplicon has to be above a minimum value; (2) the variant needs to be present in at least a minimum percentage of reads; (3) a high quality score for the base-call associated with the variant is required; (4) variants in homopolymer stretches above a certain length (e.g. 6) are treated as non-reliable calls and discarded. Clearly, the use of these ad-hoc filters is rather artificial and does not provide a trustworthy method for distinguishing between true variants and base-calling errors.

In Chapter 7 we have shown a typical example of the distribution of flowgram values corresponding to a base-call for homopolymer lengths (HPLs) of 0, 1, 2 and 3 (see Figure 7.7). Flowgram values in the tails of the respective distributions are associated with low-quality base-calls. From this observation it is clear that a variant calling method for 454 data has the potential to benefit from using the flowgram values instead of the homopolymer lengths.

This chapter is organized as follows. Section 9.2 introduces the 454 amplicon sequencing data set that is used in this chapter. The statistical method developed for variant detection from 454 amplicon sequencing data is presented in Section 9.3. The results of the analysis and an empirical assessment of the method's performance are described in Section 9.4. Finally, Section 9.5 summarizes some conclusions for this chapter.

9.2 Amplicon sequencing data on *BRCA1*- and *BRCA2*-genes

BRCA1 and *BRCA2* are two cancer tumor suppressor genes whose proteins have a function in repairing chromosomal damage and as such help to control normal cell growth. Mutations in these genes may lead to disfunctional proteins and to an increased risk for developing breast or ovarian cancer (e.g. Nathanson et al., 2001). Genetic testing by screening for mutations in these genes is an important prevention tool. To enable the testing of an increasing number of blood samples within shorter turnaround times, high-throughput screening is required (De Leeneer

et al., 2011). Also, the prospect of targeted therapeutic agents for tumors diagnosed in *BRCA1* or *BRCA2* mutation carriers, such as specific polymerase enzyme inhibitors, contribute to the expectations for genetic testing (Curtin, 2005).

The original data set consists of 454 amplicon sequencing reads of all the coding regions of the *BRCA1* and *BRCA2* genes taken from blood samples of 123 individuals, as described in De Leeneer et al. (2011). For the development of the variant calling method, a subset of these data, for which the flowgram values are still available, is used. It consists of sequencing data from 68 amplicons spread over the exons of *BRCA1* and (mostly) *BRCA2* for 19 subjects. Only loci with a reference homopolymer length of at least 4 are considered, as these are most challenging in sequence variant detection. A data example is presented in Table 9.1, which gives the first four lines for a certain read in the data set. For every unique combination of *amplicon ID* and *MID*, which is an ID tag to indicate the subject, several reads (indicated by *sequence ID*) are sequenced. At a specific homopolymer reference position there is a flowgram value and an associated base-call available for each read. The variant calling method described in Section 9.3 will be applied to the flowgram values over all reads for each unique combination of amplicon, subject and reference position.

Table 9.1: Example of the data set format. *sequence ID*: ID tag to indicate a certain read; *amplicon ID*: ID tag to indicate the sequenced amplicon of the *BRCA* gene (6th amplicon of the 5th exon of *BRCA2* in this case); *MID*: ID tag to indicate the subject; *ref pos*: position of the homopolymer on the reference sequence; *nucl*: nucleotide type at this locus; *HPLref*: homopolymer length of the reference sequence; *fg value*: the measured flowgram value corresponding with this locus for this read; *454 bc*: the base-call made by the 454 software for this homopolymer.

sequence ID	amplicon ID	MID	ref pos	nucl	HPLref	fg value	454 bc
F2V03PG01A4ADL	BRCA2_05_06	MID11	147	T	5	4.63	5
F2V03PG01A4ADL	BRCA2_05_06	MID11	157	T	6	5.78	6
F2V03PG01A4ADL	BRCA2_05_06	MID11	163	A	4	3.99	4
F2V03PG01A4ADL	BRCA2_05_06	MID11	188	T	4	3.82	4

9.3 Statistical variant detection method for 454 amplicon sequencing

Let y_{ijk} be a flowgram value measured for sequence read k ($k = 1, \ldots, K_{ij}$) at homopolymer locus j ($j = 1, \ldots, J$) of subject (MID) i ($i = 1, \ldots, I$), where I is the total number of subjects considered in the sequencing experiment, J is the total number of reference homopolymer loci for the subjects. This is usually identical for all subjects. Further, K_{ij} is the total number of observed flowgram values for locus j of subject i. This is also referred to as the coverage.

Without loss of generality, and because the estimation and hypothesis testing procedure takes place independently for each locus j and subject i, we will drop indices i and j in the remainder. Since we are considering amplicon resequencing experiments for a diploid organism, we assume that each flowgram value y_k results from a two-component normal mixture density

$$f(y_k; \boldsymbol{\theta}) = \psi_1 f_1(y_k; \mu_1, \sigma_1^2) + (1 - \psi_1) f_2(y_k; \mu_2, \sigma_2^2), \tag{9.1}$$

with $f_l(y_k; \mu_l, \sigma_l^2) = (2\pi\sigma_l^2)^{-1/2}\exp(-\frac{(y_k-\mu_l)}{2\sigma_l^2})$, where $l = 1, 2$, and $\boldsymbol{\theta} = (\mu_1, \sigma_1^2, \psi_1, \mu_2, \sigma_2^2)^T$. The mixing parameter ψ_1 denotes the probability that y_k belongs to the first normal density component; $\psi_2 = 1 - \psi_1$ is then the probability that y_k is taken from the second normal density component.

If subject i is homozygous for locus j, all y_k ($k = 1, \ldots, K_{ij}$) are assumed to belong to a single-component normal density, i.e. ψ_1 is either 0 or 1. On the other hand, in the case of heterozygosity, we expect ψ_1 and ψ_2 to be 0.5. In reality, however, the mixing proportions may deviate from these expected probabilities, e.g. due to preferential PCR amplification in the library preparation step.

9.3.1 Parameter estimation: EM algorithm

Each flowgram value y_k is supposed to result from only one of the two components. Consider now z_k, a realization of the random variable Z_k, associated with the flowgram value y_k, where

$$z_k = \begin{cases} 1 & \text{if } y_k \text{ belongs to } f_1(y_k; \mu_1, \sigma_1^2) \\ 0 & \text{if } y_k \text{ belongs to } f_2(y_k; \mu_2, \sigma_2^2) \end{cases}. \tag{9.2}$$

From Equation (9.1) it follows that $Pr\{Z_k = 1\} = \psi_1$ and $Pr\{Z_k = 0\} = 1 - \psi_1 = \psi_2$. Hence, $Z_k \sim \text{Binomial}(1, \psi_1)$. In reality, the class labels z_k are unknown. Hence, this may be considered as an incomplete-data problem. In this context, an Expectation-Maximization (EM) algorithm (Dempster et al., 1977) is often used to obtain maximum likelihood estimates of the model parameters. A comprehensive overview of EM algorithms and their extensions is given in McLachlan and Krishnan (1997). These iterative algorithms make use of the completed-data log-likelihood.

Consider the log-likelihood for the completed data of all observations for subject i at locus j, $\boldsymbol{x} = (\boldsymbol{y}, \boldsymbol{z})^T$, with $\boldsymbol{y} = (y_1, \ldots, y_{K_{ij}})$ and $\boldsymbol{z} = (z_1, \ldots, z_{K_{ij}})$,

$$\log L_c(\boldsymbol{\theta}; \boldsymbol{x}) = \sum_{l=1}^{2} \sum_{k=1}^{K_{ij}} z_k \log \psi_l + \sum_{l=1}^{2} \sum_{k=1}^{K_{ij}} z_k \log f_l(y_k; \mu_l, \sigma_l^2). \tag{9.3}$$

Suppose that the algorithm is at iteration $(m+1)$. In the E-step, the expectation of the completed-data log-likelihood $\log L_c$, given the current estimate $\boldsymbol{\theta}^{(m)}$, is calculated. This requires the computation of the expected value of Z_k given $\boldsymbol{\theta}^{(m)}$ and the observed data y_k,

$$\mathrm{E}(Z_k | y_k, \boldsymbol{\theta}^{(m)}) = \mathrm{Pr}_{\boldsymbol{\theta}^{(m)}}(Z_k | y_k) = \tau_1(y_k; \boldsymbol{\theta}^{(m)}). \tag{9.4}$$

The conditional expectation can be written as a posterior probability, which is easily calculated by applying Bayes' rule,

$$\tau_1(y_k; \boldsymbol{\theta}^{(m)}) = \frac{\psi_1^{(m)} f_1(y_k; \mu_1^{(m)}, \sigma_1^{2^{(m)}})}{\sum_{l=1}^{2} \psi_l^{(m)} f_l(y_k; \mu_l^{(m)}, \sigma_l^{2^{(m)}})}. \tag{9.5}$$

In the M-step a maximization takes place of the expected completed-data log-likelihood, given the current estimate $\boldsymbol{\theta}^{(m)}$,

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(m)}) = \sum_{l=1}^{2} \sum_{k=1}^{K_{ij}} \tau_l(y_k; \boldsymbol{\theta}^{(m)}) \log \left\{ \psi_l \; f_l(y_k; \mu_l, \sigma_l^2) \right\}. \tag{9.6}$$

The parameter estimates that maximize $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(m)})$ are computed by ($l = 1, 2$)

$$\hat{\psi}_1^{(m+1)} = \frac{1}{K_{ij}} \sum_{k=1}^{K_{ij}} \hat{\tau}_1(y_k; \boldsymbol{\theta}^{(m)}), \tag{9.7}$$

$$\hat{\mu}_l^{(m+1)} = \frac{\sum_{k=1}^{K_{ij}} \hat{\tau}_l(y_k; \boldsymbol{\theta}^{(m)}) \; y_k}{\sum_{k=1}^{K_{ij}} \hat{\tau}_l(y_k; \boldsymbol{\theta}^{(m)})}, \tag{9.8}$$

$$\hat{\sigma}_l^{2^{(m+1)}} = \frac{\sum_{k=1}^{K_{ij}} \hat{\tau}_l(y_k; \boldsymbol{\theta}^{(m)}) \; \left(y_k - \hat{\mu}_l^{(m+1)} \right)^2}{\sum_{k=1}^{K_{ij}} \hat{\tau}_l(y_k; \boldsymbol{\theta}^{(m)})}. \tag{9.9}$$

The resulting $\boldsymbol{\theta}^{(m+1)}$ is then again used in the E-step of the next iteration and the E-step and the M-step are repeated until convergence.

9.3.2 Detecting zygosity by a Wald-type test

9.3.2.1 Construction of the test statistic

Let μ_1 and μ_2 be the true mean flowgram values of the two normal densities in the two-component mixture $f(y_k;\boldsymbol{\theta})$. Further, let δ be a positive threshold value. For heterozygous loci it must hold that μ_1 and μ_2 differ from each other with a value of at least δ. Heterozygous loci can now be detected by contructing a Wald-type statistical test with hypotheses given by

$$H_0 : |\mu_1 - \mu_2| = \delta \text{ versus } H_1 : |\mu_1 - \mu_2| > \delta, \tag{9.10}$$

with $\delta > 0$.

The associated test statistic can be written as

$$T = \frac{|D| - \delta}{\hat{\sigma}_D}, \tag{9.11}$$

with $D = \hat{\mu}_1 - \hat{\mu}_2$, and $\hat{\sigma}_D = \sqrt{\hat{\sigma}^2_{\hat{\boldsymbol{\mu}},11} - 2\hat{\sigma}_{\hat{\boldsymbol{\mu}},12} + \hat{\sigma}^2_{\hat{\boldsymbol{\mu}},22}}$, which is the estimated standard error (SE) of $(\hat{\mu}_1 - \hat{\mu}_2)$. The derivation of σ_D is shown below.

$$\begin{aligned}
\text{SE}\,(\hat{\mu}_1 - \hat{\mu}_2) &= (\text{Var}\,(\hat{\mu}_1 - \hat{\mu}_2))^{1/2} \\
&= \left((1\ -1)\ \text{Var}(\hat{\boldsymbol{\mu}})\ (1\ -1)^T\right)^{1/2} \\
&= \left((1\ -1) \begin{bmatrix} \sigma^2_{\hat{\boldsymbol{\mu}},11} & \sigma_{\hat{\boldsymbol{\mu}},12} \\ \sigma_{\hat{\boldsymbol{\mu}},12} & \sigma^2_{\hat{\boldsymbol{\mu}},22} \end{bmatrix} (1\ -1)^T\right)^{1/2} \\
&= \sqrt{\sigma^2_{\hat{\boldsymbol{\mu}},11} - 2\sigma_{\hat{\boldsymbol{\mu}},12} + \sigma^2_{\hat{\boldsymbol{\mu}},22}}.
\end{aligned} \tag{9.12}$$

We now make use of the well-known property that the asymptotic variance-covariance matrix of a maximum likelihood estimator (MLE) is given by the inverse of the expected Fisher information matrix (e.g. McCullagh and Nelder, 1989). It can be estimated consistently by the inverse of the observed Fisher information matrix for the observed (incomplete) data likelihood $L(\boldsymbol{\mu};\boldsymbol{y})$ with respect to $\boldsymbol{\mu}$. This estimator is given by

$$\begin{aligned} \mathcal{I}(\hat{\boldsymbol{\mu}}; \boldsymbol{y}) &= -\partial^2 \left. \frac{\log L(\boldsymbol{\mu}; \boldsymbol{y})}{\partial \boldsymbol{\mu} \partial \boldsymbol{\mu}^T} \right|_{\boldsymbol{\mu}=\hat{\boldsymbol{\mu}}} \\ &= \begin{bmatrix} I_{11} & I_{12} \\ I_{21} & I_{22} \end{bmatrix}, \end{aligned} \tag{9.13}$$

with

$$\begin{aligned} I_{11} &= \sum_{k=1}^{K_{ij}} \frac{\hat{\tau}_1(y_k)}{\hat{\sigma}_1^2} - \sum_{k=1}^{K_{ij}} \frac{\hat{\tau}_1(y_k)\hat{\tau}_2(y_k)(y_k - \hat{\mu}_1)^2}{\hat{\sigma}_1^4}, \\ I_{22} &= \sum_{k=1}^{K_{ij}} \frac{\hat{\tau}_2(y_k)}{\hat{\sigma}_2^2} - \sum_{k=1}^{K_{ij}} \frac{\hat{\tau}_1(y_k)\hat{\tau}_2(y_k)(y_{ijk} - \hat{\mu}_2)^2}{\hat{\sigma}_2^4}, \\ I_{12} = I_{21} &= \sum_{k=1}^{K_{ij}} \hat{\tau}_1(y_k)\hat{\tau}_2(y_k) \frac{(y_k - \hat{\mu}_1)}{\hat{\sigma}_1^2} \frac{(y_k - \hat{\mu}_2)}{\hat{\sigma}_2^2}. \end{aligned}$$

The result of (9.13) can now be used in (9.12) to obtain the denominator of T.

9.3.2.2 Null distribution and p-value

The expression to calculate the p-value can be derived from the asymptotic null distribution of T. Consider first the asymptotic distribution of D under the null hypothesis,

$$D \overset{H_0}{\sim} \begin{cases} N(\delta, \sigma_D^2) & \text{if } \mu_1 > \mu_2 \\ N(-\delta, \sigma_D^2) & \text{if } \mu_1 < \mu_2. \end{cases} \tag{9.14}$$

The asymptotic null distribution of T then becomes

$$T \overset{H_0}{\sim} \begin{cases} \frac{|\sigma_D Z + \delta| - \delta}{\sigma_D} & \text{if } \mu_1 > \mu_2 \\ \frac{|\sigma_D Z - \delta| - \delta}{\sigma_D} & \text{if } \mu_1 < \mu_2, \end{cases} \tag{9.15}$$

where Z denotes a random variable with a standard normal distribution.

If $\mu_1 > \mu_2$, we have

$$T \overset{H_0}{\sim} \begin{cases} Z & \text{if } \sigma_D Z + \delta > 0 \\ -Z - \frac{2\delta}{\sigma_D} & \text{if } \sigma_D Z + \delta < 0. \end{cases} \tag{9.16}$$

Analogously, if $\mu_1 < \mu_2$

$$T \overset{H_0}{\sim} \begin{cases} Z - \frac{2\delta}{\sigma_D} & \text{if } \sigma_D Z - \delta > 0 \\ -Z & \text{if } \sigma_D Z - \delta < 0. \end{cases} \tag{9.17}$$

Because for a standard normal random variable it holds that Z and $-Z$ are equal in distribution, Equations (9.16) and (9.17) lead to the same result. The p-value can thus be calculated as follows.

$$\begin{aligned} p &= \Pr\{T > t | H_0\} \\ &= \Pr\left\{Z > t | Z > -\frac{\delta}{\sigma_D}\right\} \Pr\left\{Z > -\frac{\delta}{\sigma_D}\right\} \\ &\quad + \Pr\left\{-Z - 2\frac{\delta}{\sigma_D} > t | Z < -\frac{\delta}{\sigma_D}\right\} \Pr\left\{Z < -\frac{\delta}{\sigma_D}\right\}. \end{aligned} \tag{9.18}$$

Further, we use the asymptotic property that the variance of a consistent estimator converges to 0 with growing sample size. Hence, σ_D vanishes asymptotically and the expression $-\frac{\delta}{\sigma_D}$ approaches $-\infty$. From this result, it becomes clear that $\Pr\left\{Z > -\frac{\delta}{\sigma_D}\right\} \approx 1$ and $\Pr\left\{Z < -\frac{\delta}{\sigma_D}\right\} \approx 0$. The p-value may thus be approximated by

$$\begin{aligned} p &\approx \Pr\left\{Z > t | Z > -\frac{\delta}{\sigma_D}\right\} \\ &= 1 - \Phi(t), \end{aligned} \tag{9.19}$$

with $\Phi(.)$ the cumulative distribution function of a standard normal variable.

A p-value smaller than the significance level leads to the rejection of the null hypothesis of a homozygous locus, in favor of the alternative hypothesis that the data come from a heterozygous variant locus.

9.3.3 Penalized maximum likelihood estimation

An often encountered problem in the application of the EM algorithm for normal density finite mixture models is the singularity of the likelihood function (McLachlan and Peel, 2000). Singularities (or degeneracies) can occur in the optimization process if one of the component means

equals one of the observations in the data set and the variance becomes zero. Consequently, the likelihood goes to infinity and the MLE can not be defined. A somewhat related problem is the sensitivity to outliers when using the ordinary EM algorithm. If the data set contains one or a few outlying observations the algorithm will often tend to cluster these together in one component and assign all the other observations to the second component. The estimated variance of the normal density with the outlying observations will usually be very small. This may lead to an artificial inflation of test statistic T (9.11), possibly inducing many false positive calls. In the context of homopolymer variant detection based on 454 sequencing data, outlying flowgram values for homozygous loci occur because of sequencing errors. We wish to protect the method from calling these as heterozygous in such a situation.

A Bayesian solution for this degeneracy problem has been proposed which also reduces the sensitivity to outliers (Ridolfi and Idier, 1999). This is done by imposing an inverse gamma prior distribution on the variance parameters σ_1^2 and σ_2^2. The completed-data log-likelihood (9.3) can now be adapted to a *penalized* log-likelihood:

$$\log L_{c,P}(\boldsymbol{\mu}, \psi_1; \boldsymbol{x}, \boldsymbol{\sigma}^2) = \sum_{l=1}^{2} \sum_{k=1}^{K_{ij}} z_k \ \left\{ \log \psi_l + \log f_l(y_k; \mu_l, \sigma_l^2) + \log g(\sigma_l^2) \right\}, \tag{9.20}$$

with the inverse gamma density given by

$$g(\sigma_l^2) = \frac{\alpha^{\beta-1}}{\Gamma(\beta-1)} \frac{1}{\sigma_l^{2^\beta}} \exp\left\{ -\frac{\alpha}{\sigma_l^2} \right\} 1_{[0,+\infty)}, \tag{9.21}$$

and with $l = 1, 2$, and α and β hyperparameters. The inverse gamma distribution is known to be conjugate for the variance of a normal distribution. This conjugacy implies substantial advantages for the computation of the posterior distribution of the model parameters. Explicit formulas are still retained for estimating the parameters in the M-step of each iteration in the EM algorithm. Equations (9.7) and (9.8) remain unchanged and Equation (9.9) becomes

$$\hat{\sigma}_l^{2^{(m+1)}} = \frac{2\alpha + \sum_{k=1}^{K_{ij}} \hat{\tau}_l(y_k; \boldsymbol{\theta}^{(m)}) \ \left(y_k - \hat{\mu}_l^{(m+1)} \right)^2}{2\beta + \sum_{k=1}^{K_{ij}} \hat{\tau}_l(y_k; \boldsymbol{\theta}^{(m)})}. \tag{9.22}$$

9.4 Results

9.4.1 Analysis of the BRCA data set

To illustrate the variant calling method we analyze the BRCA data set. As already explained in Section 9.2, a separate mixture model is fitted using the flowgram values for each unique combination of amplicon, subject and reference position of a homopolymer with length of at least 4. From each fitted mixture the estimated parameters of the two normal components are used for calculating the test statistic T. A threshold value of $\delta = 0.5$ is used in the analysis. This value is motivated by the fact that in 454 base-calling flowgram values are only more or less rounded to provide the base-call. Hence, a difference in mean flowgram value of greater than 0.5 is likely to be associated with two different homopolymer lengths and thus originates from a heterozygous variant. The parameters are estimated by penalized maximum likelihood estimation. The value for both hyperparameters α and β is set at 0.4, the same value as used in Ridolfi and Idier (1999).

The normality assumption underlying the variant calling method is first checked for some typical sets of flowgram values from reference loci that are known not to contain sequence variants, following information provided in De Leeneer et al. (2011). The normal QQ plots shown in Figure 9.1 reveal no severe deviations from the normal distribution. This is confirmed by the p-values obtained for the Shapiro-Wilk test of normality (Shapiro and Wilk, 1965), which are also indicated on the plots of Figure 9.1.

All variant detection tests were conducted at the 5% significance level. This resulted in 4 detected loci containing heterozygous variants. Table 9.2 gives the results and some properties for these loci.

The interpretation of the analysis results benefits from a graphical presentation. For the 4 heterozygous variants the results are presented in Figure 9.2. The shape of the mixture density in these plots indeed suggests variation in homopolymer length at the specific locus. The bimodal structure is especially clear in the two upper plots with corresponding p-values of nearly 0.

Figure 9.3 gives two typical examples of loci for which most likely a variant would be called based on the 454 base-calls only (red bars), but which show a unimodal distribution of flowgram

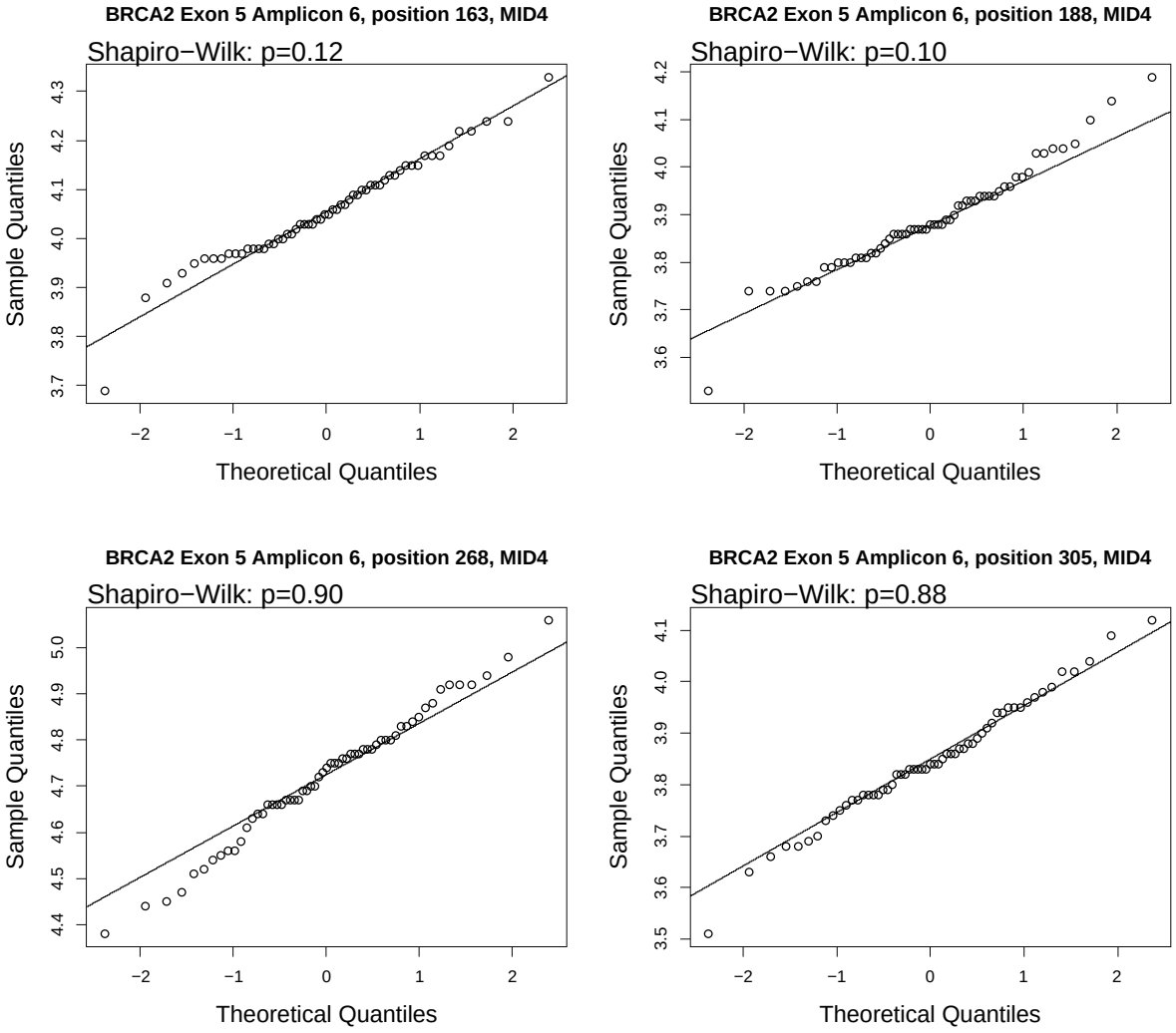


Figure 9.1: QQ plots and Shapiro-Wilk p-values for 4 typical sets of flowgram values from reference loci that do not contain sequence variants

Table 9.2: Overview of the 4 detected heterozygous homopolymer variants in the BRCA data set. The columns showing the different HPLs correspond with the frequency of base-calls determined by the 454 software for the indicated combination of amplicon ID, MID and reference position.

					HPL							
amplicon ID	MID	ref pos	K_{ij}	p-value	5	6	7	8	9	10	11	12
BRCA2_11_07	MID12	196	342	<10e-5	0	4	195	138	5	0	0	0
BRCA2_11_21	MID14	175	73	<10e-5	28	8	34	3	0	0	0	0
BRCA2_10_02	MID6	114	57	0.01173	4	35	17	1	0	0	0	0
BRCA2_03	MID2	45	22	0.01946	0	0	0	0	7	9	4	2

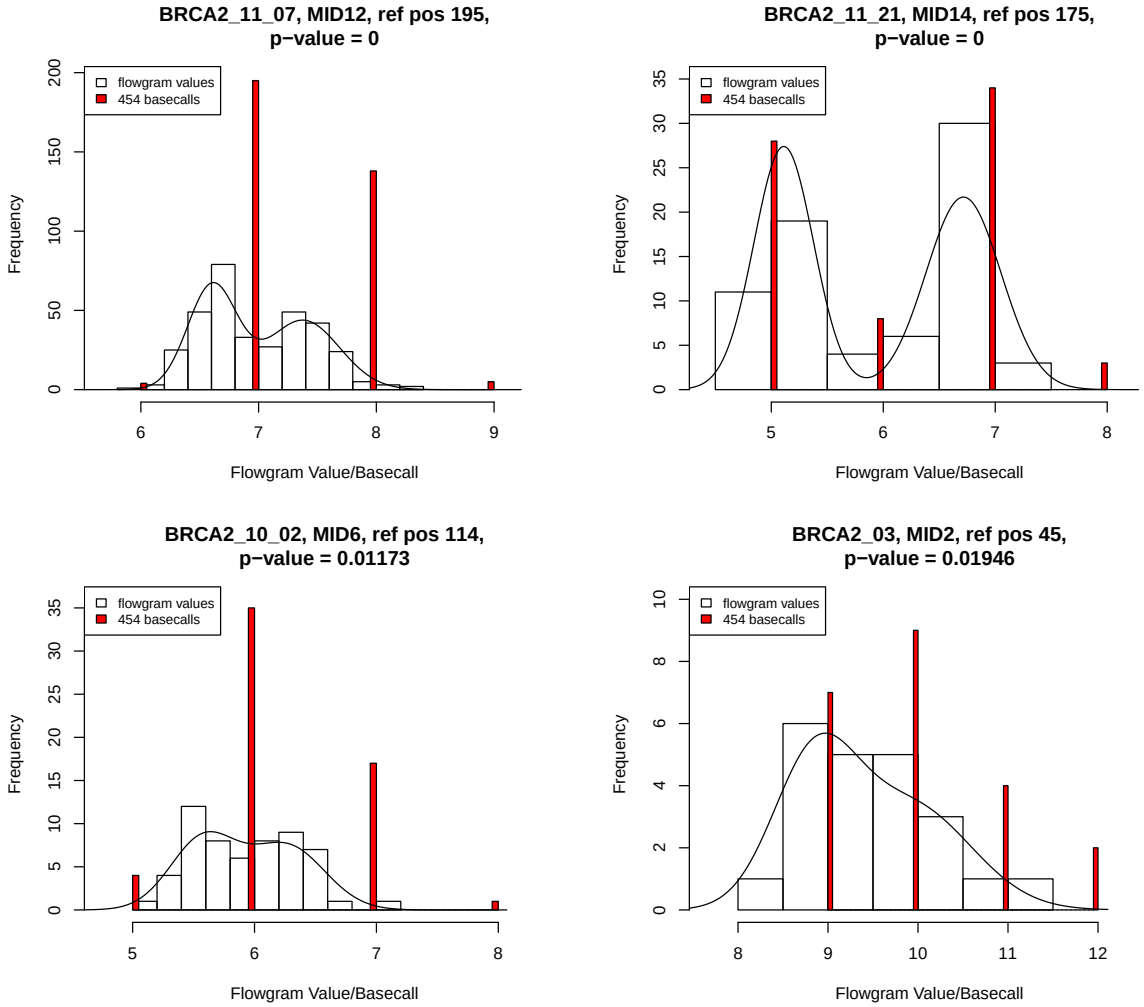


Figure 9.2: Graphical representation of the 4 detected heterozygous variants

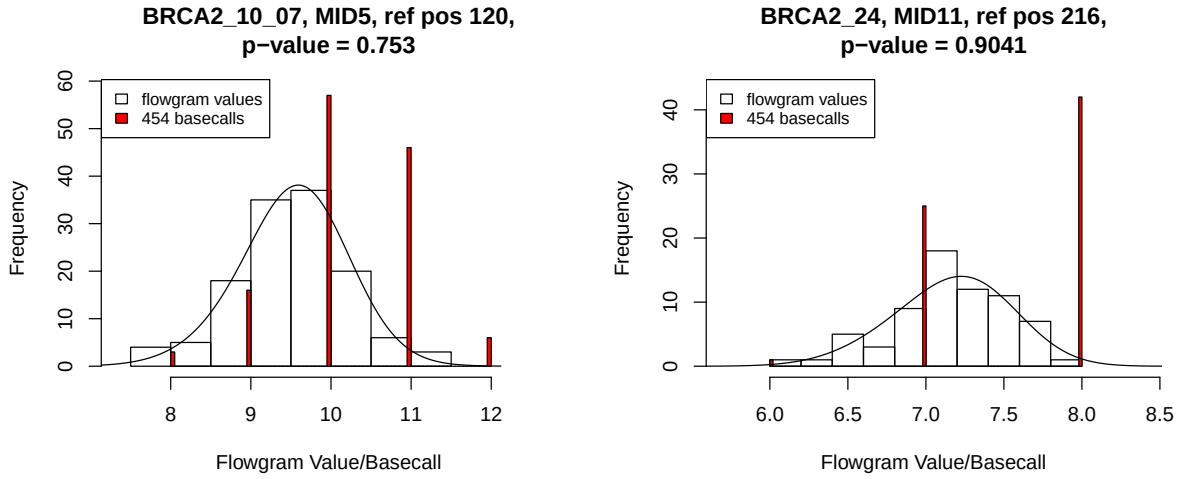


Figure 9.3: Two typical examples of called negatives that would have been called positive based on 454 base-calls

values, and hence are not called by our method. The advantage of using flowgram values is immediately clear when comparing the upper left plot of Figure 9.2 with the rightmost plot of Figure 9.3. Whereas the frequencies of the 454 base-calls seem to reflect a variant in both cases, the distribution of the flowgram values is completely different and seems to be more appropriate.

The p-values in Table 9.2 are not adjusted for multiple testing. If the FDR is controlled at 5% using the Benjamini-Hochberg method (Benjamini and Hochberg, 1995), only the two loci with the lowest p-values are still significant.

9.4.2 Empirical evaluation of the method's performance

Although the plots shown for the BRCA data suggest that the variant calling method leads to reasonable results, a proper performance evaluation is needed. One way to validate the results is to conduct Sanger sequencing to the subject samples and use this as a *gold standard*. However, these data are often not available. Alternatively, it is possible to simulate *known-truth* data ourselves. Hence, a simulation study is set up to assess the sensitivity or true positive rate (TPR), as defined in Equation (3.50), and the true negative rate or specificity (SPC), as defined in Equation (3.51), of the proposed method.

Two typical samples of real flowgram values are taken from the BRCA data set. The loci associated with these flowgram values are known to contain no sequence variants, following De Leeneer et al. (2011). The flowgram values of the first sample correspond with a HPL of 7, those of the second with a HPL of 8. The situation of a heterozygous variant (alternative hypothesis) is simulated by subsampling from the two flowgram value data sets with a certain mixing probability ψ_1, whereas the situation of no variant (null hypothesis) is obtained by subsampling from only one of the two flowgram data sets. In this study, the data set with flowgram values corresponding with HPL 7 is used. The former sampling situation will allow us to obtain the sensitivity, while the specificity will be calculated from the latter. When applying this strategy it is of paramount importance to avoid confounding factors to influence the results as much as possible. Therefore, the two samples are taken from the same subject, and at a reference position that corresponds to a similar position on the read (position 67 for HPL 7, position 91 for HPL 8). The latter condition is an attempt to minimize the well-known impact of the position

of the read on the base-call error rate because of variations in the flowgram value distribution (e.g. Chapter 8 and Brockman et al., 2008).

The simulations are conducted for scenarios involving the following settings: (A) threshold values δ between 0.05 and 1.2; (B) mixing probabilities ψ_1 of 0.2, 0.3, 0.4, 0.5, 0.6, 0.7 and 0.8; (C) sample sizes or coverages K_{ij} of 20, 50, 100, 150 and 200. For each combination, 1000 simulations are performed. In each run the variant calling method provides a p-value. The proportion of significant p-values (significance level of 5%) is then used to calculate the TPR and FPR. Subsequently, the results are combined for the construction of a ROC curve. Each point on this curve corresponds with a different value for δ.

9.4.2.1 Effect of mixing probability and coverage

Figures 9.4, 9.5 and 9.6 show the ROC curves for the method based on the ordinary MLE (left panels) and based on the penalized MLE (right panels), for $\psi_1 = 0.5$, $\psi_1 = 0.6$ and $\psi_1 = 0.2$, respectively.

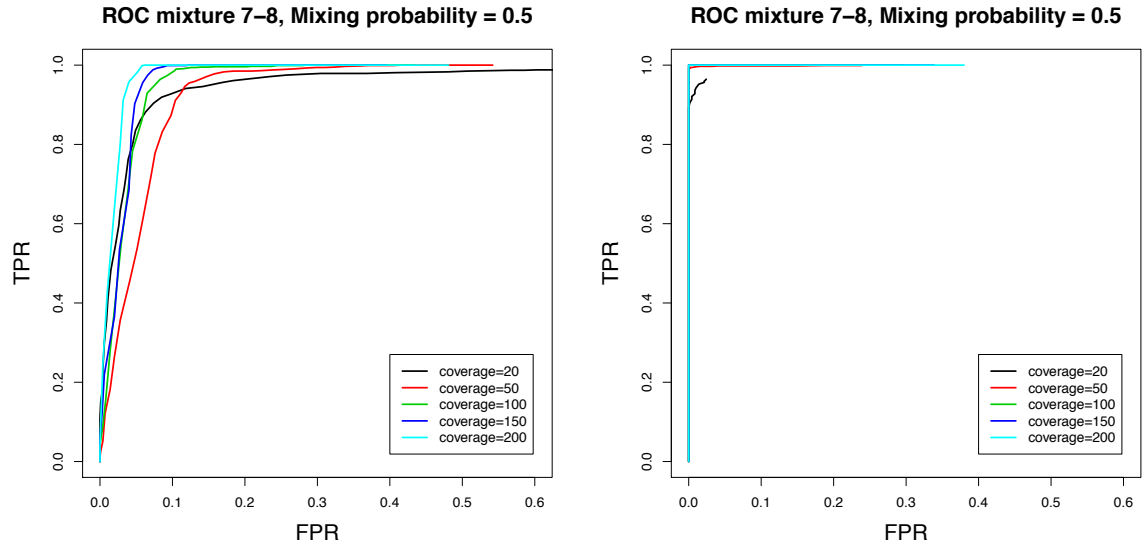


Figure 9.4: ROC curve for mixing probability $\psi_1 = 0.5$ based on the ordinary MLE (left panel) and the penalized MLE (right panel) of the mixture model with HPLs 7 and 8

The ROC curves indicate that the method based on the penalized MLE does a good job in attaining a large TPR while keeping the FPR very small. The penalized MLE method also clearly outperforms the method based on the ordinary MLE. The method works best for a mixing probability of 0.5, which is the mixing probability in an ideal sequencing experiment. If

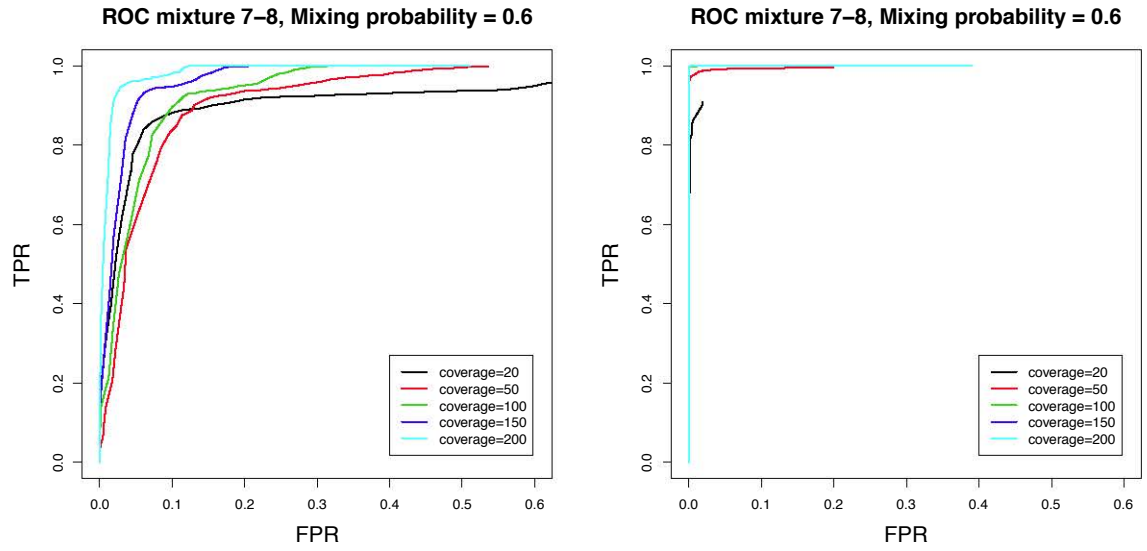


Figure 9.5: ROC curve for mixing probability $\psi_1 = 0.6$ based on the ordinary MLE (left panel) and the penalized MLE (right panel) of the mixture model with HPLs 7 and 8

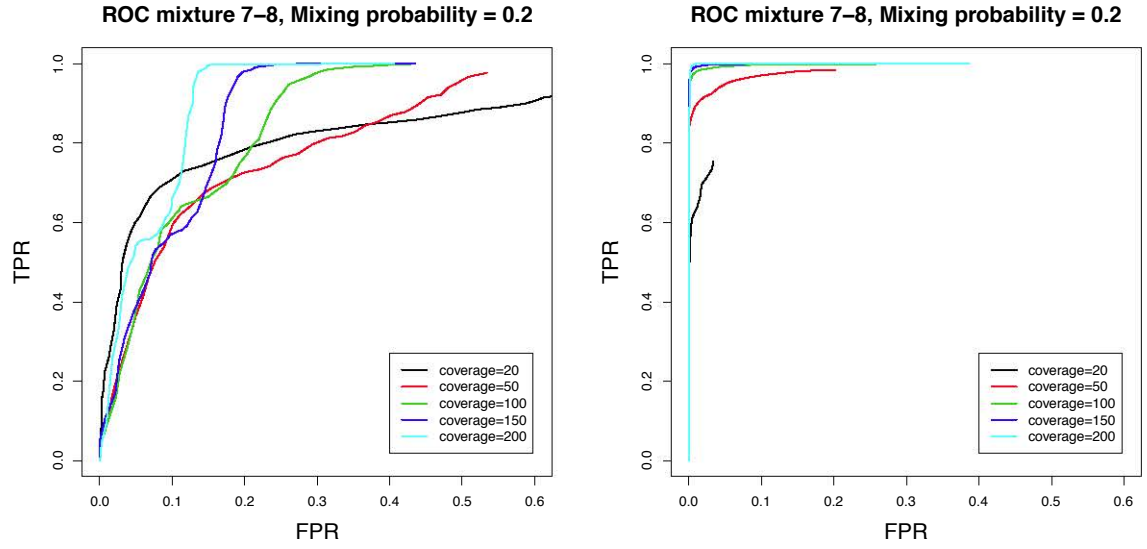


Figure 9.6: ROC curve for mixing probability $\psi_1 = 0.2$ based on the ordinary MLE (left panel) and the penalized MLE (right panel) of the mixture model with HPLs 7 and 8

the difference from this desired value increases, the performance of the method becomes worse. This is obvious from the ROC curves based on the other mixing probabilities in the simulation study. Furthermore, the performance of the method improves with increasing coverages. A coverage of only 20 is clearly insufficient to guarantee trustworthy results, whereas the method already performs quite well for a coverage of 50, especially for mixing probabilities close to 0.5. Similar plots can be made for mixtures with other HPLs. In all further analyses the results will be based on the penalized MLE.

9.4.2.2 Effect of threshold value

A difficulty in the application of the methodology that has not been discussed yet is the choice of a robust threshold value δ. One may expect that ideally different threshold values should be applied when testing different HPLs. To examine this, a new series of simulations is set up. In these simulations we explore the TPR and FPR as a function of threshold value in the analysis of a series of different HPLs. Just as before, real flowgram values from the BRCA data set that contain no variants are used for simulation. We have sampled from flowgram values corresponding to homopolymers of lengths 4, 5, 6, 7 and 8 for the determination of the FPRs, and from mixtures of flowgram values associated with homopolymers of lengths 4-5, 5-6, 6-7 and 7-8 to calculate the TPRs. In the simulations, we assume a mixing probability of 0.5 and the sample size or coverage is varied at 20, 50, 100 and 200. The resulting plots are given in Figure 9.7.

As expected, the simulation results indicate that the TPR drops from 1 to 0 with increasing threshold value. It seems that the larger the coverage of the sequencing experiment is, the larger will be the threshold value at which the TPR starts to decrease. From coverage 50 and larger, however, this increase seems only marginal. At coverage 20 the TPR starts to decrease almost immediately and for longer homopolymers (HPL 7 and 8) even never reaches the value 1 at threshold 0. The plots also suggest that the larger the coverage is, the smaller the range of threshold values is in which the decrease of TPR from 1 to 0 takes place. It also seems that the TPR starts to decrease at a smaller threshold value when dealing with longer homopolymers. In general, the FPR decreases by increasing threshold value. For short homopolymers (HPL 4 and 5) the FPR is kept at 0 over the whole range of threshold values, which is not the case for longer homopolymers. The plots suggest an increasing FPR when the HPL increases. The FPR also increases by increasing coverage and thus approximates 0 only at larger threshold values.

Based on the plots, a threshold value between 0.5 and 0.7 seems the best choice when dealing with settings that fall within the range of those in the simulation study. For threshold values in this range a large TPR is combined with a small FPR. The results imply that it might be beneficial to slightly increase the threshold for sequencing experiments with large coverages.

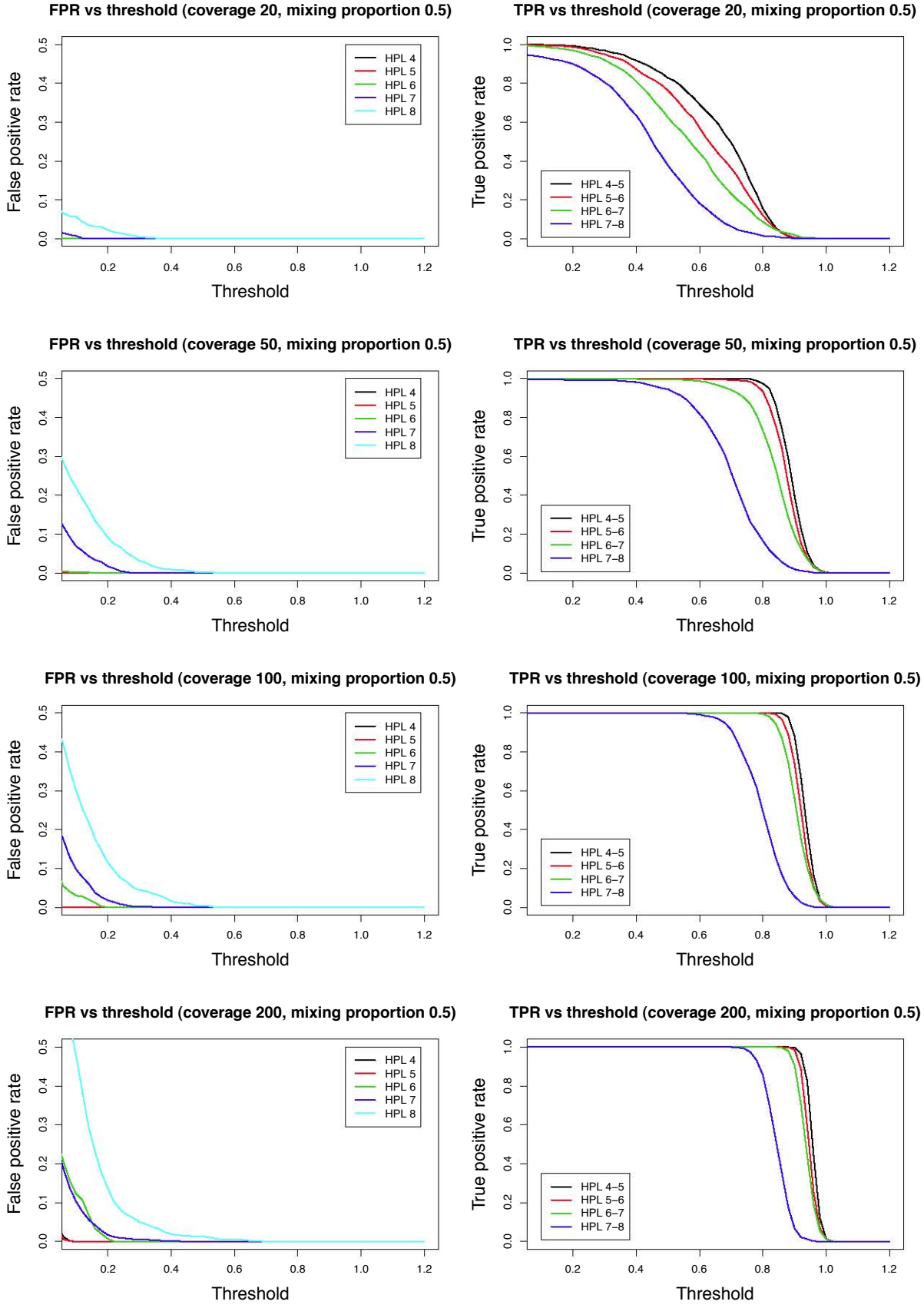


Figure 9.7: TPR and FPR as a function of threshold for different HPLs at coverages 20, 50, 100 and 200

9.4.2.3 Sensitivity to hyperparameter values α and β

In the penalized maximum likelihood procedure an inverse gamma prior distribution is imposed on the variance parameters of the two normal components of the mixture. In line with what was proposed in Ridolfi and Idier (1999), the hyperparameters α and β were set at 0.4 in all preceding analyses. To evaluate whether this is a reasonable choice, we now examine the sensitivity of the performance of the variant calling method to the hyperparameter values. Figure 9.8 shows the densities of inverse gamma distributed random variables for a selection of different hyperparameter values.

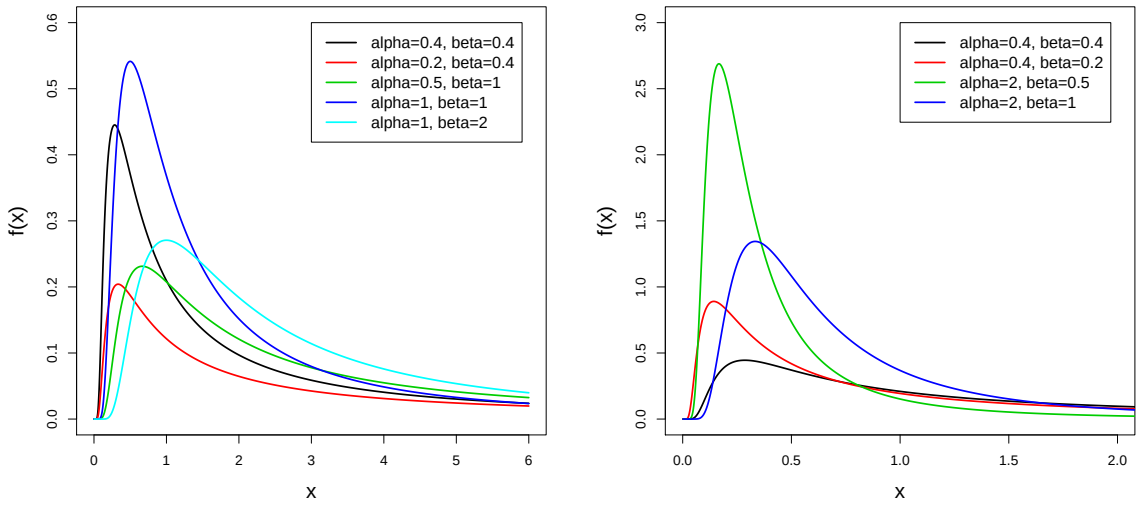


Figure 9.8: Inverse gamma densities for a selection of different hyperparameters α and β

A simulation study is conducted using the same flowgram value data as in Section 9.4.2.1. Figure 9.9 shows the ROC curves for the same selection of hyperparameters as displayed in Figure 9.8 for coverages 50 (left panel) and 200 (right panel). The threshold values δ are varied between 0.1 and 1.2. All simulations are run with a mixing probability of 0.5. It is clear that the curves are almost identical for different hyperparameter choices. This indicates that the analysis is very robust to different choices for α and β.

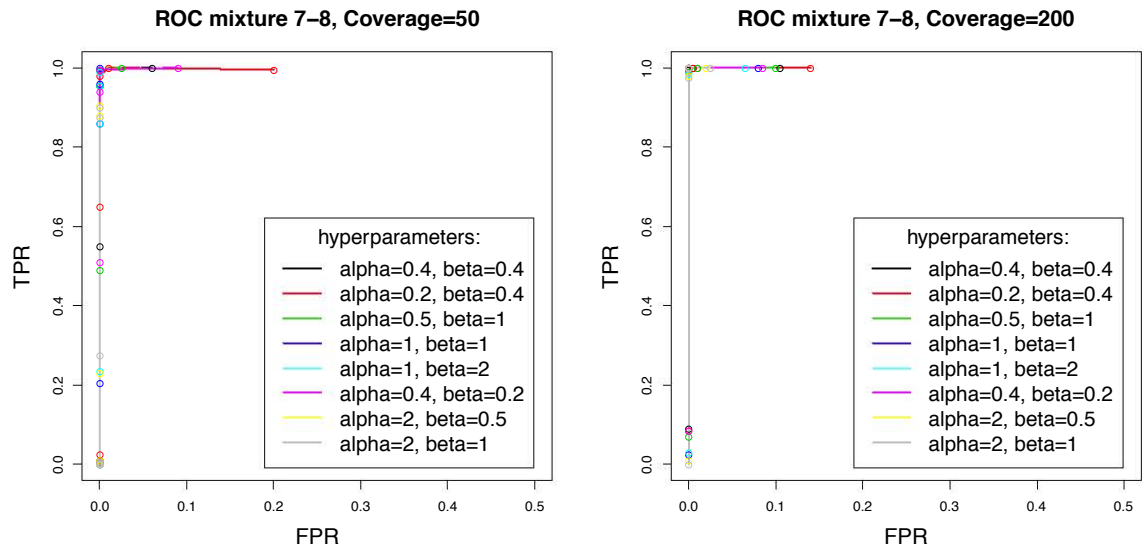


Figure 9.9: ROC curves for a selection of hyperparameters of the inverse gamma prior distribution of the mixture model with HPLs 7 and 8, for coverages 50 (left panel) and 200 (right panel), and with mixing probability $\psi_1 = 0.5$

9.5 Conclusion

With the maturing of new high-throughput DNA sequencing technologies, interest has grown to exploit their tremendous potential to apply them for personalized diagnostics and medicine. In this chapter we have proposed a statistical method to detect heterozygous sequence variants in homopolymeric regions of diploid organisms with 454 sequencing data. A two-component normal mixture model is fitted to the flowgram values at each genomic locus using a penalized maximum likelihood framework. The difference in component means is subsequently tested against a specified threshold value. The method is illustrated on an amplicon sequencing data set interrogating the *BRCA* genes. Simulation experiments reveal that the method works well in terms of sensitivity and specificity. The performance is best for mixing probabilities of 0.5 and a threshold value between 0.5 and 0.7. The method also improves with increasing coverages.

Chapter 10

Discussion, conclusions and future research perspectives for Part II

10.1 Discussion and conclusions

In this second part of the dissertation we have focused on the analysis of next-generation sequencing (NGS) data produced by the 454 platform, which is one of the prominent players among the NGS technologies. In particular, we have developed a statistical method for two distinct challenges at different stages in the data-analytic pipeline. To a large extent both problems are caused by difficulties specifically encountered with 454 sequencing for determining the correct length of homopolymers in the DNA sequence. At the start of the pipeline the base-calling of 454 sequencing data has been considered, while more downstream of the data flow we have developed a method for the detection of homozygosity and heterozygosity in homopolymeric DNA regions of diploid organisms.

An alternative method for base-calling of 454 sequencing data has been proposed based on a weighted Hurdle Poisson model. The method is referred to as HPCall. Its probabilistic framework enables a seamless integration of base-calling and quality score assignment, which are now conducted simultaneously. For a given cycle and nucleotide, the probability for each HPL is estimated conditional on read-specific covariates, and the call corresponds to the HPL with the maximum probability. In this way, the height of the maximal probability provides direct information about the base-calling uncertainty and can thus be used as a measure for the base-calling

quality. Moreover, in the case of a miscall, the second largest probability indicates whether an undercall or an overcall is more likely. This information is important for the downstream analysis of sequencing data. However, it is completely lacking when using Phred-like quality scores produced by current 454 base-callers. The distributions of maximum base-calling probabilities associated with a miscall are more evenly distributed between 0.5 and 1 than in the case of a correct call, for which it is very often nearly 1. This suggests that relatively small maximum probabilities are often associated with miscalls and therefore should raise caution.

Because Phred-like quality scores are commonly used in the downstream analysis steps of NGS experiments, they are also calculated by HPCall. As a result, they can be used in the same way as 454 quality scores. They are related to the probability of not having an overcall. These *overcall* quality scores appear to compete well with the 454 quality scores, while the Pyrobayes quality scores perform clearly worse. At the same time HPCall produces considerably more high-quality scores. Since all possible base-calling probabilities are available, alternative quality scores can also be calculated based on the probability of not having an undercall. A novel summarizing quality score, the HPCall quality score, is constructed by quantifying to what extent the overcall or the undercall quality score has the smallest value at the base-called HPL. This information is coded by the sign of the quality score (minus for undercall, plus for overcall). The new quality score now contains explicit information about the direction of a possible miscall. Quality-aware sequence aligners may use these scores to provide more reliable mapping results. We have further illustrated the use of the HPCall base-calling probabilities and the Phred-like HPCall quality scores for assessing indels in sequence variant detection. In each sequencing flow, the native 454 base-caller provides a quality score for each called base, e.g. for a homopolymer of length 3, also 3 quality scores are provided. These quality scores are not informative to discriminate between potential undercalls or overcalls. Furthermore, in the situation that 0 bases are called instead of 1, no quality scores are provided by the other base-callers. Hence, no information is given about the probability that only background signal has been measured. In contrast, HPCall clearly indicates which type of miscall - undercall or overcall - is to be expected in these examples, by means of the second largest base-calling probability and the sign of the HPCall quality score.

Besides the added value of the base-calling probabilities and improved quality scores, we have shown that the prediction accuracy of HPCall exceeds that of the native 454 base-caller and of Pyrobayes. Based on the *E. coli* data set we have detected a 35% reduction of base-calling errors as compared to the current 454 base-caller. This reduction is quite stable throughout the whole HPL range. It is obtained based on a model that uses information from the preprocessed flowgram values as well as from the earlier-stage raw intensities. If only flowgram values are used, the reduction of base-calling errors is still present, though smaller. Hence, although preprocessing raw intensities to flowgram values prior to base-calling to a large extent has the merit of reducing the spatial as well as the read-specific and background optical noise in the data, it also seems to remove crucial information for the base-calling task itself. The smaller number of base-calling errors is also reflected in the smaller number of detected indels and SNPs after mapping the base-called reads to the *E. coli* reference sequence. For the calibration of the base-caller the associated HPLs of a reference sequence are used. A possible way to implement this is by adding plasmids to the sequencing experiment. The 454 sequencer uses control reads containing varying HPLs for recalibrating its native base-caller. Hence, these control reads would be very valuable for this purpose. Up to now, however, the 454 software does not allow to extract the flowgram values associated with these reads. Finally, we have found that the accuracy performance of HPCall is stable across different training data sets used to fit the model.

In Chapter 9, we have described a statistical method for the detection of DNA sequence variants from 454 sequencing data. The method is designed to detect heterozygous variants at specific homopolymeric loci of diploid organisms, with applications in a diagnostic setting. The data variability inherent to the sequencing technology is better captured by using flowgram values instead of sequence lengths as input data. By introducing this novelty, 454 base-calling uncertainties are to some extent accounted for in the variant calling. Heterozygous variants are called by fitting a two-component normal mixture model to the flowgram value data, and testing whether the difference between the two component means exceeds a certain threshold value. The parameters are estimated using penalized maximum likelihood in an EM algorithm. Penalization is accomplished by imposing an inverse gamma prior density on the variance parameters of the normal mixture.

We have applied the method on amplicon sequencing data involving *BRCA1* and *BRCA2* genes.

Several simulation experiments have been conducted to assess the method's performance in terms of sensitivity and specificity. The resulting ROC curves for different scenarios showed promising results, with a clear benefit of applying penalized over ordinary maximum likelihood estimation. A drawback of the method is its dependence on the choice of a user-defined threshold value. The simulation results suggested, however, that reasonable threshold values can easily be chosen for different predefined scenarios, such as the coverage. The method's performance was also shown not to be affected by the choice of hyperparameter values of the inverse gamma prior density.

10.2 Future research perspectives

10.2.1 Extension of methods for other homopolymer-sensitive technologies

While HPCall was primarily developed for base-calling of 454 data (see Chapter 8), it has the potential to be adapted to emerging sequencing platforms that rely on flow cycles, for which base-calling of long homopolymers is critical. The most obvious example is the Ion Torrent™ Personal Genome Machine (PGM™) system. This emerging platform uses semiconductor technology to transmit an electrical pulse from the direct detection of positively charged hydrogen ions released during the polymerization of DNA. Similar to the 454 technology, the microwells on the semiconductor chip are sequentially flooded by one of four nucleotide types. At a homopolymer position more nucleotides are incorporated resulting in the release of multiple hydrogen ions and an increase of the recorded electrical signal. After signal processing flowgram values are produced similar to 454 flowgram values. A more detailed description of this semiconductor device can be found in Rothberg et al. (2011). For these platforms, the Hurdle Poisson model framework and base-calling pipeline will remain unchanged. Only the explanatory variables used to predict the HPL will be specific for each platform. For instance, the nucleotide flow order of the Ion PGM sequencer is different from the 454 sequencer. We performed a pilot test of HPCall on the PGM sequencer of Ion Torrent using a PGM 314 *E. coli DH10B* data set. This data set was retrieved from `http://ioncommunity.lifetechnologies.com/docs/DOC-1848`. In the model

the flowgram values are used as explanatory variables. The base-called reads from HPCall and the standard PGM software are mapped to the *E. coli DH10B* reference genome using ssaha2. Figure 10.1 gives the cumulative percentage of reads as a function of number of mismatches per read for the standard Ion PGM base-caller and for HPCall applied on Ion PGM data. In the pilot study, HPCall seems a promising alternative for the base-calling of Ion PGM sequencing data as it results in more base-called reads with a small number of mismatches.

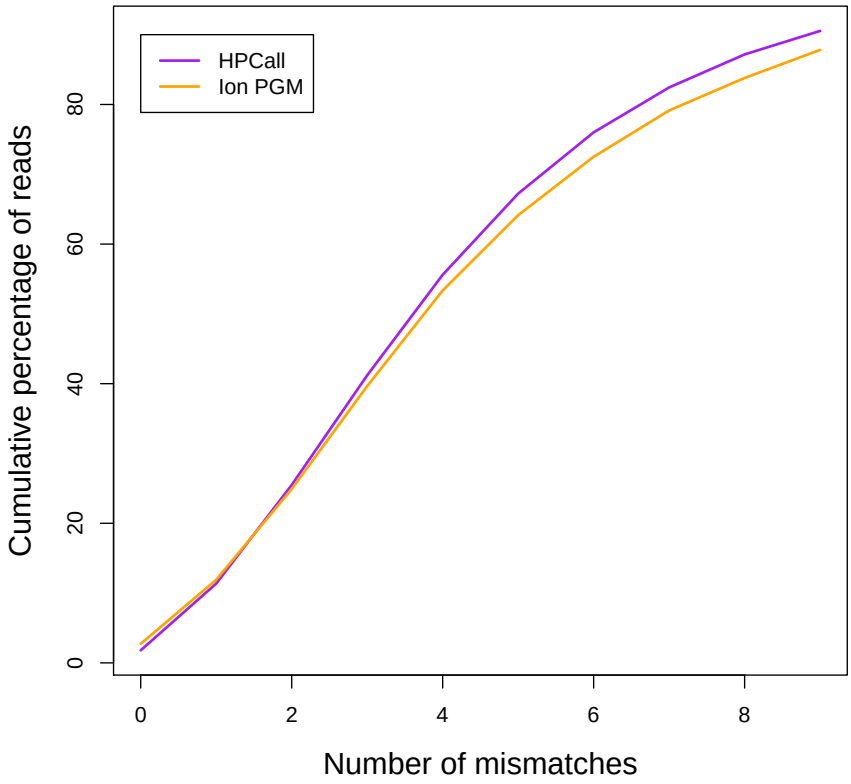


Figure 10.1: Cumulative percentage of reads as a function of mismatches per read in the mapping between the reads produced by either HPCall or the standard Ion PGM base-caller and the *E. coli DH10B* reference sequence.

Besides HPCall, also the method designed for discovering heterozygosity in homopolymeric regions (Chapter 9) has the potential to be applicable for Ion PGM data. This method uses flowgram values which are available in Ion PGM sequencing as well.

10.2.2 Use of HPCall base-calling probabilities in downstream applications

In Chapter 8 we have discussed and illustrated the added value of the base-calling probabilities predicted from the fitted Hurdle Poisson model. Based on the obtained results, it is believed that taking advantage of this additional information in downstream tasks, like mapping, genome assembly and sequence variant detection, will lead to more accurate and powerful applications.

In the following, one particular application is discussed that might be interesting for further development. A discipline in genomics that has been given increased attention in the last couple of years is *metagenomics*, which studies the genetic material from samples taken from natural habitats. Some applications of metagenomics for which 454 amplicon sequencing is commonly used include viral population dynamics (Wang et al., 2010) and the characterization of microbial communities (Huber et al., 2007). It is often of interest to estimate the population diversity of such natural habitat samples and to cluster sequences into *operational taxonomic units* (OTUs), where each OTU has a certain level of sequence difference from other OTUs. However, base-calling errors in 454 sequencing may lead to noisy reads. This makes it often difficult to distinguish between true diversity in the sample and noise introduced by the base-calling (Quince et al., 2011). Earlier studies have found that noise in 454 amplicon sequencing leads to inflated estimates of the number of OTUs (e.g. Kunin et al., 2010). Therefore, the development of methods that can effectively remove the noise in the reads is an important challenge in this area of research.

Arguably the most popular method in this regard was introduced in Quince et al. (2009) and further developed in Quince et al. (2011). This method removes 454 sequencing noise by reconstructing the true sequences and frequencies in the sample prior to OTU construction, using a mixture model. Model-based clustering is applied to the flowgrams, rather than the sequences themselves. The mixture model is used to describe the likelihood of the observed flowgrams, where each component of the mixture corresponds to a different sequence. The true sequences and their frequencies are inferred by maximizing the likelihood using an EM algorithm. In the model the flowgrams are assumed to be distributed as exponentials about the true sequences with a characteristic cluster size. More specifically, a measure for the *distance* between the observed flowgram and the *perfect* flowgram, i.e. one generated without noise and correspond-

ing with the true sequence, is modeled. This distance is based on the probability that a given flowgram $\bar{f} = (f_1, \ldots, f_M)$ of length M is generated by a sequence of nucleotides $\bar{S}$ that maps to a perfect flowgram $\bar{U} = (u_1, \ldots, u_M)$, where the u_i are integers corresponding with the HPL n. Assuming independence between the consecutive signals, this distance is then defined as (Quince et al., 2011)

$$\begin{aligned} d'(\bar{f}, \bar{U}) &= -\log\left(\prod_{i=1}^{M} \Pr(f_i|u_i = n)\right)/M \\ &= \sum_{i=1}^{M} -\log\left(\Pr(f_i|u_i = n)\right)/M \\ &= \sum_{i=1}^{M} d(f_i|u_i = n)/M. \end{aligned} \tag{10.1}$$

However, from the explorative plots on the raw intensities and flowgram values in Chapter 8, it is clear that this independence assumption does not hold. The distribution of the flowgrams depends to a certain extent on other variables such as the position in the read and the number of homopolymers in preceding and following cycles of the sequencing process. Therefore, it would be interesting to use the HPCall base-calling probabilities $\hat{\Pr}\{n|\boldsymbol{x}, \boldsymbol{y}\}$, to construct the distance measure $d'(\bar{f}, \bar{U})$. Given that these additional covariates are taken into account, this will possibly result in a better denoising of the sequencing reads and may lead to more accurate diversity estimates.

10.2.3 Extension of applicability of DNA sequence variant detection method

The DNA sequence variant detection method described in Chapter 9 currently only focuses on the detection of sequence variants at loci with HPLs of at least 4. An obvious extension of the method would be to make it applicable for the detection of variants in the whole amplicon, from start to end, and thus not only at homopolymeric positions. Hence, besides insertions and deletions in homopolymers, also substitution variants should be detected. However, this does not seem to be a big conceptual leap, because substitution variants can be considered as the insertion of a nucleotide followed by a deletion, or vice versa. The main challenge preceding the actual sequence variant analysis would be the alignment of the flowgram values correspond-

ing to the different reads of the same amplicon. Only if the alignment step is done correctly, can the flow-by-flow sequence variant detection be performed using the proposed methodology. Furthermore, the current method only distinguishes heterozygous sequence variants from homozygous loci. If a certain locus is decided to be homozygous, it is still possible that both homozygous alleles contain sequence variation with respect to the reference genome. Hence, an other interesting extension would be to integrate such a test in the existing framework.

Bibliography

Aboyoun, P., Pagès, H., and Lawrence, M. (2012). *GenomicRanges: Representation and manipulation of genomic intervals*. R package version 1.8.13.

Abramovich, F., Sapatinas, T., and Silverman, B. W. (1998). Wavelet thresholding via a Bayesian approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(4):725–749.

Andriankaja, M., Dhondt, S., De Bodt, S., Vanhaeren, H., Coppens, F., De Milde, L., Mühlenbock, P., Skirycz, A., Gonzalez, N., Beemster, G. T., and Inzé, D. (2012). Exit from proliferation during leaf development in Arabidopsis thaliana: A not-so-gradual process. *Developmental Cell*, 22(1):64–78.

Assarsson, E., Greenbaum, J. A., Sundström, M., Schaffer, L., Hammond, J. A., Pasquetto, V., Oseroff, C., Hendrickson, R. C., Lefkowitz, E. J., Tscharke, D. C., Sidney, J., Grey, H. M., Head, S. R., Peters, B., and Sette, A. (2008). Kinetic analysis of a complete poxvirus transcriptome reveals an immediate-early class of genes. *Proceedings of the National Academy of Sciences*, 105(6):2140–2145.

Barber, S., Nason, G. P., and Silverman, B. W. (2002). Posterior probability intervals for wavelet thresholding. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(2):189–205.

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

Benjamini, Y. and Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics*, 29(4):1165–1188.

Bertone, P., Stolc, V., Royce, T. E., Rozowsky, J. S., Urban, A. E., Zhu, X., Rinn, J. L., Tongprasit, W., Samanta, M., Weissman, S., Gerstein, M., and Snyder, M. (2004). Global identification of human transcribed sequences with genome tiling arrays. *Science*, 306(5705):2242–2246.

Bishop, C. (2006). *Pattern recognition and machine learning*. Springer, New York.

Blais, A. and Dynlacht, B. D. (2007). E2F-associated chromatin modifiers and cell cycle control. *Current Opinion in Cell Biology*, 19(6):658 – 662.

Bolstad, B. (2012). *preprocessCore: A collection of pre-processing functions*. R package version 1.18.0.

Bolstad, B., Irizarry, R., Östrand, M., and Speed, T. (2003). A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185–193.

Brockman, W., Alvarez, P., Young, S., Garber, M., Giannoukos, G., Lee, W. L., Russ, C., Lander, E. S., Nusbaum, C., and Jaffe, D. B. (2008). Quality scores and SNP detection in sequencing-by-synthesis systems. *Genome Research*, 18(5):763–770.

Bruce, A. G. and Gao, H.-Y. (1996). Understanding waveshrink: Variance and bias estimation. *Biometrika*, 83(4):727–745.

Carvalho, B. S. and Irizarry, R. A. (2010). A framework for oligonucleotide microarray preprocessing. *Bioinformatics*, 26(19):2363–2367.

Clement, L., De Beuf, K., Thas, O., Vuylsteke, M., Irizarry, R. A., and Crainiceanu, C. (2012). Fast wavelet based functional models for transcriptome analysis with tiling arrays. *Statistical Applications in Genetics and Molecular Biology*, 11:Iss. 1, Article 4.

Clyde, M. and George, E. I. (2000). Flexible empirical Bayes estimation for wavelets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(4):681–698.

Coppieters, F., De Wilde, B., Lefever, S., De Meester, E., De Rocker, N., Van Cauwenbergh, C., Pattyn, F., Meire, F., Leroy, B., Hellemans, J., Vandesompele, J., and De Baere, E. (2012). Massively parallel sequencing for early molecular diagnosis in Leber congenital amaurosis. *Genetics in Medicine*, 14(6):576–585.

Crick, F. H. C. (1958). On protein synthesis. *Symposia of the Society of Experimental Biology*, 12:138–163.

Crick, F. H. C. (1970). Central dogma of molecular biology. *Nature*, 227(5258):561–563.

Curtin, N. (2005). PARP inhibitors for cancer therapy. *Expert Reviews in Molecular Medicine*, 7(4):1–20.

David, L., Huber, W., Granovskaia, M., Toedling, J., Palm, C. J., Bofkin, L., Jones, T., Davis, R. W., and Steinmetz, L. M. (2006). A high-resolution map of transcription in the yeast genome. *Proceedings of the National Academy of Sciences*, 103(14):5320–5325.

De Beuf, K., De Schrijver, J., Thas, O., Van Criekinge, W., Irizarry, R. A., and Clement, L. (2012a). Improved base-calling and quality scores for 454 sequencing based on a hurdle Poisson model. *BMC Bioinformatics*, 13(1):303.

De Beuf, K., Pipelers, P., Andriankaja, M., Thas, O., Inzé, D., Crainiceanu, C., and Clement, L. (2012b). Analysis of tiling array expression studies with flexible designs in Bioconductor (waveTiling). *BMC Bioinformatics*, 13(1):234.

De Leeneer, K., Hellemans, J., De Schrijver, J., Baetens, M., Poppe, B., Van Criekinge, W., De Paepe, A., Coucke, P., and Claes, K. (2011). Massive parallel amplicon sequencing of the breast cancer genes BRCA1 and BRCA2: opportunities, challenges, and limitations. *Human Mutation*, 32(3):335–344.

De Schrijver, J., De Leeneer, K., Lefever, S., Sabbe, N., Pattyn, F., Van Nieuwerburgh, F., Coucke, P., Deforce, D., Vandesompele, J., Bekaert, S., Hellemans, J., and Van Criekinge, W. (2010). Analysing 454 amplicon resequencing experiments using the modular and database oriented variant identification pipeline. *BMC Bioinformatics*, 11(1):269.

De Veylder, L., Beeckman, T., and Inzé, D. (2007). The ins and outs of the plant cell cycle. *Nat Rev Mol Cell Biol*, 8(8):655–665.

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

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

Ding, Z., Millar, A. J., Davis, A. M., and Davis, S. J. (2007). Time for coffee encodes a nuclear regulator in the Arabidopsis thaliana circadian clock. *Plant Cell*, 19(5):1522–1536.

Donoho, D. L. and Johnstone, I. M. (1994). Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455.

Donoho, D. L. and Johnstone, I. M. (1995). Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, 90(432):1200–1224.

Durinck, S., Bullard, J., Spellman, P., and Dudoit, S. (2009a). Genomegraphs: integrated genomic data visualization with R. *BMC Bioinformatics*, 10(1):2.

Durinck, S., Moreau, Y., Kasprzyk, A., Davis, S., De Moor, B., Brazma, A., and Huber, W. (2005). BiomaRt and Bioconductor: a powerful link between biological databases and microarray data analysis. *Bioinformatics*, 21(16):3439–3440.

Durinck, S., Spellman, P., Birney, E., and Huber, W. (2009b). Mapping identifiers for the integration of genomic datasets with the R/Bioconductor package biomaRt. *Nature Protocols*, 4(8):1184–91.

Efron, B. (2003). Robbins, empirical Bayes and microarrays. *The Annals of Statistics*, 31(2):366–378.

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

Eisen, M. and Brown, P. (1999). DNA arrays for analysis of gene expression. *Methods in Enzymology*, 303:179–205.

Ewing, B. and Green, P. (1998). Base-calling of automated sequencer traces using Phred. ii. Error probabilities. *Genome Research*, 8(3):186–194.

Falcon, S. and Carvalho, B. (2012). *pdInfoBuilder: Platform design information package builder*. R package version 1.20.0.

Fan, J. and Gijbels, I. (1996). *Local polynomial modelling and its applications*. Chapman & Hall, London.

Figueiredo, M. and Nowak, R. (2001). Wavelet-based image estimation: an empirical Bayes approach using Jeffrey's noninformative prior. *Image Processing, IEEE Transactions on*, 10(9):1322 –1331.

Fodor, S., Read, J., Pirrung, M., Stryer, L., Lu, A., and Solas, D. (1991). Light-directed, spatially addressable parallel chemical synthesis. *Science*, 251(4995):767–773.

Gardner, M. J., Hubbard, K. E., Hotta, C. T., Dodd, A. N., and Webb, A. A. R. (2006). How plants tell the time. *Biochem J*, 397(1):15–24.

Gautier, L., Cope, L., Bolstad, B. M., and Irizarry, R. A. (2004). affy: Analysis of Affymetrix GeneChip data at the probe level. *Bioinformatics*, 20(3):307–315.

Gentleman, R. (2008). *R programming for bioinformatics*. CRC Press Taylor & Francis Group, Boca Raton.

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

Givens, G. and Hoeting, J. (2005). *Computational statistics*. Wiley, Hoboken.

Golub, G. and Van Loan, C. (1996). *Matrix computations, 3rd edition*. The Johns Hopkins Univ. Press, Baltimore.

Granovskaia, M., Jensen, L., Ritchie, M., Toedling, J., Ning, Y., Bork, P., Huber, W., and Steinmetz, L. (2010). High-resolution transcription atlas of the mitotic cell cycle in budding yeast. *Genome Biology*, 11(3):R24.

Gu, C. (2002). *Smoothing spline ANOVA models*. Springer-Verlag, New York.

Guan, Y. and Dy, J. (2009). Sparse probabilistic principal component analysis. *Journal of Machine Learning Research - Proceedings Track*, 5:185–192.

Haar, A. (1910). Zur theorie der orthogonalen funktionen-systeme. *Annals of Mathematics*, 69:331–371.

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

Hastie, T. J. and Tibshirani, R. J. (1990). *Generalized additive models*. Chapman & Hall, London.

Hazen, S., Naef, F., Quisel, T., Gendron, J., Chen, H., Ecker, J., Borevitz, J., and Kay, S. (2009). Exploring the transcriptional landscape of plant circadian rhythms using genome tiling arrays. *Genome Biology*, 10(2):R17.

Hollander, M. and Wolfe, D. A. (1999). *Nonparametric statistical methods, 2nd edition*. Wiley, New York.

Holt, R. A. and Jones, S. J. (2008). The new paradigm of flow cell sequencing. *Genome Research*, 18(6):839–846.

Huber, J. A., Mark Welch, D. B., Morrison, H. G., Huse, S. M., Neal, P. R., Butterfield, D. A., and Sogin, M. L. (2007). Microbial population structures in the deep marine biosphere. *Science*, 318(5847):97–100.

Huber, W., Toedling, J., and Steinmetz, L. M. (2006). Transcript mapping with high-density oligonucleotide tiling arrays. *Bioinformatics*, 22(16):1963–1970.

Huse, S., Huber, J., Morrison, H., Sogin, M., and Welch, D. (2007). Accuracy and quality of massively parallel DNA pyrosequencing. *Genome Biology*, 8(7):R143.

Ibrahim, J. G. and Laud, P. W. (1991). On Bayesian analysis of generalized linear models using Jeffreys's prior. *Journal of the American Statistical Association*, 86(416):981–986.

International Human Genome Sequencing Consortium (2004). Finishing the euchromatic sequence of the human genome. *Nature*, 431(7011):931–45.

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

Johnson, J. M., Edwards, S., Shoemaker, D., and Schadt, E. E. (2005). Dark matter in the genome: evidence of widespread transcription detected by microarray tiling experiments. *Trends in Genetics*, 21(2):93 – 102.

Johnson, N. (1949). Systems of frequency curves generated by methods of translation. *Biometrika*, 36(1):149–176.

Johnstone, I. M. and Silverman, B. W. (1997). Wavelet threshold estimators for data with correlated noise. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(2):319–351.

Kampa, D., Cheng, J., Kapranov, P., Yamanaka, M., Brubaker, S., Cawley, S., Drenkow, J., Piccolboni, A., Bekiranov, S., Helt, G., Tammana, H., and Gingeras, T. R. (2004). Novel RNAs identified from an in-depth analysis of the transcriptome of human chromosomes 21 and 22. *Genome Research*, 14(3):331–342.

Kass, R. E. and Steffey, D. (1989). Approximate Bayesian inference in conditionally independent hierarchical models (parametric empirical Bayes models). *Journal of the American Statistical Association*, 84(407):717–726.

Kunin, V., Engelbrektson, A., Ochman, H., and Hugenholtz, P. (2010). Wrinkles in the rare biosphere: pyrosequencing errors can lead to artificial inflation of diversity estimates. *Environmental microbiology*, 12:118–23.

Ledergerber, C. and Dessimoz, C. (2011). Base-calling for next-generation sequencing platforms. *Briefings in Bioinformatics*, 12(5):489–497.

Lipshutz, R. J., Fodor, S. P., Gingeras, T. R., and Lockhart, D. J. (1999). High density synthetic oligonucleotide arrays. *Nature Genetics*, 21(1):20–24.

Liu, X., Milo, M., Lawrence, N. D., and Rattray, M. (2006). Probe-level measurement error improves accuracy in detecting differential gene expression. *Bioinformatics*, 22(17):2107–2113.

Mallat, S. (1989). A theory for multiresolution signal decomposition: the wavelet representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(7):674 –693.

Mardis, E. R. (2008). The impact of next-generation sequencing technology on genetics. *Trends in Genetics*, 24(3):133–41.

Margulies, M., Egholm, M., Altman, W. E., Attiya, S., Bader, J. S., Bemben, L. A., Berka, J., Braverman, M. S., Chen, Y.-J., Chen, Z., Dewell, S. B., Du, L., Fierro, J. M., Gomes, X. V., Godwin, B. C., He, W., Helgesen, S., Ho, C. H., Irzyk, G. P., Jando, S. C., Alenquer, M. L. I., Jarvie, T. P., Jirage, K. B., Kim, J.-B., Knight, J. R., Lanza, J. R., Leamon, J. H., Lefkowitz, S. M., Lei, M., Li, J., Lohman, K. L., Lu, H., Makhijani, V. B., McDade, K. E., McKenna, M. P., Myers, E. W., Nickerson, E., Nobile, J. R., Plant, R., Puc, B. P., Ronan, M. T., Roth, G. T., Sarkis, G. J., Simons, J. F., Simpson, J. W., Srinivasan, M., Tartaro, K. R., Tomasz, A., Vogt, K. A., Volkmer, G. A., Wang, S. H., Wang, Y., Weiner, M. P., Yu, P., Begley, R. F., and Rothberg, J. M. (2005). Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, 437(7057):376–380.

McCarthy, D. J. and Smyth, G. K. (2009). Testing significance relative to a fold-change threshold is a treat. *Bioinformatics*, 25(6):765–771.

McCullagh, P. and Nelder, J. A. (1989). *Generalized linear models, 2nd edition*. Chapman & Hall, London.

McLachlan, G. and Krishnan, T. (1997). *The EM algorithm and extensions*. Wiley, New York.

McLachlan, G. and Peel, D. (2000). *Finite mixture models*. Wiley, New York.

Meinke, D. W., Meinke, L. K., Showalter, T. C., Schissel, A. M., Mueller, L. A., and Tzafrir, I. (2003). A sequence-based map of Arabidopsis genes with mutant phenotypes. *Plant Physiology*, 131(2):409–418.

Metzker, M. L. (2010). Sequencing technologies - the next generation. *Nature Reviews Genetics*, 11(1):31–46.

Mills, R. E., Pittard, W. S., Mullaney, J. M., Farooq, U., Creasy, T. H., Mahurkar, A. A., Kemeza, D. M., Strassler, D. S., Ponting, C. P., Webber, C., and Devine, S. E. (2011). Natural genetic variation caused by small insertions and deletions in the human genome. *Genome Research*, 21(6):830–839.

Mockler, T. C. and Ecker, J. R. (2005). Applications of DNA tiling arrays for whole-genome analysis. *Genomics*, 85(1):1 – 15.

Morris, J. S., Brown, P. J., Herrick, R. C., Baggerly, K. A., and Coombes, K. R. (2008). Bayesian analysis of mass spectrometry proteomic data using wavelet-based functional mixed models. *Biometrics*, 64(2):479–489.

Morris, J. S. and Carroll, R. J. (2006). Wavelet-based functional mixed models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(2):179–199.

Munch, K., Gardner, P., Arctander, P., and Krogh, A. (2006). A hidden Markov model approach for determining expression from genomic tiling micro arrays. *BMC Bioinformatics*, 7(1):239.

Nakajima, S., Sugiyama, M., and Babacan, S. (2011). On Bayesian PCA: Automatic dimensionality selection and analytic solution. In *ICML*, pages 497–504.

Nakajima, S., Sugiyama, M., and Tomioka, R. (2010). Global analytic solution for variational Bayesian matrix factorization. In *NIPS*, pages 1768–1776.

Naouar, N., Vandepoele, K., Lammens, T., Casneuf, T., Zeller, G., Van Hummelen, P., Weigel, D., Rtsch, G., Inz, D., Kuiper, M., De Veylder, L., and Vuylsteke, M. (2009). Quantitative RNA expression analysis with Affymetrix tiling 1.0R arrays identifies new E2F target genes. *The Plant Journal*, 57(1):184–194.

Narula, S. C. (1979). Orthogonal polynomial regression. *International Statistical Review*, 47(1):31–36.

Nason, G. (2005). *Wavelet methods in statistics with R (use R)*. Springer-Verlag, New York.

Nathanson, K. N., Wooster, R., and Weber, B. L. (2001). Breast cancer genetics: What we know and what we need. *Nature Medicine*, 7(5):552.

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

Nicolas, P., Leduc, A., Robin, S., Rasmussen, S., Jarmer, H., and Bessières, P. (2009). Transcriptional landscape estimation from tiling array data using a model of signal shift and drift. *Bioinformatics*, 25(18):2341–2347.

Ning, Z., Caccamo, M., and Mullikin, J. C. (2005). ssahaSNP - a polymorphism detection tool on a whole genome scale. *2005 IEEE Computational Systems Bioinformatics Conference - Workshops*, 0:251–252.

Ning, Z., Cox, A. J., and Mullikin, J. C. (2001). Ssaha: A fast search method for large DNA databases. *Genome Research*, 11(10):1725–1729.

Ogden, R. (1997). *Essential wavelets for statistical applications and data analysis*. Birchäuser, Boston.

Okamoto, M., Tatematsu, K., Matsui, A., Morosawa, T., Ishida, J., Tanaka, M., Endo, T. A., Mochizuki, Y., Toyoda, T., Kamiya, Y., Shinozaki, K., Nambara, E., and Seki, M. (2010). Genome-wide analysis of endogenous abscisic acid-mediated transcription in dry and imbibed seeds of Arabidopsis using tiling arrays. *The Plant Journal*, 62(1):39–51.

Otto, C., Reiche, K., and Hackermüller, J. (2012). Detection of differentially expressed segments in tiling array data. *Bioinformatics*, 28:1471–1479.

Pagès (2012). *BSgenome: Infrastructure for Biostrings-based genome data packages*. R package version 1.24.0.

Pagès, H., Aboyoun, P., and Lawrence, M. (2012). *IRanges: Infrastructure for manipulating intervals on sequences*. R package version 1.14.4.

Pease, A. C., Solas, D., Sullivan, E. J., Cronin, M. T., Holmes, C. P., and Fodor, S. P. (1994). Light-generated oligonucleotide arrays for rapid DNA sequence analysis. *Proceedings of the National Academy of Sciences*, 91(11):5022–5026.

Piccolboni, A. (2008). Multivariate segmentation in the analysis of transcription tiling array data. *Journal of computational biology : a journal of computational molecular cell biology*, 15(7):845–856.

Proost, S., Van Bel, M., Stercka, L., Billiaua, K., Van Parysa, T., Van de Peer, Y., and Vandepoele, K. (2009). Plaza: A comparative genomics resource to study gene and genome evolution in plants. *The Plant Cell*, 21(12):3718–3731.

Purdom, E., Simpson, K. M., Robinson, M. D., Conboy, J. G., Lapuk, A. V., and Speed, T. (2008). Firma: a method for detection of alternative splicing from exon array data. *Bioinformatics*, 24(15):1707–1714.

Qu, Y. and Xu, S. (2006). Quantitative trait associated microarray gene expression data analysis. *Molecular Biology and Evolution*, 23(8):1558–1573.

Quince, C., Lanzén, A., Curtis, T. P., Davenport, R. J., Hall, N., Head, I. M., Read, L. F., and Sloan, W. T. (2009). Accurate determination of microbial diversity from 454 pyrosequencing data. *Nature methods*, 6(9):639–641.

Quince, C., Lanzen, A., Davenport, R., and Turnbaugh, P. (2011). Removing noise from pyrosequenced amplicons. *BMC Bioinformatics*, 12(1):38.

Quinlan, A. R., Stewart, D. A., Stromberg, M. P., and Marth, G. T. (2008). Pyrobayes: an improved base caller for SNP discovery in pyrosequences. *Nature Methods*, 5(2):179–181.

R Development Core Team (2012). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Ramsay, J. and Silverman, B. (2005). *Functional data analysis, 2nd edition*. Springer-Verlag, New York.

Rattray, M., Liu, X., Sanguinetti, G., Milo, M., and Lawrence, N. D. (2006). Propagating uncertainty in microarray data analysis. *Briefings in Bioinformatics*, 7(1):37–47.

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

Rehrauer, H., Aquino, C., Gruissem, W., Henz, S. R., Hilson, P., Laubinger, S., Naouar, N., Patrignani, A., Rombauts, S., Shu, H., Van de Peer, Y., Vuylsteke, M., Weigel, D., Zeller, G., and Hennig, L. (2010). AGRONOMICS1: A new resource for Arabidopsis transcriptome profiling. *Plant Physiology*, 152(2):487–499.

Ridolfi, A. and Idier, J. (1999). Penalized maximum likelihood estimation for univariate normal mixture distributions. In *Actes du 17 e colloque GRETSI*, pages 259–262.

Ridout, M., Demetrio, C. G. B., and Hinde, J. (1998). Models for count data with many zeros. In *Proceedings of the XIXth International Biometric Conference*, Invited Papers, pages 179–192.

Ridout, M. S. and Besbeas, P. (2004). An empirical model for underdispersed count data. *Statistical Modelling*, 4(1):77–89.

Ronaghi, M. (2001). Pyrosequencing sheds light on DNA sequencing. *Genome Research*, 11(1):3–11.

Rothberg, J. and Leamon, J. (2008). The development and impact of 454 sequencing. *Nature Biotechnology*, 26(10):1117–1124.

Rothberg, J. M., Hinz, W., Rearick, T. M., Schultz, J., Mileski, W., Davey, M., Leamon, J. H., Johnson, K., Milgrew, M. J., Edwards, M., Hoon, J., Simons, J. F., Marran, D., Myers, J. W., Davidson, J. F., Branting, A., Nobile, J. R., Puc, B. P., Light, D., Clark, T. A., Huber, . M., Branciforte, J. T., Stoner, I. B., Cawley, S. E., Lyons, M., Fu, Y., Homer, N., Sedova, M., Miao, X., Reed, B., Sabina, J., Feierstein, E., Schorn, M., Alanjary, M., Dimalanta, E., Dressman, D., Kasinskas, R., Sokolsky, . T., Fidanza, J. A., Namsaraev, E., McKernan, K. J., Williams, A., Roth, G. T., and Bustillo1, J. (2011). An integrated semiconductor device enabling non-optical genome sequencing. *Nature*, 475(7356):348–352.

Royce, T. E., Rozowsky, J. S., Bertone, P., Samanta, M., Stolc, V., Weissman, S., Snyder, M., and Gerstein, M. (2005). Issues in the analysis of oligonucleotide tiling microarrays for transcript mapping. *Trends in Genetics*, 21(8):466 – 475.

Ruppert, D., Wand, M., and Carroll, R. (2003). *Semiparametric regression*. Cambridge University Press, UK.

Saha, S., Sparks, A. B., Rago, C., Akmaev, V., Wang, C. J., Vogelstein, B., Kinzler, K. W., and Velculescu, V. E. (2002). Using the transcriptome to annotate the genome. *Nature Biotechnology*, 20(5):508–512.

Samanta, M. P., Tongprasit, W., Sethi, H., Chin, C.-S., and Stolc, V. (2006). Global identification of noncoding RNAs in Saccharomyces cerevisiae by modulating an essential RNA processing pathway. *Proceedings of the National Academy of Sciences*, 103(11):4192–4197.

Sanger, F., Air, G. M., Barrell, B. G., Brown, N. L., Coulson, A. R., Fiddes, C. A., Hutchison, C. A., Slocombe, P. M., and Smith, M. (1977). Nucleotide sequence of bacteriophage phi X174 DNA. *Nature*, 265(5596):687–695.

Schadt, E., Edwards, S., GuhaThakurta, D., Holder, D., Ying, L., Svetnik, V., Leonardson, A., Hart, K., Russell, A., Li, G., Cavet, G., Castle, J., McDonagh, P., Kan, Z., Chen, R., Kasarskis, A., Margarint, M., Caceres, R., Johnson, J., Armour, C., Garrett-Engele, P., Tsinoremas, N., and Shoemaker, D. (2004). A comprehensive transcript index of the human genome generated using microarrays and computational approaches. *Genome Biology*, 5(10):R73.

Schena, M., Heller, R. A., Theriault, T. P., Konrad, K., Lachenmeier, E., and Davis, R. W. (1998). Microarrays: biotechnology's discovery platform for functional genomics. *Trends in Biotechnology*, 16(7):301 – 306.

Schena, M., Shalon, D., Heller, R., Chai, A., Brown, P. O., and Davis, R. W. (1996). Parallel human genome analysis: microarray-based expression monitoring of 1000 genes. *Proceedings of the National Academy of Sciences*, 93(20):10614–10619.

Seber, G. (1984). *Multivariate observations*. Wiley, New York.

Sémon, M. and Duret, L. (2004). Evidence that functional transcription units cover at least half of the human genome. *Trends in Genetics*, 20(5):229 – 232.

Shapiro, S. S. and Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611.

Shendure, J. and Ji, H. (2008). Next-generation DNA sequencing. *Nature Biotechnology*, 26(10):1135–1145.

Smyth, G. K. (2004). Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, 3:Iss. 1, Article 3.

Stolc, V., Samanta, M. P., Tongprasit, W., Sethi, H., Liang, S., Nelson, D. C., Hegeman, A., Nelson, C., Rancour, D., Bednarek, S., Ulrich, E. L., Zhao, Q., Wrobel, R. L., Newman, C. S., Fox, B. G., Phillips, G. N., Markley, J. L., and Sussman, M. R. (2005). Identification of

transcribed sequences in Arabidopsis thaliana by using high-resolution genome tiling arrays. *Proceedings of the National Academy of Sciences*, 102(12):4453–4458.

Stratton, M. R., Campbell, P. J., and Futreal, P. A. (2009). The cancer genome. *Nature*, 458(7239):719–724.

Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S., and Mesirov, J. P. (2005). Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 102(43):15545–15550.

Swarbreck, D., Wilks, C., Lamesch, P., Berardini, T. Z., Garcia-Hernandez, M., Foerster, H., Li, D., Meyer, T., Muller, R., Ploetz, L., Radenbaugh, A., Singh, S., Swing, V., Tissier, C., Zhang, P., and Huala, E. (2008). The Arabidopsis information resource (TAIR): gene structure and function annotation. *Nucleic Acids Research*, 36(suppl 1):D1009–D1014.

Tipping, M. E. and Bishop, C. M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622.

Toyoda, T. and Shinozaki, K. (2005). Tiling array-driven elucidation of transcriptional structures based on maximum-likelihood and Markov models. *The Plant Journal*, 43(4):611–621.

Verbeke, G. and Molenberghs, G. (2000). *Linear mixed models for longitudinal data*. Springer-Verlag, New York.

Vidakovic, B. (1998). Nonlinear wavelet shrinkage with Bayes rules and Bayes factors. *Journal of the American Statistical Association*, 93(441):173–179.

Vidakovic, B. (1999). *Statistical modeling by wavelets*. Wiley, New York.

Wand, M. and Jones, M. (1995). *Kernel smoothing*. Monographs on Statistics and Applied Probability. Chapman and Hall, London.

Wang, G. P., Sherrill-Mix, S. A., Chang, K.-M., Quince, C., and Bushman, F. D. (June 15, 2010). Hepatitis C virus transmission bottlenecks analyzed by deep sequencing. *Journal of Virology*, 84(12):6218–6228.

Wu, Z. and Irizarry, R. A. (2007). A statistical framework for the analysis of microarray probe-level data. *Annals of Applied Statistics*, 1(2):333–357.

Wu, Z., Irizarry, R. A., Gentleman, R., Martinez-Murillo, F., and Spencer, F. (2004). A model-based background adjustment for oligonucleotide expression arrays. *Journal of the American Statistical Association*, 99(468):909–917.

Yamada, K., Lim, J., Dale, J. M., Chen, H., Shinn, P., Palm, C. J., Southwick, A. M., Wu, H. C., Kim, C., Nguyen, M., Pham, P., Cheuk, R., Karlin-Newmann, G., Liu, S. X., Lam, B., Sakano, H., Wu, T., Yu, G., Miranda, M., Quach, H. L., Tripp, M., Chang, C. H., Lee, J. M., Toriumi, M., Chan, M. M. H., Tang, C. C., Onodera, C. S., Deng, J. M., Akiyama, K., Ansari, Y., Arakawa, T., Banh, J., Banno, F., Bowser, L., Brooks, S., Carninci, P., Chao, Q., Choy, N., Enju, A., Goldsmith, A. D., Gurjal, M., Hansen, N. F., Hayashizaki, Y., Johnson-Hopson, C., Hsuan, V. W., Iida, K., Karnes, M., Khan, S., Koesema, E., Ishida, J., Jiang, P. X., Jones, T., Kawai, J., Kamiya, A., Meyers, C., Nakajima, M., Narusaka, M., Seki, M., Sakurai, T., Satou, M., Tamse, R., Vaysberg, M., Wallender, E. K., Wong, C., Yamamura, Y., Yuan, S., Shinozaki, K., Davis, R. W., Theologis, A., and Ecker, J. R. (2003). Empirical analysis of transcriptional activity in the Arabidopsis genome. *Science*, 302(5646):842–846.

Yee, T. (2008). The VGAM package. *R News*, 8:28–39.

Zeller, G., Henz, S., Laubinger, S., Weigel, D., and Rätsch, G. (2008). Transcript normalization and segmentation of tiling array data. *Pacific Symposium on Biocomputing*, 13:527–538.

Summary

In the last 15 years a number of major technological advances have led to a tremendous revolution in genomics research and the emergence of the high-throughput genomics era. These new technologies provide the opportunity for biological and biomedical research to make more rapid advancements than was possible before. However, drawing meaningful information from the massive amount of data that are produced often presents a huge bottleneck. When extracting knowledge from high-throughput genomic data, statistical methods are needed in order to quantify the uncertainties inherent to the various sources of variability contained in the data. In this dissertation we have focused on different applications for two important technologies in high-throughput genomics: DNA microarrays and next-generation sequencing (NGS).

In Part I of the dissertation a statistical methodology has been proposed for transcriptome analysis with tiling microarrays, designed to detect regions of RNA expression along the genome. Tiling arrays measure transcriptional activity regardless of existing annotation and at equally spaced positions along the genome. Hence, the probe intensities can be viewed as realizations of an underlying function for RNA expression. To deal with the discontinuous and spatial heterogeneous nature of the expression data, we have adopted a wavelet-based functional modeling approach. The use of wavelets allows an efficient regularization of the expression signal without losing the ability to model local features.

In *Chapter 3* we have focused on the two-group design. The functional model that we have presented can assess transcript discovery and identify differentially expressed transcripts simultaneously. Adaptive smoothing of the effect functions is obtained by considering a Bayesian thresholding framework in which a normally distributed prior is imposed on the wavelet coefficients of these effect functions. The smoothing and error variance parameters are estimated by a marginal maximum likelihood approach. An empirical Bayes inference procedure has been

proposed, which makes use of the posterior distributions of the estimated effect functions. Both for transcript discovery and differential expression a probe-wise local Bayesian FDR is calculated. This result is associated with a predefined threshold value which enables obtaining transcriptionally affected regions that are statistically significant as well as biologically relevant. A simulation study has indicated that the wavelet-based approach outperforms the existing methods for transcript discovery and differential expression in terms of positive predictive value and specificity, while maintaining a high true positive rate. The method's use for finding potential targets in whole-genome transcription studies has been demonstrated by means of a case study on the reference plant *Arabidopsis thaliana*. The probe-wise and functional approach makes the method completely unbiased of existing annotation and therefore exploits tiling array data to their full potential.

The applicability of the wavelet-based model has been extended towards more complex experimental designs in *Chapter 4*. In particular, we have considered time-course studies, studies with more than two conditions and multiple-factor studies. The extension basically implies an appropriate adaptation of the model design matrix. A key point is to preserve the orthogonality of the design matrix to ensure analytical solutions with fast computation. In case of non-orthogonal designs, a Gram-Schmidt orthogonalization of the design matrix is conducted and the results are backtransformed to the original predictor space after estimation. A similar empirical Bayes procedure as for the two-group design has been used for inference. This procedure either occurs on the parameters themselves or on a function of the parameters, depending on the study design. The use and flexibility of the extended wavelet-based modeling approach has been illustrated on three case studies with the reference plant *Arabidopsis thaliana*. With these examples we have demonstrated the potential of the method to cope with a multitude of study designs and associated specific research questions, while still providing reliable results.

In *Chapter 5* we have discussed the implementation of the wavelet-based methods as a user-friendly R/Bioconductor package, called waveTiling. The package provides a standard analysis flow for wavelet-based transcriptome analysis on single-factor experiments with two or more biological conditions, the detection of linear and quadratic effects and circadian rhythms in time-course experiments, and the analysis of two-factor experiments or customized designs. Furthermore, it generates along-genome plots and contains functions to easily extract the transcriptionally affected genes and unannotated regions. Where possible the package uses the

standard Bioconductor S4-class data structures making it fully compatible with existing Bioconductor packages. The package also contains help functions and a manual in which the package's functions are explained and illustrated.

In Part II of the dissertation we have focused on the analysis of next-generation sequencing (NGS) data produced by the 454 platform, which is one of the prominent players among the NGS technologies. In particular, we have developed a statistical method for two distinct challenges at different stages in the data-analytic pipeline. To a large extent both problems are caused by difficulties specifically encountered with 454 sequencing for determining the correct length of homopolymers in the DNA sequence. At the start of the pipeline the base-calling of 454 sequencing data has been considered, while more downstream of the data flow we have developed a method for the detection of homozygosity and heterozygosity in homopolymeric DNA regions of diploid organisms.

In *Chapter 8* we have proposed an alternative method for base-calling of 454 sequencing data based on a weighted Hurdle Poisson model. The method is referred to as HPCall. Its probabilistic framework enables a seamless integration of base-calling and quality score assignment, which are now conducted simultaneously. For a given cycle and nucleotide, the probability for each HPL is estimated conditional on read-specific covariates, and the call corresponds to the HPL with the maximum probability. In this way, the height of the maximal probability provides direct information about the base-calling uncertainty and can thus be used as a measure for the base-calling quality. Moreover, in the case of a miscall, the second largest probability indicates whether an undercall or an overcall is more likely. Furthermore, a novel Phred-like quality score has been introduced. Unlike the traditional quality scores, these HPCall quality scores contain explicit information about the direction of a possible miscall. They may be used by quality-aware sequence aligners to provide more reliable mapping results. Besides the added value of the base-calling probabilities and improved quality scores, we have also shown that the prediction accuracy of HPCall exceeds that of current 454 base-callers.

Finally, we have described a statistical method for the detection of DNA sequence variants from 454 sequencing data (*Chapter 9*). The method is designed to detect heterozygous variants at specific homopolymeric loci of diploid organisms, with applications in a diagnostic setting. The data variability inherent to the sequencing technology is better captured by using flowgram values instead of sequence lengths as input data. By introducing this novelty, 454 base-calling

uncertainties are to some extent accounted for in the variant calling. Heterozygous variants are called by fitting a two-component normal mixture model to the flowgram value data, and testing whether the difference between the two component means exceeds a certain threshold value. The parameters are estimated using penalized maximum likelihood in an EM algorithm. Penalization is accomplished by imposing an inverse gamma prior density on the variance parameters of the normal mixture. We have applied the method on amplicon sequencing data involving *BRCA1* and *BRCA2* genes. Simulation experiments indicated that the proposed method performs well in terms of sensitivity and specificity.