Functional characterization and annotation of trait-associated genomic regions by transcriptome analysis

Dissertation

zur

Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

Promotionsgebiet Bioinformatik

Fakultät für Informatik und Elektrotechnik

Universität Rostock

vorgelegt von

Yang Du, geboren am 03. Feburar 1983 in Tianjin

aus Rostock

Rostock, 2014

urn:nbn:de:gbv:28-diss2014-0196-0

Gutachter:

Prof. Dr.-ing. Thomas Kirste (MMIS, Universität Rostock)

Prof. Dr. rer. nat. Klaus Wimmers (Leibniz-Institut für Nutztierbiologie, Dummerstorf)

Prof. Dr. rer. nat. Georg Füllen (IBIMA, Universitätsmedizin Rostock)

Prof. Dr. rer. nat. Mario Stanke (MNF, Universität Greifswald)

Datum der Einreichung: 16. Juni 2014
Datum der Verteidigung: 20. November 2014

To my beautiful daughter and beloved wife

Acknowledgements

I would like to first express my very great appreciation to my supervisor at FBN, Prof. Klaus Wimmers, for his generosity, support and patience throughout my work, for which I will always be grateful. With my deepest gratitude, I would also like to thank Prof. Thomas Kirsteand Prof. Georg Füllen, for accepting me and supporting my work, and their insightful comments and useful critiques of this research work. Without their guidance and encouragement, I would not be able to finish this dissertation.

My grateful thanks are also extended to Dr. Siriluck Ponsuksili, Dr. Eduard Murani and Dr. Nares Trakooljul, for their valuable advices and resultful discussions. It has been a great pleasure and learning opportunity to work with them in those past and on-going projects.

I am particularly grateful for the lab assistance given by Ms Hannelore Tychsen during the tiling array experiments. Despite the occasional language barrier, it has been a truly rewarding experience to work with her, and to see for oneself how experiments are done in real world.

I would like to offer my special thanks to Dr. Ronald Brunner, Frieder Hadlich, and also Peter Havemann from the IT department, for their excellent technical support with computational servers and web hosting.

I would also like to extend my thanks to many people from the university, for their openness to help, and for their suggestions and comments in many stages of my thesis preparation. It was truely uneasy for an external PhD student to walk though these process alone.

I wish to thank all my colleagues and friends at FBN, Au, Philipp, Ta, Wiebke, Tum, Milan..., it is impossible to name you all, for those German / Biology 101s, for helping me settle down and get through those tough transitions, for those laughters we shared and those fun get-togethers. Thank you all who had been kind to me and made my stay in Germany both scientifically and socially rewarding.

Last but not least, I want to specially thank my family and friends, here and at home, for their love, continuous encouragement and support during the time that I was engaged in this study.

Abstract

Functional genomics is the subject of studying biological data recorded in the complete state of a genomic system using high-throughput techniques, to describe the function of DNA and its interactions with intermediate RNA transcripts and functional protein products. One of the most crucial issues to deal with in genomics is the ambiguity arising from sequence homology. Duplicated DNA sequences of variable length commonly exist in most organisms, which impose a great challenge on the technologies used in genome research. As the two flagship high-throughput techniques used to characterize genomes and trnascriptomes and to quantify the level of various biological activities and redundancy, tiling array and next-generation sequencing both require careful handling of non-uniquely mapped features to ensure their accuracies. Thus many works have been done in the field of array probe design and in mapping sequencing reads back to reference or in *de novo* genome assembly. According to the recent result from the international collaboration effort, The ENCODE project, 80 percent of the human genome are either transcribed or biochemically functional, a number much higher than the known protein coding segments scientists used to believe even in 2003, when the human genome was fully sequenced. As a consequence of the growing availability of new high-throughput techniques, many of such novel functional fragments need to be identified and further functionally characterized. In this work, two novel implementations have been presented, which could assist in the design and data analysis of high-throughput genomic experiments. An efficient and flexible tiling probe selection pipeline utilizing the penalized uniqueness score has been implemented, which could be employed in the design of various types and scales of genome tiling task, with high coverage and resolution, while giving more control of the expected hybridization efficiency. A novel hidden semi-Markov model (HSMM) implementation is made available within the Bioconductor project, which provides a unified interface for segmenting genomic data in a wide range of research subjects. It was designed specifically for genomic data analysis, with flexible distributional assumption and optional prior learning using annotation or previous studies. The usages and performance of the two novel tools have been illustrated and evaluated using simulation and published datasets. Moreover, through an integrative and detailed case study, in which genome regions previously show to exhibit quantitative trait loci (QTL) should be characterized in terms of encoding differentially expressed genes, the two implementations have been utilized. The penalized uniqueness score was used to design 1M feature tiling arrays that covered a 18 Mb region of the porcine genome at a coverage of 49%. The HSMM was applied on the data from hybridization experiments of divergent animals enabled detecting candidate genes with trait-dependent expression.

Zusammenfassung

Die funktionale Genomik verfolgt als Ziele die allumfassende Auswertung biologischer Daten eines genomischen Systems mittels Hochdurchsatz-Technologien sowie die funktionale Charakterisierung der DNA und ihrer Wechselwirkungen mit RNA-Transkripten und funktionalen Proteinprodukten. Eine der größten Hürden in der Genomik stellt hierbei die Uneindeutigkeit durch Sequenzhomologien dar. Redundante DNA-Sequenzen variabler Länge existieren in vielen Organismen und bilden eine enorme Herausforderung an die Technologien in der genomischen Forschung. Sogenannte tiling arrays und Next-Generation-Sequenzierung sind Hochdurchsatz-Technologien, deren Daten eine sorgfältige Überprüfung und Bearbeitung hinsichtlich mehrfach kartierter Sequenzen zur Wahrung ihrer Präzision. Folglich wurden bereits viele Anstrengungen unternommen zum Erstellen von Arrayproben und beim Assemblieren von Fragmenten von Sequenzen gegen Referenzgenome bzw. bei der De-novo Assemblierung. Neueste Ergebnisse aus dem ENCODE Projekt, einer internationaler Verbundarbeit, belegen, dass 80 Prozent des menschlichen Genoms entweder transkribiert oder biochemisch funktional sind, also deutlich mehr als Wissenschaftler noch 2003 bei der Vervollständigung des Humangenoms angenommen haben. Als Konsequenz der stetig wachsenden Verfügbarkeit neuer Hochdurchsatz-Technologien müssen viele der neu gefundenen funktionalen Fragmente identifiziert und weiter funktional charakterisiert werden. In dieser Arbeit werden zwei neuartige Implementierungen präsentiert, die im Design und in der Datenanalyse von genomischen Hochdurchsatz-Experiment hilfreich sein können. Die erste Implementierung bildet eine effiziente und flexible Auswahl-Pipeline für tiling Proben basierend auf einem Eindeutigkeitsmaßmit einer Maluswertung (penalized uniqueness score), welches in vielfältigen Formen und Anwendungen für tiling Sonden ein Sondendesign mit einer hohe Abdeckungs- und Auflösungsrate ermöglicht und zudem mehr Kontrolle an erwarteter Hybridisierungeffizienz verspricht. Als zweite Implementierung wurde ein neuartiges Hidden-Semi-Markov-Modell (HSMM) im Bioconductor Projekt verfügbar gemacht, welches speziell für die genomische Datenanalyse mit flexibler Verteilungsannahme und optionalen Vorkenntnissen in Form von Annotationen oder Vorstudien die Segmentierung genomischer Daten in einer weiten Bandbreite von Forschungsvorhaben durch ihre einheitliche Schnittstelle unterstützt. Anwendbarkeit sowie Leistungsfähigkeit beider Programme sind mit Hilfe von simulierten und publizierten Daten dargestellt. In einer integrativen und detaillierten Fallstudie am Beispiel von Zuchttieren, bei dem zuvor identifizierte QTL (quantitativ trait loci) Regionen hinsichtlich differentiell exprimierter Gene charakterisiert werden sollten, wurden beide Implementationen angewendet Der penalized uniqueness score wurde genutzt um einen tiling array mit 1mio Element abzuleiten der eine 18mb große region des porci-

nen Genoms mit 49% abdeckt. Das HSMM wurde bei der Auswertung von Daten aus einem Hybridisierungsexperiment mit divergenten Tieren eingesetzt und ermöglichte die Identifizierung von merkmalsabhängig exprimierten Kandidatengene.

Acronym

A Adenine

aCGH Microarray-based comparative genomic hybridization

AUC area under the curve

BAC bacterial artificial chromosome

BLAST Basic Local Alignment Search Tool

BLAT BLAST-Like Alignment Tool

BP Biological Process

BWT Burrows—Wheeler transform

C Cytosine

CBS Circular Binary Segmentation

CC Cellular Component

CDF cumulative distribution function

cDNA complementary deoxyribonucleic acid (DNA)

CNV copy number variation

C_T threshold cycle

ChIP chromatin immunoprecipitation

DBN dynamic Bayesian network

DEPs differentially expressed probes

DEGs differentially expressed genes

DMRs differentially methylated regions

DNase deoxyribonuclease

DNA deoxyribonucleic acid

EM expectation—maximization

FDR false discovery rate

FPR false positive rate

FM-index Full-text index in Minute space

G Guanine

GC Guanine-Cytosine

GEO Gene Expression Omnibus

GO Gene Ontology

GWAS Genome Wide Association Studies

HMM Hidden Markov Model

HSMM Hidden semi-Markov Model

IPA Ingenuity pathway analysis

LINEs long interspersed nuclear elements

LOH loss of heterozygosity

LTR long terminal repeat

MAE mean absolute error

MCMC Markov chain Monte Carlo

MDS multidimensional scaling

MeDIP methylated DNA immunoprecipitation

MF Molecular Function

MUS minimum unique substring

NCBI National Center for Biotechnology Information

NGS Next-generation sequencing

ORF open reading frame

QC quality control

qPCR quantitative polymerase chain reaction

RLE run-length encoding

RMSE rooted mean squared error

ROC receiver operating characteristic

SINEs short interspersed nuclear elements

SNP single-nucleotide polymorphism

SNR signal-to-noise ratio

T Thymine

Tm melting temperature

TPR true positive rate

WHC water holding capacity

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

As an experimental science, biology used to be a time and labor intensive research subject. While in the last few decades, thanks to laboratory automation and high-throughput technologies, a new branch of contemporary genetics research, genomics, has been created, which is the studying of structure and function of the complete set of DNA from an organism. Transcriptome profiling is one of the first steps to understand complex biological processes, which helps us to move forward from genomic DNA sequences, the most basic genetic materials, to functional proteins. As the two most widely adopted high through-put technologies to survey transcriptome, genome tiling array and next-generation sequencing (NGS) give us the opportunity to unbiasedly capture the transcription activity across genomic regions. In this chapter, we will make brief introduction of these two experimental technologies, and present the key objectives of this dissertation.

1.1. Microarray

A DNA microarray or DNA chip is a highly compact assay of microscopic spots with selected specific DNA sequence, the probe, deposited or attached to a solid surface which normally taking the form of quartz slide or plastic chip. Alternative form also includes using immobilized microscopic beads without a solid platform. After hybridization, the signal intensity of the probe is then determined optically by the amount of fluorescently labeled target sample binded to the probe sequence via laser agitation. A schematic illustration of microarray is shown in Figure 1.1[1]. The application of microarry dates back to 1980s, when some hundreds or thousands of complementary DNA (cDNA) sequences

[1]Image obtained from Wikipedia

were spotted onto filter paper to study tissue or treatment specific gene expressions [1; 2; 3]. The later introduction of computer assisted image scanning and quantification, together with development of robotic spotting and in-situ oligonucleotide synthesizing eventually set the standards of modern miniaturized microarray [4].

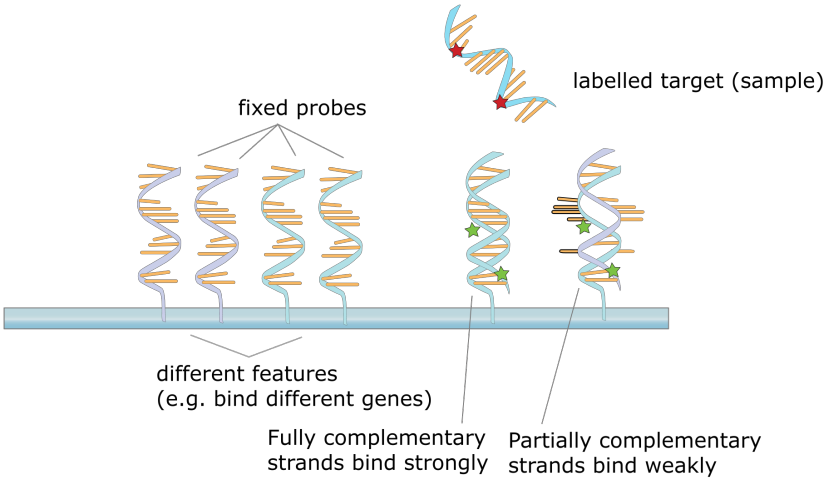


Figure 1.1.: A schematic illustration of probe hybridization mechanism of microarray.

Various types of microarray are commonly used in genetic research and medical diagnosis. Gene expression chip, the most popular microarray form, is highly cost efficient to measure expression levels of known genes and transcripts genome-wide. Another frequently adopted microarray is the single-nucleotide polymorphism (SNP) array, which is used to survey nucleotide variation in genomic DNA. SNP array is commonly applied in Genome Wide Association Studies (GWAS) of common and complex diseases. Also SNP could serve as both indicators of chromosome copy number and genotype maker, thus enabling the usage of SNP array to study copy number variation (CNV) and loss of heterozygosity (LOH) in cancer.

As another unique variety of high throughout microarray, genome tiling array targets not only known transcripts that are dispersed across the genome, but intensively covers all known contiguous regions on the genome with overlapping or evenly-spaced probes (See Figure 1.2[2]), thus being more unbiased than common gene expression arrays. Other than transcriptome profiling, tiling array also aids in discovering sites of DNA/protein interaction (chromatin immunoprecipitation (ChIP)-chip), of DNA methylation (methylated DNA immunoprecipitation (MeDIP)-chip) and of sensitivity to deoxyribonuclease (DNase)-chip) and Microarray-based comparative genomic hybridization (aCGH). Besides whole genome tiling array, region specific tiling also assists in refined transcrip-

[2]Image from Royce et al. [5]

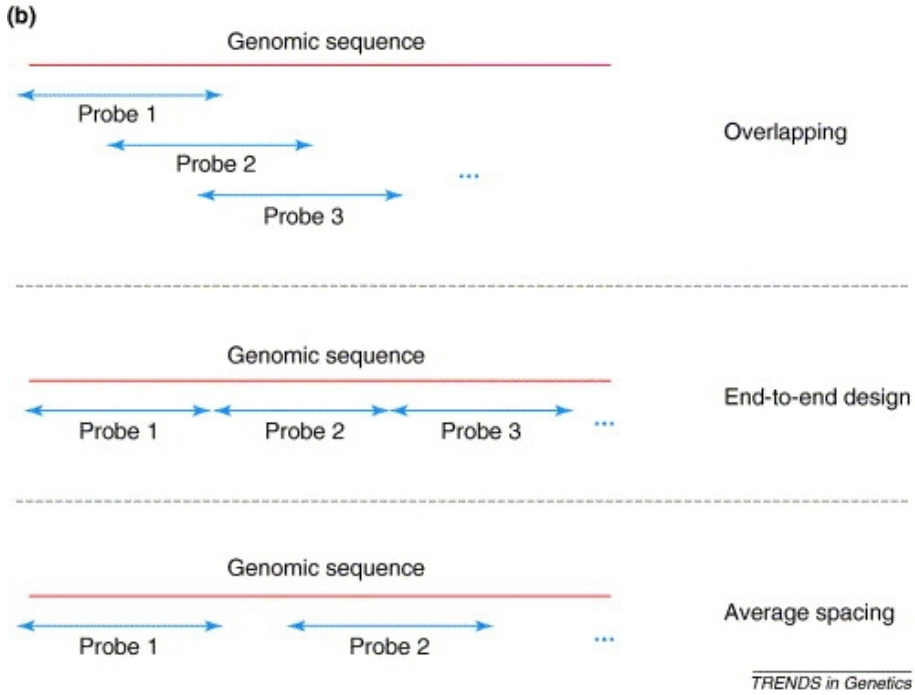


Figure 1.2.: Common tiling array probe layouts, which could be either overlapping, end-to-end or with an average spacing between neighbouring probes.

tome profiling of genome regions of interest.

Irrespective of specific microarray types, common work flow (See Figure 1.3[3]) of microarray experiment design and data analysis includes statistical power analysis, array quantification, array quality control (QC), statistical analysis and further mining of functional genomics data [6].

[3]Image from Nadon and Shoemaker [6]

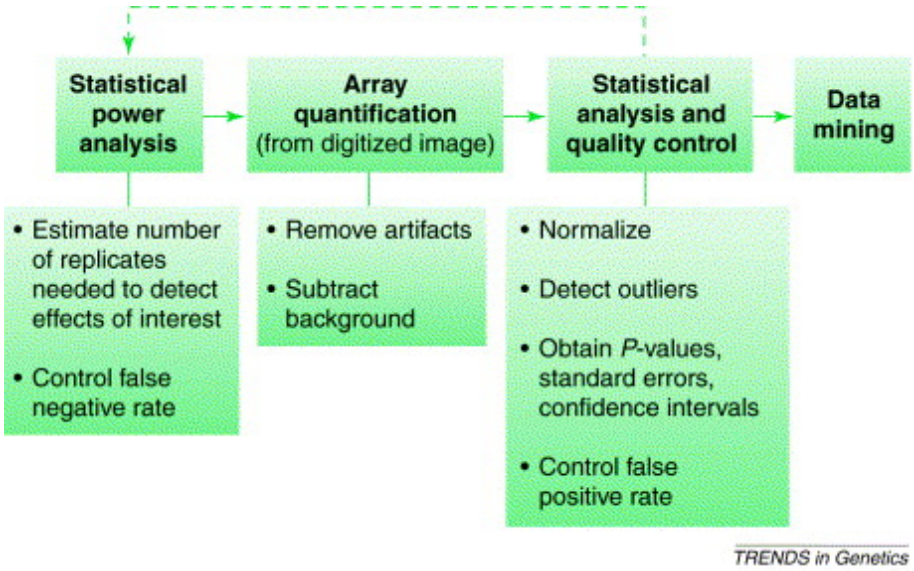


Figure 1.3.: Typical microarray experiment design and data analysis flow.

1.2. NGS

DNA sequencing is the process of sequentially determining the exact order of the four nucleotide bases—Adenine (A), Guanine (G), Cytosine (C), and Thymine (T) in the DNA molecule. To date, the most widely used sequencing method is Sanger sequencing developed by Frederick Sanger and colleagues in 1977. In recent years with advancing technology and lower costs, Next-generation sequencing (NGS) has gained unprecedented attention in genome research, which has the capability of running hundreds of thousands sequencing operations in parallel (See Figure 1.4[4] for a simplified comparison of Sanger sequencing and NGS work flow). With its inherent single-base resolution, NGS achieves higher accuracy in exon boundary mapping and can be used for SNP detection. Furthermore, its unlimited dynamic range enables detection of any subtle changes in gene expressions. Similar to genome tiling array, various types of genomics application can also be addressed by NGS — to quantify mature transcripts and small RNA using mRNA-seq, to survey transcription factor-binding sites with ChIP-seq, to study DNA methylation by MeDIP-seq, and etc [8].

Unlike microarray, for which signals of probes targeting known reference genome sequences can be easily subtracted from image background, the end products of sequencing experiments are millions of read segments which need to be quantified. The analysis of NGS data is far from mature comparing to microarray data analysis. The alignment of short reads to the reference genome, or *de novo* assembly of short reads both pose significant challenges to NGS data analysis. Further normalization, visualization and statistical modeling of genomic count data are also active fields of ongoing bioinformatics and statistical genetics research.

[4]Image from Shendure and Ji [7]

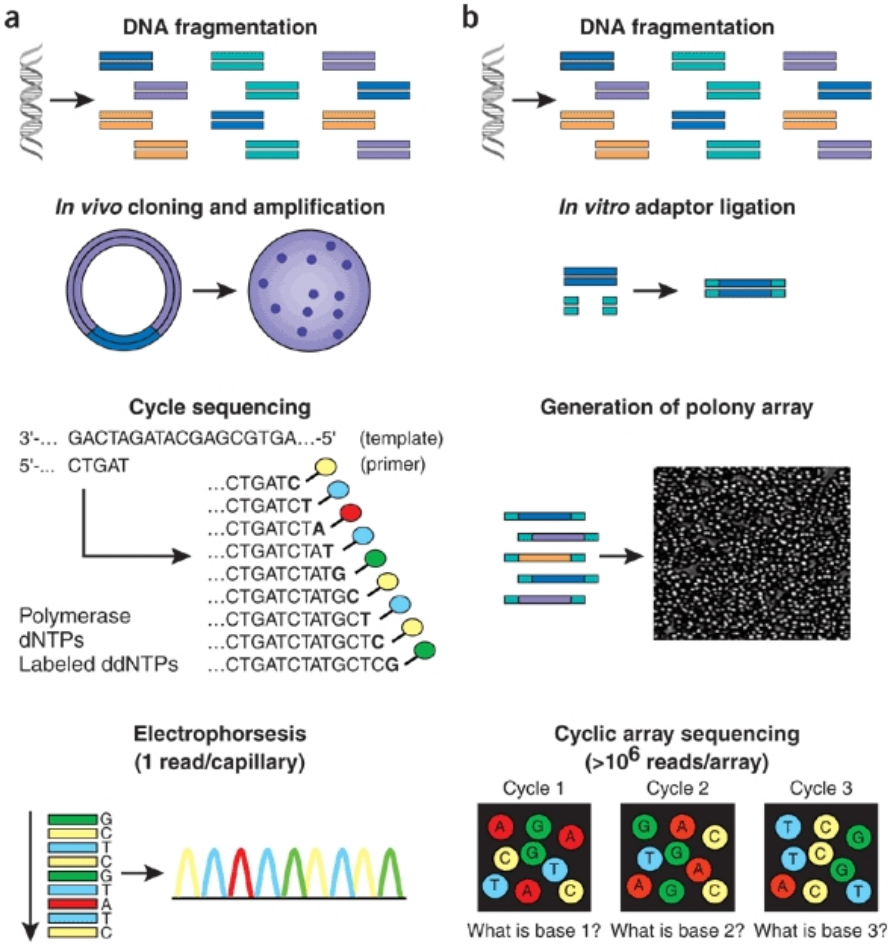


Figure 1.4.: Simplified comparison of Sanger sequencing (a) and NGS (b) work flow. The main differences include the substitution of in vivo cloning with sequencing libraries construction using clonal amplification and the array based parallel cyclic sequencing.

1.3. Aim of this work

In order to perform scientifically proof research and reveal statistically sensible findings in biology, one of the most basic requirements, the experimental conditions, environmental and biological, must be carefully designed and strictly controlled. In addition, a sufficiently large sample of qualified experiment subjects must be available for manipulation. Thus animal models provide an ideal resource for applied agricultural and experimental biology research. Also due to their immense similarities to human, genetical and physiological, animal models have been widely used to study human diseases.

Functional genomics is the subject of studying biological data recorded in the complete state of a genomic system using high-throughput techniques, to describe the function of DNA and its interactions with intermediate RNA transcripts and functional protein products. Genome tiling array and next-generation sequencing are the two flagship high-throughput technologies for current functional genomics research.

Unlike gene expression arrays or SNP arrays, high density genome wide tiling arrays are only commercially available for some model organisms, not to mention many other customary tiling arrays of specific purposes. It is also important to note that, due to the well known cross hybridization problem, which haunts microarray technology, and other probe quality issues, successful array based experiment at genome scale requires optimum and proper probe selection method. On the other hand, as the technology progressing, the study of genomics has entered the era of big data. The newly prevailing next-generation sequencing technology has the capability of generating hundreds of gigabytes per run. The accompanying high volume data demands efficient and scalable computational tools to assist in statistical modeling and data visualization.

In this work we will apply custom regional tiling array and next-generation sequencing experiments to identify and functionally characterize traits related genomic regions using farm animal model, and contribute to the software development of high-throughput computational biology.

The thesis work presented here is organized as the following,

- Chapter 2: "Methods and models", subdivided into two subsections, surveys and summarizes the common methods and issues involved in tiling array design and genomic segmentation related applications. The development made in the tilling array design using penalized uniqueness score has been published as **Yang Du** et al. [9]. The general-purpose genomic segmentation tool implemented has been published in **Yang Du** et al. [10].

- Chapter 3: "Case studies", illustrates the usage of tools developed and presented in the previous chapter through a series of experiments carried out by the research group at FBN (Leibniz Institute for Farm Animal Biology, Dummerstorf, Germany).
- Chapter 4: "Discussion and Outlooks", further reviews in detail the miscellaneous differences of the proposed methods and models with the existing ones, and discusses related issues in contemporary genomics research. Finally summarizes the works done throughout this dissertation, and discusses further improvements and outlooks in the research fields.

CHAPTER 2

Methods and Models

In this chapter, I will first review the common methodologies involved in array design, then introduce the concept and usage of penalized uniqueness score in tiling array design. In the second half, the topic will be shifted from experimental design and preparation to statistical modeling and data visualization in genomics. A general-purpose genomic segmentation model will be described and evaluated using simulation and published datasets.

2.1. Custom tiling array design

2.1.1. Common methods and issues

The fundamental bio-chemical principle of microarray technology is the hybridization between two strands of DNA formed by hydrogen bonding of complementary base pairs. Strong and reliable microarray probe signal strength then largely depends on a large number of specific complementary base paring with high sensitivity. Other influential factors contributing to hybridization efficiency are mostly nucleic acid thermodynamics related, like melting temperature (Tm), sequence base compositions, sequence complexity and propensity for secondary structure.

In a successful microarray design, selected probes should have similar hybridization efficiencies under a specified narrow band of temperature, as well as minimal potential for both self-hybridization and cross-hybridization [11]. Further guidelines for the specificity of long oligonucleotides array probe have been discussed in [12], in which the authors suggest that candidate probes should exhibit less than 75% overall sequence similarities with non-target sequences and contain no stretch of complementary sequence

longer than 15 bases.

However, these selection constraints become much more difficult to satisfy when applied to tiling probes designed for a large genome region [13], and naive method of using a uniform grid is clearly not optimum when considering hybridization efficiency. Among these contributing factors, non-specific binding or cross-hybridization is most problematic, when a non-targeted nucleotide sequence hybridizes to the designed probe. A similar situation with lower specificity also troubles NGS, when short reads need to be either aligned to a reference genome or assembled into contigs *de novo* [14].

Homogeneity and sensitivity

The melting temperature (Tm), defined as the temperature at which 50% of the oligonucleotide and its perfect complement are in duplex and the other half are in the random coil state, is essential for the success of stable hybridization, where a narrow band of Tm across all probes is highly desirable [11; 13]. It has also been shown that the melting temperature of the probe, among other oligonucleotide properties, might has the most significant impact on hybridization signal intensities [15]. In general, the melting temperature is affected by three major factors, oligo concentration, salt concentration and oligo sequence [16]. Common thermodynamics prediction models utilizing only base composition, like GC content or base counts, have been proposed and widely used in practice for short oligonucleotides [17; 18; 19]. GC content plays a crucial role in probe hybridization, since base pairings between G and C have three hydrogen bonds and are more stable compared to the A / T pairing.

$$Tm = (\#A + \#T) \times 2 + (\#G + \#C) \times 4 \tag{2.1}$$

$$Tm = 64.9 + 41 \times (\#G + \#C - 16.4) / (\#A + \#T + \#G + \#C) \tag{2.2}$$

Assuming a standard oligo concentration of 50 nM and pH neutral annealing environment, Equation (2.1) is valid for sequences shorter than 14 nucleotides [17], while Equation (2.2) is more accurate for sequences longer than 13 nucleotides [18]. Salt adjusted Tm approximation has also been considered and proposed in length specific setups [19]: Equation (2.3) is accurate when oligo length falls in the range of 18-25 mer;

when sequence is longer than 50 nucleotides Equation (2.4) gives better estimation.

$$\begin{aligned} Tm = 100.5 + 41 \times (\#G + \#C)/(\#A + \#T + \#G + \#C) \\ - 820/(\#A + \#T + \#G + \#C) + 16.6 \times log_{10}[Na+] \end{aligned} \tag{2.3}$$

$$\begin{aligned} Tm = 81.5 + 41 \times (\#G + \#C)/(\#A + \#T + \#G + \#C) \\ - 500/(\#A + \#T + \#G + \#C) + 16.6 \times log_{10}[Na+] - 0.62 \times F \end{aligned} \tag{2.4}$$

However, these simple models do not consider the actual probe sequence and base position, but rather use only the summary statistics of the sequence. The loss of information will inevitably lead to lower prediction power. Assuming a set of sequences with same length and GC content but different nucleotides arrangement, all simple models above will yield the same prediction, which would hardly be the real case. The nearest-neighbor model later proposed, taking into account of the nucleotides formation, is considered more robust and accurate [20; 21], which could also account for other influential factors like oligo concentration and ionic concentration. Thus in this work, the adapted prediction model of Tm (Equation (2.5)) using the same parameters as in [22; 23; 20] is utilized, with salt correction approximation,

$$Tm = \frac{\sum \Delta H_d + \Delta H_i}{\sum \Delta S_d + \Delta S_i + \Delta S_{self} + R \times \log C_T/b} + 16.6 \times \log[Na^+] \tag{2.5}$$

where R is the ideal gas law constant (1.987 cal/K·mol), $[Na^+]$ is the given sodium concentration, C_T is the total oligonucleotide strand concentration (b=4, if the strands are in equal concentration). Thermodynamics parameters ΔH and ΔS each represents the enthalpy (the amount of heat energy possessed by substances) change and the entropy (the amount of disorder a system exhibits) change. The subscript 'd' and 'i' indicate the di-nucleotide pairs parameter values of each nearest neighbor base pair and the initiation parameter. ΔS_{self} is the additional entropic penalty for the maintenance of the C2 symmetry of self-complementary duplexes. Values of the nearest-neighbor model parameters estimated in [22] are given in Table 2.1.

Complexity and repetitive sequence

Given that DNA / RNA sequence are both composed of only 4 possible nucleotide bases, respectively, the chance of seeing some particular pattern of nucleotides, a k-mer,

Table 2.1.: Nearest-Neighbor parameters for DNA/DNA duplex

Stack	$\Delta H^{\circ}(kcal.mol^{-1})$	$\Delta S^{\circ}(cal.mol^{-1}.K^{-1})$
5′ AA / TT 3′	-7.9	-22.2
5′ AT / TA 3′	-7.2	-20.4
5′ TA / AT 3′	-7.2	-21.3
5′ CA / GT 3′	-8.5	-22.7
5′ GT / CA 3′	-8.4	-22.4
5′ CT / GA 3′	-7.8	-21
5′ GA / CT 3′	-8.2	-22.2
5′ CG / GC 3′	-10.6	-27.2
5′ GC / CG 3′	-9.8	-24.4
5′ GG / CC 3′	-8	-19.9
The initiation parameter with G/C	0.1	-2.8
The initiation parameter with A/T	2.3	4.1
Symmetry correction	0	-1.4

repeating itself somewhere else across all chromosomes is unsurprisingly high. Assuming that we are looking at a mammalian genome like human, which contains around 3×10^9 nucleotide bases on a single strand, we want to know the probability of a 15 bp (k=15) sequence P occurs only once in the whole genome. First, there are 4^k different sequence formations, given only 4 possible bases to choose from. So for any specific location, the chance of seeing this particular 15 bp sequence P is only $1/4^{15} = 9.31E-10$, which is not very likely. However there are $N = 3E9$ locations one could check against, which would lead to on average $3E9/4^{15} = 2.79$ occurrences of this pattern, and this is only counting one strand of the DNA. The chance of a single occurrence is therefore $(1-1/4^{15})^{(3E9-1)} \times (1/4^{15}) \times 3E9 = 0.17$. In probability theory and statistics, the number of occurrence (X) of such a random pattern could be considered to follow a Binomial distribution, $B(N,p)$, with the probability of exact matching $p = 1/4^k$ and the number of comparisons equals to N. Thus by applying the cumulative distribution function (CDF) of Binomial distribution, one can easily get the probability of having this sequence more than once in the genome, $P(X > 1) = 1 - P(X \leq 1) = 0.77$, which is quite high. However when, k, the length of P increases, the success rate (p) drops exponentially, thus in turn the chance of having more than 1 occurrences decreases.

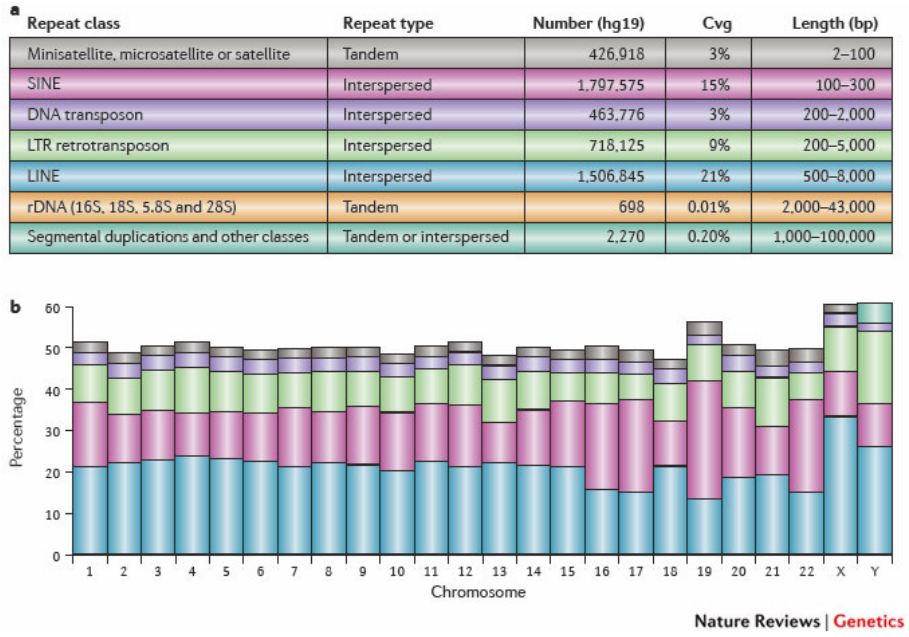
In real genomics, repetitive genomic sequences are sequences that show high degree of similarity or are identical to other parts of the genome. Their existences could be coincidental combinatorial events or products of complex cellular mechanisms. However such repetitive sequences are observed more frequently than expected. Studies have

shown for well-characterized genome like human that, nearly 50% of human genome are covered by repeats [24; 25]. Such high degrees of repetitiveness are also present in other lines of organism; for example, in plants, arabidopsis [26] and maize [27] have been found to exhibit large scale genome wide duplication. In prokaryotic microorganisms, repetitive sequences are also detected at a large proportion [28].

For these repetitive sequences, further categorization could be made depending on their sizes, positions and structural adjacency. In general, there are two types of repeats. Tandem repeats are those with the repeated copies in their immediate adjacency. Centromere and telomere of chromosome are largely comprised of tandem repeats. If the reoccurring copies of the transposable segments are located far from each other, they are termed as interspersed repeats. Due to the nature of complimentary base pairing of nucleotides, there is also another type of repeats, inverted repeats, which are sequences followed by their reverse complements, either immediately (tandem) or intervened by other random sequences (interspersed). When there is no intervening sequence between the copies, this inverted repeat is also called palindromic. According to RepeatMasker [29; 30], classification of interspersed repeats can be further characterize by 4 sub-types, short interspersed nuclear elements (SINEs), long interspersed nuclear elements (LINEs), retrovirus-like elements or long terminal repeat (LTR) and DNA transposons. There are also many forms of tandem repeats, mainly characterized by repeat length, like microsatellites, minisatellites and satellites, which are frequently used as molecular markers in forensic science and population genetics studies.

Most repeats are considered not functional, while some are involved in the evolution process [31; 32; 33], uncoupling intra- and inter-chromosomal gene conversion. Tandem repeats have also been shown to be associated with regulation of transcription factor binding [34], aging process [35], various disease forms including cancer [36; 37]. Inverted repeat and palindromic sequence, unlike most other types of repeats, are not well characterized by tools like RepeatMasker. Due to the special self-complementary structure, they can form secondary structure like stem-loop or hairpin, thus directly affecting genome stability [38]. In Figure 2.1, a summary of annotated human repetitive DNA reported by RepeatMasker are cited from [14].

Repetitive regions, as one major source of cross-hybridization and hybridization instablity, have been shown to account for a large proportion of mammalian genomes. For normal gene expression array and tilling chip used for transcriptom profiling, such repetitive segments are generally ignored in the probe selection process [13]. The most commonly adopted approach to handle repetitive regions is to exclude them using tools like RepeatMasker [29]or Window Masker [39]. However for tiling array, features reside in the repetitive proportions identified by repeat masking tools may have particular



a

Repeat class	Repeat type	Number (hg19)	Cvg	Length (bp)
Minisatellite, microsatellite or satellite	Tandem	426,918	3%	2–100
SINE	Interspersed	1,797,575	15%	100–300
DNA transposon	Interspersed	463,776	3%	200–2,000
LTR retrotransposon	Interspersed	718,125	9%	200–5,000
LINE	Interspersed	1,506,845	21%	500–8,000
rDNA (16S, 18S, 5.8S and 28S)	Tandem	698	0.01%	2,000–43,000
Segmental duplications and other classes	Tandem or interspersed	2,270	0.20%	1,000–100,000

Figure 2.1.: Summary of repetitive DNA sequences in the human genome (hg19)

significance [40]. Thus their inclusion should be considered in the chip design, and efforts have been made for the selection and interpretation of probes containing repetitive sequences [41; 42].

Specificity and probe uniqueness

Probe specificity is the most crucial and problematic factor in microarray design, many experimental techniques and analytical methods have been developed to overcome this issue. Commonly, approaches to evaluate probe uniqueness are mostly alignment based, like using Basic Local Alignment Search Tool (BLAST) [43], other researchers also employ suffix array [44] for faster indexing and matching. There are also attempts to approximate the cross-hybridization potential using thermodynamic models to assess the binding-free energy between probes and non-targeted sequences [45]. Most of these developments are done for the probe selection of gene expression array, where all known transcripts of the organism are targeted.

However when tackling tilling array designs, for mammals like us human (*Homo sapiens*), or other domestic animals like cow (*Bos taurus*) and pig (*Sus scrofa*), the total amount of DNA contained within one copy of a single genome is around 3 billion base pairs. To check if one particular query sequence is unique among the whole genome essentially involves approximately $2 \times 3 \times 10^9$ comparisons, not to mention that for a high-density chip the number of probe sequence easily exceeds 1 million. For those alignment based

methods, repetitive searches are accumulatively slow and are not readily capable to handle large scale tiling tasks. A simplest form of suffix arrays for the reference genome can be implemented in $\frac{1}{8} \times n \times \log_2 n$ bytes in space, which leads to a memory consumption of around 23 GB. Although those suffix array or suffix tree based approaches can deliver matched pattern in log-linear or linear time, the intensive memory usage could render most modern desktop PC infeasible. Thus a compressed index structure like Full-text index in Minute space (FM-index) [46], which is efficient in both query time and space consumption, is an ideal alternative for biological sequence analysis.

2.1.2. BWT and FM-index

FM-index is a compressed full-text sub-string index structure pairing the Burrows—Wheeler transform (BWT) [47] with suffix array, originally introduced by Ferragina, P. and Manzini, G.. The BWT was proposed by Burrows, M. and Wheeler, David J., which is a technique that reversibly rearrange the sequence in a way that similar characters from reoccurring subsequence would be sorted together. The nature of having consecutive runs of repetitive characters allows easy compression using schemes like run-length encoding (RLE).

Here I will first illustrate how BWT works and its pairing with suffix array via R code (see Appendix (A)). The R code here is only used for illustration purpose, thus may not be well optimized for real implementation. The transform function $bwt()$ starts with appending an extra character like '$' , which is lexicographically smaller than any of the characters present in S, to the beginning of the input sequence S. Then construct a matrix M with rows representing all possible cyclic shifts of S, then have rows of M sorted lexicographically. The BWT transformation of the input, T, is then the last column of the sorted matrix M. The original sequence S can also be reconstructed from the BWT transformed string T, via the inversion function $ibwt()$.

On the other hand, a suffix of S could be denoted as $S[i, N]$, where $i = 1, 2, \ldots, N$, with the starting position i as pointer to each suffix $S[i, N]$. A suffix array could then be constructed by sorting all suffixes in the lexicographical order, and assign the associated pointer to each array element. The core of FM-index is the pairing of BWT with suffix array, their connection could be easily seen if we place the suffix array next to the BWT transformed sequence. The jth character in the BWT transformed sequence is just the character one position before the jth suffix. For example, considering $S =$ 'mississippi', the only 'm' in the BWT encoded sequence T is ranked at the 5th position, while the corresponding suffix has an index of 2. The character before that suffix with index 2 is just the first character of the $(2-1)$th suffix which lies in the 6th row.

```
> exStr<-'mississippi'
> cbind(suffixArray(exStr), T=strsplit(bwt(exStr), '')[[1]])
               S  i T
1 |            $ 12 i
2 |           i$ 11 p
3 |        ippi$  8 s
4 |     issippi$  5 s
5 |  ississippi$  2 m
6 | mississippi$  1 $
7 |          pi$ 10 p
8 |         ppi$  9 i
9 |       sippi$  7 s
10|    sissippi$  4 s
11|      ssippi$  6 i
12|   ssissippi$  3 i
```

When looking at exact pattern matching problem, any matched pattern is essentially a prefix or a suffix of the full sequence. For a suffix array, since rows of suffixes have been sorted lexicographically, all occurrences of the pattern will be stacked together in consecutive rows. With the help of another two auxiliary data utilities, C and Occ, any suffix starts with the given query pattern can then be returned using backward search. $C[c]$ is a look-up table containing the number of occurrences of characters in the full sequence which are alphabetically smaller than c. $Occ(k, c)$ is a function which counts the occurrences of the character c in the k-th prefix of T, $T[1, k]$. Here I pre-computed all possible values of $Occ(k, c)$ as a matrix for the example sequence.

```
> occMat(bwt(exStr))
    $ i m p s
1 | 0 1 0 0 0
2 | 0 1 0 1 0
3 | 0 1 0 1 1
4 | 0 1 0 1 2
5 | 0 1 1 1 2
6 | 1 1 1 1 2
7 | 1 1 1 2 2
8 | 1 2 1 2 2
9 | 1 2 1 2 3
10| 1 2 1 2 4
```

```
11|  1 3 1 2 4
12|  1 4 1 2 4
> cMat(exStr)
$ i m p s
0 1 5 6 8
```

For a query pattern P of length p, the start index s and end index e of the sorted suffix array which contains P can be returned in a maximum of p steps. The searching starts with the last character of the query pattern, $P[p]$, with $s = 1$ and $e = p$. The two index are then iteratively recalculated using the following mapping scheme for each character $P[i]$ in the pattern, $s' = C[c] + Occ(s-1, P[i]) + 1$ and $e' = C[c] + Occ(e, P[i])$. The mapping process is illustrated in the example below with $P =$ 'iss'. At the last step ($p = 3$), the start and end index of suffix which starts with 'iss' have been located, $s = 4$ and $e = 5$.

```
               S  i T   0 1 2 3
1 |            $ 12 i | s
2 |           i$ 11 p |
3 |        ippi$  8 s |
4 |     issippi$  5 s |       s
5 |  ississippi$  2 m |       e
6 | mississippi$  1 $ |
7 |          pi$ 10 p |
8 |         ppi$  9 i |
9 |       sippi$  7 s |   s
10|    sissippi$  4 s |
11|      ssippi$  6 i |     s
12|   ssissippi$  3 i | e e e
```

2.1.3. Penalized uniqueness score

Gräf et al. [42] proposed the idea of using uniqueness score (U), which is the total number of minimum unique substring (MUS) in a given range, for cross-hybridization control in tiling probe selection. Following their original definition, a genome sequence in question is called G, which is a part of the whole genome assembly GS. If a substring X of G occurs only once in GS and each substring of X occurs more than once in GS, then this substring X is called a minimum unique substring (MUS) of G. At each position of G, if the substring from this position to the end of G is unique within GS, then there

exists one shortest unique prefix starting from that position, which is called a minimum unique prefix (MUP) at that position. The uniqueness score is then determined by counting the distinct end positions of MUP within the region.

```
> MUP(exStr, exStr)

     MUP.length m i s s i s s i p p i
1 *|          1 m
2  |          5   i s s i s
3  |          4     s s i s
4 *|          3       s i s
5  |          5         i s s i p
6  |          4           s s i p
7  |          3             s i p
8 *|          2               i p
9 *|          2                 p p
10*|          2                   p i
11 |          0                     -
```

In the code chunk above, using the same example string S from last section, the function *MUP* returns the length of the MUP at each position for string 'mississippi', with no MUP found at the last position. The searching of MUP takes the advantage of the relationship between suffix and prefix, since each prefix of the sequence is a suffix of the reversed sequence, and thus could be efficiently calculated with FM-index. Each minimum unique substring within are indicated by asterisk ($*$). In real implementation, MUP are efficiently located via an external library *GenomeTools* [48].

However, in this definition of the uniqueness score, the author only considered the absolute number of MUS without accounting for the distribution of length and coverage within the probe range. Hypothetically, the first half of a probe could contain no MUS, while the remainder might harbor a large number of MUS (which could be up to the user-specified cut-off). In certain extreme cases, for longer oligonucleotide probes, the original definition would give a score higher than the specified threshold, even though the actual coverage of these MUS is only 50%, and cross-hybridization could still occur. Another possible scenario is that compared to sequences with shorter MUS, probes with longer and near-window-sized MUS are more vulnerable to cross hybridization. A schematic illustration of the 2 aforementioned cases is shown in Figure 2.2.

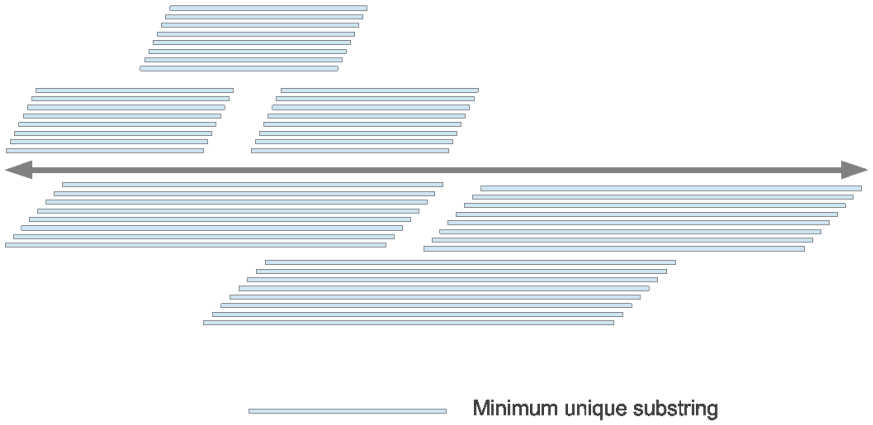


Figure 2.2.: Schematic diagram of low MUS coverage and long MUS. Hypothetically extreme cases showing low MUS coverage (above) and long MUS (below).

Hence the penalized uniqueness score U_p is defined as the following in Equation (2.6),

$$U_p = U \times C_{mus} \times (1 - mean(L_{mus})/L_{max}) \tag{2.6}$$

C_{mus} is the proportion of the probe covered by all MUS within. L_{mus} is the length of individual MUS within the candidate probe. L_{max} is the maximum possible length of MUP, which defaults to 30 and could be changed in the MUP searching step according to specific technical limits posed by the chip manufacture. U is the number of MUS within the candidate probe. So in the best case scenario, all MUS cover the whole probe and the coverage will induce no penalty on the uniqueness score. And regarding the average length of MUS, it will normally give a coefficient between 0 (only if all MUS within the probe have maximal length) and 1 (only when there is no MUS available). Thus as a rational number instead of integer count, the U_p provides a wider dynamic range.

To further illustrate the potential role of MUS coverage and length in probe uniqueness, the Agilent catalog array Human Whole Genome ChIP-on-Chip Set 244K (available via Agilent's eArray, see Appendix (B)) were fetched, which in total contains 5930500 probes spanning the whole human genome (hg19:GRCh37). All probes were processed for the two uniqueness scores and other oligonucleotide quality related measures. In addition, BLAST-Like Alignment Tool (BLAT) [49] was used to find hybridization-quality alignments, which is defined as having at most one gap, at least 60% identity of the probe length, and gap length or mismatched bases not more than 3. Such alignments are considered to be hybridized well, thus the number of qualified alignment could be

Table 2.2.: Summary of Agilent ChIP-on-Chip Set probe properties

Measurement	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
C_{mus} [a]	0.23	0.96	0.97	0.97	0.98	1
$mean(L_{mus})$ [b]	13.26	16.91	17.32	17.33	17.76	30
U [c]	0	21	24	23.06	26	38
U_p [d]	0	8.46	9.53	9.41	10.49	16.06
No. BLAT Hits	1	1	1	1.015	1	112
Probe Length	45	58	60	57.4	60	60
GC content (%)	10	26.67	33.33	34.19	39.66	86.67
Tm ($^\circ C$) [e]	58.29	67.98	70.73	70.59	73.08	92.07
Palindromic content (%) [f]	0	19.23	20	22.39	26.67	100
Repetitive proportion(%)	0	0	0	4.95	0	100
Max Base (%) [g]	25	35	38.33	39.13	43.33	71.67

[a] the percentage of the probe region covered by MUS
[b] the average length of all MUS within the probe
[c] the original uniqueness score
[d] the penalized uniqueness score
[e] the melting temperature evaluated using Equation (2.5)
[f] maximal proportion of inverted repeat (IR)
[g] maximal proportion of the four nucleotide bases

used as an indicator for cross hybridization potential. According to the chromosomal coordinates provided in the GEO files shipped together, probes were back-mapped to the same version of RepeatMasker-masked genome sequence (hg19:GRCh37). Proportion of masked bases was calculated for each mapped probe.

In Table 2.2, the calculated probe characteristics are summarized, which gives an overview of the general probe quality of the chip (figures of all parameters' distribution could be found in Additional files of [9]. In general, the catalog probes show similar hybridization efficiency, with an inter quartile range of melting temperature from 67.98$^\circ C$ to 73.08$^\circ C$; low cross hybridization potential, with an average uniqueness score (U) of 23.06 (median 24) and the penalized uniqueness score (U_p) of 9.412 (median 9.53); limited self-hybridization potential, having a mean palindromic content of 22.39% (median 20%). Back-mapping of Agilent chip targets to the reference sequence also suggests inclusion of repetitive sequence, with 411292 probes having repetitive proportion greater than zero, and 199471 probes being completely masked as repetitive.

After BLAT alignment, 99.4% of the probes found to have unique quality alignment, while only 34274 probes had been mapped to multiple locations. A back-to-back boxplot (Figure 2.3) was made for the two scores to visualize the general group differences

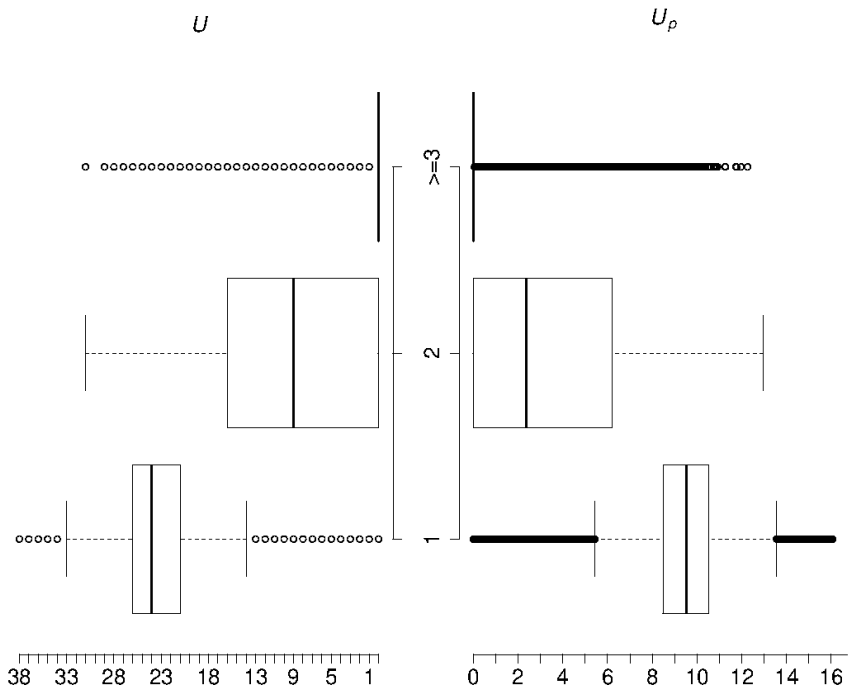


Figure 2.3.: Back-to-back box-plot of original uniqueness score (U, left) and the penalized uniqueness score (U_p, right) distribution within different BLAT hits groups (1, aligned to 1 position; 2, aligned to 2 positions; ≥ 3, aligned to at least 3 positions)

in their distributions. Both plots show similar overall pattern, probes with unique alignment tend to have higher scores than those with multiple hits. A visible difference of the grouping effect between the two scores could be found that, for the penalized uniqueness score the difference of median between group 2 and 3 is much lower than the difference of median between group 2 and group 1.

In Table 2.3, some exploratory probes are selected from this chip which are potentially vulnerable for cross-hybridization. The four 60 bp probes all show relatively high U values, close to the average level (mean=23.06, median=24) of all probes on the chip, and multiple quality alignments. The first two probes show relatively low coverage, which are far below the first quantile of all probes. The other two probes show relatively long average MUS length, which are above the third quantile (17.76) of all probes. However, when looking at the penalized uniqueness score, all four of them exhibit relatively low level of U_p, below the first quantile (8.46) of all probes on the chip. It is then suggestive that the penalized uniqueness score is more sensitive in assessing cross-hybridization potential, when taking into account of the size and positional distribution of MUS in the analysis.

Table 2.3.: Example of problematic 60-mer probes from Agilent ChIP-on-Chip Set

Probe ID	No. BLAT Hits [a]	MUS Coverage [b]	MUS Average Length [c]	U [d]	U_p [e]
A_17_P01761220	2	75%	17.55	22	6.85
A_17_P17428106	2	73.33%	16.42	24	7.97
A_17_P04386305	2	98.33%	20.86	22	6.59
A_17_P10898621	12	91.67%	20.86	22	6.14

[a] Number of hybridization-quality alignment to the reference genome.

[b] The percentage of the probe region covered by MUS.

[c] The average length of all MUS within the probe.

[d] The original uniqueness score.

[e] The penalized uniqueness score.

2.1.4. Tiling probe selection algorithm

Tiling probes can be selected in the most straightforward way, either using an end-to-end fashion or with a fixed distance or overlap between neighboring tiles. However, these simple strategies will easily encounter problems like cross-hybridization and low hybridization potential, problems that would eventually contaminate the data. Also, instead of having a high coverage up to 100% initially, the number of probes with valid signals could fall significantly after data processing. Thus an optimized and uniform tiling path is highly desirable [13]. Most of the available methods employ a window approach, which first divide the whole target region into non-overlapping fixed-size windows, and then select optimal probes within each window. Thus the resolution of the probe mapping will depend on the initial window size, this approach is preferred in CHIP-on-chip design when studying protein-DNA interaction or when high probe density is not of interest. However when mapping transcriptome, overlapping probes will provide better resolution in locating exon boundaries. To gain more control of the expected quality while giving better resolution and coverage, the following design strategy and pipeline which embed the previously defined penalized uniqueness score is presented. Full selection parameters could be seen in Table 2.4.

The algorithm, namely *OTAD*, searches for candidate probes in an intuitive growing fashion, from the 5′ of the sequence to the 3′ end. In general, neighboring probes is made to have a fixed size of overlap, if the targeted probe satisfied the user-specified constraints. Otherwise, the adjacent positions overlapping with 1 nucleotide more or less would be tested, and the search would keep shifting until the next valid probe is found or the boundary of the genomic region is reached. The shifting and checking is done intelligently to avoid unnecessary calculation.

Table 2.4.: Parameters of tiling probe selection algorithm

Parameter	Type	Description
-h		print help
-P	Integer	maximal parallel process (default: 1)
-f	Directory	path of folder containing fasta files with extension '.(m)fa(.gz)'
-w	Char	strand to be designed 'b' (both), '+', or '-' (default: +)
-l	Integer	maximal length of the probe (like: 60)
-v	Integer	maximal shrinking of probe length (default: 0)
-o	Integer	maximal length of overlapping (like: 20)
-u	Integer	minimal uniqueness score (like: 21)
-U	Real	minimal penalized uniqueness score (like: 9)
-T	Real	range of Tm calculated with nearest-neighbor model (like: 70-80)
-G	Real	range of GC content (default: 0.2-0.6)
-s	Integer	maximal single nucleotide repeats (default: 6)
-d	Integer	maximal di-nucleotide repeats (default: 4)
-b	Real	maximal proportion of each bases (default: 0.6)
-c	Integer	maximal number of synthesis cycles allowed (default: 148)
-p	Real	maximal proportion of palindromic sequence (like: 0.3)
-r	Real	maximal proportion of repetitive masked bases (like: 0.1)
		then the files in the input folder must contain 'mfa(.gz)'

Pseudo code of the detailed selection mechanism is shown in Appendix (C). An evenly-spaced, non-overlapping tiling path can also be achieved by specifying a negative value for the overlap size. Another separate option of variable probe length can be combined with the overlapping option to compensate for coverage in regions where fixed-length probes cannot be placed. The variable length design is also advantageous in selecting isothermal probes to reduce sequence bias [15]. Strand-specific design is also possible; if both strands are present on the same array, offsets between reverse complimentary tiles on the two strands are determined internally.

2.1.5. Penalized uniqueness score evaluation

Comparison of sensitivity and specificity

To further validate and evaluate the relative performance of the penalized uniqueness score against the uniqueness score, the previously processed Agilent Human ChIP-on-Chip Set was used. Receiver operating characteristic (ROC) measuring the trade-off between sensitivity and specificity was adopted to directly compare the discriminative power in non-unique probe classification. After the BLAT alignment, number of

hybridization-quality alignment was determined for each probe. In order to construct the curve, a series of values ranging from the minimum to the maximum of the uniqueness score are used as cut-offs to predict whether the probe has only one hybridization-quality alignment or more. If the probe has a uniqueness score higher than the cut-off, then it is classified as positive and estimated to have only one quality alignment, and negative otherwise. Sensitivity is defined as the true positive rate (TPR), which is the number of probes identified as positive and indeed having only one quality alignment divided by the number of probes with an alignment score equalled to one. Specificity is defined as 1 minus the false positive rate (FPR), which is the number of probes identified as positive but having more than one quality alignment divided by the number of probes actually having one quality alignment. Curves for both scores are overlaid in Figure 2.4. It could be seen that both scores work quite well and strongly deviate from the diagonal. However a visible trace of difference could be observed at the upper left corner, suggesting a clear gain of advantage by the penalized uniqueness score.

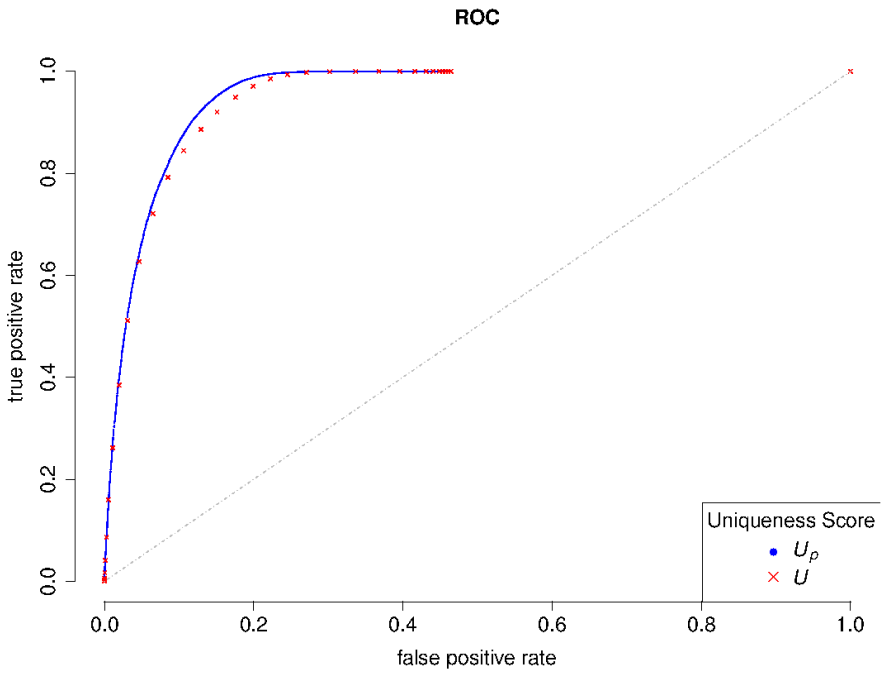


Figure 2.4.: ROC curves of using original uniqueness score (U, red) and the penalized uniqueness score (U_p, blue) for BLAT hits group classification.

Benchmarking of public array data

To further illustrate the discriminative power of the penalized uniqueness score, array data from public repository were evaluated. One popular platform from Agilent

was chosen, human whole-genome expression array 4x44K (ArrayExpress platform ID: A-AGIL-28), which contains 41000 unique probes. 14 single color array datasets are randomly selected from this platform, all having proper replicate (at least 2) for each factor level. Probes were scored using the penalized uniqueness score. For each experiment, like running a practical microarray analysis, pre-processing and filtering were done using Bioconductor[50] package *Agi4x44PreProcess* Pedro Lopez-Romero [51]. Experimental designs were derived according to individual experiment description. Procedural and parametric settings of background-correction, normalization and filtering were set to default, and common to all experiments. The filtering in *Agi4x44PreProcess* is done sequentially: first, control probes are filtered; then probes with signal not well above the local background are filtered; the third criteria is by the Agilent's FeatureExtraction flag 'gIsFound'; for the next step, probes with signal not well above the negative controls are filtered. So far, the remaining probes all have detectable and valid signal. In the next stage, over-saturated probes are removed, which could be linked to severe cross-hybridization or possible contamination to the slide; In the end, population outliers and non-uniform outliers are filtered, which could also be related to cross-hybridization or other variations in experimental conditions. In this work, probes filtered out in the last 2 stages were considered as potential victims of cross-hybridization and had them further investigated for uniqueness.

Table 2.5.: Summary of experiments from ArrayExpress

EXP_ID	Array No.	Factor level	Filtered No.	Filtered average [a]	Filtered std. dev.	Normal average [a]	t-test p-value
E-GEOD-22072	5	2	18	9.306 / 8.798	3.235	9.131 / 10.04	NA[b]
E-GEOD-23131	31	3	61	8.735 / 9.796	3.492	9.132 / 10.04	NA
E-GEOD-23558	32	2	136	4.784 / 3.503	4.246	9.146 / 10.05	2.20E-016
E-GEOD-23697	70	2	1	0 / 0	NA	9.131/ 10.04	NA
E-GEOD-24536	52	5	11	7.508 / 9.957	4.973	9.132 / 10.04	NA
E-GEOD-25623	32	3	1	4.655 / 4.655	NA	9.131/ 10.04	NA
E-GEOD-27915	20	5	6	9.633 / 9.899	2.824	9.131/ 10.04	NA
E-GEOD-29288	132	9	9	8.641 / 9.957	3.965	9.131/ 10.04	NA
E-GEOD-32155	21	7	23	8.48 / 9.613	3.642	9.132 / 10.04	NA
E-GEOD-32988	48	10	145	6.618 / 8.636	4.383	9.14 / 10.04	1.34E-010
E-GEOD-33264	49	16	31	9.428 / 9.957	2.76	9.131 / 10.04	NA
E-GEOD-35635	57	3	1	12.79 / 12.79	NA	9.131 / 10.04	NA
E-GEOD-35756	32	8	458	4.229 / 3.396	3.876	9.187 / 10.04	2.20E-016
E-GEOD-37827	87	29	32	9.078 / 9.701	2.837	9.131 / 10.04	NA

[a] U_p average column is formated as 'mean / median'.

[b] NA stands for 'Not Applicable' due to low number of filtered probe.

In the end, summary statistics of the penalized uniqueness score for filtered probes were derived, in Table 2.5, which are over-saturated or outlying and thus could be related to cross-hybridization. For the analyzed experiments, the number of probe filtered varies from 1 to 458. Interestingly, for all except one experiment (E-GEOD-35635), those filtered-out probes exhibit lower average U_p score and thus are less unique according to the uniqueness measurement. Difference in group mean of penalized uniqueness score was assessed using formal statistical test. In order to achieve statistical power of 0.9 when using a two sided t-test to detect a mean difference of 0.5 with standard deviation of 1, it would require at least 85 observations in each sample. Therefore t-test are only performed for those experiments which has more than 85 probes filtered out. For all three qualified experiments, significantly lower uniqueness are detected for those filtered-out probes. So re-analyzing public array data provided further support of using the penalized uniqueness score to discriminate non-specific probes for microarray design and data analysis.

2.1.6. Design comparison with commercial array

To address the coverage and resolution of the proposed probe selection pipeline, once again the previously processed Agilent Human ChIP-on-Chip Set was used. To simplify the comparison, only those probes targeting chromosome 22 (GenBank: NC_000022.10) were chosen rather than the whole set. The reference genome sequence was downloaded from the NCBI archive. Knowing that, for experiments like CHIP-chip, the sheared chromatin fragment is generally around 500bp [52], using short oligonucleotides (< 100-mer) makes it more cost-efficient to choose an optimal distance between tiles than to solely increase the number of probes and the tiling density. To compare with Agilent's catalog design, the distribution of the distance between neighboring tiles in the catalog array is summarized, with a median of 202 nucleotides and a mean of 426 nucleotides between adjacent probes. The probe length also varies, ranging from 45-mer to 60-mer (median=53 and mean=52.85). With the proposed implementation, several overlapping sizes (from -325 nt to -250 nt) have been tested to make the overall probe number close to the Agilent catalog design. The minimum probe length was set to 45-mer, while initial probe length always starts at 60- mer. For probe melting temperature, in Gräf et al. [42] they used the simple GC model like Equation (2.4) and tested two ranges of temperatures, $73-76^{\circ}C$ and $77-80^{\circ}C$, which in turn correspond to a GC content range of around 30-50% for a 50-mer oligo. In contrast, by considering the Tm distribution in the Agilent catalog array, a Tm range of $69-74^{\circ}C$ was selected for the nearest neighbor model Equation (2.5) and combined with a controlled GC content range of 30-50% as the criteria for optimal hybridization efficiency. For cross-hybridization control, a pe-

nalized uniqueness score of 9 was used as the empirical cut-off. A threshold of 30% for palindromic content was used, while all other parameters were kept as the following: no base proportion higher than 60%, single nucleotide repeats not exceeding 6 and not having more than 4 di-nucleotide repeats.

With the proposed implementation, it took around 2 hours on a 3.0 Ghz single core PC to finish the process, without enabling the parallel mode. In Table 2.6, the Agilent catalog design and several custom designs are summarized, showing that the proposed flexible strategy could achieve higher coverage with fewer and, on average, longer probes. The coverage was assessed using two types of measurement, the raw non-redundant bases covered (ambiguous base 'N' adjusted) and the total length of unit-sized (1000 bp) windows in which at least one probe was placed.

Table 2.6.: Design summary and coverage comparison

Design	Probe No.	Probe Len. [c]	Inter-Probe Dist. [d]	Base Coverage	Window Coverage [e]
Agilent [a]	73373	53 / 52.85	202 / 426	11.11%	51.25%
-o -250 [b]	75036	60 / 57.16	246 / 411.9	12.29%	53.60%
-o -275	72087	60 / 57.11	269 / 431.1	11.80%	53.66%
-o -300	69491	60 / 57.02	292 / 449.4	11.35%	53.74%
-o -325	67074	60 / 56.92	313 / 467.7	10.94%	53.81%

[a] Agilent Human Whole Genome ChIP-on-Chip Set 244K (Chromosome 22 only)

[b] Overlapping size set to -250, probe length range [45, 60], Tm range [69, 74], U_p range [9, Inf), palindromic content range [0, 30%], GC content range [30%, 50%]

[c] Average probes length in nucleotide, cell is formatted as median / mean

[d] Average inter-probe distance in nucleotide, cell is formatted as median / mean

[e] Using the length of all unit-sized (1000 bp) windows in which at least one probe was placed

2.1.7. Uniqueness of palindromic sequence

With increasing potential for self-hybridization, palindrome sequences play an important structural role in the biogenesis of microRNA [53; 54; 55] and also have other functional characteristics like acting as restriction enzyme sites [56]. Interestingly, unlike in Gräf et al. [42], where the author claimed that palindromic sequences are more unique in the genome, an opposing relationship between palindromes (measured as the maximal proportion of inverted repeat) and uniqueness are observed (Figure 2.5): Agilent's catalog probes with higher palindromic content tends to have lower mean uniqueness scores, which causes a left-shifting of their distributions when using a high palindrome cutoff. Small yet significant correlations could also be detected for both uniqueness scores [U, -0.0736428 (p<2.2e-16); U_p, -0.1050423 (p<2.2e-16)] and BLAT hits [0.00209 (p=3.578e-

07)] with palindromic content. However, the distribution of BLAT scores is extremely left-tilted, since most probes are aligned only once, therefore, despite supporting our findings, the correlation test may not be valid.

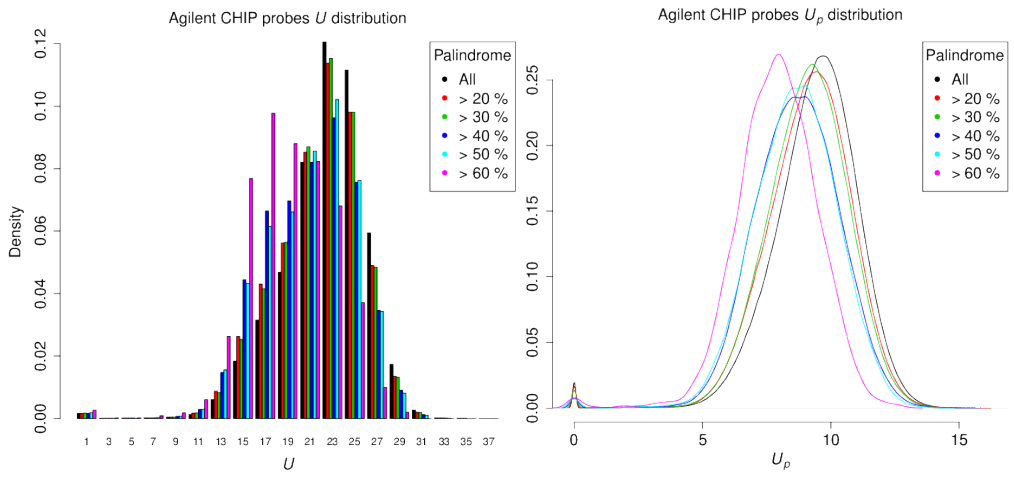


Figure 2.5.: Distributions of the original uniqueness score (U, left) and penalized uniqueness score (U_p, right) of the Agilent Human Whole Genome ChIP-on-Chip Set 244K (1 to25) probes, using different level of palindromic content as cut-off.

To investigate further the relationship between sequence uniqueness and palindromic content, data from a separate experiment were utilized, in which the proposed implementation has been used to design tiling arrays for several chromosome regions of the pig genome. For one of them, on *Sus Scrofa* chromosome 14 (GenBank: NC_010456.3) from 74377028 bp to 78176022 bp, the profiles of previously listed design parameters were evaluated for all possible 60-mer probes in the indicated region. In total 3328358 candidates were evaluated.

For the two uniqueness scores (Figure 5), unlike for the Agilent catalog array, their distributions are far from normal, both having a peak at zero and a flat, uniform interval followed by a narrow, bell-shaped region. This "twin-peak" distribution makes any formal statistical tests infeasible. However, by thresholding on palindromic content, the density of uniqueness scores were determined for candidates with palindromic content higher than the cut-off; in this way, sequences with higher palindromic content are directly visualized, which are subject to removal in the probe selection process. When using a higher cut-off of palindromic content, the trend of differences resembles that observed for the Agilent chip: the density curves tilt left, with both peaks shrinking and the saddle region raising. In particular, the proportion of sequences with close to 0 uniqueness score remains large in the high-palindrome group (>60%), yet the high

uniqueness peak vanishes.

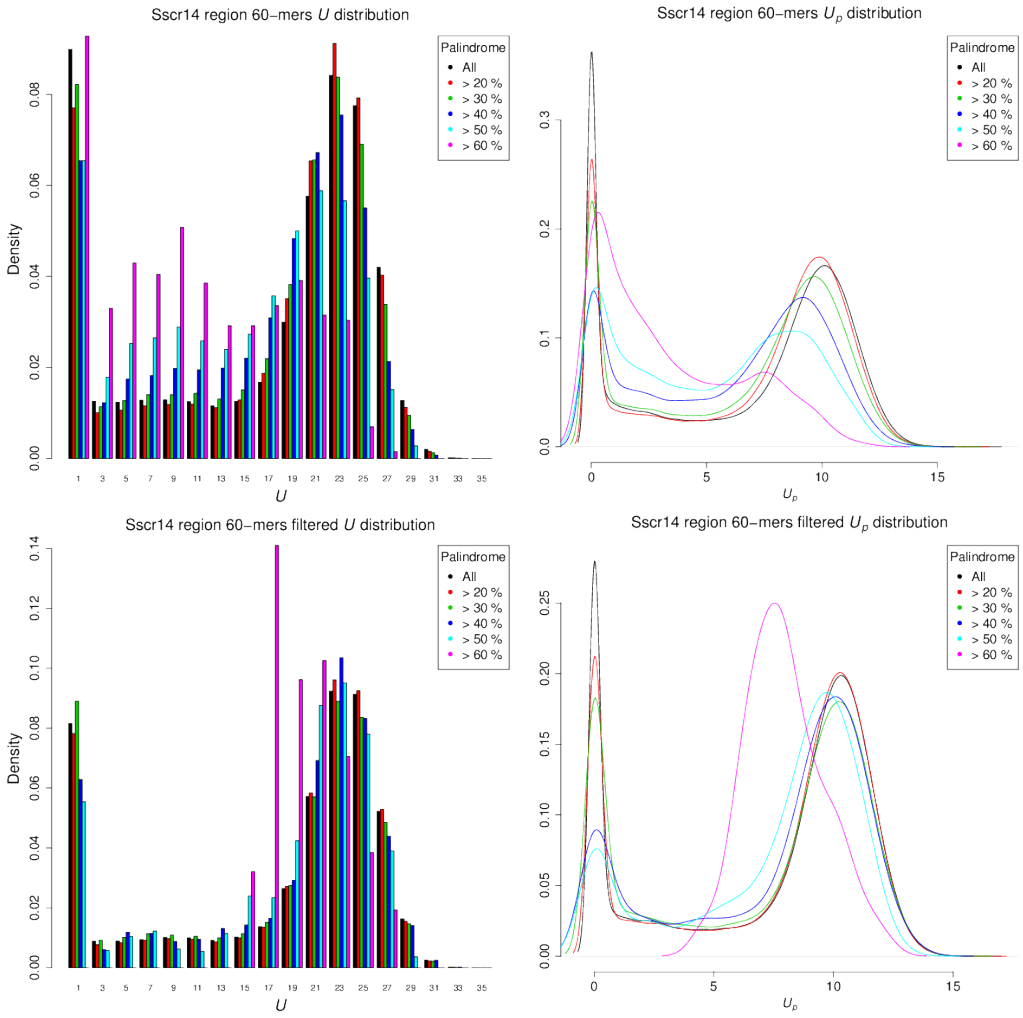


Figure 2.6.: Distributions of the original uniqueness score (upper-left) and penalized uniqueness score (upper-right) using different level of palindromic content as cut-off for 60-mer candidate probes on pig chromosome 14 (*Sus Scrofa* Build 10, NC_010456.3) region from 74377028 bp to 78176022 bp, and on the lower panel shows their distributions after filtering with standard probe selection criteria.

Finally, candidates violating those probe selection parameters were further filtered out, which include a narrow band of melting temperature ($69 - 74^{\circ}C$), moderate GC content (30%-50%), no base exceeding 60%, and no single and di-nucleotide repeats exceeding the thresholds of 6 and 4 respectively. After filtering, a similar pattern persists, yet is not so pronounced, suggesting that the filtering removed more candidates with low uniqueness scores. One particular feature is that the candidates with high palindrome (>60%) and 0 uniqueness scores were removed by filtering.

The results suggest that with higher palindromic content the sequence tends to have a lower uniqueness score, contrary to what has been previously claimed in [42]. Aside from nucleotide sequence, studied the palindrome in protein sequence using a linguistic measurement, in which they also related palindromes with low sequence complexity. Under the defined uniqueness measurement, lower complexity normally leads to fewer and longer MUS in the region, resulting in a lower penalized uniqueness score. The experimental observations made here could also be explained in plain theory, since a highly palindromic sequence will share a large identical segment with its reverse complement strand; thus there should be fewer unique substrings found in such regions.

2.2. Genomic segmentation

2.2.1. Common methods and issues

With high throughput experiments like tiling array and massively parallel sequencing, large scale genomic data are growing at an unforeseeable velocity. Researchers applying these experiments often look at genome-wide data searching for continuous homogeneous segments or signal peaks, which would represent regulatory regions [57; 58], transcripts [59; 60; 61; 62] or genome regions of deletion or amplification [63; 64]. The objective of these investigations could be generalized as the segmentation problem of partitioning the genome into non-overlapping homogeneous segments and assign biologically sensible class to each segment. As a long standing statistical problem, segmentation models have been widely studied. The origin of this field can be traced back to quality control in the manufacturing industry and the introduction of control charts by Walter Shewhart in 1920s, in which a 2 states model was built under the assumption of the process being homogeneously normal.

Various models and computational tools have been proposed to handle either the general segmentation problem or particular types of partitioning task in genomics. Some of the most addressed areas are copy number analysis with aCGH [65; 66; 67; 68; 69; 70; 71; 72; 73; 74] or SNP array [75; 76; 77; 78; 79], transcriptom profiling [80; 81] and protein-binding site detection [82; 83] with tiling array. In recent years, growing efforts have been devoted to the development of computational tools to deal with read-count data generated from next-generation sequencing (NGS) [84; 85; 86; 87; 88; 89].

Many of these computational tools utilize Hidden Markov Model (HMM) [65; 69; 75; 76; 78; 79; 82; 86; 88], since its natural capability of solving segmentation task and simultaneous labeling. However it is not straightforward for a standard HMM to take into account of a very basic property of genomic data—the physical position of the feature. To my knowledge, there have been some but limited attempts to incorporate this positional information into HMM [90; 82; 69; 78; 75; 79] or to adopt more complex dynamic Bayesian network (DBN) models [89]. Hidden semi-Markov Model (HSMM) on the other hand, as a generalized form of HMM, could be applied to take advantage of the extra information. Indeed, It has been presented to model aCGH data using HSMM [74], however without actually utilizing the positional information and the implementation is no longer publicly available.

In this section, I will introduce a novel HSMM implementation designed for various types of genomic segmentation applications. Its performance will be evaluated via

simulation benchmarking with other published tools. Various use cases will also be illustrated using published datasets.

2.2.2. Hidden semi-Markov Model

HMM was introduced by Baum and Petrie in the late 1960s. Soon afterwards, it has been widely used in engineering for speech or handwriting recognition [92; 93]. The application of HMM in biological sequence analysis exploded as the growing efforts in DNA-sequencing been made along the initiation of Human Genome Project [94], particularly in identifying RNA secondary structure and inferring phylogenies of different organismal DNA sequences.

HSMM, as an extension of HMM, was first proposed by Ferguson in the early 1980s, later but also in the field of speech recognition and signal processing. Its applications in biology and genomics are limited. Guédon et al. used HSMM to study branching and flowering patterns in plants. A variant of HSMM, termed Generalized Hidden Markov Model, was employed in GENSCAN [90] for gene prediction. It has also been used in protein structure prediction [97]. A full review of the HSMM methodology and its applications could be found in [98].

In the R/Bioconductor package *biomvRCNS*, a novel HSMM implementation is made available in function *biomvRhsmm* [10], which is specially designed to handle genomic data and tailored to serve as a general segmentation tool for various types of genomic profile, arising from both traditional microarray-based experiments and the recent NGS platform, with native support for modeling spatial pattern carried by genomic position.

Model definition

To start with, I will make a brief summary of the concepts involved and introduce the hidden semi-Markov model formulation. For some experimental data X, we have a vector of observations $x_t = (x_t^1, \ldots, x_t^N)$ made for N samples at each time or position t, $t = 1, \ldots, T$. At each t, there is an underlying unobserved state $S_t \in S = \{1, \ldots, J\}$, which depends only on the previous states chain at $t-1$, thus forming a length T discrete Markov chain with a finite number J possible states. The initial state probability is determined by distribution π, $\pi_j = P(S_1 = j)$, $j = 1, \ldots, J$, with $\sum_{j=1}^{J} \pi_j = 1$ and $\pi_j \geq 0$. The conditional probability distribution of the observed variable x_t given the unobserved (or hidden) state J, $b_j(x_t) = P(X_t = x_t \mid S_t = j)$, is controlled via the emission probability distribution B. The transition probability distribution A, governing

the probability of moving from one state to another, is formulated as $a_{ij} = P(S_{t+1} = j \mid S_t = i)$, with $\sum_{j=1}^{J} a_{ij} = 1$ and $a_{ij} \geq 0$. Thus a HMM can be defined by $\theta = (\pi, A, B)$, for which a schematic of the model parametrization is shown in Figure 2.7.

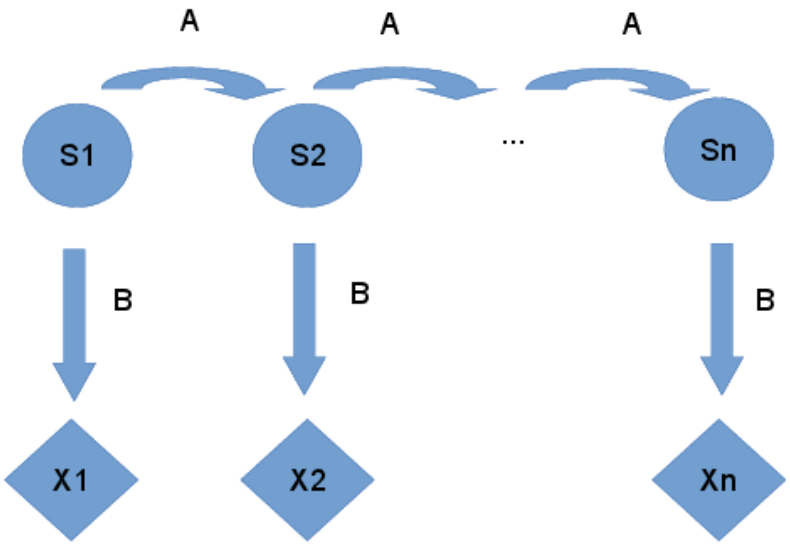


Figure 2.7.: Schematic of HMM parametrization

A semi-Markov chain could be considered as a two-layer mixture, an embedded first-order Markov chain representing the transitions between distinct states—which follows the standard definition of HMM—and an occupancy distribution attached to each non-absorbing state of the embedded first-oder Markov chain.

The discrete state occupancy distribution or the sojourn distribution, D, is defined as the probability of spending u consecutive time steps in state j,

$$d_j(u) = P(S_{t+u+1} \neq j, S_{t+u-v} = j, v = 0, \ldots, u-2 \mid S_{t+1} = j, S_t \neq j), \quad u = 1, \ldots, M_j \tag{2.7}$$

where M_j denotes the upper bound to the time spent in state j. For a normal HMM, the sojourn time could be simply deducted to $d_j(u) = a_{jj}^{u-1}(1 - a_{jj})$, which is geometrically distributed. HSMM, with the sojourn distribution explicitly specified using a common distribution or non-parametrically estimated using a pseudo sample, could be defined by $\theta = (\pi, A, B, D)$. A complete likelihood of the HSMM is given in Guédon [99] with survivor function $D_j(u) = \sum_{v \geq u} d_j(v)$ representing the sojourn time spent in the last

state and number of distinct states R,

$$\begin{aligned} L(\theta) = \pi_{S_1} d_{S_1}(u_1) \left\{ \prod_{r=2}^{R} P(S_r \mid S_{r-1}) d_{S_r}(u_r) \right\} \\ \cdot P(S_R \mid S_{R-1}) D_{S_R}(u_R) \prod_{t-1}^{T} P(X_t \mid S_t) \end{aligned} \tag{2.8}$$

where S_r is the rth state visited in the first-order Markov chain transition and S_R is the last states visited.

With likelihood function defined (Equation (2.8)), the optimal model parameters θ could then be estimated using the expectation—maximization (EM) algorithm. A forward-backward algorithm for the estimation step and a Viterbi algorithm to derive the most likely state sequence are explained in Guédon [99], where the author also shows the possibility of replacing the non-parametric M-step of the EM algorithm in sojourn distribution parameters re-estimation with a parametric M-step in practice, to simplify model and prevent over-fitting.

2.2.3. Estimation of hidden semi-Markov model

For the estimation step, as have been illustrated in Guédon [99], the forward recursion is first given by,

$$\begin{aligned} F_j(t) &= P(S_{t+1} \neq j, S_t = j \mid X_1^t = x_1^t) \\ &= \frac{b_j(x_t)}{N_t} \left[\sum_{u=1}^{t} \left\{ \prod_{v=1}^{u-1} \frac{b_j(x_{t-v})}{N_{t-v}} \right\} d_j(u) \sum_{i \neq j} a_{ij} F_i(t-u) \right. \\ &\quad \left. + \left\{ \prod_{v=1}^{t} \frac{b_j(x_{t-v})}{N_{t-v}} \right\} d_j(t+1) \pi_j \right], \end{aligned} \tag{2.9}$$

where $t = 1, \ldots, T-1$, $j = 1, \ldots, J$, and N_t is the normalizing factor, which could be derived during the forward recursion using Equation (2.10). $X_1^t = x_1^t$ is the shorthand form of $(X_1 = x_1, X_2 = x_2, \ldots, X_t = x_t)$, the same analogous abbreviation is also used for $S_1^t = s_1^t$. For the last state visited when $t = T$, the exact duration of the stay is unknown, however using the minimal staying time, the sojourn density $d_j(u)$ could be

replaced by the survivor function $D_j(u)$.

$$\begin{aligned} N_t &= P(X_t = x_t \mid X_1^{t-1} = x_1^{t-1}) \\ &= \sum_j b_j(x_t) \Bigg[\sum_{u=1}^{t} \left\{ \prod_{v=1}^{u-1} \frac{b_j(x_{t-v})}{N_{t-v}} \right\} D_j(u) \sum_{i \neq j} a_{ij} F_i(t-u) \\ &+ \left\{ \prod_{v=1}^{t} \frac{b_j(x_{t-v})}{N_{t-v}} \right\} D_j(t+1) \pi_j \Bigg], \end{aligned} \tag{2.10}$$

The smoothed probability $L_j(t) = P(S_t = j \mid X_1^t = x_1^t)$ at each position for a hidden semi-Markov chain can be decomposed and written as,

$$\begin{aligned} L_j(t) &= P(S_t = j \mid X_1^t = x_1^t) \\ &= L1_j(t) + L_j(t+1) - P(S_{t+1} = j, S_t \neq j \mid X_1^T = x_1^T), \end{aligned} \tag{2.11}$$

where $L1_j(t) = P(S_{t+1} \neq j, S_t = j \mid X_1^T = x_1^T) = B_j(t) F_j(t)$ gives the conditional independence between future and past at transition between distinct states, which also provides the entry point for the backward recursion. $L_j(T)$ is initialized as $L_j(T) = P(S_T = j \mid X_1^T = x_1^T) = F_j(T)$ for $t = T$ and all j.

The backward recursion is done by pre-calculating another auxiliary variable, $G_j(t+1)$, which helps reduce the complexities of the forward-backward procedure to $O(JT(J+T))$ time and $O(JT)$ space in the worst case. $L1_j(t)$ and the third term in Equation (2.11) could then be written as,

$$L1_j(t) = \left\{ \sum_{k \neq j} G_k(t+1) a_{jk} \right\} F_j(t), \tag{2.12}$$

$$P(S_{t+1} = j, S_t \neq j \mid X_1^T = x_1^T) = G_j(t+1) \sum_{i \neq j} a_{ij} F_i(t), \tag{2.13}$$

where $G_j(t+1) = \sum_{u=1}^{T-t} G_j(t=1, u)$, and

$$\begin{aligned} G_j(t+1, u) &= \frac{L1_j(t+u)}{F_j(t+u)} \left\{ \prod_{v=0}^{u-1} \frac{b_j(x_{t+u-v})}{N_{t+u-v}} \right\} d_j(u), u = 1, \ldots, T-1-t, \\ G_j(t+1, T-t) &= \left\{ \prod_{v=0}^{T-1-t} \frac{b_j(x_{T-v})}{N_{T-v}} \right\} D_j(T-t). \end{aligned} \tag{2.14}$$

For the parameter re-estimation step, the initial probabilities and transition probabilities could be updated at each EM iteration,

$$\hat{\pi_j} = P(S_1 = j|X_1^T = x_1^T;\theta) = L_j(1), \tag{2.15}$$

$$\hat{a_{ij}} = \frac{\sum_{t=1}^{T-1} G_j(t+1)a_{ij}F_i(t)}{\sum_{t=1}^{T-1} L1_i(t)}. \tag{2.16}$$

Using components calculated during the forward-backward run, the updates of state occupancy probabilities and emission probabilities are done as the following, Equation (2.17), depending on the assumptions imposed on the emission distribution ($I(x_t)$) and sojourn distribution. Equation (2.18) gives the non-parametric E-step as shown in [99].

$$\hat{b_j}(x_t) = \frac{\sum_{t=1}^{T} L_j(t)I(x_t)}{\sum_{t=1}^{T} L_j(t)}, \tag{2.17}$$

$$\hat{d_j}(u) = \frac{\eta_{ju}}{\sum_{t=0}^{T-1} L1_j(t) + L_j(T)}, \tag{2.18}$$

where the quantities η_{ju} could be computed during the backward procedure as in Equation (2.19).

$$\begin{aligned}\eta_{ju} = &\sum_{t=1}^{T-1} P(S_{t+u+1\neq j, S_{t+u-v}=j, v=0,\ldots,u-1}, S_t \neq j \mid X_1^T = x_1^T;\theta) \\ &+ P(S_u \neq j, S_{u-v} = j, v = 1,\ldots,u|X_1^T = x_1^T;\theta)\end{aligned} \tag{2.19}$$

The first term in Equation (2.19) could be further re-written, when $u \leq T-1-t$,

$$P(S_{t+u+1\neq j, S_{t+u-v}=j, v=0,\ldots,u-1}, S_t \neq j \mid X_1^T = x_1^T;\theta) = G_j(t+1,u)\sum_{i\neq j} a_{ij}F_i(t), \tag{2.20}$$

and for $u > T-1-t$,

$$P(S_{t+u+1\neq j, S_{t+u-v}=j, v=0,\ldots,u-1}, S_t \neq j \mid X_1^T = x_1^T;\theta) = \left\{\prod_{v=0}^{T-1-t} \frac{b_j(x_{T-v})}{N_{T-v}}\right\} d_j(u)\sum_{i\neq j} a_{ij}F_i(t) \tag{2.21}$$

The second term in Equation (2.19) could also be represented using pre-computed products, when $u \leq T$,

$$P(S_u \neq j, S_{u-v} = j, v = 1,\ldots,u|X_1^T = x_1^T;\theta) = \frac{L1_j(u-1)}{F_j(u-1)}\left\{\prod_{v=1}^{u} \frac{b_j(x_{u-v})}{N_{u-v}}\right\} d_j(u)\pi_j, \tag{2.22}$$

and for $u > T$,

$$P(S_u \neq j, S_{u-v} = j, v = 1, \ldots, u | X_1^T = x_1^T; \theta) = \left\{ \prod_{v=1}^{T} \frac{b_j(x_{T-v})}{N_{T-v}} \right\} d_j(u) \pi_j. \tag{2.23}$$

The quantities η_{ju} can also be treated as a pseudo-sample of some selected parametric sojourn distributions. Thus other parametric re-estimations basing on continuous and discrete distributions like Gamma, Poisson, and Negative Binomial distribution can be done *ad hoc*, using point estimation methods like moment estimator or maximum likelihood estimator, with additional shift parameter d to control the minimum stay duration in a state. The shift parameter d is determined by assessing possible values, $1, \ldots, min(u \mid \eta_{ju} > 0)$, of which gives the maximum likelihood of the re-estimated sojourn mass.

To obtain the most likely state sequence, a Viterbi procedure, using a similar forward recursion, could be applied by defining the quantities $\alpha_j(t)$ as maximum conditional likelihood of having a transition of state after the current t, $\alpha_j(t) = \max_{S_1,\ldots,S_t} P(S_{t+1} \neq j, S_t = j, S_1^{t-1} = s_1^{t-1}, X_1^t = x_1^t)$. For $t \leq T-1$, $\alpha_j(t)$ could be re-written as,

$$\begin{aligned} \alpha_j(t) = & \max_{S_1,\ldots,S_{t-1}} P(S_{t+1} \neq j, S_t = j, S_1^{t-1} = s_1^{t-1}, X_1^t = x_1^t) \\ = & b_j(x_t) \max \left[\left\{ \prod_{v=1}^{t} b_j(x_{t-v}) \right\} d_j(t+1) \pi_j, \right. \\ & \left. \max_{1 \leq u \leq t} \left[\left\{ \prod_{v=1}^{u-1} b_j(xt-v) \right\} d_j(u) \max_{i \neq j} \{ p_{ij} \alpha_i(t-u) \} \right] \right]. \end{aligned} \tag{2.24}$$

While for $t = T$ again using $D_j(t)$, the right censoring of the sojourn time in the last state visited, $\alpha_j(T)$ is formulated as the following,

$$\begin{aligned} \alpha_j(T) = & \max_{S_1,\ldots,S_{T-1}} P(S_T = j, S_{T-1} = j, S_1^{t-1} = s_1^{t-1}, X_1^T = x_1^T) \\ = & b_j(x_T) \max \left[\left\{ \prod_{v=1}^{T} b_j(x_{T-v}) \right\} D_j(T+1) \pi_j, \right. \\ & \left. \max_{1 \leq u \leq T} \left[\left\{ \prod_{v=1}^{u-1} b_j(xT-v) \right\} D_j(u) \max_{i \neq j} \{ p_{ij} \alpha_i(T-u) \} \right] \right]. \end{aligned} \tag{2.25}$$

Thus the most likely state sequence associated with the observed data sequence could then be backtracked by finding the state j which maximize $\alpha_j(t)$.

The E-step of the forward-backward EM procedure and the Viterbi algorithm described in Guédon [99] has been implemented as C library in the R/Bioconductor package *biomvRCNS*, serving as the core of our proposed hidden semi-Markov segmentation model.

2.2.4. R Implementation

The batch function *biomvRhsmm* accepts both basic R data matrix and more encapsulated object like *GenomicRanges* [100] as input, for better interfacing with other Bioconductor classes and methods. The function will sequentially process each region identified by the distinctive sequence names in the positional input. A second layer of stratification is introduced by a grouping argument, assigning each profile to a group, which could be used to reflect experimental design. Sample columns within the same group could be treated simultaneously in the modeling process as well as iteratively. The assumption is that profiles from the same group could be considered homogeneous, thus processed together. This joint analysis is only possible for emission distribution type set to multivariate normal distribution or multivariate t distribution. Additionally there is a built-in automatic grouping method by hierarchical clustering.

Priors of the sojourn distribution parameters will be initialized as flat or estimated from other related data source by calling function *sojournAnno*. State number could be either assigned explicitly or inferred during the sojourn learning. The model complexity is limited by a constant of M, which denotes the upper bound to the time spent in a state, very similar to the approach adapted in the segmentation model in *tilingArray* [81]. The constant could be explicitly given by the argument *maxk* or inferred by another constant *maxbp* together with positional information. The modeling of sojourn time is done using the positional information like genomic distance between markers, and regresses to a rank-based position setting, like the original design in [99], when positional information is not available. Starting state probabilities will be initialized as a flat vector. Initial parameters for the emission distribution could be estimated using different levels of quantile of the input or via a clustering process, assuming different states tend to have different levels of emitted signal.

The function will then call the C library to compute the smoothed state probability profile in the E-step, after which model parameters will be re-estimated in an M-step. Eventually, the most likely state sequence could be inferred from the smoothed state probability profile or estimated with the Viterbi algorithm. The complexity of the forward-backward algorithm used in the E-step and the Viterbi algorithm is $O(JT(J + T))$ time and $O(JTM)$ space in the worst case.

After the batch run, results will be combined and returned together with input data plus model parameters as an object of class *biomvRCNS*, for which a plot method has been implemented to provide integrative visualization of the segmentation results with optional annotation. To relax the high memory burden from NGS data of base pair resolution, RLE is used for the storage and handling of sequencing count data. Since the mapping distribution of sequencing features is normally sparse across the genome, which is due to the existence of large intergenic gaps between transcribed or functional regions.

2.2.5. Simulation benchmarking

In order to show the reliability and relative performance of the proposed model, the implementation `biomvRhsmm` has been compared with several other state-of-the-art segmentation algorithms (Table 2.7) in **Yang Du** et al. [10], using a similar approach as in Lai et al. [101], by calculating the receiver operating characteristic (ROC) curves on simulated data.

Some of the models reviewed in Lai et al. [101] have evolved over the years. Venkatraman and Olshen presented a faster and modified version of Circular Binary Segmentation (CBS) in R/Bioconductor package *DNAcopy*. Picard et al. extended the univariate dynamic programming procedure [68] to joint analysis of multiple CGH profiles in R package *cghseg*, and adopted the modified Bayesian information criterion [102] for model selection. The unsupervised hidden Markov model described in R package *aCGH* [65] (labeled *HMM* hereafter) was also included, and the local adaptive weights smoothing procedure in R package *GLAD* [66] in this comparison, which are considered to be the early efforts in the field. Thus they could serve as baselines in the comparison, and also to show advances and development in the field.

In recent years, several new methods and computational tools have also been introduced. In R package *bcp*, Erdman and Emerson implemented an efficient Bayesian change point model described by Barry and Hartigan [103]. Ben-Yaacov and Eldar suggested an ultrafast segmentation model based on wavelet decomposition and thresholding in R package *HaarSeg*. Marioni et al. implemented a heterogeneous hidden Markov model *bioHMM* in R/Bioconductor package *snapCGH*, which can utilize positional information or clone quality in the modeling process, thus could be considered as an extension of the *HMM* in package *aCGH*. Among these models, there has been no comparison study between *bcp*, *bioHMM* and *HaarSeg* in recent literature.

Table 2.7.: List of segmentation algorithms compared

ID	Reference	Method	R package
bcp	Erdman and Emerson [72]	Product partition model	bcp_3.0.1
bioHMM	Marioni et al. [69]	Heterogeneous HMM	snapCGH_1.30.0
CBS	Venkatraman and Olshen [70]	Circular Binary Segmentation	DNAcopy_1.34.0
cghseg	Picard et al. [73]	Joint CGH segmentation	cghseg_1.0.1
GLAD	Hupé et al. [66]	Adaptive Weights Smoothing	GLAD_2.24.0
HaarSeg	Ben-Yaacov and Eldar [71]	Wavelet decomposition	HaarSeg_0.0.3
HMM	Fridlyand et al. [65]	Homogeneous HMM	aCGH_1.38.0
hsmm	**Yang Du** et al. [10]	Hidden semi-Markov model	biomvRCNS_1.3.1

Data Simulation

For the data simulation, I tried to make it conceptually similar to the scenario one may encounter in real experiments. For copy number studies using CGH or using sequencing with matched case control sample, three states are commonly assumed, and regions of copy gain and loss are of major interest whose size may range from about 1 kb to some megabases [104]. For this purpose, pools of segments for each state was first created; lengths of the segments were sampled from three Poisson distributions, with lambda equals to 20, 270 and 10, respectively. The distance between data points was assumed to be regular and equals to 1. Signal intensities were sampled from three Normal distributions, $N_1(r, 1)$, $N_2(2 \times r, 1)$, $N_3(3 \times r, 1)$ for each state, respectively, with state mean controlled via a ratio factor r varying from 1 to 3 at a step of 1. Segments from different states were then randomly sampled and joined together to form one data sequence.

For sequencing data, in order to check for splicing and novel transcripts or detect peaks for transcript factor binding sites, one would be mainly interested in distinguishing the true expression signal from the background. Normally, annotated coding or non-coding transcripts are relatively much shorter comparing to intergenic regions. In this case, we also first created pools of segments for three virtual states, intergenic, short and relatively lowly expressed gene and protein coding sequence with high abundance; length of the segments were sampled from three Poisson distributions, with mean parameter λ equals to 285, 5 and 10, respectively. Signal intensities for each segment were then sampled from three pools of Poisson distribution, $P_1(1)$, $P_2(r)$, $P_3(r^2)$, with mean parameter λ controlled via a ratio parameter r varying from 1.5 to 2 at a step of 0.25 for each pool of segments. Segments from different states were then randomly sampled and joined together to form one data sequence, representing one targeted region.

R code for data simulation is available in Appendix (D).

Performance comparison

The proposed HSMM was compared with several well tested segmentation algorithms, all of which are available as R packages. Since different algorithms tend to be tuned differently to suit their own methodologies for better sensitivity, I did not attempt to alter their default settings and fed only the simulated signals without other information to the models, thus achieving an essentially fair comparison and mimicking a common use case for normal users.

Using simulated data with varying levels of inter-state ratio r, which is conceptually similar to signal-to-noise ratio (SNR); since for both simulations, states with extreme values are of interests, thus the differences in mean between the extreme states and the intermediate states could be considered as signal, while the variation associated with the intermediate state could be considered as noise. I calculated the TPR and the FPR over 10000 iterations (100 simulations for each of the 100 random segments formations) of simulation for each level of r. The TPR is defined as the number of points which are from the states of interest and fall into the predicted states of interest divided by the total number of points from the states of interest. The FPR is defined as the number of points, which are not from the states of interest but fall into the predicted states of interest divided by the total number of points not from the states of interest. The true states of interest depend on the type of simulation, for normal data in simulation 1, this is assigned to the first and the third state namely the abnormal state separately; for count data in simulation 2, this is assigned to the third state, which is used to represent signal peak. The prediction is done by comparing the estimated segment mean with a threshold (t) varying from the maximum to the minimum of the simulated signal value. For abnormal state of gain in simulation 1 and peak in simulation 2, segment with estimated value above the threshold is considered as positive; while for state of loss in simulation 1, segment with estimated value below the threshold is considered as positive. Definition of TPR and FPR are formulated in Equation (2.26).

$$\begin{aligned} TPR_{loss} &= \frac{N(x < t | s = 1)}{N(s = 1)}, FPR_{loss} = \frac{N(x < t | s \neq 1)}{N(s \neq 1)} \\ TPR_{gain|s2} &= \frac{N(x > t | s = 3)}{N(s = 3)}, FPR_{gain|s2} = \frac{N(x > t | s \neq 3)}{N(s \neq 3)} \end{aligned} \tag{2.26}$$

All calculations were carried out in the statistical language R (version 3.0.1). area under the curve (AUC) was estimated using Bioconductor package *ROC* [105]. The

system used for benchmarking is a standard 64-bits Linux desktop with Intel Core i7 3.07 GHz and 6 GB DDR3 memory.

R code for performance evaluation is also available in Appendix (D).

Benchmarking results

After two extensive simulation runs, the resulting ROC curves under different signal to noise ratios for all compared models are shown in Figure 2.9. In Figure 2.8, two sets of randomly simulated data (chosen from the 50th random grid formation and the 50th iteration of that formation), one from each simulation run (using the intermediate r level, 2 for simulation 1 and 1.75 for simulation 2), have been illustrated as an example together with estimated segments from competing models.

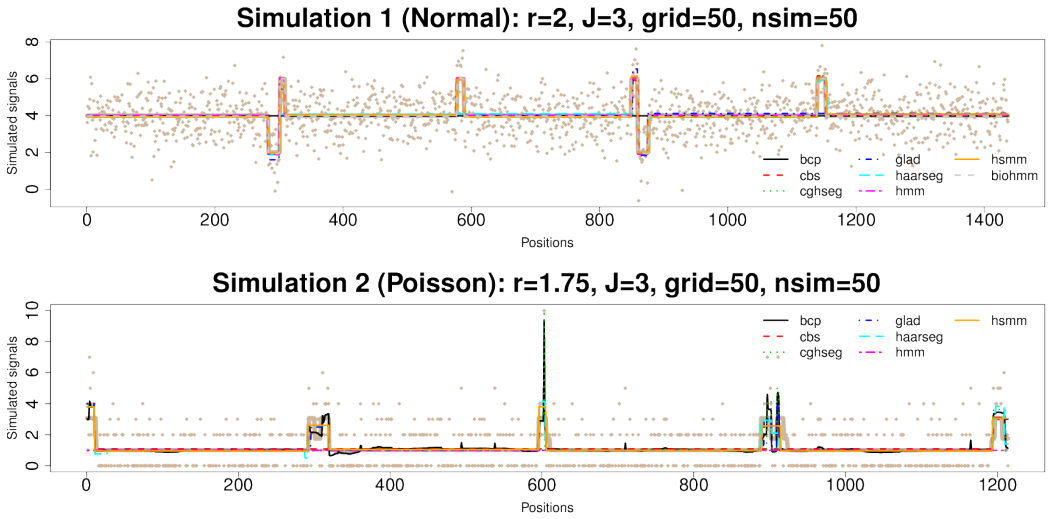


Figure 2.8.: Two sets of randomly simulated data (chosen from the 50th random grid formation and the 50th iteration of that formation), one for each simulation run (using the intermediate r level, 2 for simulation 1 and 1.75 for simulation 2) have been illustrated as an example together with estimated segments from competing models. Segments are represented using the estimated segment averages. The true underlying grid used for data simulation is shown as the solid line in beige.

In simulation 1 (Table 2.8), most algorithms except for *HMM* perform comparably well at intermediate and low noise scenarios. The difference in detecting gain and loss is consistent with our simulation setup, where the loss region is intentionally set to be relatively longer making it much easier to detect. In general the competing algorithms could be categorized into three classes, with our model, *bioHMM* and *HaarSeg* top the charts, colsely followed by *CBS* and *cghseg*, and the other three algorithms perform less

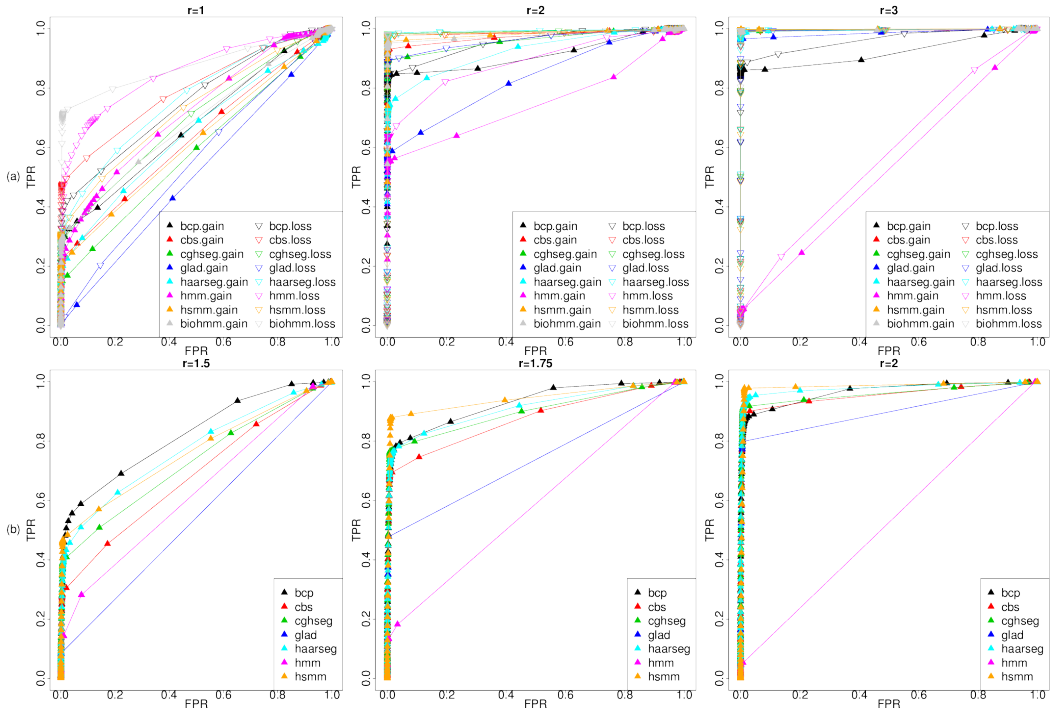


Figure 2.9.: ROC curves for segmentation algorithms comparison under different signal to noise settings (r). The curves were generated by measuring the sensitivity and the specificity at different threshold levels. The x-axis and y-axis show the FPR and the TPR respectively. The upper panel (a) shows the simulation 1 which is similar to an aCGH analysis, and the lower panel (b) shows the simulation 2 which is similar to peak identification using NGS. Compared algorithms are color coded as indicated in the figure legend, while the up-triangle represents segment of gain in simulation 1 and peak in simulation 2, and hollow down-triangle represents segment of loss in simulation 1. Models are labeled using lower case letters of their name. Our proposed model is coded as *'hsmm'* for simplicity and the *HMM* in package *aCGH* is labeled as *'hmm'*.

satisfactorily. It is worth mentioning that, in simulation 1, *bioHMM* has surprisingly high power in high noise setting. However, such advantage essentially disappears when signals get stronger. This could be related to its model selection process, where it attempts to assign higher number of states thus more segments to compensate for the random noise. There is also a clear difficulty for *HaarSeg* to detect short gain segments, which could be related to the default model setting that is not well adapted to short aberrations [71].The behavior of *bcp* indicates that in order to achieve higher power there is an inevitably loss in specificity, even with high signal to noise setting.

For smoothing algorithm like *GLAD*, it only operates well under higher signal to noise ratio. Due to the smoothing, segments boundaries became less accurate. And as mentioned in Lai et al. [101], *GLAD* is sensitive to single outliers, which explains the deficiency of sensitivity in detecting gain region even for low noise cases. HMM gets quite high AUC when high noise exists ($r = 1$) in simulation 1, and performs comparably worse when signals are stronger, eventually fails to identify most segments. This is in accordance with Lai et al. [101], where *HMM* failed to identify any region in Glioblastoma Multiforme (GBM) data. It also fails to make any meaningful segmentation in simulation 2.

Table 2.8.: Area under the ROC curves of simulation data 1

Sim 1	r=1		r=2		r=3		weighted
	AUC_g[a]	AUC_l[a]	AUC_g	AUC_l	AUC_g	AUC_l	avg.rank[b]
bcp	0.675 3	0.758 0	0.912 0	0.956 0	0.921 1	0.963 4	6.416 5
bioHMM	0.685 6	0.875 2	0.977 1	0.990 2	0.995 1	0.997 6	3.408 4
CBS	0.633 3	0.795 9	0.974 0	0.985 5	0.996 1	0.995 4	4.491 1
cghseg	0.586 5	0.696 3	0.960 2	0.991 8	0.996 0	0.998 1	4.621 1
HaarSeg	0.649 7	0.763 7	0.923 4	0.993 7	0.995 9	0.998 4	3.887 0
GLAD	0.506 6	0.548 8	0.833 0	0.962 9	0.986 1	0.996 6	6.907 6
HMM	0.717 6	0.854 8	0.749 5	0.887 2	0.526 4	0.573 6	6.589 8
hsmm	0.619 5	0.729 7	0.982 2	0.988 7	0.999 1	0.998 5	3.528 4

[a] AUC_g and AUC_l are AUC for simulated gain and loss segments respectively for each r.

[b] Weighted avg.rank is calculated as $n + 1 - \sum_{j=1}^{j=c} AUC_i \times rank^j(AUC_i)/c$ for each model i, where c is the number of AUC columns and n is the number of competing models.

In simulation 2 (Table 2.9), when data is a mixture of Poisson distributions, I failed to run *bioHMM* due to an error in a foreign function call to the C library. I have to assume that the implementation cannot work on discrete count data. However all other

implementations are still operable and achieve similar performance as in simulation 1. Though the mean parameter for Poisson data simulation is not considerably large, the normal approximation could still achieve reasonably good power. Nonetheless, our explicit modeling of count data is still advantageous for segmenting count data, which has the highest weighted average rank (Table 2.9), followed by *bcp* and *HaarSeg*. Comparing to *HaarSeg*, the power boost for *bcp* essentially occurs under higher false positive rate. It could be seen that algorithms like *bcp* perform better when stronger signal exists, which could be due to the normal error assumption in the model.

Table 2.9.: Area under the ROC curves of simulation data 2

Sim 2	$AUC_{r=1.5}$	$AUC_{r=1.75}$	$AUC_{r=2}$	weighted avg.rank[a]
bcp	0.821 9	0.928 1	0.965 6	2.616 6
CBS	0.682 8	0.874 4	0.954 9	5.487 9
cghseg	0.729 2	0.898 3	0.959 5	4.550 7
GLAD	0.541 8	0.736 7	0.897 1	6.730 2
HaarSeg	0.772 8	0.911 4	0.978 7	2.977 9
HMM	0.624 3	0.587 3	0.526 0	7.212 7
hsmm	0.762 3	0.942 4	0.984 9	2.232 4

[a] Weighted avg.rank is calculated as $n + 1 - \sum_{j=1}^{j=c} AUC_i \times rank^j(AUC_i)/c$ for each model i, where c is the number of AUC columns and n is the number of competing models.

For both simulations, as has been shown in Lai et al. [101], *cghseg* and *CBS* perform consistently well under various scenarios. Three of the newly introduced methods, *bcp*, *bioHMM* and *HaarSeg* also achieve comparable or better performance, whereas the HSMM consistently ranks among the top 3 performing algorithms when considering AUC. Across the two simulations, *GLAD* and *HMM* are considered to process lowest power. Concerning computation time, *HaarSeg* is the fastest algorithm among all implementations, by a factor of 50-100, while *bioHMM* is the slowest due to its internal model selection process. *bcp* is the second slowest, as a result of long Markov chain Monte Carlo (MCMC) run. The processing time of the HSMM is similar to *cghseg*, and is slower comparing to *CBS*, which is about two times faster.

Concerning overall accuracy of estimated segments number, Occam's razor states that the best model should be the simplest yet still retaining the same power. For both simulations, on average 14 segments were joined into one sequence. In simulation 1, the HSMM achieves the lowest rooted mean squared error (RMSE) and the mean absolute error (MAE); whereas in simulation 2, HSMM finds fewer segments with the median number of detected segments only 6 across three noise levels. Taking into account the

Table 2.10.: Processing time and error estimates of the compared models

		Simulation 1			Simulation 2		
	avg.t[a]	maxcp[b]	MAE[c]	RMSE[d]	maxcp	MAE	RMSE
hsmm	0.25645	13	5.756	3.476	9	18.73	6.31
bcp	1.46298	NA[e]	NA	NA	NA	NA	NA
bioHMM	6.96811	192	8.655	7.376	NA	NA	NA
CBS	0.12168	15	7.444	4.178	10	20.762	7.068
cghseg	0.28938	13	9.059	4.783	17	14.83	5.533
GLAD	0.23725	11	13.128	6.071	12	22.139	7.668
HaarSeg	0.00268	27	12.896	4.984	22	10.117	4.018
HMM	0.28008	386	94.666	80.792	365	144.83	97.178

[a] avg.t is calculated as the mean run time of 2000 simulation iterations.

[b] maxcp is the maximal number of segments produced across 3 SNR settings.

[c] $MAE = \sum |est.no.seg - true.no.seg| / n$.

[d] $RMSE = \sqrt{\sum (est.no.seg - true.no.seg)^2 / n}$.

[e] NA indicates that the measurement is not applicable for this algorithm. For bcp, the model output posterior means for each position that does not tend to form segments with constant mean. For bioHMM, the model cannot be run, thus no results were collected.

power advantage of the HSMM method in the sensitivity analysis, it indicates that the estimated segments boundaries are more accurate in HSMM. This could be due to the fact that the simulated aberrant segments are sampled from the same distribution and the sojourn modeling in the HSMM clearly takes advantage of such property. In the second simulation, *HaarSeg* achieves lowest error estimates for both RMSE and MAE. Moreover, *cghseg* also has similar error estimates as the HSMM. Like *HaarSeg*, this is essentially achieved by fitting more segments. As has been pointed out in [68], assumptions of the mean-variance relationship imposed on the model may lead to more segments in order to satisfy such requirements.

2.2.6. Segmentation of copy number profiles

Microarray-based comparative genomic hybridization has been used to study DNA copy number aberrations, and has been considered as an effective diagnostic tool in medical genetics and cancer research. Extracted from packge *DNACopy* [70], the `coriell` data contains two aCGH studies (GM05296 and GM13330) of Corriell cell lines taken from Snijders et al. [63], with which I will first illustrate the usage of the proposed HSMM in copy number analysis. In particular, the data contains normalized copy-number ratios between cancer cell strains and normal reference DNA, in total with 2271 mapped features across 22 autosomes and chromosome X. To get started, we first build a *GRanges* object from *data.frame*, one can also supply a data matrix with optional positional information as input.

```
> data('coriell', package='biomvRCNS')
> head(coriell, n=3)
       Clone Chromosome Position Coriell.05296 Coriell.13330
1 GS1-232B23          1        1      0.000359      0.207470
2 RP11-82d16          1      469      0.008824      0.063076
3 RP11-62m23          1     2242     -0.000890      0.123881

> xgr<-GRanges(seqnames=coriell[,2],
+  IRanges(start=coriell[,3], width=1, names=coriell[,1]))
> values(xgr)<-DataFrame(coriell[,4:5], row.names=NULL)
> xgr<-sort(xgr)
> head(xgr, n=3)
GRanges with 3 ranges and 2 metadata columns:
             seqnames       ranges strand | Coriell.05296 Coriell.13330
                <Rle>    <IRanges>  <Rle> |     <numeric>     <numeric>
  GS1-232B23        1 [   1,    1]      * |      0.000359       0.20747
  RP11-82d16        1 [ 469,  469]      * |      0.008824      0.063076
  RP11-62m23        1 [2242, 2242]      * |      -0.00089      0.123881
  ---
```

```
  seqlengths:
    1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
   NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
```

Then by passing the input object to `biomvRhsmm`, the copy number states will be estimated using the hidden-semi Markov model. The batch function will sequentially process each chromosome identified by their unique *seqnames*.

```
> rhsmm<-biomvRhsmm(x=xgr, maxbp=1E5, J=3, soj.type='gamma',
+  com.emis=T, emis.type='norm', prior.m='quantile')
> show(rhsmm)
Object is of class: 'biomvRCNS'
List of parameters used in the model:
J, maxk, maxbp, maxgap, soj.type, emis.type, q.alpha, r.var, iterative, cMethod, maxit, tol,
grp, cluster.m, avg.m, prior.m, trim, na.rm, soj.par, emis.par

The segmented ranges:
GRanges with 102 ranges and 3 metadata columns:
        seqnames            ranges strand   |        SAMPLE STATE        AVG
           <Rle>         <IRanges>  <Rle>   |         <Rle> <Rle>      <Rle>
    [1]        1 [     1, 108746]      *   | Coriell.05296     2  0.0091220
    [2]        1 [112204, 218166]      *   | Coriell.05296     2  0.0138270
    [3]        1 [110293, 110293]      *   | Coriell.05296     1 -0.0791300
    [4]        1 [220439, 240001]      *   | Coriell.05296     1 -0.0083905
    [5]        1 [     1,  36207]      *   | Coriell.13330     3  0.0874010
    ...      ...              ...    ... ...           ...   ...        ...
   [98]       22 [ 20553,  33001]      *   | Coriell.13330     3   0.130433
   [99]       23 [     1, 155001]      *   | Coriell.05296     3   0.676184
  [100]       23 [     1,  98906]      *   | Coriell.13330     2  -0.053510
  [101]       23 [125572, 155001]      *   | Coriell.13330     2  -0.012260
  [102]       23 [103194, 122966]      *   | Coriell.13330     1  -0.101480
  ---
  seqlengths:
    1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
   NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
```

In the above run, we limited the model complexity by setting the *maxbp* to $1E5$, which will restrict the maximum evaluated sojourn length to *maxbp*. *J* is the number of states in the HSMM, in which case three states can be assumed for aCGH studies, copy loss region, normal region, or duplicated region.

Argument *emis.type* controls the distribution of emission probability, in this case the log2 ratio of aCGH data is considered to follow Normal distribution. The emission density could be estimated using all data or only data on the respective region or chro-

mosome (identified by unique `seqnames`), controlling via *com.emis*. In this case, the ratios cross chromosomes are directly comparable, thus *com.emis* was set to true. The prior of the emission parameters could be controlled by supplying *q.alpha* and *r.var* with *prior.m='quantile'*, or automatically determined through a clustering process with *prior.m='cluster'*.

The function will then call C codes and estimate the most likely state sequence, with either *cMethod='F-B'* or *cMethod='Viterbi'*. The *F-B* method (default) uses a forward-backward algorithm described in Guédon [99], which gives a smooth state sequence, whereas the Viterbi algorithm with *cMethod='Viterbi'* will use the state profile estimated by the forward-backward algorithm and rebuild the most likely state sequence. The parameter *maxit* controls the maximum iteration of the EM algorithm. When assessing aCGH data, the quantile method should be able to give a good estimation of the emission density priors, one can also adjust *q.alpha* and *r.var* for better control over the mean-variance relationships in extreme states. Since we are not training a prediction model, but trying to derive the most likely state sequence, one iteration of the EM procedure is sufficient.

The function returns an object of class *biomvRCNS*, in which the `res` slot is a *GRanges* object containing the summary of each estimated segments. There are three meta columns: column `SAMPLE` gives the column name of which sample this segment belongs to; column `STATE`, the estimated state for each segment, the lower state number represents state with lower mean value, thus in this example, a state of 1 could represent region of deletion and 3 for region of duplication, whereas state 2 could be considered copy neutral; column `AVG`, gives the segment average value, which could take the form of (trimmed) mean or median controlled by *avg.m*. The original input is also kept and returned in slot `x` with the estimated most likely state assignment and associated probability.

A `plot` method has been implemented for *biomvRCNS* object using R/Bioconductor package *Gviz*, by default the `plot` method tries to output graphics to multiple EPS/PDF files for each chromosome region and sample. Multiple samples could also be overlaid on the same image, by passing *sampleInOne=TRUE* in the `plot` method. In Figure 2.10, a copy loss region on chromosome 11 from sample Coriell.05296 is shown.

```
> obj<-biomvRGviz(exprgr=xgr[seqnames(xgr)=='11', 'Coriell.05296'],
+  seggr=rhsmm@res[mcols(rhsmm@res)[,'SAMPLE']=='Coriell.05296'])
```

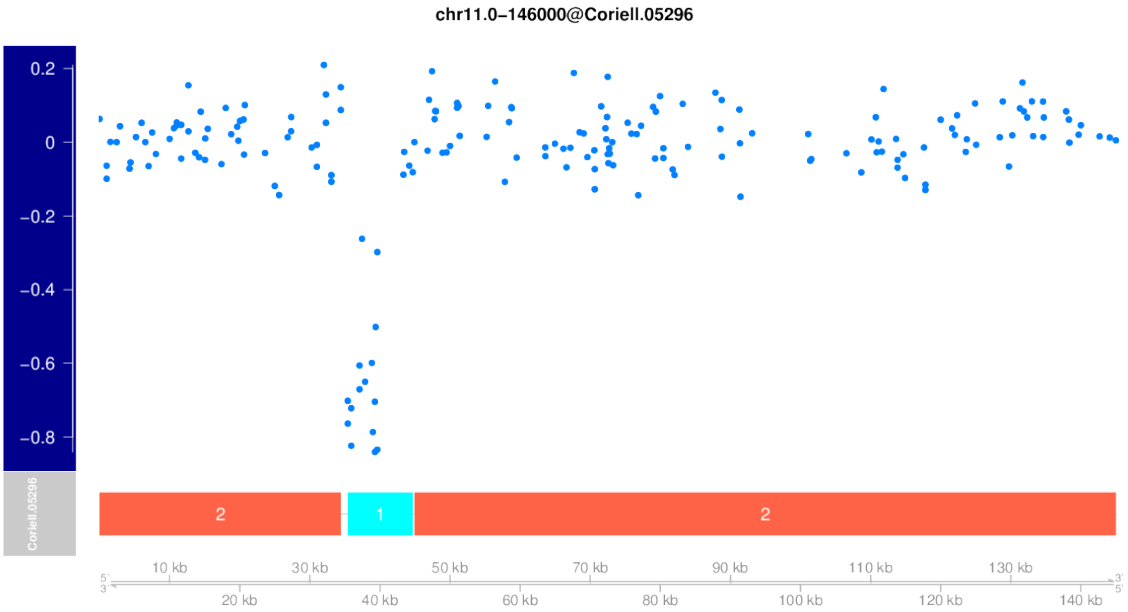


Figure 2.10.: Estimated copy number states of sample 05296 from the Coriell aCGH dataset. A state of 1 could represent region of deletion and 3 for region of duplication, whereas state 2 could be considered copy neutral.

2.2.7. Transcript detection with mRNA-seq data from ENCODE

The newly prevailing NGS technology has enabled the deep profiling of transcriptome at an unprecedented depth, allowing base-pair resolution detection of novel transcripts and splicing events. Recent study has shown that thousands of unannotated long non-coding RNAs are transcriptionally active [106].

In this section, I will illustrate the usage of `biomvRhsmm` in transcriptome mapping. The data contains gene expressions and transcript annotations in the region of the human TP53 gene (chr17:7,560,001-7,610,000 from the Human February 2009 (GRCh37/hg19) genome assembly), which is part of the long RNA-seq data generated by ENCODE [107] / Cold Spring Harbor Lab, containing 2 cell types (GM12878 and K562) with 2 replicates each. The libraries were sequenced on the Illumina GAIIx platform as paired-ends for 76 or 101 cycles for each read. The average depth of sequencing is 200 million reads (100 million paired-ends). The data were mapped against hg19 using Spliced Transcript Alignment and Reconstruction (STAR).

To generate local read counts, alignment files were pulled from UCSC ([1]) using R / Bioconductor package *Rsamtools*. And subsequently reads were counted in each non-

[1]http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeCshlLongRnaSeq/

overlapping unit sized window for the region. In the pre-compiled data `encodeTP53` , a window size of 25 bp was used with the chunk of code below.

```
> winsize<-25
> cgr<-GRanges("chr17", strand='-',
+  IRanges(start=seq(7560001, 7610000, winsize), width =winsize))
> bf<-system.file("extdata", "encodeFiles.txt", package = "biomvRCNS")
> bamfiles<-read.table(bf, header=T, stringsAsFactors=F)
> library(Rsamtools)
> which<-GRanges("chr17", IRanges(7560001, 7610000))
> param<-ScanBamParam(which=which, what=scanBamWhat())
> for(i in seq_len(nrow(bamfiles))){
+  frd<-scanBam(bamfiles[i,1], param=param)
+  frdgr<-GRanges("chr17", strand=frd[[1]]$strand,
+  IRanges(start=frd[[1]]$pos , end = frd[[1]]$pos+frd[[1]]$qwidth-1))
+  mcols(cgr)<-DataFrame(mcols(cgr), DOC=countOverlaps(cgr, frdgr))
+ }
```

Alternatively one can also operate on base pair resolution, in which case a *Rle* object should be preferred to store the count data for lower memory footprint and better efficiency. Also to speed things up, one could set *useMC=T* to enable parallel processing of multiple `seqnames`, the number of parallel process could be set by *options(mc.cores=n)*.

```
> cgr<-GRanges("chr17", strand='-',
+  IRanges(seq(7560001, 7610000), width=1))
> bf<-system.file("extdata", "encodeFiles.txt", package = "biomvRCNS")
> bamfiles<-read.table(bf, header=T, stringsAsFactors=F)
> library(Rsamtools)
> which<-GRanges("chr17", IRanges(7560001, 7610000))
> param<-ScanBamParam(which=which, flag=scanBamFlag(isMinusStrand=TRUE))
> for(i in seq_len(nrow(bamfiles))){
+  cod<-coverage(BamFile(bamfiles[i,1]), param=param)[['chr17']][7560001:7610000]
+  mcols(cgr)<-DataFrame(mcols(cgr), DOC=cod)
+ }
```

The pre-compiled data `encodeTP53` also includes the regional annotation of TP53 RNAs isoforms, `gmgr`, which were derived from the manually curated ENCODE Gene Annotations (GENCODE) [2], and subset to only isoforms of TP53 gene and neighboring genes in the region.

```
> af<-system.file("extdata", "gmodTP53.gff", package = "biomvRCNS")
```

[2] http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeGencodeV4/)

```
> gtfsub<-read.table(af, fill=T, stringsAsFactors=F)
> idx<-gtfsub[,3]=='CDS' | gtfsub[,3]=='UTR'
> gmgr<-GRanges("chr17", IRanges(start=as.integer(gtfsub[idx, 4]), end=as.integer(gtfsub[idx, 5]),
+  names=gtfsub[idx, 13]), strand=gtfsub[idx, 7], TYPE=gtfsub[idx, 3])
```

We first load the encodeTP53 data, pool the read counts for each cell type and add 1 to the base count to increase stability.

```
> data(encodeTP53, package='biomvRCNS')
> cgr<-encodeTP53$cgr
> gmgr<-encodeTP53$gmgr
> mcols(cgr)<-DataFrame(
+  Gm12878=1+rowSums(as.matrix(mcols(cgr)[,1:2])),
+  K562=1+rowSums(as.matrix(mcols(cgr)[,3:4])) )
```

For count data from sequencing, the *emis.type* could be set to either 'pois' or 'nbinom', though 'pois' is preferred for sharp boundary detection. For the sojourn settings, instead of using the uninformative flat prior, we here use estimates from other data source as a prior. We load the *TxDb.Hsapiens.UCSC.hg19.knownGene* (version 2.10.1) known gene database, and pass the *TranscriptDb* object to parameter *xAnno*. Then internally sojourn parameters and state number *J* will be estimated from *xAnno* by calling function sojournAnno. When given a *TranscriptDb* object to *xAnno*, state number would be set to 3 and each represents 'intergenic', 'intron' and 'exon', respectively. One can also supply a named *list* object with initial values for parameters of distribution specified by *soj.type*. Using the sojourn parameters estimated from the known transcripts database, one can visualize the sojourn density and compare it with the empirical distribution of different features, like in Figure 2.11 where Gamma distribution was used. There is no confirmed biological justification for using any specific parametric distribution in modeling length of genome units. The Gamma distribution used here could be considered relevant, since Gamma distribution has been frequently used to model waiting time. For emission, given the highly dispersed nature of count data, we set the prior for emission mean to be more extreme, with *q.alpha=0.01*.

```
> library(TxDb.Hsapiens.UCSC.hg19.knownGene)
> txdb<-TxDb.Hsapiens.UCSC.hg19.knownGene
> sojournAnno(txdb)
$type
[1] "gamma"

$fttypes
[1] "intergenic" "intron"     "exon"
```

```
$J
[1] 3

$shape
[1] 0.07911853 0.08877242 0.16583121

$scale
[1] 2180784.013   70520.286    2053.809

> rhsmm<-biomvRhsmm(x=cgr, xAnno=txdb, maxbp=1E3, soj.type='gamma',
+  emis.type='pois', prior.m='quantile', q.alpha=0.01)
> rhsmm@res[mcols(rhsmm@res)[,'STATE']=='exon']
GRanges with 52 ranges and 3 metadata columns:
       seqnames              ranges strand   |  SAMPLE STATE   AVG
          <Rle>           <IRanges>  <Rle>   |   <Rle> <Rle> <Rle>
   [1]    chr17 [7571801, 7572125]      -   | Gm12878  exon   312
   [2]    chr17 [7572251, 7572350]      -   | Gm12878  exon    96
   [3]    chr17 [7572426, 7572550]      -   | Gm12878  exon    61
   [4]    chr17 [7572601, 7572625]      -   | Gm12878  exon    60
   [5]    chr17 [7572851, 7573050]      -   | Gm12878  exon   127
   ...      ...                 ...    ... ...     ...   ...   ...
  [48]    chr17 [7588951, 7589400]      -   |    K562  exon  20.0
  [49]    chr17 [7589426, 7589525]      -   |    K562  exon   6.0
  [50]    chr17 [7589676, 7589825]      -   |    K562  exon   9.0
  [51]    chr17 [7590701, 7590800]      -   |    K562  exon  14.5
  [52]    chr17 [7592026, 7592050]      -   |    K562  exon   6.0
  ---
  seqlengths:
   chr17
      NA
```

As in the ENCODE guide [108], the study identified the p53 isoform observed in K562 cells has a longer 3'UTR than the isoform seen in the GM12878 cell line. So here we plot our model estimates and consider the third state, namely 'exon', to represent detected transcripts. And the HSMM model clearly picked up the extra transcripts of the K562 cell line at the 3'UTR. Now we can also locate those novel detected fragments in K562 cell line comparing to the annotation and those detected in Gm12878 cell line. One can then follow up these findings either by gene structure prediction using local nucleotides composition or by experimental validation.

```
> g<-mcols(rhsmm@res)[,'STATE']=='exon' & mcols(rhsmm@res)[,'SAMPLE']=='Gm12878'
> k<-mcols(rhsmm@res)[,'STATE']=='exon' & mcols(rhsmm@res)[,'SAMPLE']=='K562'
> exon<-mcols(rhsmm@res)[,'STATE']=='exon'
> obj<-biomvRGviz(exprgr=cgr[,'K562'], gmgr=gmgr, seggr=rhsmm@res[exon],
```

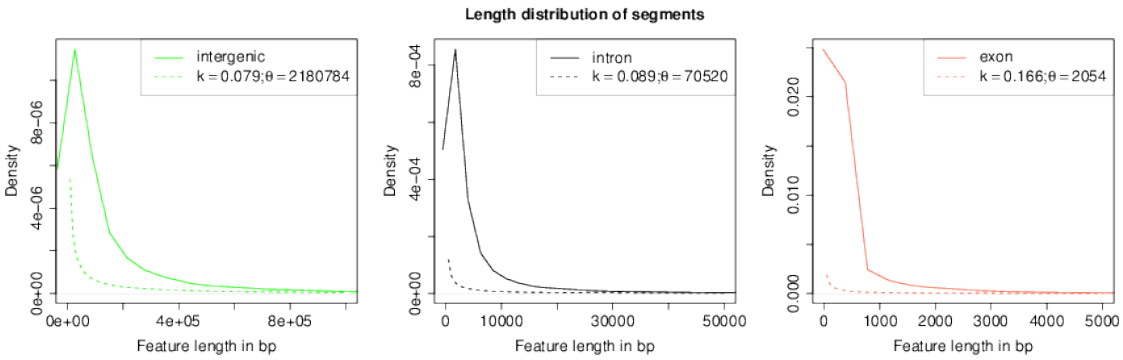


Figure 2.11.: Sojourn distribution parameters estimated using Gamma distribution from known gene database . The solid curve gives the segment length distribution of each feature type, 'intergenic', 'intron' and 'exon'. The dotted line gives the estimated density of each state. k and θ are the shape and scale parameters for Gamma distribution.

```
+ plotstrand='-', regionID='TP53', tofile=FALSE)

> nK2gm<-findOverlaps(rhsmm@res[k], gmgr)@queryHits
> nK2G<-findOverlaps(rhsmm@res[k], rhsmm@res[g])@queryHits
> rhsmm@res[k][setdiff(seq_len(sum(k)), unique(c(nK2G, nK2gm)))]
GRanges with 19 ranges and 3 metadata columns:
       seqnames              ranges strand   | SAMPLE STATE   AVG
          <Rle>           <IRanges>  <Rle>   |  <Rle> <Rle> <Rle>
   [1]    chr17 [7569151, 7569225]      -   |   K562  exon     9
   [2]    chr17 [7569651, 7569925]      -   |   K562  exon    15
   [3]    chr17 [7570301, 7570550]      -   |   K562  exon    16
   [4]    chr17 [7570751, 7570850]      -   |   K562  exon    10
   [5]    chr17 [7570901, 7571000]      -   |   K562  exon     8
   ...      ...                 ...     ... ...     ...   ...   ...
  [15]    chr17 [7587201, 7587225]      -   |   K562  exon     8
  [16]    chr17 [7588826, 7588850]      -   |   K562  exon     6
  [17]    chr17 [7589426, 7589525]      -   |   K562  exon     6
  [18]    chr17 [7589676, 7589825]      -   |   K562  exon     9
  [19]    chr17 [7592026, 7592050]      -   |   K562  exon     6
  ---
  seqlengths:
   chr17
      NA
```

After the model run, one also gets access to the updated sojourn and emission distribution parameters, which could be used to generate summary of the states or used as parameter input in other related modelings.

```
1> rhsmm@param$soj.par['chr17',]
```

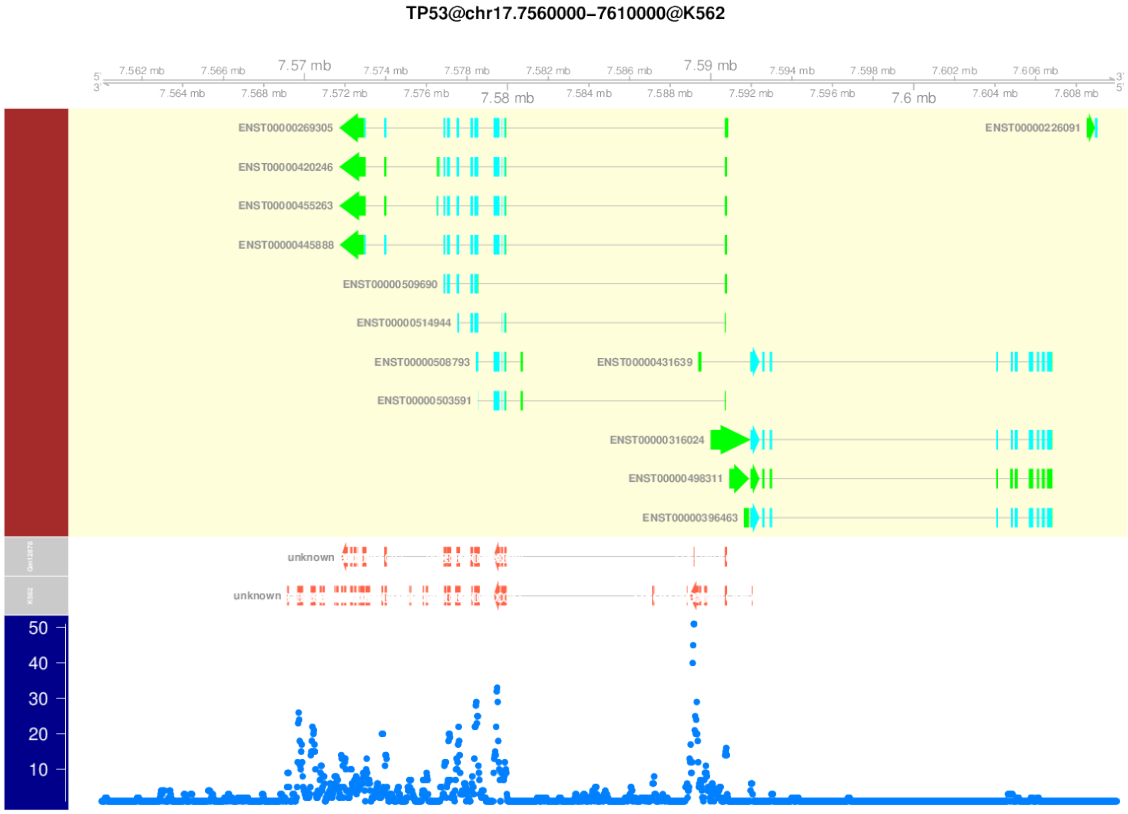


Figure 2.12.: Novel splicing detected in the UTR of TP53 gene in K562 sample. On the upper panel, the annotated CDS (cyan) and UTR (green) elements within the region are illustrated and grouped by transcript. The two rows (Gm12878 and K562) in the center present the segments labeled as 'exon' in the HSMM estimation. The lower scatter plot shows the read coverage in K562 sample.

```
$Gm12878
$Gm12878$shape
[1] 0.5963197 0.6214228 1.9855437

$Gm12878$scale
[1] 464.05423 456.93630  65.81886

$K562
$K562$shape
[1] 0.4838741 0.5038419 0.8982046

$K562$scale
[1] 484.0769 477.5418 113.6743

1> rhsmm@param$emis.par['chr17',]
$Gm12878
$Gm12878$mu
[1]   7.056160   8.156769 191.523558

$K562
$K562$mu
[1]  1.972035  2.112671 12.101558
```

2.2.8. Detection of differentially methylated regions

Differentially methylated regions (DMRs) are genomic regions with different methylation status, i.e. variable degree of DNA methylation between different samples, which has been considered to have regulatory functions for gene transcription [109] and is associated with cell differentiation and proliferation [110; 111]. Such regions could be surveyed using high-throughput technology like tiling array [112] and sequencing [113]. As an example, we include a set of data extracted from *BiSeq* [114], which contains a small subset of a published study [115], comprising intermediate differential methylation results prior to DMRs detection. We first load the `variosm` data,

```
> data(variosm, package='biomvRCNS')
```

The data contains a *GRanges* object `variosm` with two meta columns: `meth.diff`, methylation difference between the two sample groups; `p.val`, significance level from the Wald test. Our model could be applied on data from other pipelines as well, using similar data input. In the *BiSeq* work-flow, they use an approach similar to the max-gap-min-run algorithm to define DMRs boundaries, by prior filtering and comparing

the differential test statistics with a user specified significance level in the candidate regions. The positional information of methylation sites is taken into account by locating and testing highly correlated cluster regions in the filtering process. With `biomvRhsmm`, we utilize both types of information to detect DMRs: (1) the difference in the methylation ratio and (2) the significance level from differential test. The methylation difference gives information about the directionality of the change as well as the size, and the significance level gives the confidence in claiming differential events. We implicitly ask the model to give 3 states, since J is default to 3. Regarding the methylation ratio (`meth.diff`), these levels may represent hypomethylated regions, undefined null regions, or hypermethylated regions, respectively. When modeling significance levels (`p.val`), these states would represent highly confident regions, lowly confident regions or null results. For both scenarios, we are more interested in extreme states, where we have consistent direction of differences and low P-values. However, the distribution of `p.val` and `meth.diff` are both non-uniform and asymmetric around 0 (`meth.diff`) and 0.5 (`p.val`), we thus enable the cluster mode for emission prior initialization by setting *prior.m='cluster'*. The 'cluster' mode will employ the method described in Kaufman and Rousseeuw [116] to divide data into clusters, then using the centroid of each cluster to represent the mean parameter, and also for the variance structure or other distributional parameters can be estimated using the corresponding clusters. Due to the non-uniformly located CpG sites, one may split inter-spreading long segments with parameter *maxgap=100*.

```
> rhsmm<-biomvRhsmm(x=variosm, maxbp=100, prior.m='cluster', maxgap=100)
> hiDiffgr<-rhsmm@res[mcols(rhsmm@res)[,'STATE']!=2
+  & mcols(rhsmm@res)[,'SAMPLE']=='meth.diff']
> dirNo<-mcols(hiDiffgr)[,'STATE']=='1' & mcols(hiDiffgr)[,'AVG']>0 |
+  mcols(hiDiffgr)[,'STATE']=='3' & mcols(hiDiffgr)[,'AVG']<0
> hiDiffgr<- hiDiffgr[!dirNo]
> loPgr<-rhsmm@res[mcols(rhsmm@res)[,'STATE']==1
+  & mcols(rhsmm@res)[,'SAMPLE']=='p.val']
> DMRs<-reduce(intersect(hiDiffgr, loPgr), min.gapwidth=100)
> idx<-findOverlaps(variosm, DMRs, type='within')
> mcols(DMRs)<-DataFrame(cbind(TYPE='DMR', aggregate(as.data.frame(mcols(variosm[idx@queryHits])),
+  by=list(DMR=idx@subjectHits), FUN=median)[,-1]))
> names(DMRs)<-paste0('DMRs', seq_along(DMRs))
> DMRs
GRanges with 5 ranges and 3 metadata columns:
        seqnames            ranges strand |     TYPE   meth.diff        p.val
           <Rle>         <IRanges>  <Rle> | <factor>   <numeric>    <numeric>
  DMRs1     chr1 [875227, 875470]      * |      DMR  0.31947418 6.677193e-06
  DMRs2     chr1 [876807, 876958]      * |      DMR -0.06108219 6.500328e-02
  DMRs3     chr1 [877684, 877738]      * |      DMR -0.06123008 2.844639e-02
```

```
  DMRs4     chr2 [ 46126,  46280]      * |      DMR  0.41008524 1.818530e-07
  DMRs5     chr2 [ 46389,  46558]      * |      DMR  0.44823172 1.890819e-06
  ---
  seqlengths:
   chr1 chr2
     NA   NA
> plot(rhsmm, gmgr=DMRs, tofile=FALSE)
```

After the model fitting, by intersecting regions with extreme `meth.diff` and regions with low `p.val`, we can locate those detected DMRs, returned with their average `meth.diff` and `p.val`. Comparing to the regions detected in the *BiSeq* vignette, the two sets of regions are largely similar except for two regions: (chr1:872335,872386), which in our case the `meth.diff` has not been considered high enough due to the highly asymmetric distribution of `meth.diff`; another region (chr2:46915,46937) resides in the tail of chromosome 2 with low density of methylation sites, which has been sorted into the intermediate state due to the lack of support from both the emission level and the sojourn time. However, it is worth mentioning that due to the filtering applied in their workflow, they built wider regions out of a smaller set of more significant sites; whereas in our case, the regions are more refined and especially we identified two hypomethylated regions (chr1:876807,876958 and chr1 :877684,877738). The two segmented profiles are visualized in Figure 2.13 using the default `plot` method.

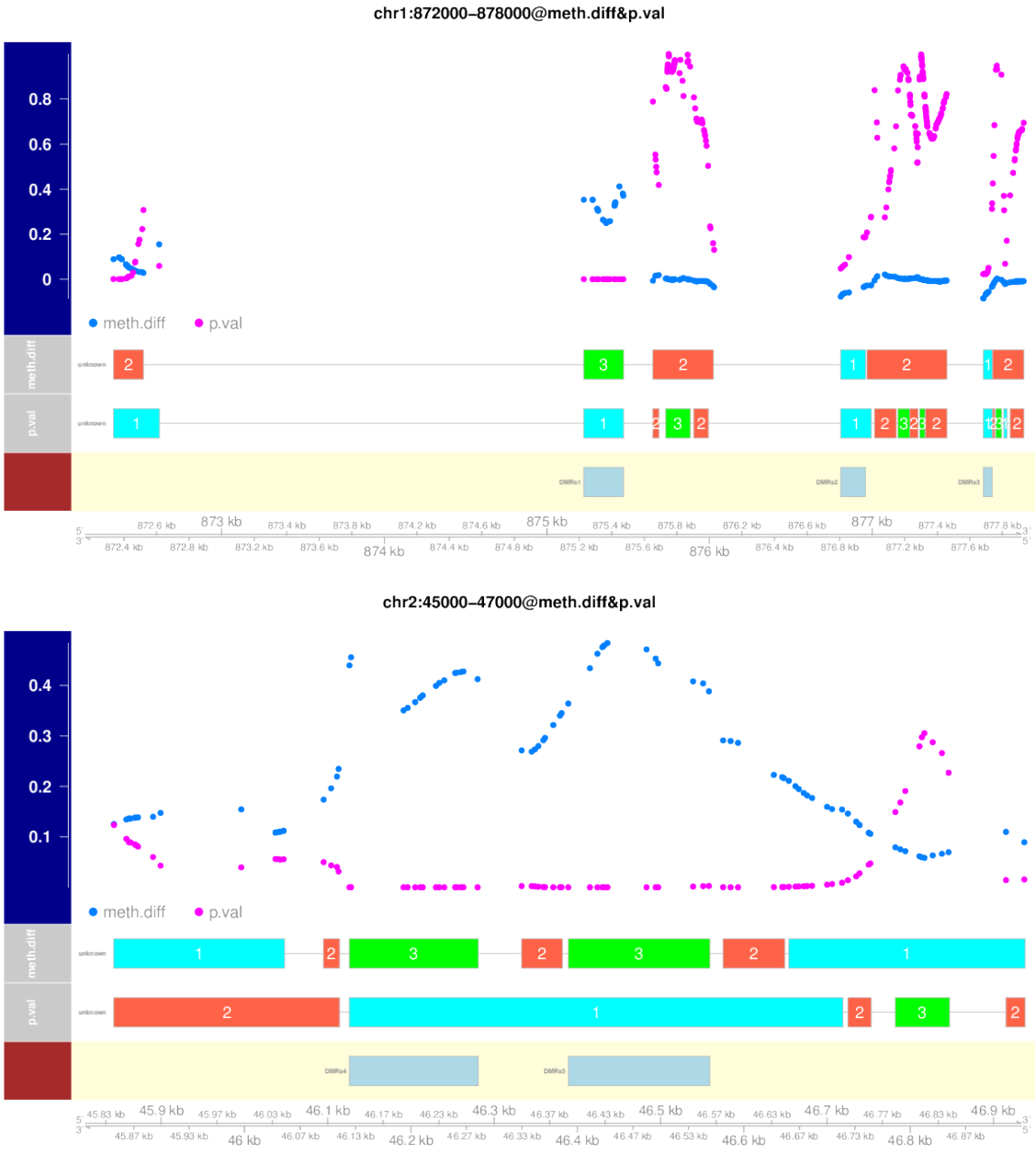


Figure 2.13.: Detected differentially methylated regions (DMRs) in the example data, together with estimated segmentation profiles. DMRs could be located by intersecting resulting states '1' or '3' in `meth.diff` and segments '1' in `p.val`, indicated by boxes in the third row.

CHAPTER 3

Case studies

In this chapter, we will employ the previously described methods in our animal experiments, to study water holding capacity (WHC), an economically important meat quality trait, which also shares many similarities with pathological processes associated with muscle injury. The objective is to use high-throughput technologies to survey regions with known association with the trait, and to detect candidate genes or novel transcriptional units which show differential regulation status between phenotypic groups. Further their potential functional involvement in the related biological process will be discussed.

3.1. Characterizing traits related regions using custom tiling array

3.1.1. Animals and materials

Genomic DNA and phenotypic records were obtained from animals of an experimental F2 population based on a reciprocal cross of Duroc $\times$ Pietrain (DuPi, $n = 417$) as well as the commercial cross-breed and performance tested animals Pietrain $\times$ [German Large White $\times$ German Landrace] (PiF1, $n = 481$). The commercial cross-breed populations (PiF1) represent the typical end product in the German market. They were from different breeding organizations and did not exhibit any genetic link for many generations.

The pigs were slaughtered at a commercial abattoirs and carcass and meat quality data were collected according to guidelines of the German performance test. Meat quality traits analyzed in this study cover indicators of WHC including meat color at 24 h

p.m. (OPTO), drip loss (DRIP), thawing loss (THAW), cooking loss (COOK), pH at 45 min p.m. (pH1), pH at 24 h p.m. (pH24), conductivity at 45 min p.m. (CON1) and conductivity at 24 p.m. (CON24). Meat for conductivity, color, and pH at 24 h p.m. was stored at 4°C in the slaughterhouse. Conductivity and pH-value were measured by using Star-series equipment (Rudolf Matthaeus Company, Germany) in M. longissimus dorsi between 13th/14th ribs. Drip loss was scored based on a bag-method with a size standardized sample from the M. longissimus dorsi collected at 24 h *post mortem* that was weighed, suspended in a plastic bag, held at 4°C for 48 h, and thereafter re-weighed [117; 118]. To determine cooking loss, a loin cube was taken from the M. longissimus dorsi, weighed, placed in a polyethylene bag, and incubated in water at 75°C for 50 min and the solid portion was re-weighed. Thawing loss was determined similarly after at least 24 h freezing at -20°C. Drip loss, cooking loss, and thawing loss were calculated as a percentage of weight loss based on the start weight of a sample. The numbers of records, mean values, and standard deviations are shown in Table 3.1.

For the tiling array experiment, 11 animals were selected for the DuPi population and 12 were chosen in the PiF1 population (Table 3.2) which show extreme drip loss differences. Samples are labeled as 'HI' if showing large value of drip loss, which in turn indicates low water holding capacity.

Table 3.1.: Population summaries and phenotype data measured with means and standard deviations

	DuPi	PiF1
Number of animals	417	481
Number of sires	5	10
Number of litters	44	232
meat color at 24 h p.m. (OPTO)	68.57 ± 5.69	70.39 ± 8.83
pH at 45 min p.m. (pH1)	6.56 ± 0.21	6.24 ± 0.26
pH at 24 h p.m. (pH24)	5.51 ± 0.10	5.57 ± 0.11
conductivity at 45 min p.m. (CON1)	4.36 ± 0.62	2.91 ± 0.60
conductivity at 24 p.m. (CON24)	2.82 ± 0.85	3.45 ± 0.95
drip loss (DRIP)	2.10 ± 0.96	1.94 ± 0.79
thawing loss (THAW)	8.09 ± 1.98	9.08 ± 3.97
cooking loss (COOK)	24.97 ± 2.13	25.39 ± 2.07

Table 3.2.: Experimental panel of tiling array samples

DuPi				PiF1			
SampleID	Drip-Loss group	Batch	Sex	SampleID	Drip-Loss group	Batch	Sex
14	HI	1	Male	36	HI	9	Male
15	LO	1	Male	199	HI	9	Female
6	LO	2	Female	82	HI	10	Female
9	LO	2	Male	83	LO	10	Female
10	HI	2	Female	204	HI	11	Female
11	HI	2	Male	424	LO	11	Male
17	HI	3	Female	434	HI	11	Male
18	LO	3	Female	205	LO	12	Female
20	HI	3	Male	559	HI	12	Male
28	HI	3	Female	579	LO	12	Male
19	LO	4	Male	36	HI	13	Male
14	HI	5	Male	234	LO	13	Female
15	LO	5	Male	424	LO	13	Male
20	HI	5	Male	83	LO	14	Female
28	HI	5	Female	204	HI	14	Female
6	LO	6	Female	205	LO	15	Female
9	LO	6	Male	82	HI	15	Female
17	HI	6	Female	234	LO	15	Female
10	HI	7	Female	204	HI	16	Female
11	HI	7	Male	434	HI	16	Male
19	LO	7	Male	261	LO	16	Male
14	HI	8	Male	83	LO	17	Female
15	LO	8	Male	261	LO	17	Male
				559	HI	17	Male
				579	LO	18	Male
				82	HI	18	Female
				424	LO	18	Male

Candidate regions

Earlier, QTLs for drip loss were identified on SSC5 and SSC18 in the DuPi population [119; 120]. In order to further characterize the nature of the QTLs for drip loss identified on SSC5 and SSC18 obtained in the DuPi population, region-specific bacterial artificial chromosome (BAC) arrays were constructed for expression profiling in these QTL regions [121]. To map these QTL regions, ends sequence of BAC features were retrieved from National Center for Biotechnology Information (NCBI) and aligned to *Sus scrofa* 10 (GCF_000003025.4, NCBI Build 3.1) using NCBI BLAST (v2.2.25) [43], top alignments with at least 80% identity were kept. Overall covered regions were then derived using all mapped BAC clones.

Beside the QTL regions, we also managed to include several genomic regions containing SNPs associated with meat quality traits including drip loss and expression traits highly correlated with WHC [122; 123]. Using all these SNPs, haplotype blocks were constructed with HapView v4.2 [124], using Solid Spine of LD approach with default parameters and filtering. By cross referencing common significant SNPs found in the GWAS and eQTL results, 35 block regions were further located.

Genomic coordinates of all candidate regions are shown in Table 3.3, together with an extra gene region of interest. The regions covers 18 Mb of genomic sequences comprised of 254 annotated transcripts representing 234 genes (according to Ensembl *Sus Scrofa* 10.2 release 71).

3.1.2. Tiling array design and processing

Initially genomic sequences for candidate regions were obtained based on *Sus scrofa* 10 (GCF_000003025.4, NCBI Build 3.1). Tiling probes were selected using *OTAD* [9] for both strands of the DNA, with the following parameters: predicted nearest-neighbour melting temperature between 62 to 82 degree; probe length between 60 bp and 45 bp; maximal overlapping size of neighboring probes as 20 bp; minimal penalized uniqueness score of 9; GC content between 15% and 60%; single nucleotide repeats not exceeding 7 and di-nucleotide repeats not exceeding 4; no base exceeds 60% in probe composition; maximal palindrome content of 40%. In total, 957208 qualified probes provide a coverage of 49% of our targeted regions.

Tiling array chips were manufactured using Agilent's SurePrint G3 Custom Gene Expression Microarray 1x1M (Agilent Technology, USA). Hybridization was made in house using Tecan HS400 Pro (Tecan Group Ltd., Switzerland) according to Agilent's

Table 3.3.: Candidate genomic regions to be tiled on the array

Ssc	from	to	ContigLength	Ssc	from	to	ContigLength
1	116599604	116643478	43875	8	78660962	78760549	99588
1	154956889	154979315	22427	9	66651045	67130079	479035
1	12629603	12689880	60278	9	67784010	68134921	350912
1	121107995	121595489	487495	12	74980	564152	489173
1	289904792	290058951	154160	12	25024663	25105512	80850
2	9736615	9960856	224242	12	25709780	25822853	113074
2	72155901	72405040	249140	13	24413648	24766488	352841
2	134222204	134350467	128264	13	33925637	34300217	374581
3	87379656	87816579	436924	13	34320248	34507047	186800
3	31866257	32220932	354676	13	66102694	66383309	280616
4	16438095	16566860	128766	14	76010942	76391082	380141
4	104379984	104675931	295948	14	51441662	51917769	476108
5	14154240	14225798	71559	14	113098480	113567260	468781
5	80424455	80428262	3808	14	152715621	153193779	478159
6	124842118	124895462	53345	15	128766900	129185062	418163
6	47852177	48278540	426364	17	13354121	13629686	275566
7	2967727	3024065	56339	5	3522373	7742411	4223998
7	94648080	95004183	356104	18	53586438	58018224	4534183
7	130657961	130805381	147421	2	151050000[a]	151200000[a]	150001

[a] Chromosomal coordinates relative to Ensembl *Sus Scrofa* 10.2 release 71 genomic sequence assembly

protocol, with 2-3 technical replicates for each biological sample as have been shown in Table 3.2. Processed arrays were scanned using Tecan PowerScanner (Tecan Group Ltd., Switzerland). Array signal intensities and local backgrounds were then summarized using Array-Pro Analyzer 6.3 (Meida Cybernetics Inc., USA), with potentially unreliable spots marked for removal.

3.1.3. Tiling array data analysis

Pre-processing of tiling array data

The analysis was conducted initially on the probe level, using R/Bioconductor package *limma* (version 3.16.8) [125]. Prior to processing, probes marked as unreliable were removed separately within each population. Given the nature of tiling array, with which large number of probes should show no differential expression, background correction and inter-array normalization were done with R/Bioconductor package *vsn* (version 3.28.0) [126], using spike-in controls and with local background subtracted. Due to the relatively long time span of array processing and the two animal sources, the two distinct populations were separately analyzed and batch effects were removed using the `ComBat` function in R/Bioconductor package *sva* (version 3.6.0) [127]. QC were done separately for each population with R/Bioconductor package *arrayQualityMetrics* (version 3.16.0) [128] after preprocessing. Detailed QC reports can be found in Appendix (E). Processed signals were then filtered to exclude lowly expressed probes. We first got the 95% percentile of the negative control probes on each array, and latter probes were kept for those with at least 10% higher expression than the negative controls on at least two arrays. Probes were latter remapped to Ensembl *Sus Scrofa* 10.2 release 71 with BLAT (v34) [49] using only exact matches for further analysis.

Correlation with previous Affymetrix GeneChip

Previously, gene expression profiling of DuPi (GSE11193 [129] and GSE10204 [122]) and PiF1 (GSE32112 [130; 131]) animals have been conducted using Affymetrix Porcine gene expression chip (GEO platform ID: GPL3533). We thus attempted to compare the transcript specific tiling array results with previous Affymetrix expression profiles. To accomplish this, first target and consensus sequences of Affymetrix array probeset were obtained via Affymetrix chip annotation. The Affymetrix chip contains in total 24123 probesets, among which 23935 are expression profiling targets together with 124 controls sets and 64 mapping reporters, while the tilling array constructed covers 113 genes

that are not represented on the Affymetrix GeneChips.

For consensus sequences, first, genomic coordinates of probesets with unique Ensembl gene ID assignment were located using the recent annotation (Ensembl 10.2.71); for those without ID assignment, consensus sequences of their transcripts were used to align to the genome assembly (10.2.71) with BLAT. Top alignment with highest match size was kept as the target range. After that, a second BLAST run was carried out to fill those unmapped ones after the first BLAT alignment. Sequences without acceptable mapping so far were ignored (16 not mapped). Similarly for Affymetrix target sequence, which is only a subset close to the 3' of consensus sequence, it is more directly comparable with our tiling probes. All target sequences were aligned to the genome assembly (10.2.71) using BLAT, unmapped ones were then similarly processed (31 not mapped). Correlations between tiling array probes and Affymetrix probesets were assessed using those tiling probes which overlap with aligned Affymetrix feature. For each mapped Affymetrix feature, normalized signals of overlapping tiling array probes across samples from respective population were pooled using median. Similarly, normalized signals of Affymetrix probesets were also averaged across samples using median. In Table 3.4, Pearson's correlation test has been carried out and summarized for the three previous Affymetrix experiments, using both consensus and target sequence matches with experiment specific populations. Clearly, our tiling array probes show good agreement with those Affymetrix probesets aligned to our tiling regions. The correlations are also consistently higher when matching with mapped target sequence, as have been expected. In Figure 3.1, correlations of common features between the two platforms have been illustrated using target sequence matches.

Table 3.4.: Correlation between Affymetrix GeneChip and customary tiling array

GEO	Population	Correlation[a]	P-value[a]	Correlation[b]	P-value[b]
GSE11193	DuPi	0.5570741	< 2.2e-16	0.4274864	2.22e-15
GSE10204	DuPi	0.5552044	< 2.2e-16	0.4345732	6.661e-16
GSE32112	PiF1	0.6161726	< 2.2e-16	0.4893904	< 2.2e-16

[a] Matching aligned Affymetrix probesets with tiling probes using target sequence

[b] Matching aligned Affymetrix probesets with tiling probes using consensus sequence

Differential expression analysis

Differential expression of probes between 'HI' and 'LO' drip loss samples were assessed using the moderated t-test implemented in the `lmFit` function of *limma*. Instead of controlling the false discovery rate (FDR) with p-value adjustment, we approached this multiple testing problem differently. Thanks to the overlapping tiling array design,

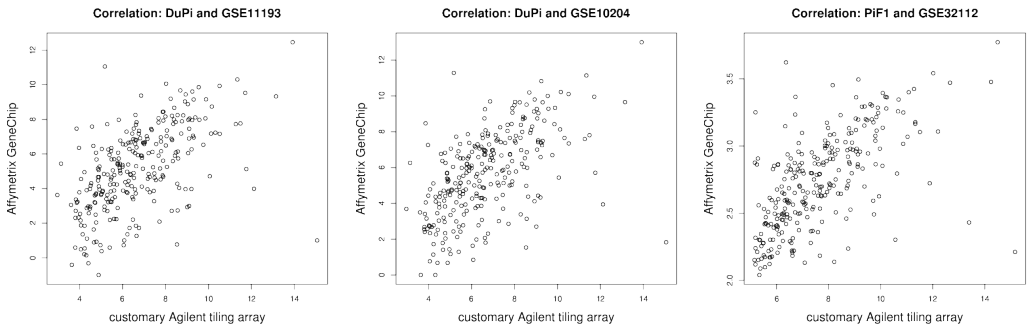


Figure 3.1.: Correlation between Affymetrix GeneChip and customary tiling array, using common feature matched by Affymetrix probeset target sequence.

neighboring probes are overlapping with an average size of 20 bp, thus consecutive probes could ideally form a continuous tiling path with high density. In our pipe line, we first applied a threshold of 0.05 on the nominal p-value computed by *limma*. Using this initial set of significant probes we then attempted to form continuous regions if neighboring probes (overlapping or with a gap smaller than 25 bp) show same direction of fold change. We solved this by applying a modified max-gap-min-run algorithm with a maximum gap of 25 bp and minimum probe number of 3. We also borrowed information from a concurrent mRNA-seq experiment run on the same PiF1 animals to filter regions to which there are less than 12 reads in total mapped across all samples. We will refer to these regions as significantly differentially expressed probes (DEPs) regions. Schematic illustration of DEPs region definition and rules of commonality call are shown in Figure 3.3. Incorporation of linear or quadratic terms of Guanine-Cytosine (GC) content in the linear models has also been considered and tested, with derived DEPs regions largely unchanged. There is also hardly any changes in the resulting DEPs regions when including gender of the animal as additional experimental factor. Thus for simplicity and better model interpretation, we excluded GC content and animal gender from the final model.

For common DEPs shared by the two populations, we required the overlapping pair to have consistent fold change direction. Common genes with exons overlapping with significant DEPs in both population were selected as candidate differentially expressed genes (DEGs). For genes with multiple exons covered by different DEPs regions, the fold change direction has to be consistent across all DEPs regions. Special consideration was given to those genes with exons overlapped with DEPs regions in both populations though the individual DEPs regions are not overlapping. Details of the commonality calling are illustrated in Figure 3.3. Resulting Venn diagrams for both Ensembl gene ID and non-overlapping DEPs region are shown in Figure 3.2.

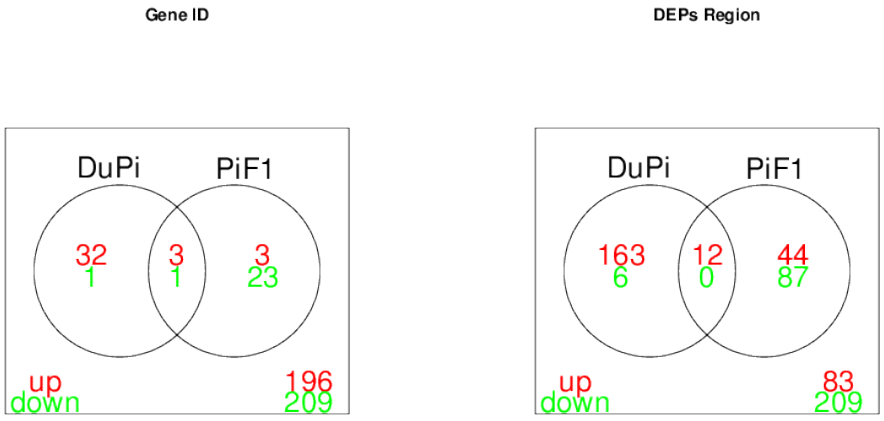


Figure 3.2.: On the left-hand-side is the venn diagram of detected differentially expressed Ensembl gene IDs, and on the right-hand-side is the venn diagram for detected differentially expressed probe regions (DEPs) of the two populations

Gene set and pathway enrichment analysis

We then attempted to conduct a Gene Ontology (GO) enrichment analysis. The regions we tiled on the array cover 234 annotated genes (according to Ensembl *Sus Scrofa* 10.2 release 71), which were used as background. Unlike in the previous Venn diagram, common genes with exons overlapping with significant DEPs region in both populations were selected as candidate DEGs (14 in total), with relaxed fold change direction constraint between populations, yet the same fold change direction rule still holds within individual population. Ensembl gene ID was then mapped to Entrez gene ID using *bioMart* [132]. Common genes overlapping with DEPs regions are listed in Table 3.5 together with their functional annotations.

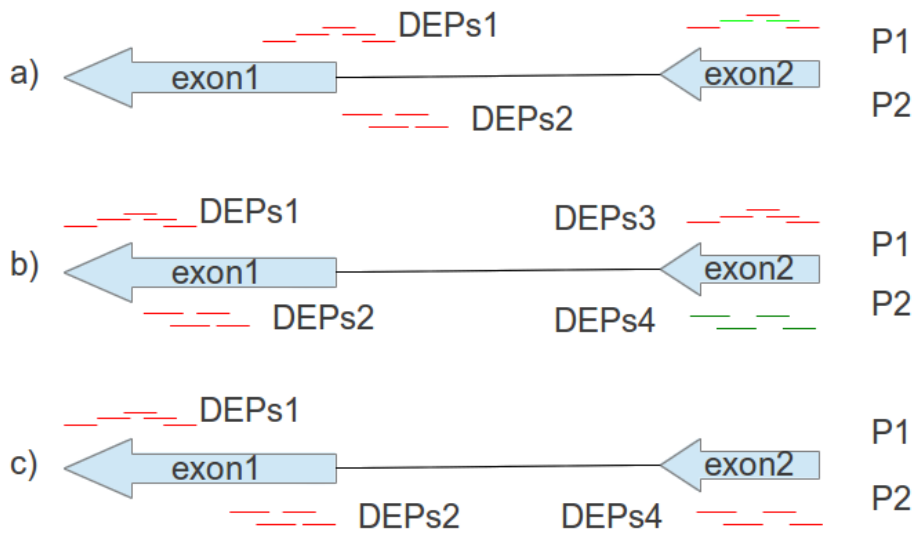


Figure 3.3.: To illustrate the definition of DEPs regions and how common DEPs regions and DEPs genes are counted, we hypothesized a gene with two exons and two experimental population P1 and P2. In panel a), two DEPs regions are formed by overlapping up-regulated probes, one for each population. In P1, the set of probes over exon2 show different fold change directions thus ignored, and the gene is considered to be up-regulated in P1. However in P2, since DEPs2 is not overlapping with exon1, we don't count the gene as regulated. Across the two population, DEPs1 and DEPs2 are then considered as common DEPs since they are overlapping and with consistent fold change direction. In panel b), DEPs1 and DEPs3 are found in P1, covering different exons of the same gene yet with consistent fold change direction, so we count the gene as up-regulated in P1. While for P2, DEPs2 and DEPs4 are identified with different fold change directions, so the gene is not counted but the DEPs calls are still valid for these two. Across the populations, DEPs1 and DEPs2 are still considered common, while for DEPs3 and DEPs4 commonality call is not made. In panel c), DEPs are detected for different exons and exon parts in the two populations but with consistent fold change direction, so we call the gene as common and up-regulated. For common DEPs call, DEPs1 and DEPs4 are not overlapping thus not considered common. However for DEPs1 and DPEs2, the two are not directly overlapping but sharing the same exon, we then consider them as a special case and call exon1 as a common DEPs region.

Table 3.5.: Common genes overlapped with DEPs regions

EnsemblGeneID	Symbol	EntrezGeneID	Description	Location	Type
ENSSSCG00000022192	NCAPD3	NA	non-SMC condensin II complex, subunit D3	Nucleus	other
ENSSSCG00000004065	TIAM2	100155014	T-cell lymphoma invasion and metastasis 2	Cytoplasm	enzyme
ENSSSCG00000006663	OTUD7B	100152569	OTU domain containing 7B	Cytoplasm	peptidase
ENSSSCG00000026968	SLC20A2	NA	solute carrier family 20 (phosphate transporter), member 2	Plasma Membrane	transporter
ENSSSCG00000000084	ATF4	100144302	activating transcription factor 4	Nucleus	transcription regulator
ENSSSCG00000006667	BOLA1	100152975	bolA homolog 1 (E. coli)	Cytoplasm	other
ENSSSCG00000000104	DDX17	100514347	DEAD (Asp-Glu-Ala-Asp) box helicase 17	Nucleus	enzyme
ENSSSCG00000000068	EP300	100156226	E1A binding protein p300	Nucleus	transcription regulator
ENSSSCG00000010470	IDE	100155309	insulin-degrading enzyme	Extracellular Space	peptidase
ENSSSCG00000010469	MARCH5	100157340	membrane-associated ring finger (C3HC4) 5	Cytoplasm	enzyme
ENSSSCG00000024983	RSL24D1	100623328	ribosomal L24 domain containing 1	Nucleus	other
ENSSSCG00000029270	SMIM19	100152445	small integral membrane protein 19	Nucleus	other
ENSSSCG00000021586	ZHX2	100152952	zinc fingers and homeoboxes 2	Nucleus	transcription regulator
ENSSSCG00000014401	NR3C1	396740	nuclear receptor subfamily 3, group C, member 1 (glucocorticoid receptor)	Nucleus	ligand-dependent nuclear receptor

Over representation of GO terms of category Biological Process (BP), Cellular Component (CC) and Molecular Function (MF) were tested with R/Bioconductor package *GOstats* (version 2.26.0)[133] for the common DEPs genes. Significantly enriched GO terms were selected with Hypergeomtric test p-values smaller than 0.05. In Table 3.6 and Table 3.7 resulting lists of over-representation of GO term category MF and BP are presented.

We also submitted the common gene IDs (irrespective of their fold change directions in the two populations) to Ingenuity pathway analysis (IPA) (Ingenuity Systems[1]). Significantly enriched canonical pathways with -log(p-value) greater than 2 are shown in Table 3.8 together with their functional annotation.

It could be seen clearly from the GO enrichment and the pathway analysis that, ATF4 and EP300, as two major contributors, are associated with several diseases and cancer related signaling pathways, and many regulatory processes. ATF4 encodes a transcription factor, which is a widely expressed DNA binding protein. Recent study has shown that forced expression of ATF4 together with other regulator could caused ATP depletion, oxidative stress and cell death [134]. Like ATF4, EP300 also encodes a transcriptional co-activator protein and regulates transcription via chromatin remodeling. It has been shown that the EP300 gene is a key player in the processes of cell proliferation and differentiation [135; 136]. Interestingly, one probeset on the previously employed Affymetrix GeneChip representing EP300 has also been found to show high expression level in low drip loss samples [129]. From the Ingenuity pathway analysis, pathway "NRF2-Mediated Oxidative Stress Response" appears to be interesting, which is related to the imbalance of oxygen supply in *post mortem* muscle cells and has direct connection to apoptosis and necrosis. Thus it could be linked to the drip loss during the conversion from muscle to meat.

[1] http://www.ingenuity.com/products/ipa

Table 3.6.: Common DEPs genes to GO MF test for over-representation

GOMFID	Pvalue	ExpNo.	No.	GeneSymbol	Term
GO:0008270	0.0002	1	6	EP300,OTUD7B,MARCH5,IDE,NR3C1,ZHX2	zinc ion binding
GO:0046914	0.0004	1	6	EP300,OTUD7B,MARCH5,IDE,NR3C1,ZHX2	transition metal ion binding
GO:0003677	0.0028	1	5	EP300,ATF4,OTUD7B,NR3C1,ZHX2	DNA binding
GO:0043565	0.0030	0	3	ATF4,NR3C1,ZHX2	sequence-specific DNA binding
GO:0035257	0.0034	0	2	EP300,DDX17	nuclear hormone receptor binding
GO:0035258	0.0034	0	2	EP300,DDX17	steroid hormone receptor binding
GO:0051427	0.0034	0	2	EP300,DDX17	hormone receptor binding
GO:0003676	0.0070	2	6	EP300,ATF4,DDX17,OTUD7B,NR3C1,ZHX2	nucleic acid binding
GO:0003712	0.0079	0	3	EP300,DDX17,ZHX2	transcription cofactor activity
GO:0046872	0.0095	2	6	EP300,OTUD7B,MARCH5,IDE,NR3C1,ZHX2	metal ion binding
GO:0000988	0.0114	1	3	EP300,DDX17,ZHX2	protein binding transcription factor activity
GO:0000989	0.0114	1	3	EP300,DDX17,ZHX2	transcription factor binding transcription factor activity
GO:0046983	0.0114	1	3	ATF4,IDE,ZHX2	protein dimerization activity
GO:0001071	0.0158	1	3	ATF4,NR3C1,ZHX2	nucleic acid binding transcription factor activity
GO:0003700	0.0158	1	3	ATF4,NR3C1,ZHX2	sequence-specific DNA binding transcription factor activity
GO:0043169	0.0165	3	6	EP300,OTUD7B,MARCH5,IDE,NR3C1,ZHX2	cation binding
GO:0097159	0.0172	3	7	EP300,ATF4,DDX17,OTUD7B,IDE,NR3C1,ZHX2	organic cyclic compound binding
GO:1901363	0.0172	3	7	EP300,ATF4,DDX17,OTUD7B,IDE,NR3C1,ZHX2	heterocyclic compound binding
GO:0003713	0.0189	0	2	EP300,DDX17	transcription coactivator activity
GO:0043167	0.0192	3	7	EP300,DDX17,OTUD7B,MARCH5,IDE,NR3C1,ZHX2	ion binding
GO:0004871	0.0210	1	3	TIAM2,IDE,NR3C1	signal transducer activity
GO:0060089	0.0210	1	3	TIAM2,IDE,NR3C1	molecular transducer activity

Table 3.7.: Common DEPs genes to GO BP test for over-representation

GOBPID	Pvalue	ExpNo.	No.	GeneSymbol	Term
GO:0030518	0.0040	0	2	EP300,NR3C1	intracellular steroid hormone receptor signaling pathway
GO:0006366	0.0042	1	5	EP300,ATF4,DDX17,NR3C1,ZHX2	transcription from RNA polymerase II promoter
GO:0048519	0.0068	2	6	EP300,ATF4,OTUD7B,MARCH5,IDE,ZHX2	negative regulation of biological process
GO:0030522	0.0116	0	2	EP300,NR3C1	intracellular receptor mediated signaling pathway
GO:0048545	0.0116	0	2	EP300,NR3C1	response to steroid hormone stimulus
GO:0006357	0.0125	1	4	EP300,ATF4,DDX17,ZHX2	regulation of transcription from RNA polymerase II promoter
GO:0045944	0.0147	1	3	EP300,ATF4,DDX17	positive regulation of transcription from RNA polymerase II promoter
GO:0006355	0.0147	2	5	EP300,ATF4,DDX17,NR3C1,ZHX2	regulation of transcription, DNA-dependent
GO:2001141	0.0147	2	5	EP300,ATF4,DDX17,NR3C1,ZHX2	regulation of RNA biosynthetic process
GO:0010628	0.0202	1	3	EP300,ATF4,DDX17	positive regulation of gene expression
GO:0045893	0.0202	1	3	EP300,ATF4,DDX17	positive regulation of transcription, DNA-dependent
GO:0014070	0.0225	0	2	EP300,NR3C1	response to organic cyclic compound
GO:0006351	0.0240	2	5	EP300,ATF4,DDX17,NR3C1,ZHX2	transcription, DNA-dependent
GO:0010468	0.0240	2	5	EP300,ATF4,DDX17,NR3C1,ZHX2	regulation of gene expression
GO:0009891	0.0268	1	3	EP300,ATF4,DDX17	positive regulation of biosynthetic process
GO:0010557	0.0268	1	3	EP300,ATF4,DDX17	positive regulation of macromolecule biosynthetic process
GO:0031328	0.0268	1	3	EP300,ATF4,DDX17	positive regulation of cellular biosynthetic process
GO:0045935	0.0268	1	3	EP300,ATF4,DDX17	positive regulation of nucleobase-containing compound metabolic process
GO:0051173	0.0268	1	3	EP300,ATF4,DDX17	positive regulation of nitrogen compound metabolic process
GO:0051254	0.0268	1	3	EP300,ATF4,DDX17	positive regulation of RNA metabolic process
GO:0033993	0.0361	0	2	EP300,NR3C1	response to lipid

Table 3.8.: Ingenuity Canonical Pathways of common DEPs genes

Ingenuity Canonical Pathways	-log(p-value)	Ratio	Molecules
Circadian Rhythm Signaling	3.68E00	5.26E-02	ATF4,EP300
Role of IL-17F in Allergic Inflammatory Airway Diseases	3.43E00	4.17E-02	ATF4,EP300
ATM Signaling	3.18E00	3.23E-02	ATF4,EP300
Estrogen-Dependent Breast Cancer Signaling	3.14E00	2.74E-02	ATF4,EP300
ERK5 Signaling	3.13E00	3.03E-02	ATF4,EP300
Hypoxia Signaling in the Cardiovascular System	3.1E00	2.99E-02	ATF4,EP300
Neurotrophin/TRK Signaling	3.08E00	2.63E-02	ATF4,EP300
Prolactin Signaling	3.01E00	2.5E-02	NR3C1,EP300
FLT3 Signaling in Hematopoietic Progenitor Cells	2.99E00	2.6E-02	ATF4,EP300
Prostate Cancer Signaling	2.9E00	2.02E-02	ATF4,EP300
Melanocyte Development and Pigmentation Signaling	2.87E00	2.15E-02	ATF4,EP300
FGF Signaling	2.87E00	2.17E-02	ATF4,EP300
NGF Signaling	2.69E00	1.68E-02	ATF4,EP300
Gαs Signaling	2.65E00	1.63E-02	ATF4,EP300
Corticotropin Releasing Hormone Signaling	2.64E00	1.45E-02	ATF4,EP300
p38 MAPK Signaling	2.6E00	1.69E-02	ATF4,EP300
Synaptic Long Term Potentiation	2.59E00	1.54E-02	ATF4,EP300
P2Y Purigenic Receptor Signaling Pathway	2.57E00	1.42E-02	ATF4,EP300
Estrogen Receptor Signaling	2.54E00	1.47E-02	NR3C1,EP300
GNRH Signaling	2.53E00	1.32E-02	ATF4,EP300
B Cell Receptor Signaling	2.35E00	1.17E-02	ATF4,EP300
Dopamine-DARPP32 Feedback in cAMP Signaling	2.33E00	1.08E-02	ATF4,EP300
CREB Signaling in Neurons	2.27E00	9.71E-03	ATF4,EP300
Ephrin Receptor Signaling	2.27E00	9.85E-03	ATF4,EP300
Dendritic Cell Maturation	2.25E00	9.57E-03	ATF4,EP300
NRF2-mediated Oxidative Stress Response	2.24E00	1.04E-02	ATF4,EP300
Calcium Signaling	2.24E00	9.39E-03	ATF4,EP300
ILK Signaling	2.21E00	1.03E-02	ATF4,EP300
ERK/MAPK Signaling	2.21E00	9.62E-03	ATF4,EP300
cAMP-mediated signaling	2.08E00	8.89E-03	ATF4,EP300
Huntington's Disease Signaling	2.06E00	8.23E-03	ATF4,EP300
Phospholipase C Signaling	2.02E00	7.6E-03	ATF4,EP300

Data deposition

Raw and processed expression data for the present study has been submitted to the NCBI Gene Expression Omnibus (GEO)[2] with the accession number GSE52384, with population specific subseries GSE50846 (DuPi) and GSE52383 (PiF1).

[2]http://www.ncbi.nlm.nih.gov/geo

3.2. Characterizing traits related regions using mRNA-seq

3.2.1. mRNA-seq preparation and preprocessing

We have also recently conducted mRNA-seq experiments on the same discordant PiF1 sibs selected for the array experiments. Two paired-end sequencing runs were done in-house using Illumina GAIIx and following standard Illumina unstranded TruSeq protocol. Resulting FASTQ files were aligned to the reference genome assembly of Ensembl *Sus Scrofa* 10.2 release 71 using *TopHat*[137] (v2.0.3) and *Bowtie* (v0.12.7.0) [138] . The sequencing data provides approximately 10X coverage of those regions previously targeted by the tiling array experiment. Experimental panel and sequencing library statistics (for only reads mapped to tiled regions) are listed in Table 3.9, in which case we have a slightly higher read depth in the 'HI' samples, summed up to 1029593 comparing to the 955932 total reads in 'LO' samples.

Table 3.9.: Experimental panel of mRNA-seq samples

SampleID	Drip-Loss group	Sex	lib.size[a]	norm.factors[b]	sizeFactors[c]
36	HI	Male	172075	1.0516186	1.1036386
199	HI	Female	168200	0.9375786	0.9252544
82	HI	Female	199407	1.0187227	1.3003812
83	LO	Female	151324	1.0572419	1.0076013
204	HI	Female	173684	0.8775684	0.9297538
205	LO	Female	150139	1.0499616	0.9278158
559	HI	Male	179061	0.9919174	1.1036646
579	LO	Male	154341	1.0684607	0.9954937
234	LO	Female	151342	0.9737174	0.8516431
424	LO	Male	164388	0.9942177	1.0667615
434	HI	Male	137166	1.0855224	0.8493345
261	LO	Male	184398	0.9176182	1.1204193

[a] Total number of reads mapped to the tiling regions for each sample

[b] Normalization factors calculated by *edgeR* (version 3.2.4)

[c] sizeFactors calculated by *DESeq* (version 1.12.1)

3.2.2. Correlation with tiling array

In order to have a way to compare our previous tiling array results with the sequencing data, a pseudo array was created which shares the same positional information as all the

probes we had earlier mapped. Pseudo array signals were then counted as the number of reads mapped to the probe range with a minimum overlap of 45 bp (irrespective of the strand direction of the probe). Reads overlapping with multiple probes were counted multiple times, due to the overlapping nature of the tiling array. As have been expected, owning to the high sparsity of transcriptional activity, a large majority of the pseudo probes have no reads mapped to. Histograms of the pseudo array probe signals in raw counts and log_2RPKM are shown in Figure 3.4. The distribution of the sum of pseudo array probe signals across samples is summarized in Table 3.10

Table 3.10.: Distribution of the sum of pseudo array probe signals across samples

	No. probes	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Raw counts	913399	0	< 2.2e-16	0	5.84	0	33180
log_2RPKM	913399	-0.3576	-0.3576	-0.3576	0.1846	-0.2321	16.6600

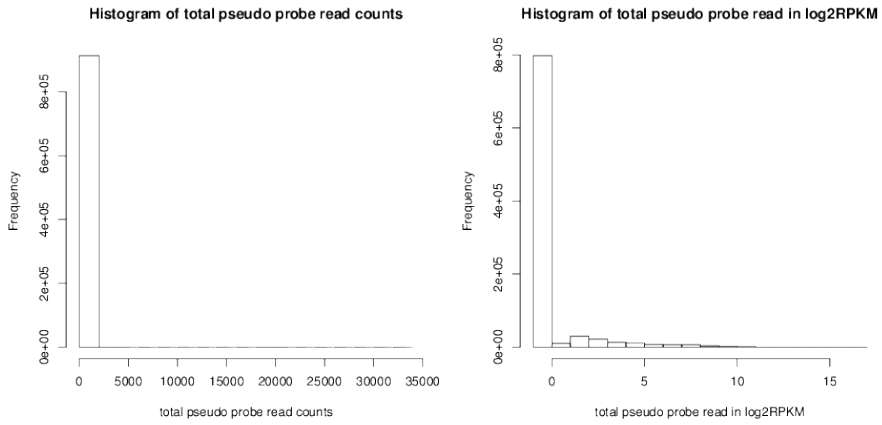


Figure 3.4.: Histogram and density plot of the pseudo array probe signal

We then checked the correlation of the raw array signal with the pseudo array counts, as well as the normalized array signal with log_2RPKM value of the pseudo array. Due to the high sparsity in the read mapping and relatively noisy array data, a set of filtering criteria have been applied in order to get a high confidence set of probes. For both raw and normalized array data, local background was first subtracted after removing unreliable spots. For the next step, similar to the array analysis in the previous section, but only probes overlapped with annotated exon and with signal which is 10% higher than the 95% percentile of the negative controls on the array in all replicates of each sample were kept. Tiling array probe signals were than averaged across replicates using arithmetic mean. On the other hand, for the pseudo array probes, only non-zero pseudo probes were kept. A two-sided Pearson's correlation test was then conducted using the intersection of these two sets of probes for each sample. Summary of the correlation

tests for all paired samples is shown in Table 3.11. It is clear that, in general, the two sets of experiment data are in good agreement with each other. Also one could see that, in Figure 3.5, for both raw expression and normalized data, the array data appears to be more noisy, showing higher variation for those with low sequencing coverage.

Table 3.11.: Correlation of the high confident probes from the tiling array and the pseudo array

	Raw data				Normalized data			
Sample ID	Probe No[a]	Array Cut-off [b]	p-value (Cor.)	Correlation	Probe No[a]	Array Cut-off [b]	p-value (Cor.)	Correlation
36	4785	137.4460	< 2.2e-16	0.6211	4903	6.4282	< 2.2e-16	0.3947
199	4800	115.0113	< 2.2e-16	0.6698	4644	6.4328	< 2.2e-16	0.3795
82	2198	106.3252	< 2.2e-16	0.5899	3968	6.5314	< 2.2e-16	0.3659
83	1815	103.8986	< 2.2e-16	0.5320	3486	6.6656	< 2.2e-16	0.3348
204	2054	97.2403	< 2.2e-16	0.4243	3345	6.5845	< 2.2e-16	0.3578
424	1930	98.9507	< 2.2e-16	0.4400	3145	6.7201	< 2.2e-16	0.3273
434	3178	110.2844	< 2.2e-16	0.5846	4277	6.6236	< 2.2e-16	0.3667
205	3899	112.7920	< 2.2e-16	0.5755	4087	6.5962	< 2.2e-16	0.3314
559	3037	105.3014	< 2.2e-16	0.4981	3553	6.6090	< 2.2e-16	0.3445
579	3469	110.7350	< 2.2e-16	0.5463	3929	6.6044	< 2.2e-16	0.3405
234	4442	127.7264	< 2.2e-16	0.5564	4374	6.4753	< 2.2e-16	0.3478
261	4487	123.6986	< 2.2e-16	0.6434	4816	6.4389	< 2.2e-16	0.3834

[a] Number of remaining high confident probes tested

[b] $1.1 \times NegCtrl_{95\%}$ after local background subtraction

3.2.3. Segmentation of mRNA-seq data

Thanks to the high read abundance we got from the mRNA-seq data, we were able to use the HSMM described earlier, `biomvRhsmm`, to segment sequencing data and detect novel transcripts with high confidence. To start with, using the ranges with probe mapped to and the gene annotation from Ensembl *Sus Scrofa* 10.2 release 71, we first derived a list of 'gap' regions complimentary to all annotated exons, which are combinations of intergenic and intron regions. Strandness of the exons were temporarily ignored to rule out antisense transcription. Alignment BAM files were then read in, and coverage profiles for each base in these 'gap' regions were evaluated and summed over all samples to get the total coverage for each base. For our paired-end reads, only those with proper mating pair were counted. We then run `biomvRhsmm` on the resulted gap coverage profile, assuming a binary states model with emission density following Negative Binomial distribution. Prior of emission density was assumed to be common across all 'gap' regions

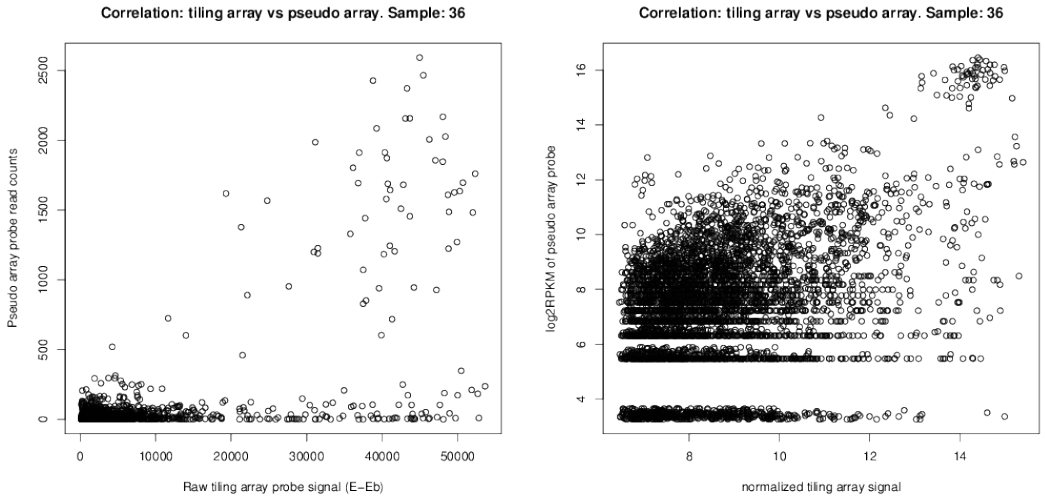


Figure 3.5.: Correlation of the high confident probes from the tiling array and the pseudo array. On the left is the raw data comparison, and normalized data on the right. Each point represents a probe on the tiling array. mRNA-seq expression levels per probe were measured using *RPKM* for normalized array comparison and raw count for raw data comparison. The normalized array data were the same as in the previous differential expression analysis.

and initialized by clustering all 'gap' nucleotides into 2 groups. Sojourn density was left as default using Gamma distribution, and maximum evaluated sojourn length set to 500 bp. Regions labeled with high emission state were further filtered with minimum average read coverage of 12, which means every base within the region has been at least covered once in all sequenced samples. In the end, 441 of such putative 'exon' have been located, among which 324 have also been found to be within the transcripts resulting from '*ab initio*' gene prediction by Ensembl (using algorithms like SNAP [139] and GENSCAN [90]) indicating their structural potentials for protein coding. Interestingly another 253 of these 441 putative 'exons' sit within the intron regions of annotated transcripts from Ensembl (Release 71) indicating novel splicing events, of which 235 'exons' also present in the '*ab initio*' prediction. These putative and transcriptionally active units will be referred to as 'HSMMGxxx' with 'xxx' as their rank numbers hereafter.

We made automatic filtering and manual inspection of the coverage profile in these intronic regions to see if any novel splicing events could be observed. Novel splicing events were sorted into three categories, intron retention, 3' UTR splicing and 5' UTR splicing. We further filtered these putative novel splicing events basing on the following criteria. For predictions made in the intron regions or outside the flanking (1000 bp) UTRs of annotated transcripts, instead of using a hard cut-off value for the coverage, we

utilized the coverage profile for the annotated exons of this transcript. If the predicted feature has a coverage greater than all annotated exons, or if not higher than all those annotated exons in the transcript, the cutoff is then determined by the maximum of a empirically selected coverage threshold 120 and the minimum coverage of all annotated exons of the associated transcript multiplied by a coefficient of 0.7. In this way the filtering will account for both lowly expressed feature and also relatively low proportion of the novel splicing event yet strong enough to be observed in our data. In the end we have concluded a list of novel splicing events from the sequencing coverage profile, where 35 intron retentions and 29 novel splicings in the UTRs were observed. Some of the novel transcripts and splicing events are shown in Figure 3.6.

We then submitted these novel splicing segments for a BLAST search for similar transcripts in the NCBI refseq_rna database. Among the 64 novel splicing transcripts, 41 were found to match to known and predicted transcripts and transcript variants with high confidence. For example, HSMMG149 is found to be similar to a transcript variants of ubiquitin specific peptidase 37 (USP37). To further confirm the novel splicing events and also their regulation status with respect to our experimental setup, sets of qPCR runs have been scheduled and will be performed in the near future. While for those novel transcriptionally active units which show low level of homology to known transcripts or proteins, their possible roles as non-coding RNA in transcription regulation could also be investigated.

3.2.4. Differential expression analysis

Feature quantification

Predicted novel transcripts and annotated genes within mapped probe regions were first exported as GTF files, which were in the next step separately supplied to *htseq-count* [3] to get read counts for annotated genes and predicted novel transcripts using mode *'intersection-nonempty'* and stranded *'no'*.

A multidimensional scaling (MDS) plot using the top 500 features with largest standard deviations across all samples is shown in Figure 3.7 to illustrate group difference. There is no clear separation of 'HI' and 'LO' samples using the first 2 leading dimensions. However, it is indicative that the 'LO' samples show more intra- and inter-group variation. There is also no gender specific expression pattern among these genes.

[3] http://www-huber.embl.de/users/anders/HTSeq/

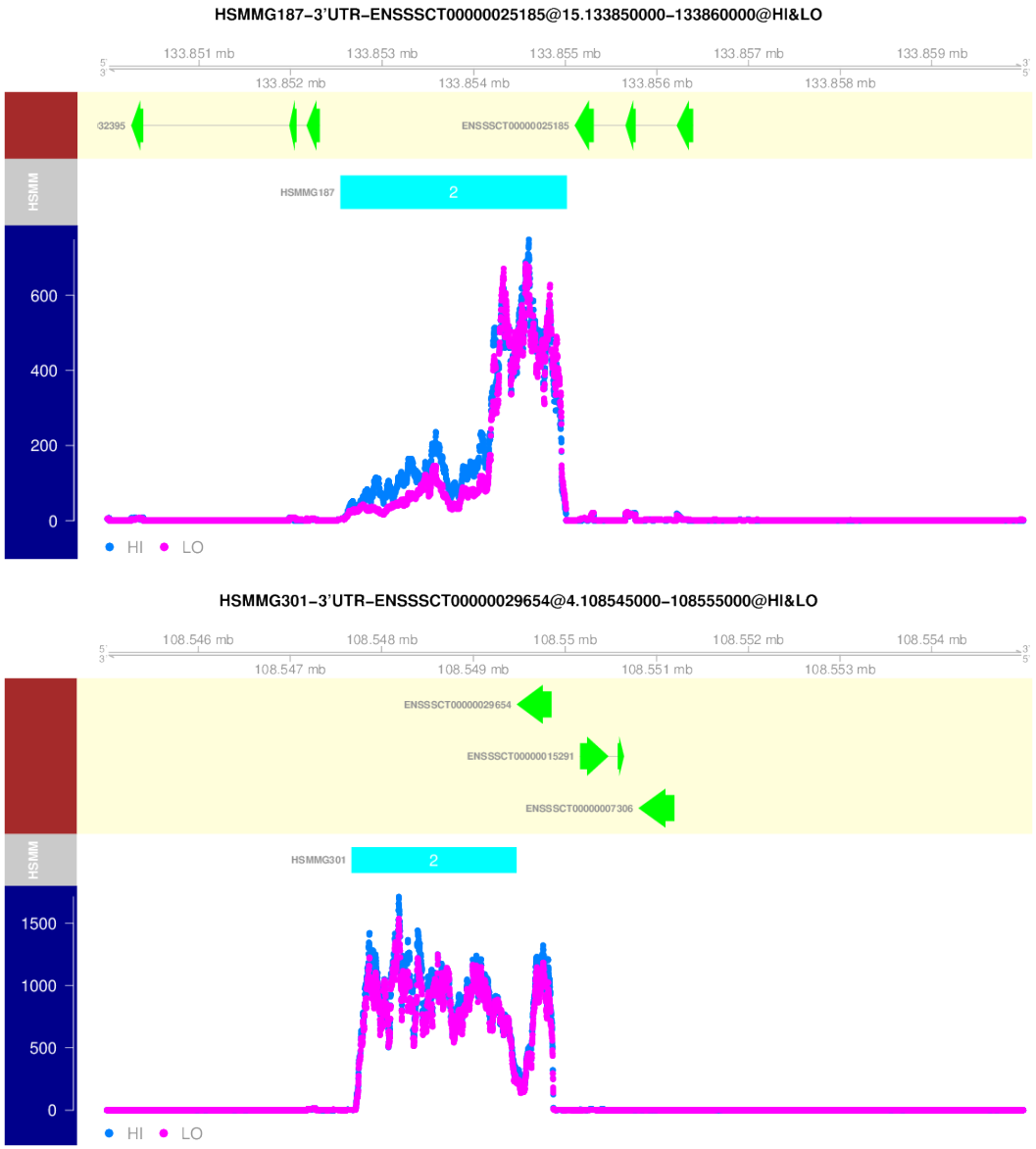


Figure 3.6.: Examples of novel splicing events are illustrated, with brown panel (green feature) representing annotated transcripts, grey panel (cyan feature) representing HSMM prediction and a scatter plot showing phenotype specific per-base converge of the region by mRNA-seq reads.

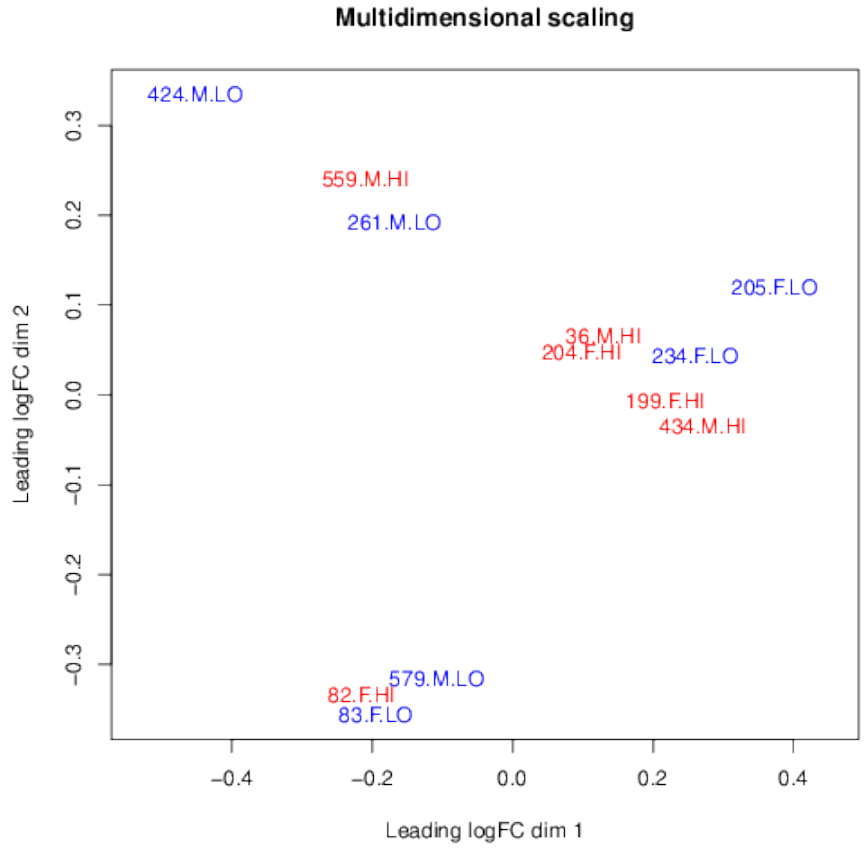


Figure 3.7.: multidimensional scaling plot of mRNA-seq samples. Each point is labeled with 'SampleID.Sex.Type', color coded red if it is 'HI' and blue otherwise.

Testing for differential expression

The count tables were then supplied to R/Bioconductor package *edgeR* (version 3.2.4) [140] and *DESeq* (version 1.12.1) [141] for differential expression analysis. In the *edgeR* pipeline, normalization factors were first computed using method 'TMM', with subsequent common and de-trended tag-wise dispersion estimation. In the end, likelihood ratio tests were conducted with negative binomial generalized log-linear model. When using *DESeq*, samples were also first normalized with the so called 'size factors'. Then dispersion parameters were estimated using 'pooled' method combined with '*sharingMode=maximum*'. Finally differential regulation status was assessed using *nbinomTest*, in which a conceptually similar negative binomial generalized log-linear model is implemented. For both methods, p-values were corrected for multiple testing with Benjamini—Hochberg methods [142]. Also we did not perform any independent filtering due to the relatively low number of annotated features in the mapped regions.

Fast skimming through the two resulting top DEGs lists also suggests low significance for most features. To proceed, an empirical threshold was determined using our previous findings. From the tiling array experiments, we have derived a list of common DEPs genes. Though most showing contradicting fold change direction with the tiling arrary results, four genes show consistent fold change direction in both populations. TIAM2, OTUD7B and SLC20A2 show up-regulation from 'LO' to 'HI' samples, while NCAPD3 shows negative regulation. Three out of the four still have consistent fold change in the mRNA-seq result except for NCAPD3 (ENSSSCG00000022192) which shows a minor up-regulation instead of the down-regulation we seen in the array experiments. For the other three genes (TIAM2, SLC20A2 and OTUD7B), OTUD7B achieves the lowest p-value in both *edgeR* and *DESeq* results, thus was chosen as baseline. From there, genes with nominal p-values lower than the p-value associated with OTUD7B on both *edgeR* and *DESeq* lists were selected as candidate DEGs. Differential expression status of candidate DEGs together with previous common DEPs genes are listed in Table 3.12. It is also worth mentioning that the top 5 ranking genes all show relatively low expression, all of which have not been identified by our previous array experiment.

Table 3.12.: Significant DEGs from mRNA-seq and differential expression status of common DEPs genes

EnsemblGeneID	Symbol	alogFC1[a]	alogFC2[b]	logFC[c]	logCPM[d]	pval[c]	fdr[c]	baseMean[e]	logFC[f]	pval[f]	fdr[f]
ENSSSCG00000030298	H1F0			6.649	5.553	1.42E-005	0.011	5.889	Inf	0.110	1
ENSSSCG00000018173	ssc-mir-425			2.366	4.057	0.023	1	0.743	2.968	0.162	1
ENSSSCG00000004067	CLDN20			1.717	4.404	0.020	1	1.471	2.011	0.141	1
ENSSSCG00000000078	TNRC6B			0.523	8.071	0.044	1	41.746	0.523	0.054	1
ENSSSCG00000011249	DLEC1			1.115	4.971	0.076	1	3.210	1.111	0.141	1
ENSSSCG00000000086	MGAT3			-1.132	4.871	0.044	1	2.787	-1.204	0.186	1
ENSSSCG00000006663	OTUD7B	0.928	0.519	0.252	11.985	0.077	1	656.450	0.256	0.173	1
ENSSSCG00000026968	SLC20A2	1.043	0.535	0.415	11.786	0.134	1	572.369	0.422	0.221	1
ENSSSCG00000022192	NCAPD3	-0.806	-0.574	0.076	8.767	0.728	1	68.716	0.079	0.712	1
ENSSSCG00000004065	TIAM2	1.250	-0.451	0.026	9.676	0.934	1	130.927	0.030	0.905	1
ENSSSCG00000008745	PROM1			-0.385	1.847	0.093	1	33.700	-0.392	0.160	1
ENSSSCG00000024960	PDGFC			0.190	3.791	0.304	1	135.877	0.188	0.295	1
ENSSSCG00000004505	SMAD3			-0.162	2.811	0.401	1	67.692	-0.163	0.471	1
ENSSSCG00000001757	WDR61			0.060	4.968	0.662	1	309.262	0.061	0.752	1
ENSSSCG00000011370	DALRD3			-0.044	5.415	0.746	1	422.308	-0.044	0.743	1
ENSSSCG00000028225	DICER1			0.003	5.803	0.980	1	553.241	0.002	0.970	1

[a] log2 fold change of gene-overlapping tiling array DEPs from 'LO' to 'HI' samples in the DuPi population

[b] log2 fold change of gene-overlapping tiling array DEPs from 'LO' to 'HI' samples in the PiF1 population

[c] calculated or estimated by *edgeR*

[d] Average log2-counts per million across all samples estimated by *edgeR*

[e] Average normalized counts across all samples estimated by *DESeq*

[f] calculated or estimated by *DESeq*

Among the candidate genes listed in the first chunk of Table 3.12, H1F0 encodes a member protein of the histone H1 family, which could be found in cells that exhibit low cell division and differentiation. In this analysis, the 'HI' samples tend to have lower level of HIF0 expression. Since muscle cells, like nerve cells and red blood cells, are highly specialized cells and normally show no further cell division, this observation could be related to initial cell number in early muscle development. CLDN20 encodes a integral membrane protein of the claudin family, which operates as a physical barrier to restrain paracellular passing of water and solutes. The relatively low counts observed in the high drip loss animals could therefore explain the phenotypic difference. TNRC6B is a recently annotated protein coding gene, which sits within close proximity of several QTLs reported previously for drip loss [119; 143; 144; 145]. It has been reported to play a role in miRNA regulated gene silencing in human [146]. DLEC1, according to GO annotation, is associated with negative regulation of cell proliferation, which according to our observation is also lowly expressed in 'HI' samples. Thus the results could support the assumption of its involvement in the early muscle development. MGAT3 encodes an enzyme, according to UniProt [147], which is one of the most important regulators involved in the biosynthesis of glycoprotein oligosaccharides. ssc-mir-425 encodes a miRNA gene of the mir-425 family. According to miRwalk [148], 6 out of 10 validated miRNA target genes of mir-425 have also been annotated in the Ensembl GTF file used for feature counting. Therefore these 6 miRNA target genes were also included in the differential analysis of mRNA-seq data (In the lower chunk of Table 3.12). Half of the annotated miRNA target genes show negative regulation in the differential expression analysis, indicating the potential miRNA interference of transcription. In a previous study, we applied weighted gene co-expression network analysis to identify co-expression modules correlated to meat quality phenotypes using miRNA chip and gene expression chip [131]. Among the 4 probesets representing mir-425 family, three mammal probesets (mmu.miR.425.star_st, hsa.miR.425.star_st , bta.miR.425.5p_st) were found to be associated with module 'green' and 'brown', which have the highest negative correlation (cor=-0.13, p=0.07) with drip loss, consistent with the differential regulation status observed here; whereas the one from frog (xtr.miR.425.5p_st) were sorted into the more neutral module which has marginal correlation with drip loss (cor=0.045, p=0.5).

3.3. Validation and calibration

3.3.1. Validating common DEPs using qPCR

For those commonly detected DEPs regions and exons with consistent fold change direction across the two populations (See Table 3.13) in the customary regional tiling array experiment, we further validated the existence of their transcripts using quantitative polymerase chain reaction (qPCR). For all these regions we first cut out the corresponding genomic sequences and then tried to locate open reading frame (ORF) using NCBI ORF Finder[4]. For those with ORF detected we then used the predicted ORF start and end position to limit the template range, and for those without predicted ORF the whole sequence was used. The sequences were then passed to Primer-BLAST [149] to select nested optimum primers for multilplex qPCR. DEPs region 11 and 12 belong to the special case mentioned earlier, where region 11 was formed in the array result of DuPi population and region 12 was the DEPs region derived with the PiF1 population. The two regions are not directly overlapping, but both sitting on the same exon (ENSSSCE00000210352) of gene OTUD7B. So we initially used the whole exon sequence as template, and primers were selected for the two predicted ORFs which closely cover the two DEPs regions.

RPL32 was selected as reference house-keeping gene. The qPCR runs were done with 2 technical replicates for the same animals we have used for the tiling array experiment in the PiF1 population. We used both target starting quantity and threshold cycle (C_T) as measurement for qPCR product and as well as for fold change calculation. One tailed T-tests were latter conducted to test for up-regulation of corrected expression levels from 'LO' to 'HI' samples. Also Pearson's correlation tests were carried out to test agreement of array intensity with qPCR expression, with one tailed p-value for correlation greater than 0. Probes within each DEPs region were averaged using median and then averaged across each technical replicates of the same animal. Results of the t-tests and correlation tests are shown in Table 3.14.

Although most t-test p-values don't reach the 5% significance level, mainly due to the power constraint when discriminating relatively small difference with limited sample size, we do observe a slight up-regulation from the 'LO' samples to the 'HI' samples and high correlation with the array signals for most regions except for region 6. Further checking the pseudo array signals also gives poor agreement with the qPCR result of DEPs region 6 (Pearson's correlation = -0.01892477). While for DEPs region 11 and

[4]http://www.ncbi.nlm.nih.gov/projects/gorf/

12, though targeting the same exon, region 11 shows poor correlation with both array expressions and pseudo array counts, while region 12 gives better agreement between the three. This could be due the fact that DEPs region 11 was initially found only in the DuPi population, whereas the sequencing and qPCR runs were both made with samples from the same PiF1 animals. All other DEPs regions exhibit rather consistent correlation between the qPCR abundance and the array (or pseudo array) signals.

Table 3.13.: Commonly detected DEPs regions

DEPs	Ssc	from	to	str	Gene[a]	antisense [b]	intron [c]	ORF [d]	Pre[e]	Gene ID[f]
1	2	151052329	151052602	+						
2	2	151154649	151154909	+		NR3C1		144-260		ENSSSCG00000014401
3	2	151052125	151052831	-				33-182		
4	2	151054005	151054257	-						
5	5	6521977	6522212	+			SUN2	2-109	GENSCAN00000016024	ENSSSCG00000000094
6	5	4797096	4797306	-						
7	5	6969095	6969413	-			MAFF			ENSSSCG00000030165
8	5	6969929	6970088	-			MAFF	36-159		ENSSSCG00000030165
9	13	67835121	67835364	-		BHLHE40				ENSSSCG00000011534
10	17	12976180	12976467	+	SLC20A2		1-117		GENSCAN00000023394	ENSSSCG00000026968
11	4	108607431	108607587	-	OTUD7B			14-250[g]	GENSCAN00000024304	ENSSSCG00000006663
12	4	108610286	108610354	-	OTUD7B			2869-3196[g]	GENSCAN00000024304	ENSSSCG00000006663

[a] The overlapping gene symbol

[b] The overlapping antisense gene symbol

[c] The gene symbol of flanking exons

[d] The predicted ORF, 'from-to', relative to the region

[e] The *'ab initio'* gene prediction ID from Ensembl

[f] The associated Ensembl gene ID from either *a*, *b* or *c*

[g] The coordinates are relative to exon (ENSSSCE00000210352) sequence of gene OTUD7B.

Table 3.14.: Correlation of qPCR expression with tiling array and pseudo array

DEPs	FC.Array[a]	FC.qPCR[a]	ddct.FC.low [b]	ddct.FC.upp[b]	p.ttest[c]	cor.array[d]	cor.p.array[e]	cor.seq[d]	cor.p.seq[e]
1	1.6944394691	1.0991813224	0.8819257	1.4961667	0.2916724	0.1644963	0.3047194	0.6015129	0.01927089
2	1.7369847646	1.0448685217	0.9162039	1.609106	0.3176784	0.551105	0.03164236	0.280298	0.1887715
3	1.8120597162	1.0708723301	0.8146865	1.3525511	0.3675234	0.5177069	0.04235509	0.7133349	0.004597438
4	1.5672160744	1.1250333629	0.8075992	1.5457302	0.2191762	0.4191602	0.08749801	0.6694957	0.008622935
5	1.6485029857	1.1037519932	0.794923	1.900149	0.3972977	0.4384501	0.07696743	0.5707567	0.02630558
6	2.0515297849	0.9394792932	0.9553236	1.061378	0.5727461	0.08364467	0.3980319	-0.01892477	0.5232752
7	1.9456112825	2.3424237269	0.8533119	5.9953955	0.06450189	0.7816312	0.001337238	0.607794	0.018019
8	1.6457891306	2.3315378904	0.7523607	5.5231968	0.05720295	0.841914	0.0002962728	0.6964516	0.005929845
9	1.7015836782	1.5494187243	0.6289844	4.2838379	0.1680828	0.5601603	0.02909788	0.7193743	0.004179944
10	1.7277209239	1.4272210147	0.6214466	1.9403242	0.112359	0.4235642	0.08501854	0.7289881	0.003574668
11	1.569848351	1.0405664796	0.7968062	1.3450867	0.3506391	0.2794228	0.1895486	0.08034809	0.4019803
12	1.569848351	1.0720118455	1.067105	1.381559	0.290765	0.3327252	0.1453134	0.3764387	0.1138918

[a] Average fold change of DEPs region probes from 'LO' to 'HI' samples across the two population, using geometric mean.

[b] Upper- and lower-bound of qPCR expression fold change from 'LO' to 'HI' samples calculated with $2^{-\Delta\Delta C_T}$ method [150]

[c] P-value of one-tailed t-test of up-regulation of qPCR expression from 'LO' to 'HI' samples

[d] Pearson's correlation between the qPCR expression and the median of DEPs probes or pseudo array probes in PiF1 population.

[e] P-value of one-tailed Pearson's correlation test

3.3.2. Calibration of previous findings using mRNA-seq

We further used mapped reads from mRNA-seq to validate and calibrate results previously derived in the tiling array experiment, by visually inspecting the mRNA-seq coverage profiles of flanking regions for each DEPs site. Unlike in the segmentation step the reads were summed cross all samples at each base, two profiles 'HI' and 'LO', aggregated over sample types, were separately generated in a similar fashion. Thus not only the relative abundance of the DEPs feature to the annotated units could be identified, but also evidences of their differential expression status.

For the three DEPs regions which locate within annotated introns of gene SUN2 (region 5) and MAFF (region 7 and 8), the coverage profiles from mRNA-seq were plotted with *biomvRGviz* from package *biomvRCNS* (Figure 3.8). One could clearly see that, comparing to the annotated exon regions, the number of reads mapped to the array DEPs regions are extremely low though still with weak transcriptional activities. This means array signals can be noisy even after normalization and filtering, while sequencing gives better dynamic range for the relative expression.

For the three DEPs regions which overlap with annotated exons of gene SLC20A2 (region 10) and gene OTUD7B (region 11 and 12), the coverage profiles from mRNA-seq are shown in (Figure 3.9). The up-regulation of these two genes from 'LO' samples to 'HI' samples are rather strong, especially considering the relatively lower sequencing depth in 'LO' samples. It could also be seen that the boundaries of regions identified by mapped probes do not coincide well with annotated exon boundaries of SLC20A2, largely due to the overlapping resolution limits. The exon where DEPs region 11 and 12 are located is rather long, spanning 5143 bp, across which variation in the individual probe hybridization efficiency makes it difficult to detect long interspersed trancriptional fragments.

Multiple DEPs regions also fall in the regions without any annotated transcripts, which match to our previously made prediction on the mRNA-seq profiles. DEPs region 6 is covered by the predicted feature HSMMG342, which is located downstream of the 3' UTR of ST13. While DEPs regions 1, 3 and 4 are parts of prediction HSMMG288, which sits outside the 3' UTR of NR3C1 and upstream of gene ARHGAP26.

Two of the DEPs regions (2 and 9) are overlapping with the opposite strand of the coding sequence of ST13 (exonID: ENSSSCE00000238865) and BHLHE40 (exonID: ENSSSCE00000102062). Both of the DEPs regions match to the end of the cooresponding exons, but not necessarily on the 3' end, which is different from the classic 3' array manufactured by Affymetrix. For BHLHE40, a predicted trancriptionally active feature

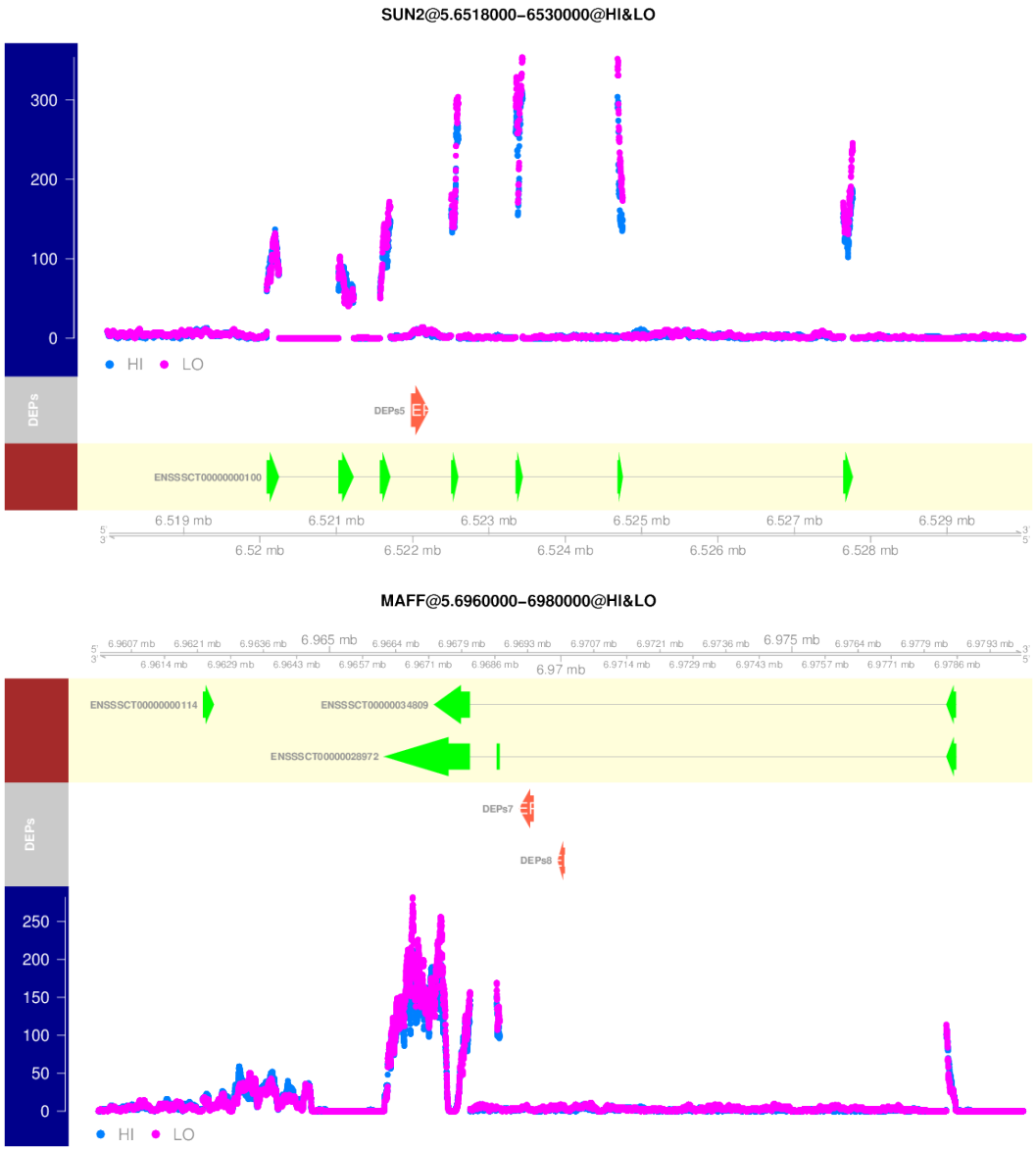


Figure 3.8.: mRNA-seq profiles of intronic DEPs regions. Scatter plot shows the per-base coverage of the region by mRNA-seq reads.

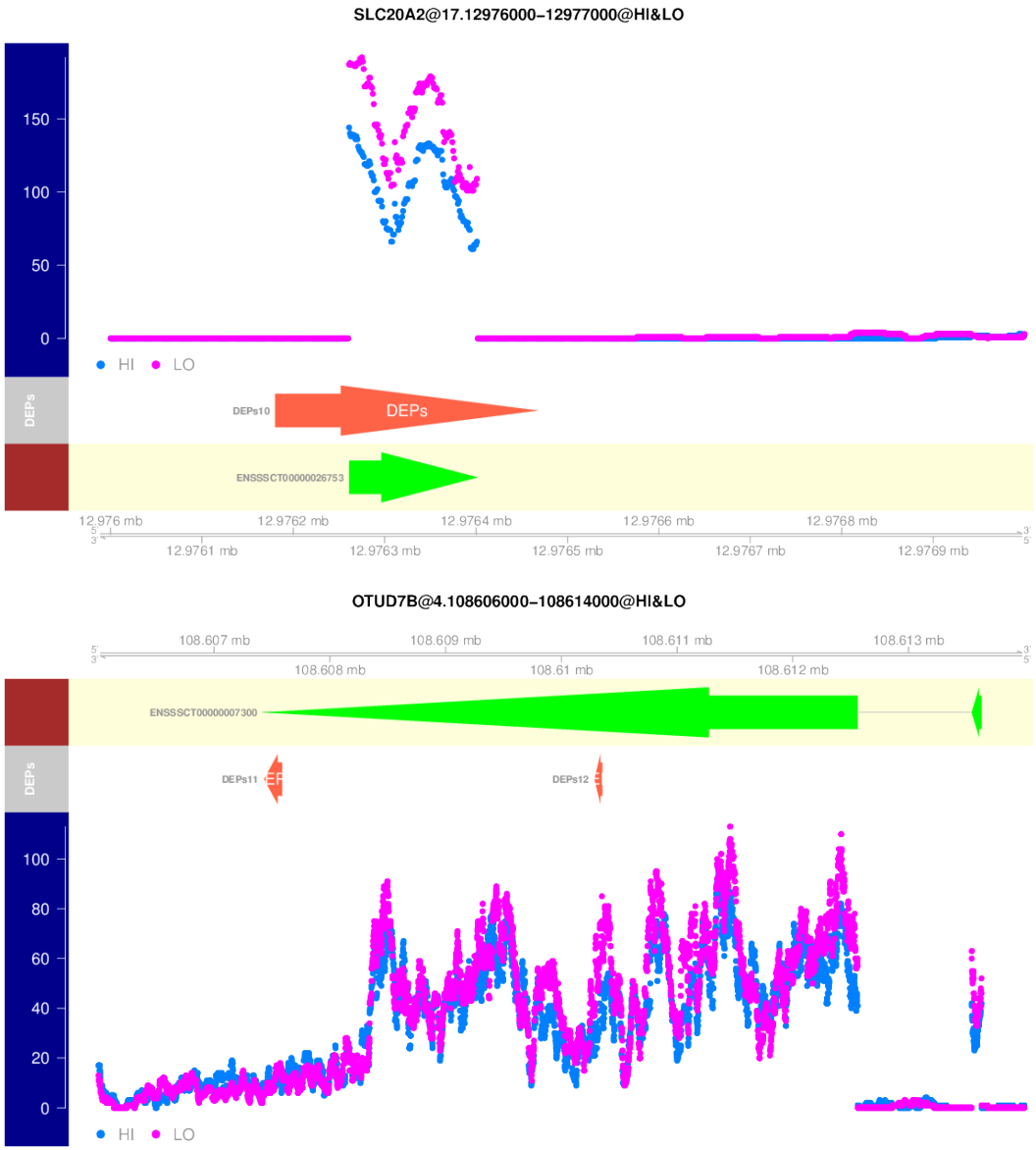


Figure 3.9.: mRNA-seq profiles of exonic DEPs regions, with brown panel (green feature) representing annotated transcripts, grey panel (cyan feature) representing common DEPs region detected in the tiling array experiments. Scatter plot shows the per-base coverage of the region by mRNA-seq reads.

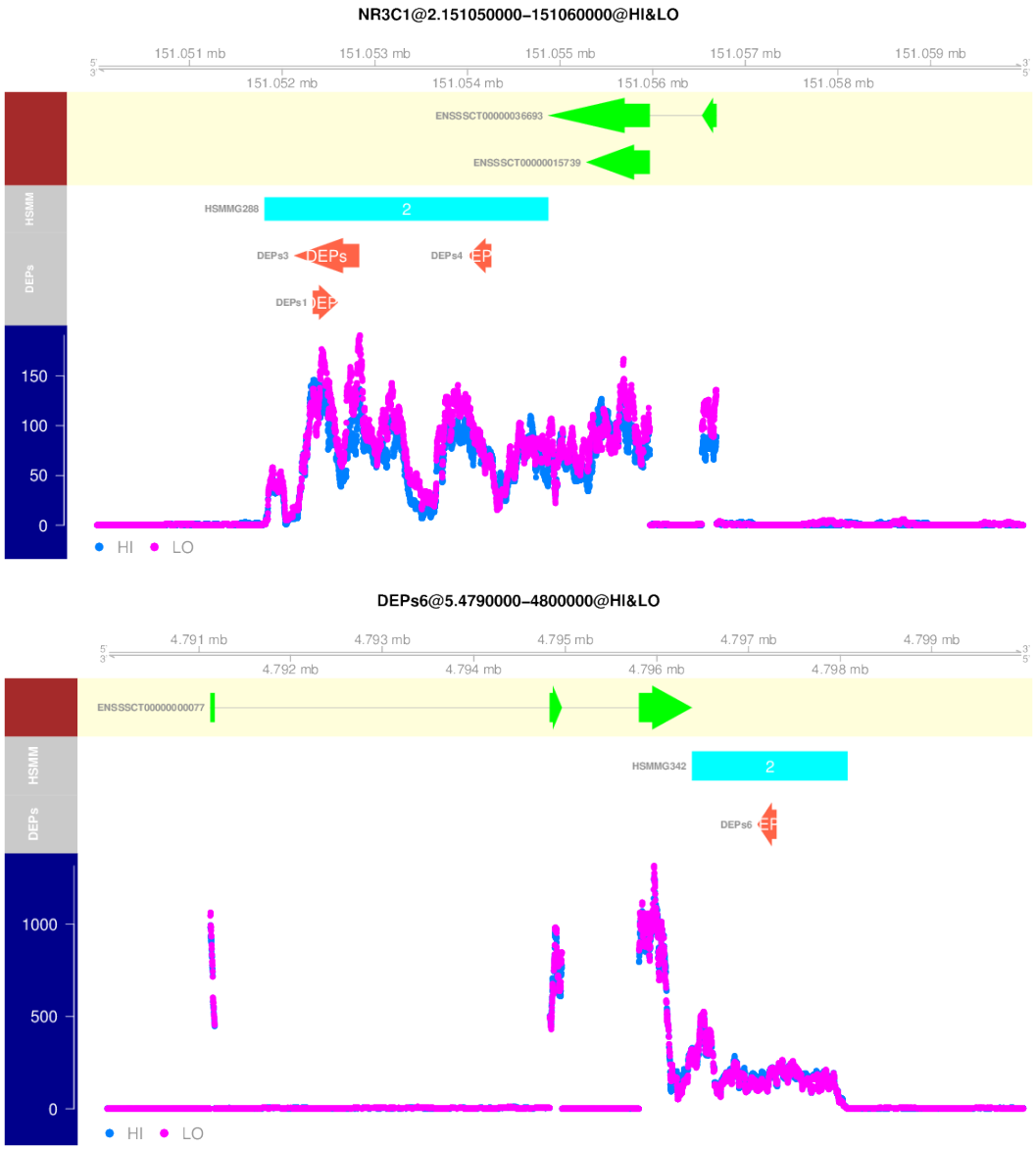


Figure 3.10.: mRNA-seq profiles of DEPs regions with HSMM prediction, with brown panel (green feature) representing annotated transcripts, grey panel (cyan feature) representing common DEPs region detected in the tiling array experiments. Scatter plot shows the per-base coverage of the region by mRNA-seq reads.

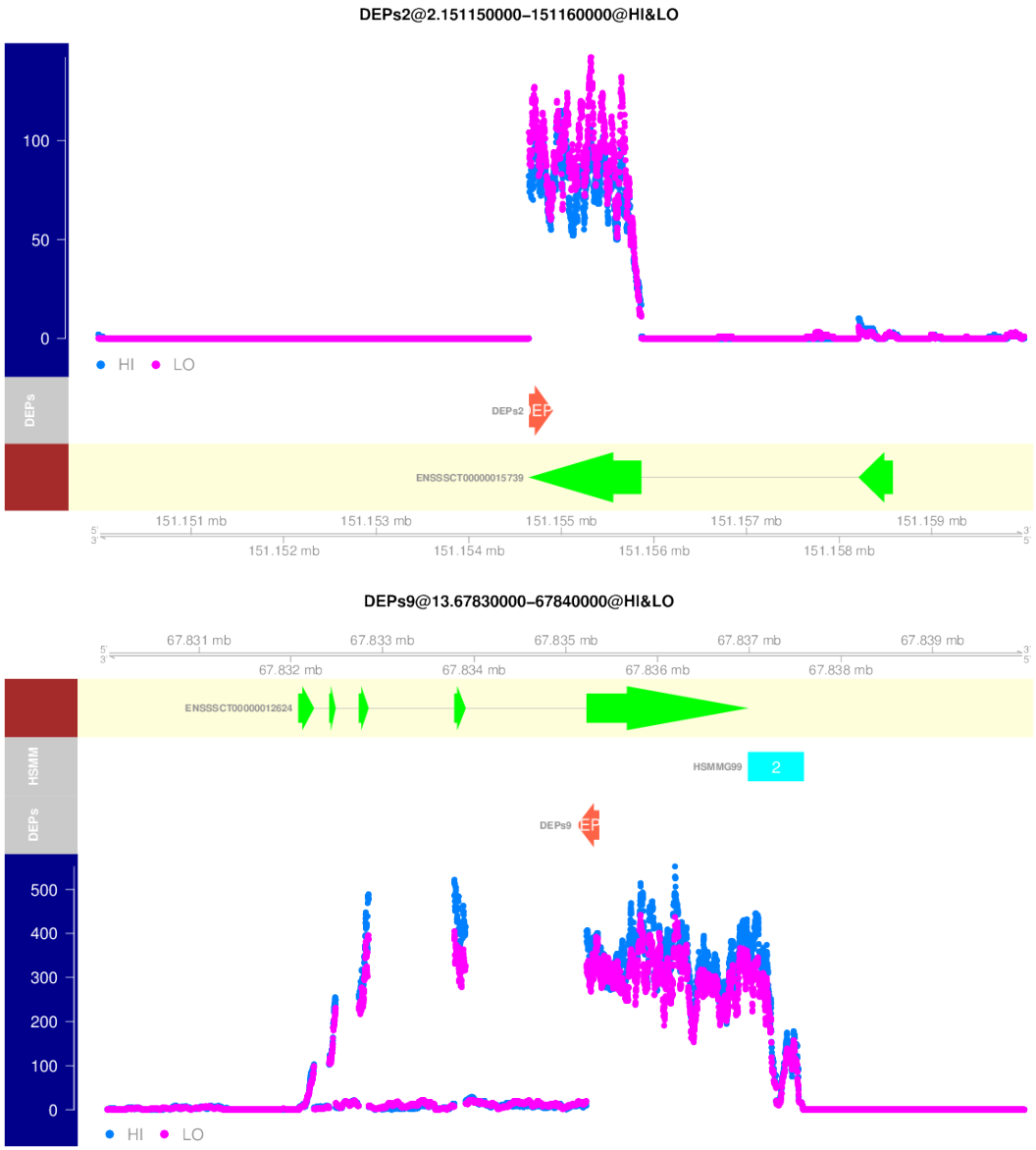


Figure 3.11.: mRNA-seq profiles of DEPs regions which match anti-sense genes, with brown panel (green feature) representing annotated transcripts, grey panel (cyan feature) representing common DEPs region detected in the tiling array experiments or novel transcriptionally active units detected by the HSMM segmentation of mRNA-seq profile. Scatter plot shows the per-base coverage of the region by mRNA-seq reads.

is also found outside the 3′ UTR.

In Table 3.15, previous mRNA-seq DEGs analysis results for exonic DEPs genes (SLC20A2 and OTUD7B), anti-sense DEPs features and DEPs overlapping prediction from previous sequencing profile segmentation are listed. For most of the DEPs features, the prevailing up-regulation in 'LO' samples estimated from the tiling array data are largely consistent with the sequencing results. DEPs feature 6, which overlaps with the strandless novel prediction HSMMG342, shows different regulation status in sequencing. A similar relationship is found for DEPs feature 9, which sits in the opposite strand of BHLHE40, while the novel splicing prediction HSMMG99 does agree with BHLHE40 in their differential expression assessment.

Table 3.15.: Differential expression status of DEPs in mRNA-seq

DEPs	Feature[a]	Relation[b]	logFC[c]	logCPM[d]	pval[c]	baseMean[e]	logFC[f]	pval[f]	Gene Info[g]
10	SLC20A2	Exon	0.414 8	11.786 1	0.133 6	572.369 5	0.422 1	0.220 8	ENSSSCG00000026968
11/12	OTUD7B	Exon	0.251 6	11.985 4	0.077 2	656.449 9	0.256 5	0.172 7	ENSSSCG00000006663
1/3/4	HSMMG288	Novel	0.361 4	11.581 9	0.093 3	495.575 2	0.366 1	0.225 5	Ssc2 [151051808, 151054872] *
6	HSMMG342	Novel	−0.014 7	12.135 6	0.921 8	728.266 8	−0.010 0	0.994 4	Ssc5 [4796382, 4798079] *
2	NR3C1	Anitisense	0.329 5	11.957 9	0.120 9	643.535 4	0.332 5	0.257 2	ENSSSCG00000014401
9	BHLHE40	Antisense	−0.150 8	13.277 8	0.589 1	1 607.023 1	−0.153 2	0.666 4	ENSSSCG00000011534
9	HSMMG99	Novel	−0.216 5	10.791 3	0.435 3	285.334 8	−0.217 6	0.540 1	Ssc13[67836985, 67837595] *

[a] Feature ID used in the sequencing reads quantification

[b] Relationship between the DEPs region and the quantified feature: Exon, overlapping with its exon; Novel, overlapping with the predicted novel transcriptional fragments; Antisense, overlapping with the opposite strand of the annotated exon.

[c] calculated or estimated by *edgeR*

[d] Average log2-counts per million across all samples estimated by *edgeR*

[e] Average normalized counts across all samples estimated by *DESeq*

[f] calculated or estimated by *DESeq*

[g] If the feature is related to a known gene, then its Ensembl gene ID is listed; if the feature is related to a novel prediction, then the positional information 'chr[start.pos, end.pos]strand' of the prediction is given.

CHAPTER 4

Discussion and Outlooks

High throughput technologies are pushing genomics forward, allowing researchers to survey genomic regions at base pair resolution. In the last few chapters, two novel methods developed during this dissertation have been presented and their individual performances and usages are illustrated. Also as a comprehensive use case, the two implementations have been employed in the experiments carried out in the related research project.

One of the most crucial issues to deal with in genomics is the ambiguity arising from sequence homology. As have been shown in the section of repetitive sequence and sequence complexity, duplicated DNA sequence of variable length are common in the genome of most life forms. However, this imposes a great challenge on the technologies used in genomic research. As the two flagship high-throughput techniques used to quantify level of various biological activities, tiling array and NGS both require careful handling of non-uniquely mapped features to ensure their accuracies. For each microarray based experiment, one has to pre-select probes and deposit them on the array surface for later hybridization reactions with labeled samples. The sequencing methods are working the other way around, where one has to later map the generated short reads back to a reference. Thus many works have been done in the field of array probe design and mapping sequencing reads back to reference. However, unlike the constant improvements made in the development of sequencing technology, like strand-specific sequencing protocols and prolonged reads, which may help improve mapping rate and also reduce ambiguity in *de novo* genome assembly, grounds for technical improvement in array design are generally limited. Probe quality in microarray largely depends on the pre-selection of uniquely aligned sequences.

The probe selection algorithm implemented in this study utilizes the proposed pe-

nalized uniqueness score as a controlling criterion of cross-hybridization, together with several other parameters for the flexible tweaking of the positional distribution, optimal hybridization efficiency, and essential constraints of sequence complexity. Profiling Agilent's catalog probes confirms the appropriateness of our parameter settings. The intention of the algorithm is to allow a part of the probe to overlap with its neighboring tiles, giving higher coverage and resolution for experiments, like a targeted sequencing library preparation. However, it can also be used to design CHIP-chip experiments, which require distantly-spaced probes across large genomic regions. To achieve this, the selection algorithm behaves slightly differently when a negative overlap is specified, and it will not attempt to shift to the 5′, if that would induce overlapping. Studies have shown that sequence polymorphisms may affect probe hybridization efficiency [151; 152; 153]. Thus common SNPs defined in databases, like dbSNP [154], could be excluded in the input FASTA files by cutting the region into two new regions, or SNP could be masked with lowercase letters and controlled like repetitive regions. The design is done on a per-sequence basis thus the memory requirement of the implementation depends on the largest sequence contig in question. For each sequence header defined in the source FASTA files, to calculate the uniqueness score all corresponding MUP entries in the prefix file have to be imported and processed in RAM; each entry will need two integers for the position and the length of the prefix, respectively, to be stored in the array. This in turn suggests that for large genome tiling design - e.g., when covering a whole mammalian chromosome, which could have a length of 3×10^8 bp, the memory requirement will exclude the possibility of applying the method on a normal desktop PC. Additionally, the algorithm runs in linear time with various parameters affecting the exact time expectation. Yet further parallelization is easily available either on a per-sequence basis given sufficient memory or on a per-chunk basis, since, in our implementation, the individual sequence is initially pieced into segments with continuous non-ambiguous bases and sequentially processed. Simple per-chunk test utilizing 4 parallel processes further reduced the running time to approximately 45 min for the same designing task that is presented in the coverage comparison section with Agilent CHIP-on-chip set.

Compared to traditional BLAST-like alignment methods, this definition of the penalized uniqueness score makes less parametric model assumptions for the homologous estimation, letting it be more sequence-driven and less sensitive to arbitrary parameter settings. The calculation of the score makes usage of GenomeTools [48], which is memory relaxed, and inherently benefit from the computational efficiency of the FM-index [46]. As a rational variable, it provides a more continuous distribution and a wider dynamic range for the uniqueness measurement without further increase in the computational complexity, while showing higher sensitivity and specificity over the original

count score, which only takes discrete integers ranging from 0 to the user specified maximal length of MUP. The score itself, like the original count score, measures the degree of uniqueness and dissimilarity of the sequence to the rest of the genome, which means the lower the score the higher possibility for non-specific binding. Thus, it could be further factored in the background correction model or at the normalization step prior to the downstream array data analysis, to correct for cross-hybridization noises using the uniqueness score. Such a correction model could take the typical form as the exponential-normal convolution model in RMA algorithm [155; 156], by adding an additional term for non-specific binding noise.

In recent years with advancing technology and lower costs, NGS starts to replace array based experiments in large scale genomic experiments. Despite prevailing trend of NGS, tiling array remains more cost-effective for large samples which typically provide higher and more reliable statistical power — therefore, cross-platform collaboration between deep sequencing data and array data has been considered [157]. Additionally, relatively few ongoing research projects seek to solve biological questions on a whole-genome scale, so it is more likely that several linked QTL regions or intervals identified in genome-wide association studies are pursued in detail, like I have shown in the case study. Such specific interests can also be addressed by using capture oligos for targeted enrichment of DNA fragments representing the genomic region of interest [158]. These capture oligos might be applied either in solution-based or microarray-based methods, thus combining the two platforms in the array assisted targeted sequencing approaches, where targeted regions could be tiled on DNA capture arrays, and the hybridization products could be used in the follow-up sequencing library preparation [159; 160; 161]. This customized tailoring tool selectively enriches only the regions of interest and provides the opportunity to reduce both cost and processing time, while retaining high sensitivity through high-throughput technology. Also as have been illustrated in the case study here and confirmed by other researchers [162] that, tiling array and RNA-seq can provide complementary results in transcriptome profiling.

Moving one step forward into functional genomic data analysis, data generated from experiments like expression tiling array and mRNA-seq requires to be first quantified using biological sensible unit typically taking the form of annotation, which many curated databases may provide and only carries known information up to date. According to the result from ENCODE project, 80 percents of the human genome are either transcribed or biochemically functional [163]. In agreement with our regional investigation in the case study, we also identified numbers of potentially coding and non-coding novel transcripts. In applications other than transcriptome profiling, feature quantification could be done through a naive "window" methods, like the one used here in the CHIP-on-chip

design comparison. Later on, signals quantified using these units can then be sequentialized using the associated genomic positions. Then the detection of consecutive units which exhibit homogeneous signals could be solved using segmentation models.

The segmentation problem in general is involved in many types of biological experiment, and could naturally fit into the hidden Markov model framework with segment boundaries modeled as transitions between hidden states. As a generalization of hidden Markov model, HSMM allows the sojourn distribution to be specified other than the Geometric distribution implicitly used in common HMM. Given the complexity of the genome, such an implicit assumption could be easily violated. Though the true underlying sojourn distributions involving various genomic features remains unknown, the HSMM implementation gives more flexible options in the modeling and thus might provide more insights.

In this implementation `biomvRhsmm`, several types of sojourn distribution are implemented. For example, with Gamma distributed sojourn, the neighboring position will tend to stay in the same state, and transit to other states if far apart. Different from the original design in Guédon [99], the R suite implemented in this work utilizes the positional information that naturally comes with most genomic features for the sojourn density estimation. Such an integrative approach is advantageous comparing to simply using the rank of their positions since mapping positions are not always uniformly distributed and the spatial patterns may be of interest in experiments like DMRs detection. It also differs from those models which embed these positions in a non-parametric fashion like `BioHMM` [69] and *QuantiSNP* [75], or the "instability-selection" model for LOH analysis [78; 79], which all employ variations of exponential function to account for genomic distance. The HSMM is considered more close to the DBN model employed in Segway [89], yet being less experiment specific and easier to interpret, not to mention the convenient communication with other analytical and visualization tools within the Bioconductor community.

The explosion of data availability also provides another possibility of learning from previous studies to benefit one's current work. Other than the flat prior commonly used in Bayesian inference, prior information for the sojourn density could be estimated from annotation or previous studies, thus be effectively utilized together with positional information of features to guide the estimation of the most likely state sequence. With its full probabilistic model, various emission densities are provided, enabling the model to handle normally distributed data from traditional array platform as well as count data from sequencing experiment. The proposed model has also been applied on well-studied aCGH dataset from Coriell cell lines [63] and RNA-seq data generated by ENCODE project [108; 107] to illustrate its functionalities. As have been shown with

these experimental datasets, the underlying data distributions could be more complex than any particular parametric distributions can recover. Thus it would be beneficial to explore the usage of higher-order mixture of distributions to model the emission density or the sojourn density.

Like each industry revolution in the human history, new technologies are the driving force to reshape human recognition of science and our interaction with the world. The future of genome biology, as an experimental science, will inevitably depend on the technological renovation. As has been observed in the last few years, advent of NGS has made microarrays largely replaced by this paradigm-shifting tool in many sectors of genome research, for the additional information on genetic structure and variation it conveys. However sequencing data analysis and modeling is still in its early age. In many areas, like differentially expressed gene (DEG) analysis, general consensus has not been reached and gold standard has not been set. Advanced analytical methods and statistical models to depict complex genomic data are highly needed and are under active research. Thus for contemporary genome research, combining the power of different technologies, and embedding prior knowledge in the optimized design of experiment, in complex data analysis and interpretation, eventually with independent validation of scientific findings, provides a unique opportunity to better understand genomics and conduct reproducible research.

Bibliography

[1] Leonard H. Augenlicht and Diane Kobrin. Cloning and screening of sequences expressed in a mouse colon tumor. *Cancer Research*, 42(3):1088–1093, 1982.

[2] D A Kulesh, D R Clive, D S Zarlenga, and J J Greene. Identification of interferon-modulated proliferation-related cdna sequences. *Proceedings of the National Academy of Sciences*, 84(23):8453–8457, 1987.

[3] L H Augenlicht, J Taylor, L Anderson, and M Lipkin. Patterns of gene expression that characterize the colonic mucosa in patients at genetic risk for colonic cancer. *Proceedings of the National Academy of Sciences*, 88(8):3286–3289, 1991.

[4] Mark Schena, Dari Shalon, Ronald W. Davis, and Patrick O. Brown. Quantitative Monitoring of Gene Expression Patterns with a Complementary DNA Microarray. *Science*, 270(5235):467–470, 1995. 10.1126/science.270.5235.467.

[5] Thomas E. Royce, Joel S. Rozowsky, Paul Bertone, Manoj Samanta, Viktor Stolc, Sherman Weissman, Michael Snyder, and Mark Gerstein. Issues in the analysis of oligonucleotide tiling microarrays for transcript mapping. *Trends in Genetics*, 21(8):466–475, 2005.

[6] Robert Nadon and Jennifer Shoemaker. Statistical issues with microarrays: processing and analysis. *Trends in Genetics*, 18(5): 265–271, 2002.

[7] Jay Shendure and Hanlee Ji. Next-generation DNA sequencing. *Nat Biotech*, 26(10):1135–1145, 2008. 10.1038/nbt1486.

[8] Wendy Weijia Soon, Manoj Hariharan, and Michael P. Snyder. High-throughput sequencing for biology and medicine. *Mol Syst Biol*, 9, 2013. 10.1038/msb.2012.61.

[9] **Yang Du**, Eduard Murani, Siriluck Ponsuksili, and Klaus Wimmers. Flexible and efficient genome tiling design with penalized uniqueness score. *BMC Bioinformatics*, 13(1):323, 2012. ISSN 1471-2105. doi: 10.1186/1471-2105-13-323.

[10] **Yang Du**, Eduard Murani, Siriluck Ponsuksili, and Klaus Wimmers. biomvRhsmm: Genomic segmentation with hidden semi-Markov model. *BioMed Research International*, 2014, 2014. ISSN 2314-6133. doi: 10.1155/2014/910390.

[11] David J. Lockhart, Helin Dong, Michael C. Byrne, Maximillian T. Follettie, Michael V. Gallo, Mark S. Chee, Michael Mittmann, Chunwei Wang, Michiko Kobayashi, Heidi Norton, and Eugene L. Brown. Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nat Biotech*, 14(13):1675–1680, 1996. 10.1038/nbt1296-1675.

[12] Michael D. Kane, Timothy A. Jatkoe, Craig R. Stumpf, Jia Lu, Jeffrey D. Thomas, and Steven J. Madore. Assessment of the sensitivity and specificity of oligonucleotide (50mer) microarrays. *Nucleic Acids Research*, 28(22):4552–4557, 2000. 10.1093/nar/28.22.4552.

[13] Paul Bertone, Valery Trifonov, Joel S. Rozowsky, Falk Schubert, Olof Emanuelsson, John Karro, Ming-Yang Kao, Michael Snyder, and Mark Gerstein. Design optimization methods for genomic DNA tiling arrays. *Genome research*, 16(2):271–281, 2006. doi: 10.1101/gr.4452906.

[14] Todd J. Treangen and Steven L. Salzberg. Repetitive DNA and next-generation sequencing: computational challenges and solutions. *Nat Rev Genet*, 13(2):146–146, 2012. doi: 10.1038/nrg3164.

[15] Hairong Wei, Pei Fen Kuan, Shulan Tian, Chuhu Yang, Jeff Nie, Srikumar Sengupta, Victor Ruotti, Gudrun A. Jonsdottir, Sunduz Keles, James A. Thomson, and Ron Stewart. A study of the relationships between oligonucleotide properties and hybridization signal intensities from NimbleGen microarray datasets. *Nucleic Acids Research*, 36(9):2926–2938, 2008.

[16] Richard Owczarzy, Bernardo G. Moreira, Yong You, Mark A. Behlke, and Joseph A. Walder. Predicting Stability of DNA Duplexes in Solutions Containing Magnesium and Monovalent Cations. *Biochemistry*, 47(19):5336–5353, 2008.

[17] Julius Marmur and Paul Doty. Determination of the base composition of deoxyribonucleic acid from its thermal denaturation temperature. *Journal of molecular biology*, 5(1):109–118, 1962.

[18] R. Bruce Wallace, J. Shaffer, R. F. Murphy, J. Bonner, T. Hirose, and K. Itakura. Hybridization of synthetic oligodeoxyribonucleotides to $\phi\chi$ 174 DNA: the effect of single base pair mismatch. *Nucleic Acids Research*, 6(11):3543–3558, 1979. 10.1093/nar/6.11.3543.

[19] P. M. Howley, M. A. Israel, M. F. Law, and M. A. Martin. A rapid method for detecting and mapping homology between heterologous DNAs. Evaluation of polyomavirus genomes. *Journal of Biological Chemistry*, 254(11):4876–4883, 1979.

[20] John SantaLucia. A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics. *Proceedings of the National Academy of Sciences*, 95(4):1460–1465, 1998.

[21] Alejandro Panjkovich and Francisco Melo. Comparison of different melting temperature calculation methods for short DNA sequences. *Bioinformatics*, 21(6):711–722, 2005.

[22] John SantaLucia, Hatim T. Allawi, and P. Ananda Seneviratne. Improved Nearest-Neighbor Parameters for Predicting DNA Duplex Stability. *Biochemistry*, 35(11):3555–3562, 1996.

[23] Hatim T. Allawi and John SantaLucia. Thermodynamics and NMR of Internal G·T Mismatches in DNA. *Biochemistry*, 36(34): 10581–10594, 1997.

[24] Carl W. Schmid and Prescott L. Deininger. Sequence organization of the human genome. *Cell*, 6(3):345–358, 1975.

[25] Mark A. Batzer and Prescott L. Deininger. Alu repeats and human genomic diversity. *Nat Rev Genet*, 3(5):370–379, 2002. 10.1038/nrg798.

[26] The Arabidopsis Genome Initiative. Analysis of the genome sequence of the flowering plant *Arabidopsis thaliana*. *Nature*, 408 (6814):796–815, 2000. 10.1038/35048692.

[27] Patrick S. Schnable, Doreen Ware, Robert S. Fulton, Joshua C. Stein, Fusheng Wei, Shiran Pasternak, Chengzhi Liang, Jianwei Zhang, Lucinda Fulton, Tina A. Graves, Patrick Minx, Amy Denise Reily, Laura Courtney, Scott S. Kruchowski, Chad Tomlinson, Cindy Strong, Kim Delehaunty, Catrina Fronick, Bill Courtney, Susan M. Rock, Eddie Belter, Feiyu Du, Kyung Kim, Rachel M. Abbott, Marc Cotton, Andy Levy, Pamela Marchetto, Kerri Ochoa, Stephanie M. Jackson, Barbara Gillam, Weizu Chen, Le Yan, Jamey Higginbotham, Marco Cardenas, Jason Waligorski, Elizabeth Applebaum, Lindsey Phelps, Jason Falcone, Krishna Kanchi, Thynn Thane, Adam Scimone, Nay Thane, Jessica Henke, Tom Wang, Jessica Ruppert, Neha Shah, Kelsi Rotter, Jennifer Hodges, Elizabeth Ingenthron, Matt Cordes, Sara Kohlberg, Jennifer Sgro, Brandon Delgado, Kelly Mead, Asif Chinwalla, Shawn Leonard, Kevin Crouse, Kristi Collura, Dave Kudrna, Jennifer Currie, Ruifeng He, Angelina Angelova, Shanmugam Rajasekar, Teri Mueller, Rene Lomeli, Gabriel Scara, Ara Ko, Krista Delaney, Marina Wissotski, Georgina Lopez, David Campos, Michele Braidotti, Elizabeth Ashley, Wolfgang Golser, HyeRan Kim, Seunghee Lee, Jinke Lin, Zeljko Dujmic, Woojin Kim, Jayson Talag, Andrea Zuccolo, Chuanzhu Fan, Aswathy Sebastian, Melissa Kramer, Lori Spiegel, Lidia Nascimento, Theresa Zutavern, Beth Miller, Claude Ambroise, Stephanie Muller, Will Spooner, Apurva Narechania, Liya Ren, Sharon Wei, Sunita Kumari, Ben Faga, Michael J. Levy, Linda McMahan, Peter Van Buren, Matthew W. Vaughn, et al. The B73 Maize Genome: Complexity, Diversity, and Dynamics. *Science*, 326(5956):1112–1115, 2009. doi: 10.1126/science.1178534.

[28] Nam-Hyuk Cho, Hang-Rae Kim, Jung-Hee Lee, Se-Yoon Kim, Jaejong Kim, Sunho Cha, Sang-Yoon Kim, Alistair C. Darby, Hans-Henrik Fuxelius, Jun Yin, Ju Han Kim, Jihun Kim, Sang Joo Lee, Young-Sang Koh, Won-Jong Jang, Kyung-Hee Park, Siv G. E. Andersson, Myung-Sik Choi, and Ik-Sang Kim. The Orientia tsutsugamushi genome reveals massive proliferation of conjugative type IV secretion system and host—cell interaction genes. *Proceedings of the National Academy of Sciences*, 104(19):7981–7986, 2007. doi: 10.1073/pnas.0611553104.

[29] Arian FA Smit, Robert Hubley, and Phil Green. RepeatMasker Open-3.0, 1996.

[30] Arian F. A. Smit. The origin of interspersed repeats in the human genome. *Current Opinion in Genetics & Development*, 6(6): 743–748, 1996.

[31] Jerzy Jurka, Vladimir V. Kapitonov, Oleksiy Kohany, and Michael V. Jurka. Repetitive Sequences in Complex Genomes: Structure and Evolution. *Annual Review of Genomics and Human Genetics*, 8(1):241–259, 2007.

[32] Philip M. Kim, Hugo Y.K. Lam, Alexander E. Urban, Jan O. Korbel, Jason Affourtit, Fabian Grubert, Xueying Chen, Sherman Weissman, Michael Snyder, and Mark B. Gerstein. Analysis of copy number variants and segmental duplications in the human genome: Evidence for a change in the process of formation in recent evolutionary history. *Genome research*, 18(12):1865–1874, 2008.

[33] Roy J. Britten. Transposable element insertions have strongly affected human evolution. *Proceedings of the National Academy of Sciences*, 2010.

[34] Patricia Martin, Katherine Makepeace, Stuart A. Hill, Derek W. Hood, and E. Richard Moxon. Microsatellite instability regulates transcription factor binding and gene expression. *Proceedings of the National Academy of Sciences of the United States of America*, 102 (10):3800–3804, 2005. 10.1073/pnas.0406805102.

[35] Kathrin Reichwald, Chris Lauber, Indrajit Nanda, Jeanette Kirschner, Nils Hartmann, Susanne Schories, Ulrike Gausmann, Stefan Taudien, Markus Schilhabel, Karol Szafranski, Gernot Glockner, Michael Schmid, Alessandro Cellerino, Manfred Schartl, Christoph Englert, and Matthias Platzer. High tandem repeat content in the genome of the short-lived annual fish nothobranchius furzeri: a new vertebrate model for aging research. *Genome Biology*, 10(2):R16, 2009.

[36] Michael Mitas. Trinucleotide repeats associated with human disease. *Nucleic Acids Research*, 25(12):2245–2253, 1997. doi: 10.1093/nar/25.12.2245.

[37] Iordanis I. Arzimanoglou, Fred Gilbert, and Hugh R. K Barber. Microsatellite instability in human solid tumors. *Cancer*, 82(10): 1808–1820, 1998.

[38] Irina Voineagu, Vidhya Narayanan, Kirill S. Lobachev, and Sergei M. Mirkin. Replication stalling at unstable inverted repeats: Interplay between DNA hairpins and fork stabilizing proteins. *Proceedings of the National Academy of Sciences*, 105(29):9936–9941, 2008.

[39] Aleksandr Morgulis, E. Michael Gertz, Alejandro A. Schäffer, and Richa Agarwala. WindowMasker: window-based masker for sequenced genomes. *Bioinformatics*, 22(2):134–141, 2006. 10.1093/bioinformatics/bti774.

[40] Joost H. A. Martens, Roderick J. O'Sullivan, Ulrich Braunschweig, Susanne Opravil, Martin Radolf, Peter Steinlein, and Thomas Jenuwein. The profile of repeat-associated histone lysine methylation states in the mouse epigenome. *EMBO J*, 24(4):800–812, 2005. 10.1038/sj.emboj.7600545.

[41] E. Ryder, R. Jackson, A. Ferguson-Smith, and S. Russell. MAMMOT—a set of tools for the design, management and visualization of genomic tiling arrays. *Bioinformatics*, 22(7):883–884, 2006.

[42] Stefan Gräf, Fiona G. G. Nielsen, Stefan Kurtz, Martijn A. Huynen, Ewan Birney, Henk Stunnenberg, and Paul Flicek. Optimized design and assessment of whole genome tiling arrays. *Bioinformatics*, 23(13):i195–i204, 2007. doi: 10.1093/bioinformatics/btm200.

[43] Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, 1990.

[44] Udi Manber and Gene Myers. Suffix Arrays: A New Method for On-Line String Searches. *SIAM Journal on Computing*, 22(5): 935–948, 1993.

[45] Sophie Lemoine, Florence Combes, and Stéphane Le Crom. An evaluation of custom microarray applications: the oligonucleotide design challenge. *Nucleic Acids Research*, 37(6):1726–1739, 2009.

[46] Ferragina, P. and Manzini, G. Opportunistic data structures with applications. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, page 390, 796543, 2000. IEEE Computer Society.

[47] Burrows, M. and Wheeler, David J. A block-sorting lossless data compression algorithm. Technical report, Digital Equipment Corporation, 1994.

[48] Gordon Gremme, Sascha Steinbiss, and Stefan Kurtz. GenomeTools: A Comprehensive Software Library for Efficient Processing of Structured Genome Annotations. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 10(3):645–656, 2013.

[49] W. James Kent. BLAT—The BLAST-Like Alignment Tool. *Genome research*, 12(4):656–664, 2002. 10.1101/gr.229202.

[50] Robert Gentleman, Vincent Carey, Douglas Bates, Ben Bolstad, Marcel Dettling, Sandrine Dudoit, Byron Ellis, Laurent Gautier, Yongchao Ge, Jeff Gentry, Kurt Hornik, Torsten Hothorn, Wolfgang Huber, Stefano Iacus, Rafael Irizarry, Friedrich Leisch, Cheng Li, Martin Maechler, Anthony Rossini, Gunther Sawitzki, Colin Smith, Gordon Smyth, Luke Tierney, Jean Yang, and Jianhua Zhang. Bioconductor: open software development for computational biology and bioinformatics. *Genome Biology*, 5(10):R80, 2004.

[51] Pedro Lopez-Romero. *Agi4x44PreProcess: PreProcessing of Agilent 4x44 array data*, 2012. R package version 1.20.0.

[52] Michael J. Buck and Jason D. Lieb. ChIP-chip: considerations for the design, analysis, and application of genome-wide chromatin immunoprecipitation experiments. *Genomics*, 83(3):349–360, 2004.

[53] Rosalind C. Lee, Rhonda L. Feinbaum, and Victor Ambros. The *C. elegans* heterochronic gene lin-4 encodes small RNAs with antisense complementarity to lin-14. *Cell*, 75(5):843–854, 1993.

[54] Brenda J. Reinhart, Frank J. Slack, Michael Basson, Amy E. Pasquinelli, Jill C. Bettinger, Ann E. Rougvie, H. Robert Horvitz, and Gary Ruvkun. The 21-nucleotide let-7 RNA regulates developmental timing in *Caenorhabditis elegans*. *Nature*, 403(6772):901–906, 2000. 10.1038/35002607.

[55] Mariana Lagos-Quintana, Reinhard Rauhut, Winfried Lendeckel, and Thomas Tuschl. Identification of Novel Genes Coding for Small Expressed RNAs. *Science*, 294(5543):853–858, 2001.

[56] Mikhail S. Gelfand and Eugene V. Koonin. Avoidance of palindromic words in bacterial and archaeal genomes: a close connection with restriction enzymes. *Nucleic Acids Research*, 25(12):2430–2439, 1997.

[57] Tong Ihn Lee, Nicola Rinaldi, François Robert, Duncan Odom, Ziv Bar-Joseph, Georg Gerber, Nancy Hannett, Christopher Harbison, Craig Thompson, Itamar Simon, et al. Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science Signaling*, 298(5594):799, 2002.

[58] Akiko Doi, In-Hyun Park, Bo Wen, Peter Murakami, Martin J Aryee, Rafael Irizarry, Brian Herb, Christine Ladd-Acosta, Junsung Rho, Sabine Loewer, et al. Differential methylation of tissue-and cancer-specific CpG island shores distinguishes human induced pluripotent stem cells, embryonic stem cells and fibroblasts. *Nature genetics*, 41(12):1350–1353, 2009.

[59] Paul Bertone, Viktor Stolc, Thomas E Royce, Joel S Rozowsky, Alexander E Urban, Xiaowei Zhu, John L Rinn, Waraporn Tongprasit, Manoj Samanta, Sherman Weissman, et al. Global identification of human transcribed sequences with genome tiling arrays. *Science*, 306(5705):2242–2246, 2004.

[60] Viktor Stolc, Manoj Pratim Samanta, Waraporn Tongprasit, Himanshu Sethi, Shoudan Liang, David C Nelson, Adrian Hegeman, Clark Nelson, David Rancour, and Sebastian Bednarek. Identification of transcribed sequences in Arabidopsis thaliana by using high-resolution genome tiling arrays. *Proceedings of the National Academy of Sciences of the United States of America*, 102(12):4453–4458, 2005.

[61] Lior David, Wolfgang Huber, Marina Granovskaia, Joern Toedling, Curtis J. Palm, Lee Bofkin, Ted Jones, Ronald W. Davis, and Lars M. Steinmetz. A high-resolution map of transcription in the yeast genome. *Proceedings of the National Academy of Sciences*, 103(14):5320–5325, 2006.

[62] Zhong Wang, Mark Gerstein, and Michael Snyder. RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, 10(1):57–63, 2009.

[63] Antoine M Snijders, Norma Nowak, Richard Segraves, Stephanie Blackwood, Nils Brown, Jeffrey Conroy, Greg Hamilton, Anna Katherine Hindle, Bing Huey, Karen Kimura, et al. Assembly of microarrays for genome-wide measurement of DNA copy number. *Nature genetics*, 29:263–264, 2001.

[64] Peter H Sudmant, Jacob O Kitzman, Francesca Antonacci, Can Alkan, Maika Malig, Anya Tsalenko, Nick Sampas, Laurakay Bruhn, Jay Shendure, Evan E Eichler, et al. Diversity of human copy number variation and multicopy genes. *Science*, 330(6004): 641–646, 2010.

[65] Jane Fridlyand, Antoine M. Snijders, Dan Pinkel, Donna G. Albertson, and Ajay N. Jain. Hidden Markov models approach to the analysis of array CGH data. *Journal of Multivariate Analysis*, 90(1):132 – 153, 2004. Special Issue on Multivariate Methods in Genomic Data Analysis.

[66] Philippe Hupé, Nicolas Stransky, Jean-Paul Thiery, François Radvanyi, and Emmanuel Barillot. Analysis of array CGH data: from signal ratio to gain and loss of DNA regions. *Bioinformatics*, 20(18):3413–3422, 2004.

[67] Adam B. Olshen, E. S. Venkatraman, Robert Lucito, and Michael Wigler. Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics*, 5(4):557–572, 2004.

[68] Franck Picard, Stephane Robin, Marc Lavielle, Christian Vaisse, and Jean-Jacques Daudin. A statistical approach for array CGH data analysis. *BMC Bioinformatics*, 6(1):27, 2005.

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

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

[71] Erez Ben-Yaacov and Yonina C. Eldar. A fast and flexible method for the segmentation of aCGH data. *Bioinformatics*, 24(16): i139–i145, 2008.

[72] Chandra Erdman and John W. Emerson. A fast Bayesian change point analysis for the segmentation of microarray data. *Bioinformatics*, 24(19):2143–2148, 2008.

[73] Franck Picard, Emilie Lebarbier, Mark Hoebeke, Guillem Rigaill, Baba Thiam, and Stéphane Robin. Joint segmentation, calling, and normalization of multiple CGH profiles. *Biostatistics*, 12(3):413–428, 2011.

[74] Jiarui Ding and S. P. Shah. Robust hidden semi-Markov modeling of array CGH data. In *Bioinformatics and Biomedicine (BIBM), 2010 IEEE International Conference on*, pages 603–608, 2010.

[75] Stefano Colella, Christopher Yau, Jennifer M. Taylor, Ghazala Mirza, Helen Butler, Penny Clouston, Anne S. Bassett, Anneke Seller, Christopher C. Holmes, and Jiannis Ragoussis. QuantiSNP: an Objective Bayes Hidden-Markov Model to detect and accurately map copy number variation using SNP genotyping data. *Nucleic Acids Research*, 35(6):2013–2025, 2007.

[76] Kai Wang, Mingyao Li, Dexter Hadley, Rui Liu, Joseph Glessner, Struan F.A. Grant, Hakon Hakonarson, and Maja Bucan. PennCNV: An integrated hidden Markov model designed for high-resolution copy number variation detection in whole-genome SNP genotyping data. *Genome research*, 17(11):1665–1674, 2007.

[77] Tatiana Popova, Elodie Manie, Dominique Stoppa-Lyonnet, Guillem Rigaill, Emmanuel Barillot, and Marc Stern. Genome Alteration Print (GAP): a tool to visualize and mine complex cancer genomic profiles obtained by SNP arrays. *Genome Biology*, 10(11):R128, 2009.

[78] Rameen Beroukhim, Ming Lin, Yuhyun Park, Ke Hao, Xiaojun Zhao, Levi A. Garraway, Edward A. Fox, Ephraim P. Hochberg, Ingo K. Mellinghoff, Matthias D. Hofer, Aurelien Descazeaud, Mark A. Rubin, Matthew Meyerson, Wing Hung Wong, William R. Sellers, and Cheng Li. Inferring Loss-of-Heterozygosity from Unpaired Tumors Using High-Density Oligonucleotide SNP Arrays. *PLoS Comput Biol*, 2(5):e41, 2006.

[79] Robert B Scharpf, Giovanni Parmigiani, Jonathan Pevsner, and Ingo Ruczinski. Hidden markov models for the assessment of chromosomal alterations using high-throughput snp arrays. *The annals of applied statistics*, 2(2):687, 2008.

[80] Antonio Piccolboni. Multivariate segmentation in the analysis of transcription tiling array data. *Journal of Computational Biology*, 15(7):845–856, 2008.

[81] Wolfgang Huber, Joern Toedling, and Lars M. Steinmetz. Transcript mapping with high-density oligonucleotide tiling arrays. *Bioinformatics*, 22(16):1963–1970, 2006.

[82] Jiang Du, Joel S. Rozowsky, Jan O. Korbel, Zhengdong D. Zhang, Thomas E. Royce, Martin H. Schultz, Michael Snyder, and Mark Gerstein. A supervised hidden markov model framework for efficiently segmenting tiling array data in transcriptional and chIP-chip experiments: systematically incorporating validated biological knowledge. *Bioinformatics*, 22(24):3016–3024, 2006.

[83] Joern Toedling, Oleg Sklyar, and Wolfgang Huber. Ringo - an R/Bioconductor package for analyzing ChIP-chip readouts. *BMC Bioinformatics*, 8(1):221, 2007.

[84] Yong Zhang, Tao Liu, Clifford Meyer, Jerome Eeckhoute, David Johnson, Bradley Bernstein, Chad Nusbaum, Richard Myers, Myles Brown, Wei Li, and X Shirley Liu. Model-based Analysis of ChIP-Seq (MACS). *Genome Biology*, 9(9):R137, 2008.

[85] Christiana Spyrou, Rory Stark, Andy Lynch, and Simon Tavare. BayesPeak: Bayesian analysis of ChIP-seq data. *BMC Bioinformatics*, 10(1):299, 2009.

[86] Zhaohui Qin, Jianjun Yu, Jincheng Shen, Christopher Maher, Ming Hu, Shanker Kalyana-Sundaram, Jindan Yu, and Arul Chinnaiyan. HPeak: an HMM-based algorithm for defining read-enriched regions in ChIP-Seq data. *BMC Bioinformatics*, 11(1):369, 2010.

[87] Thomas J. Hardcastle, Krystyna A. Kelly, and David C. Baulcombe. Identifying small interfering RNA loci from high-throughput sequencing data. *Bioinformatics*, 28(4):457–463, 2012. doi: 10.1093/bioinformatics/btr687.

[88] Jason Ernst and Manolis Kellis. ChromHMM: automating chromatin-state discovery and characterization. *Nature Methods*, 9(3): 215–216, 2012.

[89] Michael M Hoffman, Orion J Buske, Jie Wang, Zhiping Weng, Jeff A Bilmes, and William Stafford Noble. Unsupervised pattern discovery in human chromatin structure through genomic segmentation. *Nature Methods*, 9(5):473–476, 2012.

[90] Chris Burge and Samuel Karlin. Prediction of complete gene structures in human genomic DNA. *Journal of molecular biology*, 268 (1):78–94, 1997.

[91] Leonard Baum and Ted Petrie. Statistical Inference for Probabilistic Functions of Finite State Markov Chains. *The Annals of Mathematical Statistics*, 37(6):1554–1563, 1966.

[92] J. Baker. The DRAGON system–An overview. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 23(1):24–29, 1975.

[93] F. Jelinek, L. Bahl, and R. Mercer. Design of a linguistic statistical decoder for the recognition of continuous speech. *Information Theory, IEEE Transactions on*, 21(3):250–256, 1975.

[94] Richard Durbin, Sean Eddy, Anders Krogh, and Graeme Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

[95] Jack D. Ferguson. Variable duration models for speech. pages 143–179, 1980.

[96] Y. Guédon, D. Barthélémy, Y. Caraglio, and E. Costes. Pattern Analysis in Branching and Axillary Flowering Sequences. *Journal of Theoretical Biology*, 212(4):481–520, 2001.

[97] Zafer Aydin, Yucel Altunbasak, and Mark Borodovsky. Protein secondary structure prediction for a single-sequence using hidden semi-Markov models. *BMC Bioinformatics*, 7(1):178, 2006.

[98] Shun-Zheng Yu. Hidden semi-Markov models. *Artificial Intelligence*, 174(2):215–243, 2010.

[99] Yann Guédon. Estimating Hidden Semi-Markov Chains from Discrete Sequences. *Journal of Computational and Graphical Statistics*, 12(3):604–639, 2003. ISSN 10618600. doi: 10.2307/1391041.

[100] Michael Lawrence, Wolfgang Huber, Hervé Pagés, Patrick Aboyoun, Marc Carlson, Robert Gentleman, Martin T. Morgan, and Vincent J. Carey. Software for Computing and Annotating Genomic Ranges. *PLoS Comput Biol*, 9(8):e1003118, 2013.

[101] Weil R. Lai, Mark D. Johnson, Raju Kucherlapati, and Peter J. Park. Comparative analysis of algorithms for identifying amplifications and deletions in array CGH data. *Bioinformatics*, 21(19):3763–3770, 2005.

[102] Nancy R. Zhang and David O. Siegmund. A Modified Bayes Information Criterion with Applications to the Analysis of Comparative Genomic Hybridization Data. *Biometrics*, 63(1):22–32, 2007.

[103] Daniel Barry and John A Hartigan. A Bayesian analysis for change point problems. *Journal of the American Statistical Association*, 88(421):309–319, 1993.

[104] PawełStankiewicz and James R. Lupski. Structural Variation in the Human Genome and its Role in Disease. *Annual Review of Medicine*, 61(1):437–455, 2010. doi: 10.1146/annurev-med-100708-204735. PMID: 20059347.

[105] Vince Carey and Henning Redestig. *ROC: utilities for ROC, with uarray focus*, 2013. R package version 1.36.0.

[106] Matthew J. Hangauer, Ian W. Vaughn, and Michael T. McManus. Pervasive Transcription of the Human Genome Produces Thousands of Previously Unidentified Long Intergenic Noncoding RNAs. *PLoS Genet*, 9(6):e1003569, 06 2013. doi: 10.1371/journal.pgen.1003569.

[107] The ENCODE Project Consortium. The ENCODE (ENCyclopedia of DNA elements) project. *Science*, 306(5696):636–640, 2004.

[108] The ENCODE Project Consortium. A User's Guide to the Encyclopedia of DNA Elements (ENCODE). *PLoS Biol*, 9(4):e1001046, 04 2011. doi: 10.1371/journal.pbio.1001046.

[109] En Li, Timothy H. Bestor, and Rudolf Jaenisch. Targeted mutation of the DNA methyltransferase gene results in embryonic lethality. *Cell*, 69(6):915–926, 1992.

[110] Wolf Reik, Wendy Dean, and Jörn Walter. Epigenetic Reprogramming in Mammalian Development. *Science*, 293(5532):1089–1093, 2001. doi: 10.1126/science.1063443.

[111] Rafael A. Irizarry, Christine Ladd-Acosta, Bo Wen, Zhijin Wu, Carolina Montano, Patrick Onyango, Hengmi Cui, Kevin Gabo, Michael Rongione, Maree Webster, Hong Ji, James B. Potash, Sarven Sabunciyan, and Andrew P. Feinberg. The human colon cancer methylome shows similar hypo- and hypermethylation at conserved tissue-specific CpG island shores. *Nat Genet*, 41(2): 178–186, 2009. doi: 10.1038/ng.298.

[112] Xu Zhang, Shinhan Shiu, Andrew Cal, and Justin O. Borevitz. Global Analysis of Genetic, Epigenetic and Transcriptional Polymorphisms in *Arabidopsis thaliana* Using Whole Genome Tiling Arrays. *PLoS Genet*, 4(3):e1000032, 2008.

[113] Shawn J. Cokus, Suhua Feng, Xiaoyu Zhang, Zugen Chen, Barry Merriman, Christian D. Haudenschild, Sriharsa Pradhan, Stanley F. Nelson, Matteo Pellegrini, and Steven E. Jacobsen. Shotgun bisulphite sequencing of the Arabidopsis genome reveals DNA methylation patterning. *Nature*, 452(7184):215–219, 2008. doi: 10.1038/nature06745.

[114] Katja Hebestreit, Martin Dugas, and Hans-Ulrich Klein. Detection of Significantly Differentially Methylated Regions in Targeted Bisulfite Sequencing Data. *Bioinformatics*, 2013.

[115] Till Schoofs, Christian Rohde, Katja Hebestreit, Hans-Ulrich Klein, Stefanie Göllner, Isabell Schulze, Mads Lerdrup, Nikolaj Dietrich, Shuchi Agrawal-Singh, Anika Witten, Monika Stoll, Eva Lengfelder, Wolf-Karsten Hofmann, Peter Schlenke, Thomas Büchner, Klaus Hansen, Wolfgang E. Berdel, Frank Rosenbauer, Martin Dugas, and Carsten Müller-Tidow. DNA methylation changes are a late event in acute promyelocytic leukemia and coincide with loss of transcription factor binding. *Blood*, 121(1): 178–187, 2013.

[116] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009.

[117] Karl. O. Honikel. *How to Measure the Water-Holding Capacity of Meat? Recommendation of Standardized Methods*, volume 38 of *Current Topics in Veterinary Medicine and Animal Science*, chapter 11, pages 129–142. Springer Netherlands, 1987.

[118] Karl. O. Honikel. *Water-holding Capacity of Meat*, chapter 18, page 389. CABI, 2004.

[119] G. Liu, D. G. J. Jennen, E. Tholen, H. Juengst, T. Kleinwächter, M. Hölker, D. Tesfaye, G. Ün, H. J. Schreinemachers, E. Murani, S. Ponsuksili, J. J. Kim, K. Schellander, and K. Wimmers. A genome scan reveals QTL for growth, fatness, leanness and meat quality in a Duroc-Pietrain resource population. *Animal Genetics*, 38(3):241–252, 2007.

[120] Guisheng Liu, Jong Kim, Elisebeth Jonas, Klaus Wimmers, Siriluck Ponsuksili, Eduard Murani, Chirawath Phatsara, Ernst Tholen, Heinz Juengst, Dawit Tesfaye, Ji Chen, and Karl Schellander. Combined line-cross and half-sib qtl analysis in duroc—pietrain population. *Mammalian Genome*, 19(6):429–438, 2008.

[121] Klaus Wimmers, Eduard Murani, and Siriluck Ponsuksili. Pre-and postnatal differential gene expression with relevance for meat and carcass traits in pigsâĂŞa review. *Anim Sci Pap Rep*, 28:115–122, 2010.

[122] Siriluck Ponsuksili, Elisabeth Jonas, Eduard Murani, Chirawath Phatsara, Tiranun Srikanchai, Christina Walz, Manfred Schwerin, Karl Schellander, and Klaus Wimmers. Trait correlated expression combined with expression QTL analysis reveals biological pathways and candidate genes affecting water holding capacity of muscle. *BMC Genomics*, 9(1):367, 2008.

[123] Siriluck Ponsuksili, Eduard Murani, Manfred Schwerin, Karl Schellander, and Klaus Wimmers. Identification of expression QTL (eQTL) of genes expressed in porcine M. longissimus dorsi and associated with meat quality traits. *BMC genomics*, 11(1):572, 2010.

[124] J. C. Barrett, B. Fry, J. Maller, and M. J. Daly. Haploview: analysis and visualization of LD and haplotype maps. *Bioinformatics*, 21(2):263–265, 2005.

[125] Gordon K Smyth. Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Stat Appl Genet Mol Biol*, 3(1):3, 2004.

[126] Wolfgang Huber, Anja von Heydebreck, Holger Sültmann, Annemarie Poustka, and Martin Vingron. Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, 18(suppl 1):S96–S104, 2002.

[127] W. Evan Johnson, Cheng Li, and Ariel Rabinovic. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics*, 8(1):118–127, 2007.

[128] Audrey Kauffmann, Robert Gentleman, and Wolfgang Huber. arrayQualityMetrics–a bioconductor package for quality assessment of microarray data. *Bioinformatics*, 25(3):415–6, 2009.

[129] Siriluck Ponsuksili, Eduard Murani, Chirawath Phatsara, Elisabeth Jonas, Christina Walz, Manfred Schwerin, Karl Schellander, and Klaus Wimmers. Expression Profiling of Muscle Reveals Transcripts Differentially Expressed in Muscle That Affect Water-Holding Capacity of Pork. *Journal of Agricultural and Food Chemistry*, 56(21):10311–10317, 2008.

[130] Siriluck Ponsuksili, **Yang Du**, Eduard Murani, Manfred Schwerin, and Klaus Wimmers. Elucidating molecular networks that either affect or respond to plasma cortisol concentration in target tissues of liver and muscle. *Genetics*, 192(3):1109–1122, 2012. ISSN 0016-6731. doi: 10.1534/genetics.112.143081.

[131] Siriluck Ponsuksili, **Yang Du**, Frieder Hadlich, Puntita Siengdee, Eduard Murani, Manfred Schwerin, and Klaus Wimmers. Correlated mRNAs and miRNAs from co-expression and regulatory networks affect porcine muscle and finally meat properties. *BMC Genomics*, 14(1):533, 2013. ISSN 1471-2164. doi: 10.1186/1471-2164-14-533.

[132] Arek Kasprzyk. BioMart: driving a paradigm change in biological data management. *Database*, 2011, 2011.

[133] S. Falcon and R. Gentleman. Using GOstats to test gene lists for GO term association. *Bioinformatics*, 23(2):257–258, 2007.

[134] Jaeseok Han, Sung Hoon Back, Junguk Hur, Yu-Hsuan Lin, Robert Gildersleeve, Jixiu Shan, Celvie L. Yuan, Dawid Krokowski, Shiyu Wang, Maria Hatzoglou, Michael S. Kilberg, Maureen A. Sartor, and Randal J. Kaufman. ER-stress-induced transcriptional regulation increases protein synthesis leading to cell death. *Nat Cell Biol*, 15(5):481–490, 2013.

[135] Hiroaki Kawasaki, Richard Eckner, Tso-Pang Yao, Kazunari Taira, Robert Chiu, David M. Livingston, and Kazunari K. Yokoyama. Distinct roles of the co-activators p300 and CBP in retinoic-acid-induced F9-cell differentiation. *Nature*, 393(6682):284–289, 1998. 10.1038/30538.

[136] Tso-Pang Yao, Suk P. Oh, Miriam Fuchs, Nai-Dong Zhou, Lian-Ee Ch'ng, David Newsome, Roderick T. Bronson, En Li, David M. Livingston, and Richard Eckner. Gene Dosage Dependent Embryonic Development and Proliferation Defects in Mice Lacking the Transcriptional Integrator p300. *Cell*, 93(3):361–372, 1998.

[137] Cole Trapnell, Lior Pachter, and Steven L. Salzberg. TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics*, 25(9): 1105–1111, 2009.

[138] Ben Langmead, Cole Trapnell, Mihai Pop, and Steven Salzberg. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology*, 10(3):R25, 2009.

[139] Ian Korf. Gene finding in novel genomes. *BMC Bioinformatics*, 5(1):59, 2004.

[140] Mark D. Robinson, Davis J. McCarthy, and Gordon K. Smyth. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140, 2010.

[141] Simon Anders and Wolfgang Huber. Differential expression analysis for sequence count data. *Genome Biology*, 11(10):R106, 2010.

[142] Yoav Benjamini and Yosef Hochberg. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995.

[143] Tiranun Srikanchai, Eduard Murani, Klaus Wimmers, and Siriluck Ponsuksili. Four loci differentially expressed in muscle tissue depending on water-holding capacity are associated with meat quality in commercial pig herds. *Molecular Biology Reports*, 37(1): 595–601, 2010.

[144] A. M. Ramos, R. H. Pita, M. Malek, P. S. Lopes, S. E. F. GuimarÃčes, and M. F. Rothschild. Analysis of the mouse high-growth region in pigs. *Journal of Animal Breeding and Genetics*, 126(5):404–412, 2009.

[145] H. Thomsen, H. K. Lee, M. F. Rothschild, M. Malek, and J. C. M. Dekkers. Characterization of quantitative trait loci for growth and meat quality in a cross between commercial breeds of swine. *Journal of Animal Science*, 82(8):2213–2228, 2004.

[146] Daniela Lazzaretti, Isabelle Tournier, and Elisa Izaurralde. The C-terminal domains of human TNRC6A, TNRC6B, and TNRC6C silence bound transcripts independently of Argonaute proteins. *RNA*, 15(6):1059–1066, 2009. 10.1261/rna.1606309.

[147] The UniProt Consortium. Activities at the Universal Protein Resource (UniProt). *Nucleic Acids Research*, 42(D1):D191–D198, 2014.

[148] Harsh Dweep, Carsten Sticht, Priyanka Pandey, and Norbert Gretz. miRWalk — Database: Prediction of possible miRNA binding sites by "walking" the genes of three genomes. *Journal of Biomedical Informatics*, 44(5):839–847, 2011.

[149] Jian Ye, George Coulouris, Irena Zaretskaya, Ioana Cutcutache, Steve Rozen, and Thomas Madden. Primer-BLAST: A tool to design target-specific primers for polymerase chain reaction. *BMC Bioinformatics*, 13(1):134, 2012.

[150] Kenneth J. Livak and Thomas D. Schmittgen. Analysis of Relative Gene Expression Data Using Real-Time Quantitative PCR and the $2^{-\Delta\Delta C_T}$ Method. *Methods*, 25(4):402–408, 2001.

[151] Elzbieta Sliwerska, Fan Meng, Terence P. Speed, Edward G. Jones, William E. Bunney, Huda Akil, Stanley J. Watson, and Margit Burmeister. Snps on chips: The hidden genetic code in expression arrays. *Biological Psychiatry*, 61(1):13–16, 2007.

[152] David Benovoy, Tony Kwan, and Jacek Majewski. Effect of polymorphisms within probe—target sequences on olignonucleotide microarray experiments. *Nucleic Acids Research*, 36(13):4417–4423, 2008.

[153] Eric R. Gamazon, Wei Zhang, M. Eileen Dolan, and Nancy J. Cox. Comprehensive survey of snps in the affymetrix exon array using the 1000 genomes dataset. *PLoS ONE*, 5(2):e9366, 2010.

[154] S. T. Sherry, M. H. Ward, M. Kholodov, J. Baker, L. Phan, E. M. Smigielski, and K. Sirotkin. dbSNP: the NCBI database of genetic variation. *Nucleic Acids Research*, 29(1):308–311, 2001. 10.1093/nar/29.1.308.

[155] Rafael A. Irizarry, Bridget Hobbs, Francois Collin, Yasmin D. BeazerâĂŘBarclay, Kristen J. Antonellis, Uwe Scherf, and Terence P. Speed. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4(2):249–264, 2003.

[156] Monnie McGee and Zhongxue Chen. Parameter Estimation for the Exponential-Normal Convolution Model for Background Correction of Affymetrix GeneChip Data. *Statistical Applications in Genetics and Molecular Biology*, 5(1):1–25, 2006.

[157] Ashish Agarwal, David Koppstein, Joel Rozowsky, Andrea Sboner, Lukas Habegger, LaDeana Hillier, Rajkumar Sasidharan, Valerie Reinke, Robert Waterston, and Mark Gerstein. Comparison and calibration of transcriptome data from RNA-Seq and tiling arrays. *BMC genomics*, 11(1):383, 2010.

[158] Marcel Margulies, Michael Egholm, William E. Altman, Said Attiya, Joel S. Bader, Lisa A. Bemben, Jan Berka, Michael S. Braverman, Yi-Ju Chen, Zhoutao Chen, Scott B. Dewell, Lei Du, Joseph M. Fierro, Xavier V. Gomes, Brian C. Godwin, Wen He, Scott Helgesen, Chun He Ho, Gerard P. Irzyk, Szilveszter C. Jando, Maria L. I. Alenquer, Thomas P. Jarvie, Kshama B. Jirage, Jong-Bum Kim, James R. Knight, Janna R. Lanza, John H. Leamon, Steven M. Lefkowitz, Ming Lei, Jing Li, Kenton L. Lohman, Hong Lu, Vinod B. Makhijani, Keith E. McDade, Michael P. McKenna, Eugene W. Myers, Elizabeth Nickerson, John R. Nobile, Ramona Plant, Bernard P. Puc, Michael T. Ronan, George T. Roth, Gary J. Sarkis, Jan Fredrik Simons, John W. Simpson, Maithreyan Srinivasan, Karrie R. Tartaro, Alexander Tomasz, Kari A. Vogt, Greg A. Volkmer, Shally H. Wang, Yong Wang, Michael P. Weiner, Pengguang Yu, Richard F. Begley, and Jonathan M. Rothberg. Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, 437(7057):376–380, 2005. 10.1038/nature03959.

[159] Thomas J. Albert, Michael N. Molla, Donna M. Muzny, Lynne Nazareth, David Wheeler, Xingzhi Song, Todd A. Richmond, Chris M. Middle, Matthew J. Rodesch, Charles J. Packard, George M. Weinstock, and Richard A. Gibbs. Direct selection of human genomic loci by microarray hybridization. *Nat Meth*, 4(11):903–905, 2007. 10.1038/nmeth1111.

[160] David T. Okou, Karyn Meltz Steinberg, Christina Middle, David J. Cutler, Thomas J. Albert, and Michael E. Zwick. Microarray-based genomic selection for high-throughput resequencing. *Nat Meth*, 4(11):907–909, 2007. 10.1038/nmeth1109.

[161] Gregory J. Porreca, Kun Zhang, Jin Billy Li, Bin Xie, Derek Austin, Sara L. Vassallo, Emily M. LeProust, Bill J. Peck, Christopher J. Emig, Fredrik Dahl, Yuan Gao, George M. Church, and Jay Shendure. Multiplex amplification of large sets of human exons. *Nat Meth*, 4(11):931–936, 2007. 10.1038/nmeth1110.

[162] Sunitha Kogenaru, Qing Yan, Yinping Guo, and Nian Wang. RNA-seq and microarray complement each other in transcriptome profiling. *BMC genomics*, 13(1):629, 2012.

[163] The ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414): 57–74, 2012. 10.1038/nature11247.

APPENDIX A

R code of using BWT and suffix array to perform backward search

```
# calculate the BWT
bwt<-function(x, ins='$'){
  s<-unlist(strsplit(x, ''))
  if(ins %in% s)
    stop("can't have 'ins' in the string!")
  # append $
  s<-c(s, ins)
  n<-length(s)
  # create matrix of rotation
  tab<-unname(t(cbind(s,
      sapply(2:n, function(x) c(s[x:n], s[1:(x-1)]))
    )))
  # sort and return
  tab<-tab[do.call(order, lapply(1:n, function(i) tab[,i])),]
  return(paste(tab[,n], collapse=''))
}

# calculate the inverse of BWT
ibwt<-function(t, ins='$'){
  s<-unlist(strsplit(t, ''))
  if(table(s)[ins]!=1)
    stop("occurance of 'ins' doesnot equal to 1!")
  n<-length(s)
  tab<-matrix('', n, n)
  for(i in n:1){
    # insert column
    tab[,i]<-s
    # sort row
    tab<-tab[do.call(order, lapply(1:n, function(i) tab[,i])),]
```

A. R code of using BWT and suffix array to perform backward search

```
  }
  # return row ends with $
  return(paste(tab[which(tab[,n]==ins),-n], collapse=''))
}

# create suffix array as a data.frame
suffixArray<-function(x, ins='$'){
  x<-unlist(strsplit(x, ''))
  if(ins %in% x)
    stop("can't have 'ins' in the string!")
  x<-c(x, ins)
  n<-length(x)
  # create suffixes
  suf<-sapply(1:n, function(i) {
    paste(x[i:n], collapse='')
  })
  #sort
  suf<-data.frame(S=suf[order(suf)], i=order(suf),
    stringsAsFactors=F)
  return(suf)
}

# total number of char which are alphabetically smaller than c
cMat<-function(x, ins='$'){
  s<-unlist(strsplit(x, ''))
  if(ins %in% s)
    stop("can't have ins in the string!")
  s<-c(s, ins)
  n<-length(s)
  tab<-table(s)
  c<-cumsum(tab)-as.integer(tab)
  return(c)
}

# the matrix gives occurrences of character c in the BWT prefix.
occMat<-function(t, ins='$'){
  t<-unlist(strsplit(t, ''))
  if(table(t)[ins]!=1)
    stop("occurance of 'ins' doesnot equal to 1!")
  n<-length(t)
  odrchr<-names(table(t))
  occ<-sapply(odrchr, function(c) cumsum(t==c))
  return(occ)
}

# search pattern p in x
BackwardSearch<-function(P, S, ins='$') {
```

```
  P<-unlist(strsplit(P, ''))
  if(ins %in% P)
    stop("can't have 'ins' in the string!")
  S<-unlist(strsplit(S, ''))
  if(!all(P %in% S))
    return("pattern contains char not found in x!")
  T<-bwt(S, ins=ins)
  Occ<-occMat(T, ins=ins)
  C<-cMat(S, ins=ins)
  p<-length(P)
  c<-P[p]
  s<-C[c]+1
  e<-C[c]+Occ[nrow(Occ), c]
  while(s<=e && p>=2){
    c<-P[p-1]
    s<-C[c]+Occ[s-1, c]+1
    e<-C[c]+Occ[e, c]
    p<-p-1
  }
  if (e<s)
    return('no occurrence!')
  else
    return(unname(c(s, e)))
}

MUP<-function(P, S, ins='$'){
  P<-unlist(strsplit(P, ''))
  if(ins %in% P)
    stop("can't have 'ins' in the string!")
  S<-unlist(strsplit(S, ''))
  if(!all(P %in% S))
    return("pattern contains char not found in x!")
  T<-bwt(S, ins=ins)
  Occ<-occMat(T, ins=ins)
  C<-cMat(S, ins=ins)
  p<-length(P)
  n<-length(S)
  mup<-rep(0, p)
  pp<-data.frame(matrix(NA, p, p+1))
  colnames(pp)<-c('MUP.length', S)
  for( i in seq_len(p)){
    s<-1+1
    e<-n+1
    j<-i
    while(j<=p && s<e){
      c<-P[j]
      s<-C[c]+Occ[s-1, c]+1
```

A. R code of using BWT and suffix array to perform backward search

```
        e<-C[c]+Occ[e, c]
        j<-j+1
      }
      if (e <= s)    {
        mup[i]<-  j - i
      }
      pp[i, ]<-c(mup[i], rep('␣', i-1),
        if(mup[i]>0) S[i:(i+mup[i]-1)] else '-',
        rep('␣', p-i-mup[i]+1))
    }
    print(pp)
    return(mup)
}
```

APPENDIX B

Agilent Human Whole Genome ChIP-on-Chip Set 244K design ID

https://earray.chem.agilent.com/earray/

ADID16060
ADID16063
ADID16066
ADID16068
ADID16069
ADID16070
ADID16071
ADID16072
ADID16073
ADID16074
ADID16075
ADID16076
ADID16138
ADID16077
ADID16139
ADID16140
ADID16141
ADID16142
ADID16143
ADID16144
ADID16145

B. Agilent Human Whole Genome ChIP-on-Chip Set 244K design ID

ADID16146
ADID16147
ADID16148
ADID16149

APPENDIX C

Pseudo code of tiling probe selection algorithm

```
start from the 5'
while still enough space to place a probe
  if probe temperature is too high
    if all position to the 5' has been checked
      jump to last checked position  to 3', shift 1 nt to 3' and re-check
    else if probe length can be shorter
      probe length minus 1 and re-check
    else if  it is ok to shift to 5'
      shift 1 nt to 5' and re-check
  else if probe temperature is too low or not unique enough
    if all possible positions to the 5' has been checked
      jump to last checked position  to 3', shift 1 nt to 3' and re-check
    else if probe length can be shorter and ok to shift to 5'
      probe length plus 1, shift 1 nt to 5' and re-check
    else if it is ok to shift to 5'
      shift 1 nt to 5' and re-check
  if other remaining checks failed
    if all position to the 5' has been checked
      jump to last checked position to 3', shift 1 nt to 3' and re-check
    else if probe length can be shorter and ok to shift to 5'
      probe length plus 1, shift 1 nt to 5' and re-check
    else if it is ok to shift to 5'
      shift 1 nt to 5' and re-check
```

```
  tile found
  move to next candidate position
end loop
```

APPENDIX D

R code of segmentation data simulation and benchmarking

```
################################################
# helper functions for the simulation
################################################
reportROC<-function(roc, er=1, J=3, nstep=100,
  CorS='S', toPlot=T, main='SNR', pcex=2)
{
  #CorS, cgh or sequencing
  if(CorS=='S'){
    srs<-seq(-1, er^(J-1)+2, length.out=nstep)
  } else {
    srs<-seq(-1, er*(J-1)+2, length.out=nstep)
  }
  nsim<-nrow(roc[[1]][['t']])
  ngrid<-length(roc)
  methods<-names(roc[[1]][['t']])
  # get run time
  cat('algorithm names : ', methods,'\n', sep='\t')
  avgt<-colMeans( do.call(rbind, do.call(rbind, roc)[,'t']))
  cat('average run time: ', avgt,'\n', sep='\t')
  # get summary of break point no.
  print(apply(do.call(rbind, do.call(rbind, roc)[,'ncp']), 2, summary))
  # get error estimate
  segmse<-apply(
    do.call(rbind, lapply(roc, function(ll) ll$ncp-length(ll$L))),
    2, function(x) sum(x^2)/nsim/ngrid)
  cat('MSE for no. segs: ', segmse,'\n', sep='\t')
  segame<-apply(
    do.call(rbind, lapply(roc, function(ll) ll$ncp-length(ll$L))),
    2, function(x) sum(abs(x))/nsim/ngrid)
```

D. R code of segmentation data simulation and benchmarking

```
cat('AME␣for␣no.␣segs:␣', segame,'\n', sep='\t')
# calc pos and true pos
rmethods<-names(roc[[1]][['A']][[1]])
tp<-Reduce('+', lapply(roc,
  function(ll) sapply(rmethods,
    function(m) sapply(seq_len(nstep),
      function(i) sum(sapply(seq_len(nsim),
        function(n) ll$A[[n]][,m]>srs[i] & rep(ll$S, times=ll$L)==3))))))
p<-Reduce('+', lapply(roc,
  function(ll) sapply(rmethods,
    function(m) sapply(seq_len(nstep),
      function(i) sum(sapply(seq_len(nsim),
        function(n) ll$A[[n]][,m]>srs[i]))))))
if(CorS=='C'){
  ltp<-Reduce('+', lapply(roc,
    function(ll) sapply(rmethods,
      function(m) sapply(seq_len(nstep),
        function(i) sum(sapply(seq_len(nsim),
          function(n) ll$A[[n]][,m]<srs[i] & rep(ll$S, times=ll$L)==1))))))
  lp<-Reduce('+', lapply(roc,
    function(ll) sapply(rmethods,
      function(m) sapply(seq_len(nstep),
        function(i) sum(sapply(seq_len(nsim),
          function(n) ll$A[[n]][,m]<srs[i]))))))
}
#get AUC stat
cat('\nalgorithm␣runned:␣', rmethods,'\n', sep='\t')
aucg<-sapply(rmethods, function(m) callAUC(
  tpr=tp[,m]/sum(sapply(roc, function(ll) sum(ll$L[ll$S=='3']))/nsim,
  fpr=(p[,m]-tp[,m])/sum(sapply(roc,function(ll) sum(ll$L[ll$S!='3']))/nsim)
)
cat('AUC␣for␣gain␣/␣3:␣', aucg, '\n',sep='\t')
if(CorS=='C'){
  aucl<-sapply(rmethods, function(m) callAUC(
    tpr=ltp[,m]/sum(sapply(roc, function(ll) sum(ll$L[ll$S=='1']))/nsim,
    fpr=(lp[,m]-ltp[,m])/sum(sapply(roc,function(ll) sum(ll$L[ll$S!='1']))/
      nsim)
  )
  cat('AUC␣for␣loss␣/␣1:␣', aucl, '\n',sep='\t')
}
# order names for ploting
rmethods<-sort(rmethods)
if(CorS=='C'){
  rmethods<-rmethods[c(1, 3:8, 2)]
}
if(toPlot){
  # seperate gain and loss
```

```
    colors<-palette()[1:8]; colors[7]<-'orange'
    # plot background
    plot(c(0,0), c(1,1), col='white', xlab='FPR', ylab='TPR',
      xlim=c(0,1), ylim=c(0,1), main=main, cex.lab=pcex,
      cex.axis=pcex, cex.main=pcex, cex.sub=pcex)
    for(i in seq_along(rmethods)){
      m<-rmethods[i]
      points(
        (p[,m]-tp[,m])/sum(sapply(roc, function(l) sum(l$L[l$S!='3']))/nsim,
        tp[,m]/sum(sapply(roc, function(l) sum(l$L[l$S=='3']))/nsim,
        col=colors[i], cex=pcex, pch=17, type='b')
      if(CorS=='C'){
        points(
          (lp[,m]-ltp[,m])/sum(sapply(roc, function(l) sum(l$L[l$S!='1']))/
            nsim,
          ltp[,m]/sum(sapply(roc, function(l) sum(l$L[l$S=='1']))/nsim,
          col=colors[i], cex=pcex, pch=6, type='b')
      }
    }
    rmethods[rmethods=='mine']<-'hsmm'
    gandl<-c('gain','loss')
    if(CorS=='C'){
      legend('bottomright', cex=3, ncol=2,
        pasteo(rep(rmethods, times=2),'.',rep(gandl, each=length(rmethods))),
        col=colors[rep(seq_along(rmethods), times=2)],
        pch=rep(c(17,6),each=length(rmethods)))
    } else {
      legend('bottomright', rmethods, col=colors[seq_along(rmethods)],
        pch=17, cex=pcex)
    }
  }
}

# create auc obj for test
callAUC<-function(tpr, fpr){
  rocobj<-new('rocc', sens=tpr, spec=1-fpr,
    caseLabel="case", markerLabel="marker")
  return(AUC(rocobj))
}

# sim univariate series
simUvSegDat<-function(n, j, param, seed=NULL){
  if(!is.null(seed)) set.seed(seed)
  x<-switch(param$type,
    norm = rnorm(n, mean=param$mean[j], sd=param$sd[j]),
    t = rt(n, ncp=param$ncp[j], df=param$df[j]),
    gamma = rgamma(n, shape=param$shape[j], scale=param$scale[j]),
```

```
    pois = rpois(n, lambda=param$lambda[j]),
    nbinom = rnbinom(n, mu=param$mu[j], size=param$size[j]))
}
# batch sim data and segment
simROCDataE<-function(J=3, nsim=4, ngrid=2, soj, emis, er,
  seed=832314, pool=20, toPlot=FALSE, bioHMM=FALSE){
  #check J to see if it conforms with soj and emis
  if(!is.null(soj)){
    if(! soj$type %in% c('gamma', 'pois', 'nbinom')){
      stop("soj type not supported")
    }
  }
  if(!is.null(emis)){
    if(! emis$type %in% c('norm', 't', 'pois','nbinom')){
      stop("emis type not supported")
    }
    paraLen<-sapply(emis, length)
    if(!all(paraLen[names(paraLen)!='type']==J)){
      stop("incorrect length for the emis parameter")
    }
  }
  # create pool of segment length for J states
  segLen<-sapply(seq_len(J), function(j) simUvSegDat(pool, j, soj, seed=seed))
  nnres<-lapply(seq_len(ngrid), function(g) {
    # draw segments, and pile up
    sel<-sample(seq_len(J), size=pool, replace=T)
    S<-runValue(Rle(sel))
    nsel<-length(S)
    L<-segLen[(S-1)*pool+1:nsel]
    true.cp<-cumsum(L)
    true.seg<-IRanges(start=c(1, true.cp[-(length(true.cp))]+1), end=true.cp)
    true.state<-rep(S, times=L)
    roc<-list(L=L, S=S, E=numeric(), pdf=character(), A=list(),
      ncp=data.frame(matrix(0, ncol=8, nrow=nsim)),
      t=data.frame(matrix(0, ncol=8, nrow=nsim))
    )
    colnames(roc[['t']])<-colnames(roc[['ncp']])<-
      c('bcp', 'biohmm', 'cbs', 'cghseg', 'glad', 'haarseg', 'hmm', 'mine')

    erf<-seq_len(J)-1
    if(emis$type == 'pois'){
      emis$lambda <-er^erf
    } else if (emis$type == 'norm'){
      emis$mean <- er*(erf+1)
      emis$sd <- rep(ifelse(er>=1, 1, er), J)
    } else if (emis$type == 't'){
      emis$ncp <- er*(erf+1)
```

```
  emis$df <- rep(ifelse(er>=1, 1, er), J)
} else {
  stop('emis$type not supported yet!')
}
for(n in seq_len(nsim)){
  E<-unlist(sapply(seq_along(S),
    function(i) simUvSegDat(L[i], S[i], emis)))
  roc$E<-cbind(roc$E, E)
  if(toPlot){
    filename<-pasteo('sim.', gsub(':', '-', date()) ,'.pdf')
    pdf(filename)
    ts.plot(E, type='p')
    dev.off()
    roc$pdf<-c(roc$pdf, filename)
  }
  xx<-as.matrix(E, ncol=1)
  ssxx<-ssxx

  ## biomvRhsmm
  mine.t<-system.time(
    mine.res<-biomvRhsmm(x=xx, maxk=min(500, nrow(xx)-1),
      emis.type=emis$type, soj.type='gamma', prior.m='quantile',
      q.alpha=0.05,r.var=0.75, avg.m='mean'))
  mine.res@res<-sort(mine.res@res)
  mine.cp<-end(mine.res@res)
  roc$t[n,'mine']<-mine.t[3]
  roc$ncp[n,'mine']<-length(mine.cp)
  rm(mine.t)

  ## bcp
  bcp.t<-system.time(bcp.res<-bcp(E))
  bcp.cp<-cumsum(runLength(Rle(bcp.res$posterior.mean)))
  roc$t[n,'bcp']<-bcp.t[3]
  roc$ncp[n,'bcp']<-length(bcp.cp)
  rm(bcp.t)

  ### CBS - DNAcopy
  cbs.obj<-CNA(xx, maploc=ssxx, chrom='sseq')
  cbs.t<-system.time(cbs.res<-DNAcopy::segment(cbs.obj))
  cbs.cp<-cbs.res$output$loc.end
  roc$t[n,'cbs']<-cbs.t[3]
  roc$ncp[n,'cbs']<-length(cbs.cp)
  rm(cbs.obj, cbs.t)

  ### MAlist obj - snapCGH
  ma.obj<-list()
  ma.obj$design<-1
```

D. R code of segmentation data simulation and benchmarking

```
ma.obj$M<-xx
ma.obj$genes<-data.frame(Chr='sseq', Position=ssxx, Start=ssxx, End=ssxx)
class(ma.obj)<-'MAList'

### bioHMM - snapCGH, when not using distance, revert to HMM
if(bioHMM){
  biohmm.t<-system.time(biohmm.res<-runBioHMM(ma.obj, useCloneDists=T))
  biohmm.cp<-cumsum(runLength(Rle(biohmm.res$state)))
  roc$t[n,'biohmm']<-biohmm.t[3]
  roc$ncp[n,'biohmm']<-length(biohmm.cp)
  rm(biohmm.t)
}

### HMM - snapCGH wraper for aCGH
hmm.t<-system.time(hmm.res<-runHomHMM(ma.obj))
hmm.cp<-cumsum(runLength(Rle(hmm.res$state)))
roc$t[n,'hmm']<-hmm.t[3]
roc$ncp[n,'hmm']<-length(hmm.cp)
rm(hmm.t, ma.obj)

###GLAD - original
profV<-data.frame(PosOrder=ssxx, LogRatio=xx, PosBase=ssxx, Chromosome='
    999')
profileCGH<-list(profileValues = profV)
class(profileCGH) <- "profileCGH"
glad.t<-system.time(glad.res <- glad(profileCGH))
glad.cp<-c(glad.res$BkpInfo$PosBase, sum(L))
roc$t[n,'glad']<-glad.t[3]
roc$ncp[n,'glad']<-length(glad.cp)
rm(profileValues, glad.t, profileCGH)

### multiseg - cghseg
cgh.obj <- new("CGHdata",Y=as.data.frame(xx))
CGHo <- new("CGHoptions")
cghseg.t<-system.time(cghseg.res<-multiseg(cgh.obj,CGHo))
cghseg.cp<-cghseg.res@mu[[1]][,'end']
roc$t[n,'cghseg']<-cghseg.t[3]
roc$ncp[n,'cghseg']<-length(cghseg.cp)
rm(cgh.obj, cghseg.t, CGHo)

### haarseg
haarseg.t<-system.time(haarseg.res<-haarSeg(E))
roc$t[n,'haarseg']<-haarseg.t[3]
roc$ncp[n,'haarseg']<-nrow(haarseg.res$SegmentsTable)
rm(haarseg.t)

if(bioHMM){
```

```
        roc$A<-c(roc$A, list(data.frame(
          bcp=bcp.res$posterior.mean,
          biohmm=biohmm.res$M.predicted,
          cbs=rep(cbs.res$output$seg.mean, times=cbs.res$output$num.mark),
          cghseg=rep(cghseg.res@mu[[1]][,'mean'],
            times=(cghseg.res@mu[[1]][,'end']-cghseg.res@mu[[1]][,'begin']+1)),
          haarseg=haarseg.res$Segmented,
          glad=glad.res$profileValues$Smoothing,
          hmm=hmm.res$M.predicted,
          mine=rep(as.numeric(mcols(mine.res@res)[,'AVG']),
            times=width(mine.res@res))
        )))
      } else {
        roc$A<-c(roc$A, list(data.frame(
          bcp=bcp.res$posterior.mean,
          cbs=rep(cbs.res$output$seg.mean, times=cbs.res$output$num.mark),
          cghseg=rep(cghseg.res@mu[[1]][,'mean'],
            times=(cghseg.res@mu[[1]][,'end']-cghseg.res@mu[[1]][,'begin']+1)),
          haarseg=haarseg.res$Segmented,
          glad=glad.res$profileValues$Smoothing,
          hmm=hmm.res$M.predicted,
          mine=rep(as.numeric(mcols(mine.res@res)[,'AVG']),
            times=width(mine.res@res))
        )))
      }
        cat('layout ', g, ' simulation ', n, ' finished\n')
      }
    return(roc)
  })
  return(nnres)
}

################################################
# models comparision with simulated data
################################################
library(DNAcopy)
library(bcp)
library(cghseg)
library(biomvRCNS)
library(aCGH)
library(HaarSeg)
library(GLAD)
library(snapCGH)
library(ROC)
seed<-832314
nsim<-100; ngrid=100;
```

D. R code of segmentation data simulation and benchmarking

```
# pois count seq, state 3 of interest
ers<-c(1.5, 1.75, 2)
soj<-list(type='pois', lambda=c(285, 5, 10), shift=c(0,0,0))
emis<-list(type='pois')
for(er in ers){
  roc<-simROCDataE(nsim=nsim, ngrid=ngrid, soj=soj,
    emis=emis, seed=seed, er=er)
  recName<-pasteo('emis.',emis$type,'.roc.nsim.',
    nsim,'.ngrid.',ngrid,'.er.', er, '.RData')
  save(roc, seed, nsim, ngrid, er, soj, emis, file=recName)
  dirName<-pasteo('emis.',emis$type,'.roc.nsim.',
    nsim,'.ngrid.',ngrid,'.er.', er)
  dir.create(dirName)
  system(paste('mv␣*.pdf␣./', dirName, '/', sep=''))
}

# normal ratio cgh, state 1 and 3 of interest
ers<-c(1,2,3)
nsim<-100; ngrid=100;
soj<-list(type='pois', lambda=c(20, 270, 10), shift=c(0,0,0))
emis<-list(type='norm')
for(er in ers){
  roc<-simROCDataE(nsim=nsim, ngrid=ngrid, soj=soj,
    emis=emis, seed=seed, er=er, bioHMM=T)
  recName<-pasteo('emis.',emis$type,'.roc.nsim.',
    nsim,'.ngrid.',ngrid,'.er.', er, '.RData')
  save(roc, seed, nsim, ngrid, er, soj, emis, file=recName)
  dirName<-pasteo('emis.',emis$type,'.roc.nsim.',
    nsim,'.ngrid.',ngrid,'.er.', er)
  dir.create(dirName)
  system(paste('mv␣*.pdf␣./', dirName, '/', sep=''))
}

# integrate output log and plot
ers<-c(1,1.5,1.75, 2, 3)
nstep<-100;nsim<-100; ngrid=100;
logName<-pasteo('roc.nsim.',nsim,'.ngrid.',ngrid,'.er.',
  paste(ers, collapse=','), '.nstep.',nstep,'.log')
figName<-pasteo('roc.nsim.',nsim,'.ngrid.',ngrid,'.er.',
  paste(ers, collapse=','), '.nstep.',nstep,'.eps')
sink(logName)
setEPS()
postscript(figName, paper='special', fonts=c("sans"),
  colormodel="rgb", height=20, width=10*3)
par(mfrow=c(2,3))
for(e.t in c('norm', 'pois')){
  if(e.t=='pois') {
```

```
    ers<-c(1.5,1.75,2)
  } else {
    ers<-c(1,2,3)
  }
  for(er in ers){
    recName<-pasteo('emis.',e.t,'.roc.nsim.',
      nsim,'.ngrid.',ngrid,'.er.', er, '.RData')
    load(recName)
    cat('\n\n', recName, '\n')
    if(match(er, ers)==1){
      par(mar=c(4,12,3,1))
    } else {
      par(mar=c(4,6,3,1))
    }
    if(e.t == 'norm'){
      reportROC(roc, er=er, CorS='C',nstep=nstep,
       main=pasteo('r=', er), pcex=3)
      if(match(er, ers)==1){
        text(par("usr")[1]-0.1, 0.5,
          pasteo('(', letters[match(e.t, c('norm', 'pois'))], ')'),
          srt = 360, xpd = TRUE, pos = 2, cex=3)
      }
    } else {
      reportROC(roc, er=er, CorS='S',nstep=nstep,
        main=pasteo('r=', er), pcex=3)
      if(match(er, ers)==1){
        text(par("usr")[1]-0.1, 0.5,
          pasteo('(', letters[match(e.t, c('norm', 'pois'))], ')'),
          srt = 360, xpd = TRUE, pos = 2, cex=3)
      }
    }
  }
}
dev.off()
sink()

#####################################################
# to create example image of one simulation
#####################################################
nn<-50; J=3; nstep<-100;nsim<-100; ngrid=100;
erf<-seq_len(J)-1
colors<-palette()[1:8]; colors[7]<-'orange'
figName<-pasteo('roc.example.col2.nsim.',nn,'.ngrid.',nn,'.eps')
setEPS()
postscript(figName, paper='special', fonts=c("sans"),
  colormodel="rgb", height=10*1, width=10*2)
par(mfrow=c(2,1))
```

D. R code of segmentation data simulation and benchmarking

```
for(e.t in c('norm', 'pois')){
  # use prior parameter
  if(e.t == 'pois'){
    er<-1.75
    m <-er^erf
    main<-'Simulation␣2␣(Poisson):␣r=1.75,␣J=3,␣grid=50,␣nsim=50'
    lpos<-'topright'
  } else {
    er <- 2
    m <- er*(erf+1)
    main<-'Simulation␣1␣(Normal):␣r=2,␣J=3,␣grid=50,␣nsim=50'
    lpos<-'bottomright'
  }
  recName<-pasteo('emis.',e.t,'.roc.nsim.',
    nsim,'.ngrid.',ngrid,'.er.', er, '.RData')
  load(recName)
  # plot simulated data points
  plot(seq_len(nrow(roc[[nn]]$E)), roc[[nn]]$E[,nn], col='bisque3',
    xlab='Positions', pch=18, cex=1,ylab='Simulated␣signals',
    main=main, cex.main=3, cex.lab=1.5, cex.axis=2)
  # plot underlying true segments
  lines(seq_len(nrow(roc[[nn]]$A[[nn]])),
    as.vector(Rle(m[roc[[nn]]$S], roc[[nn]]$L)), col='bisque3', lwd=8)
  # for each model
  mods<-colnames(roc[[nn]]$A[[nn]])
  mods<-sort(mods)
  if(e.t == 'norm'){
    mods<-mods[c(1, 3:8, 2)]
  }
  for(i in seq_along(mods)){
    lines(seq_len(nrow(roc[[nn]]$A[[nn]])),
      roc[[nn]]$A[[nn]][,mods[i]], col=colors[i], lty=i, lwd=3, cex=2)
  }
  # legend
  mods[mods=='mine']<-'hsmm'
  legend(lpos, mods, col=colors[seq_along(mods)],
    lty=seq_along(mods), ncol=3, bty = "n", cex=1.5, lwd=3)
}
dev.off()
```

APPENDIX E

Tiling array QC reports

The quality controls were done separately for each population using *arrayQualityMetrics* after preprocessing.

The first set of reports is of samples from population 1 (DuPi). Among all samples only the 17th array has been marked as outlying due to its relatively large distance from the others, though it is still clustered close enough to its technical replicate of array 7. Considering all other tests returned normal, so we kept it and treated it not as a real outlier but rather a side effect of the batch bias removal process.

The second set is for population 2 (PiF1), for which parallel mRNA-seq runs have been done on the same animals. Similarly some arrays marked for relatively large distance to other arrays were kept considering the majority of tests have passed.

E. Tiling array QC reports

arrayQualityMetrics report for yesobj

Browser compatibility

This report uses recent features of HTML 5. Functionality has been tested on these browsers: Firefox 10, Chrome 17, Safari 5.1.2

- Array metadata and outlier detection overview

	array	sampleNames	*1	*2	*3	Type	Batch	SampleID	SampleNumber	FileName	Slot
☐	1	D027038				HI	1	D027038	14	254451610064-532-Area1-B1-2013-01-15.txt	1
☐	2	D027039				LO	1	D027039	15	254451610063-532-Area1-B1-2013-01-15.txt	3
☐	3	D023016				LO	2	D023016	6	254451610062-532-Area1-B2-2013-01-17.txt	1
☐	4	D026030				LO	2	D026030	9	254451610061-532-Area1-B2-2013-01-17.txt	2
☐	5	D026048				HI	2	D026048	10	254451610060-532-Area1-B2-2013-01-17.txt	3
☐	6	D026053				HI	2	D026053	11	254451610059-532-Area1-B2-2013-01-17.txt	4
☐	7	D036028				HI	3	D036028	17	254451610058-532-Area1-B3-2013-01-18.txt	1
☐	8	D036029				LO	3	D036029	18	254451610057-532-Area1-B3-2013-01-18.txt	2
☐	9	D036058				HI	3	D036058	20	254451610056-532-Area1-B3-2013-01-18.txt	3
☐	10	D049051				HI	3	D049051	28	254451610055-532-Area1-B3-2013-01-18.txt	4
☐	11	D027038.1				HI	5	D027038	14	254451610050-532-Area1-B5-2013-01-23.txt	1
☐	12	D027039.1				LO	5	D027039	15	254451610049-532-Area1-B5-2013-01-23.txt	2
☐	13	D036058.1				HI	5	D036058	20	254451610048-532-Area1-B5-2013-01-23.txt	3
☐	14	D049051.1				HI	5	D049051	28	254451610047-532-Area1-B5-2013-01-23.txt	4
☐	15	D023016.1				LO	6	D023016	6	254451610046-532-Area1-B6-2013-01-24.txt	1
☐	16	D026030.1				LO	6	D026030	9	254451610045-532-Area1-B6-2013-01-24.txt	2
☑	17	D036028.1	x			HI	6	D036028	17	254451610079-532-Area1-B6-2013-01-24.txt	3
☐	18	D036029.1				LO	6	D036029	18	254451610078-532-Area1-B6-2013-01-24.txt	4
☐	19	D026048.1				HI	7	D026048	10	254451610077-532-Area1-B7-2013-01-25.txt	1
☐	20	D026053.1				HI	7	D026053	11	254451610076-532-Area1-B7-2013-01-25.txt	2
☐	21	D036046				LO	7	D036046	19	254451610075-532-Area1-B7-2013-01-25.txt	3
☐	22	D027038.2				HI	8	D027038	14	254451610073-532-Area1-B8-2013-01-29.txt	1
☐	23	D027039.2				LO	8	D027039	15	254451610072-532-Area1-B8-2013-01-29.txt	2

The columns named *1, *2, ... indicate the calls from the different outlier detection methods:

1. outlier detection by Distances between arrays
2. outlier detection by Boxplots
3. outlier detection by MA plots

The outlier detection criteria are explained below in the respective sections. Arrays that were called outliers by at least one criterion are marked by checkbox selection in this table, and are indicated by highlighted lines or points in some of the plots below. By clicking the checkboxes in the table, or on the corresponding points/lines in the plots, you can modify the selection. To reset the selection, reload the

HTML page in your browser.

At the scope covered by this software, outlier detection is a poorly defined question, and there is no 'right' or 'wrong' answer. These are hints which are intended to be followed up manually. If you want to automate outlier detection, you need to limit the scope to a particular platform and experimental design, and then choose and calibrate the metrics used.

Section 1: Between array comparison

- Figure 1: Distances between arrays.

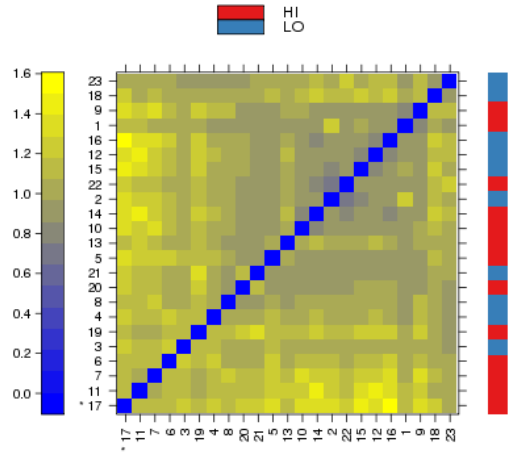


Figure 1 (PDF file) shows a false color heatmap of the distances between arrays. The color scale is chosen to cover the range of distances encountered in the dataset. Patterns in this plot can indicate clustering of the arrays either because of intended biological or unintended experimental factors (batch effects). The distance d_{ab} between two arrays *a* and *b* is computed as the mean absolute difference (L_1-distance) between the data of the arrays (using the data from all probes without filtering). In formula, d_{ab} = mean | M_{ai} - M_{bi} |, where M_{ai} is the value of the *i*-th probe on the *a*-th array. Outlier detection was performed by looking for arrays for which the sum of the distances to all other arrays, $S_a = \Sigma_b d_{ab}$ was exceptionally large. One such array was detected, and it is marked by an asterisk, *.

- Figure 2: Outlier detection for Distances between arrays.

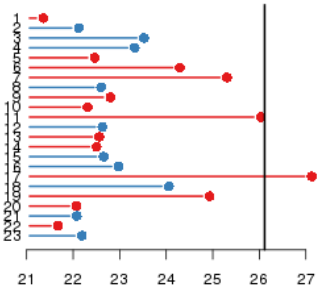


Figure 2 (PDF file) shows a bar chart of the sum of distances to other arrays S_a, the outlier detection criterion from the previous figure. The bars are shown in the original order of the arrays. Based on the distribution of the values across all arrays, a threshold of 26.1 was determined, which is indicated by the vertical line. One array exceeded the threshold and was considered an outlier.

- Figure 3: Principal Component Analysis.

E. Tiling array QC reports

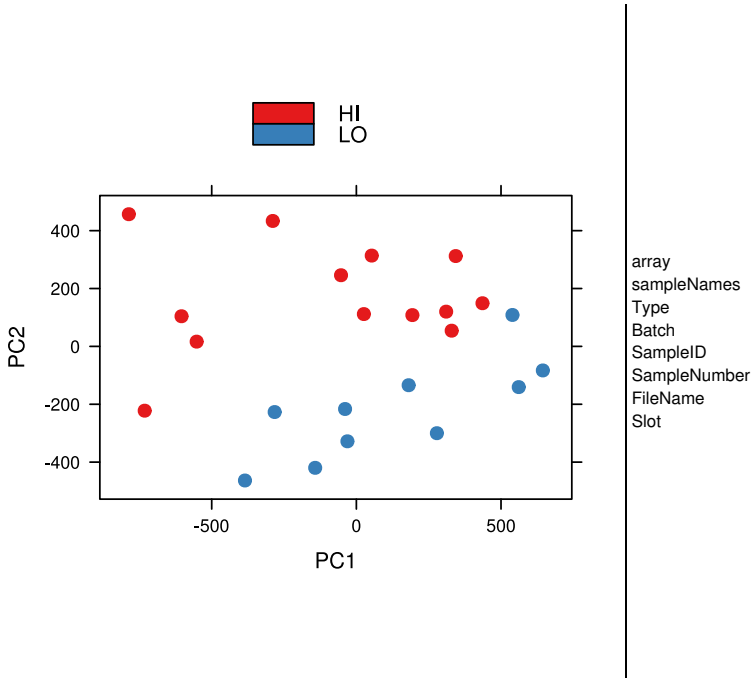


Figure 3 (PDF file) shows a scatterplot of the arrays along the first two principal components. You can use this plot to explore if the arrays cluster, and whether this is according to an intended experimental factor, or according to unintended causes such as batch effects. Move the mouse over the points to see the sample names.
Principal component analysis is a dimension reduction and visualisation technique that is here used to project the multivariate data vector of each array into a two-dimensional plot, such that the spatial arrangement of the points in the plot reflects the overall data (dis)similarity between the arrays.

Note: the figure is static - enhancement with interactive effects failed. This is either due to a version incompatibility of the 'SVGAnnotation' R package and your version of 'Cairo' or 'libcairo', or due to plot misformating. Please consult the Bioconductor mailing list, or contact the maintainer of 'arrayQualityMetrics' with a reproducible example in order to fix this problem.

Section 2: Array intensity distributions

- Figure 4: Boxplots.

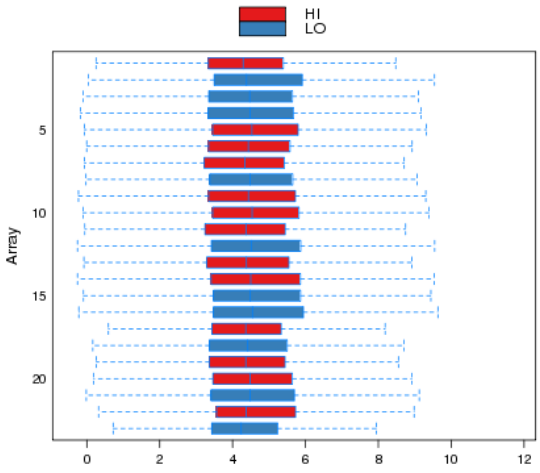


Figure 4 (PDF file) shows boxplots representing summaries of the signal intensity distributions of the arrays. Each box corresponds to one array. Typically, one expects the boxes to have similar positions and widths. If the distribution of an array is very different from the others, this may indicate an experimental problem. Outlier detection was performed by computing the Kolmogorov-Smirnov statistic K_a between each array's distribution and the distribution of the pooled data.

- Figure 5: Outlier detection for Boxplots.

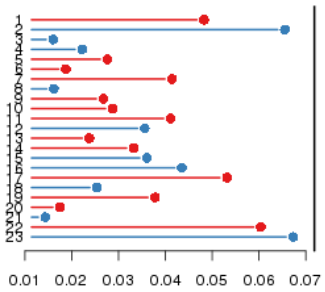


Figure 5 (PDF file) shows a bar chart of the Kolmogorov-Smirnov statistic K_a, the outlier detection criterion from the previous figure. The bars are shown in the original order of the arrays. Based on the distribution of the values across all arrays, a threshold of 0.0718 was determined, which is indicated by the vertical line. None of the arrays exceeded the threshold and was considered an outlier.

- Figure 6: Density plots.

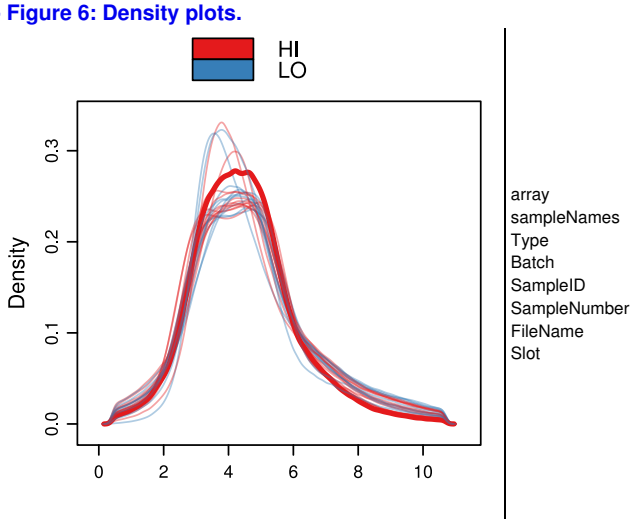


Figure 6 (PDF file) shows density estimates (smoothed histograms) of the data. Typically, the distributions of the arrays should have similar shapes and ranges. Arrays whose distributions are very different from the others should be considered for possible problems. Various features of the distributions can be indicative of quality related phenomena. For instance, high levels of background will shift an array's distribution to the right. Lack of signal diminishes its right right tail. A bulge at the upper end of the intensity range often indicates signal saturation.

Section 3: Variance mean dependence

- Figure 7: Standard deviation versus rank of the mean.

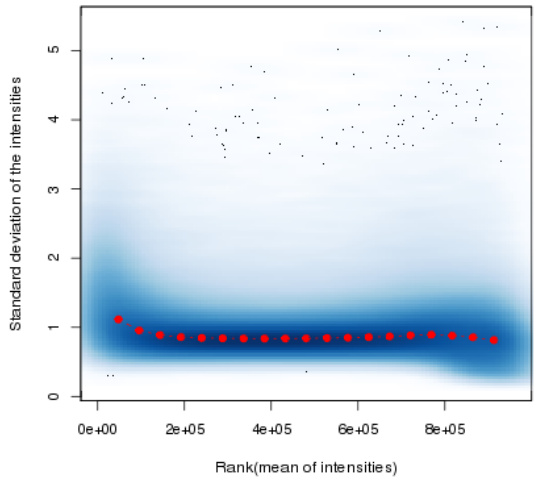


Figure 7 (PDF file) shows a density plot of the standard deviation of the intensities across arrays on the *y*-axis versus the rank of their mean on the *x*-axis. The red dots, connected by lines, show the running median of the standard deviation. After normalisation and transformation to a logarithm(-like) scale, one typically expects the red line to be approximately horizontal, that is, show no substantial trend. In some cases, a hump on the right hand of the x-axis can be observed and is symptomatic of a saturation of the intensities.

Section 4: Individual array quality

- Figure 8: MA plots.

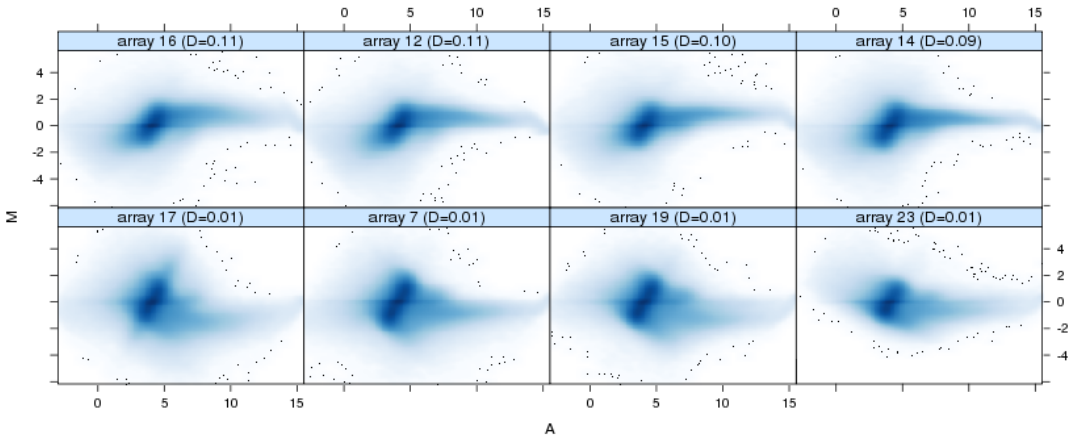


Figure 8 (PDF file) shows MA plots. M and A are defined as:
$M = \log_2(I_1) - \log_2(I_2)$
$A = 1/2\ (\log_2(I_1)+\log_2(I_2))$,
where I_1 is the intensity of the array studied, and I_2 is the intensity of a "pseudo"-array that consists of the median across arrays. Typically, we expect the mass of the distribution in an MA plot to be concentrated along the M = 0 axis, and there should be no trend in M as a function of A. If there is a trend in the lower range of A, this often indicates that the arrays have different background intensities; this may be addressed by background correction. A trend in the upper range of A can indicate saturation of the measurements; in mild cases, this may be addressed by non-linear normalisation (e.g. quantile normalisation).
Outlier detection was performed by computing Hoeffding's statistic D_a on the joint distribution of A and M for each array. Shown are first the 4 arrays with the highest values of D_a, then the 4 arrays with the lowest values. The value of D_a is shown in the panel headings. 0 arrays had D_a>0.15 and were marked as outliers. For more information on Hoeffing's *D*-statistic, please see the manual page of the function `hoeffd` in the `Hmisc` package.

- Figure 9: Outlier detection for MA plots.

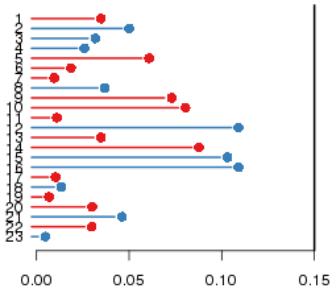


Figure 9 (PDF file) shows a bar chart of the D_a, the outlier detection criterion from the previous figure. The bars are shown in the original order of the arrays. A threshold of 0.15 was used, which is indicated by the vertical line. None of the arrays exceeded the threshold and was considered an outlier.

This report has been created with arrayQualityMetrics 3.16.0 under R version 3.0.0 (2013-04-03).

(Page generated on Fri Jul 12 11:17:52 2013 by hwriter)

E. Tiling array QC reports

arrayQualityMetrics report for yesobj

Browser compatibility

This report uses recent features of HTML 5. Functionality has been tested on these browsers: Firefox 10, Chrome 17, Safari 5.1.2

- Array metadata and outlier detection overview

	array	sampleNames	*1	*2	*3	Type	Batch	SampleID	SampleNumber	FileName	Slot
☐	1	36				HI	9	36	36	254451610041-532-Area1-B9-2013-04-16.txt	1
☐	2	199				HI	9	199	199	254451610042-532-Area1-B9-2013-04-16.txt	3
☐	3	82				HI	10	82	82	254451610080-532-Area1-B10-2013-05-07.txt	1
☐	4	83				LO	10	83	83	254451610067-532-Area1-B10-2013-05-07.txt	3
☐	5	204				HI	11	204	204	254451610068-532-Area1-B11-2013-05-14.txt	1
☑	6	424	x			LO	11	424	424	254451610081-532-Area1-B11-2013-05-14.txt	3
☐	7	434				HI	11	434	434	254451610082-532-Area1-B11-2013-05-14.txt	4
☐	8	205				LO	12	205	205	254451610043-532-Area1-B12-2013-05-15.txt	1
☐	9	559				HI	12	559	559	254451610065-532-Area1-B12-2013-05-15.txt	3
☐	10	579				LO	12	579	579	254451610066-532-Area1-B12-2013-05-15.txt	4
☐	11	36.1				HI	13	36	36	254451610083-532-Area1-B13-2013-05-16.txt	1
☐	12	234				LO	13	234	234	254451610084-532-Area1-B13-2013-05-16.txt	3
☐	13	424.1				LO	13	424	424	254451610085-532-Area1-B13-2013-05-16.txt	4
☐	14	83.1				LO	14	83	83	254451610087-532-Area1-B14-2013-05-17.txt	3
☐	15	204.1				HI	14	204	204	254451610088-532-Area1-B14-2013-05-17.txt	4
☐	16	205.1				LO	15	205	205	254451610089-532-Area1-B15-2013-05-22.txt	1
☐	17	82.1				HI	15	82	82	254451610090-532-Area1-B15-2013-05-22.txt	3
☐	18	234.1				LO	15	234	234	254451610091-532-Area1-B15-2013-05-22.txt	4
☐	19	204.2				HI	16	204	204	254451610092-532-Area1-B16-2013-05-23.txt	1
☐	20	434.1				HI	16	434	434	254451610093-532-Area1-B16-2013-05-23.txt	3
☐	21	261				LO	16	261	261	254451610094-532-Area1-B16-2013-05-23.txt	4
☐	22	83.2				LO	17	83	83	254451610095-532-Area1-B17-2013-05-24.txt	1
☐	23	261.1				LO	17	261	261	254451610096-532-Area1-B17-2013-05-24.txt	3
☑	24	559.1	x			HI	17	559	559	254451610097-532-Area1-B17-2013-05-24.txt	4
☐	25	579.1				LO	18	579	579	254451610098-532-Area1-B18-2013-05-29.txt	1
☐	26	82.2				HI	18	82	82	254451610099-532-Area1-B18-2013-05-29.txt	3
☑	27	424.2	x			LO	18	424	424	254451610100-532-Area1-B18-2013-05-29.txt	4

The columns named *1, *2, ... indicate the calls from the different outlier detection methods:

1. outlier detection by Distances between arrays

2. outlier detection by Boxplots
3. outlier detection by MA plots

The outlier detection criteria are explained below in the respective sections. Arrays that were called outliers by at least one criterion are marked by checkbox selection in this table, and are indicated by highlighted lines or points in some of the plots below. By clicking the checkboxes in the table, or on the corresponding points/lines in the plots, you can modify the selection. To reset the selection, reload the HTML page in your browser.

At the scope covered by this software, outlier detection is a poorly defined question, and there is no 'right' or 'wrong' answer. These are hints which are intended to be followed up manually. If you want to automate outlier detection, you need to limit the scope to a particular platform and experimental design, and then choose and calibrate the metrics used.

Section 1: Between array comparison

- Figure 1: Distances between arrays.

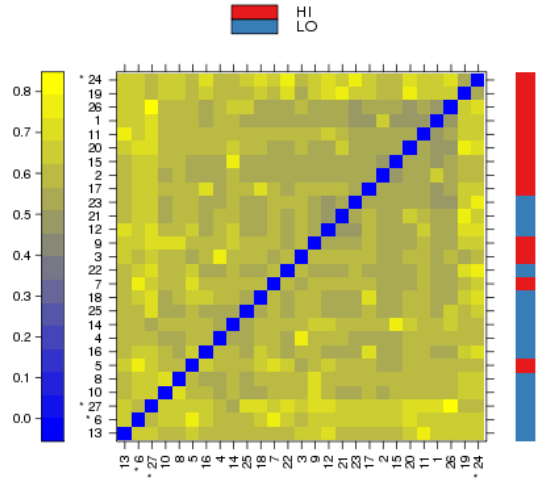


Figure 1 (PDF file) shows a false color heatmap of the distances between arrays. The color scale is chosen to cover the range of distances encountered in the dataset. Patterns in this plot can indicate clustering of the arrays either because of intended biological or unintended experimental factors (batch effects). The distance d_{ab} between two arrays *a* and *b* is computed as the mean absolute difference (L_1-distance) between the data of the arrays (using the data from all probes without filtering). In formula, d_{ab} = mean | M_{ai} - M_{bi} |, where M_{ai} is the value of the *i*-th probe on the *a*-th array. Outlier detection was performed by looking for arrays for which the sum of the distances to all other arrays, $S_a = \Sigma_b d_{ab}$ was exceptionally large. 3 such arrays were detected, and they are marked by an asterisk, *.

- Figure 2: Outlier detection for Distances between arrays.

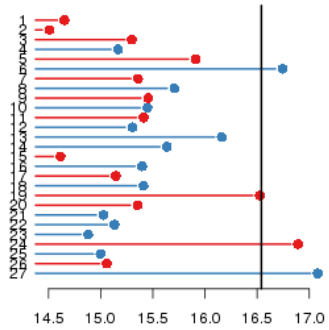


Figure 2 (PDF file) shows a bar chart of the sum of distances to other arrays S_a, the outlier detection criterion from the previous figure. The bars are shown in the original order of the arrays. Based on the distribution of the values across all arrays, a threshold of 16.5 was determined, which is indicated by the vertical line. 3 arrays exceeded the threshold and were considered outliers.

- Figure 3: Principal Component Analysis.

E. Tiling array QC reports

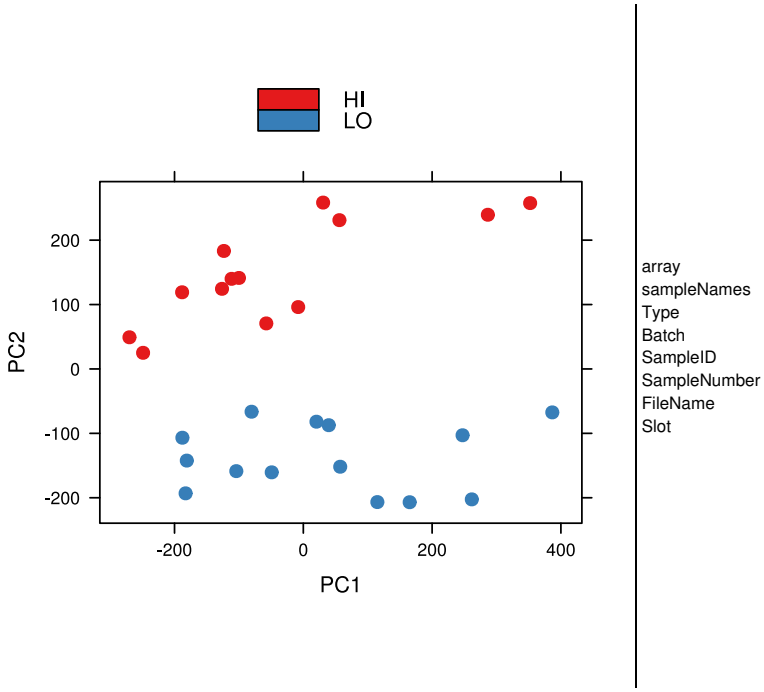


Figure 3 (PDF file) shows a scatterplot of the arrays along the first two principal components. You can use this plot to explore if the arrays cluster, and whether this is according to an intended experimental factor, or according to unintended causes such as batch effects. Move the mouse over the points to see the sample names.
Principal component analysis is a dimension reduction and visualisation technique that is here used to project the multivariate data vector of each array into a two-dimensional plot, such that the spatial arrangement of the points in the plot reflects the overall data (dis)similarity between the arrays.

Note: the figure is static - enhancement with interactive effects failed. This is either due to a version incompatibility of the 'SVGAnnotation' R package and your version of 'Cairo' or 'libcairo', or due to plot misformating. Please consult the Bioconductor mailing list, or contact the maintainer of 'arrayQualityMetrics' with a reproducible example in order to fix this problem.

Section 2: Array intensity distributions

- Figure 4: Boxplots.

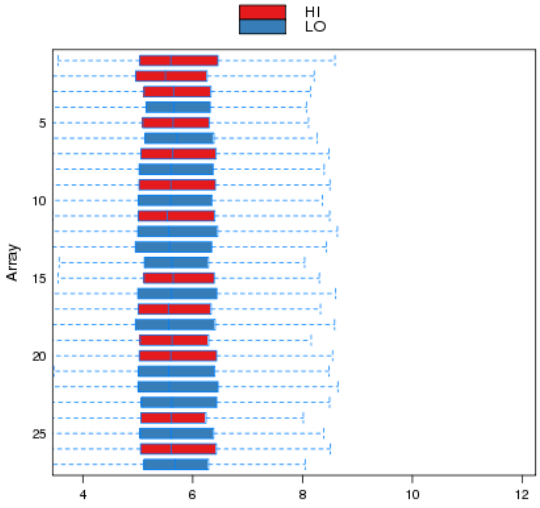


Figure 4 (PDF file) shows boxplots representing summaries of the signal intensity distributions of the arrays. Each box corresponds to one array. Typically, one expects the boxes to have similar positions and widths. If the distribution of an array is very different from the others, this may indicate an experimental problem. Outlier detection was performed by computing the Kolmogorov-Smirnov statistic K_a between each

array's distribution and the distribution of the pooled data.

- Figure 5: Outlier detection for Boxplots.

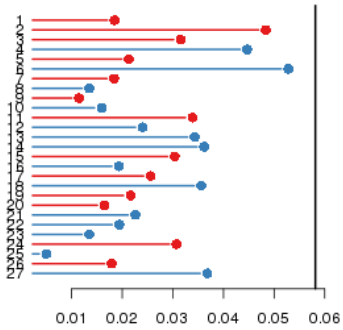


Figure 5 (PDF file) shows a bar chart of the Kolmogorov-Smirnov statistic K_a, the outlier detection criterion from the previous figure. The bars are shown in the original order of the arrays. Based on the distribution of the values across all arrays, a threshold of 0.0582 was determined, which is indicated by the vertical line. None of the arrays exceeded the threshold and was considered an outlier.

- Figure 6: Density plots.

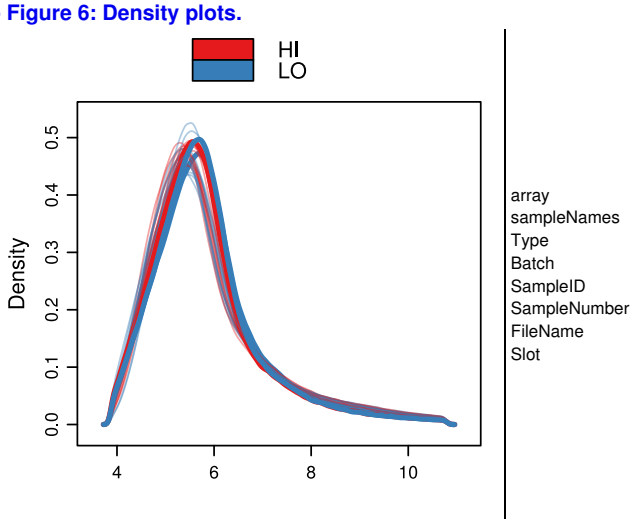


Figure 6 (PDF file) shows density estimates (smoothed histograms) of the data. Typically, the distributions of the arrays should have similar shapes and ranges. Arrays whose distributions are very different from the others should be considered for possible problems. Various features of the distributions can be indicative of quality related phenomena. For instance, high levels of background will shift an array's distribution to the right. Lack of signal diminishes its right right tail. A bulge at the upper end of the intensity range often indicates signal saturation.

Section 3: Variance mean dependence

- Figure 7: Standard deviation versus rank of the mean.

E. Tiling array QC reports

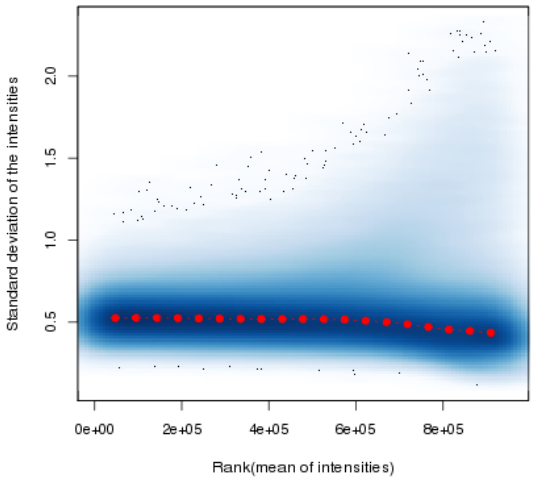


Figure 7 (PDF file) shows a density plot of the standard deviation of the intensities across arrays on the *y*-axis versus the rank of their mean on the *x*-axis. The red dots, connected by lines, show the running median of the standard deviation. After normalisation and transformation to a logarithm(-like) scale, one typically expects the red line to be approximately horizontal, that is, show no substantial trend. In some cases, a hump on the right hand of the x-axis can be observed and is symptomatic of a saturation of the intensities.

Section 4: Individual array quality

- Figure 8: MA plots.

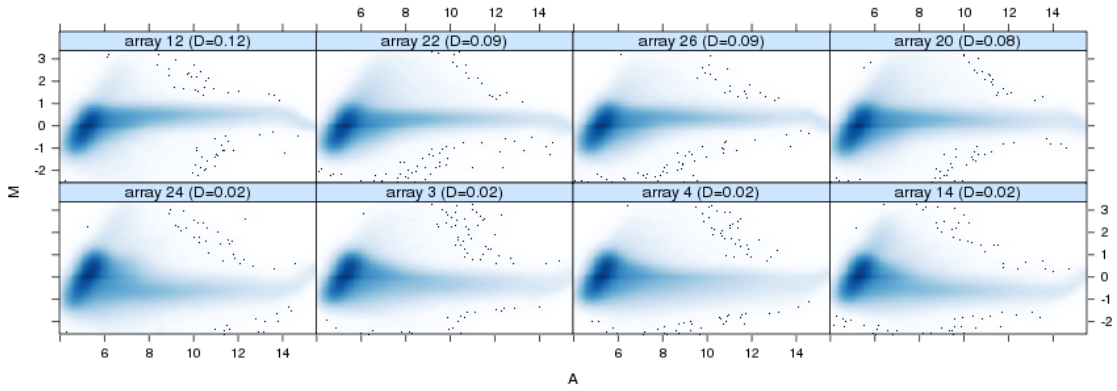


Figure 8 (PDF file) shows MA plots. M and A are defined as:
$M = \log_2(I_1) - \log_2(I_2)$
$A = 1/2\ (\log_2(I_1)+\log_2(I_2))$,
where I_1 is the intensity of the array studied, and I_2 is the intensity of a "pseudo"-array that consists of the median across arrays. Typically, we expect the mass of the distribution in an MA plot to be concentrated along the M = 0 axis, and there should be no trend in M as a function of A. If there is a trend in the lower range of A, this often indicates that the arrays have different background intensities; this may be addressed by background correction. A trend in the upper range of A can indicate saturation of the measurements; in mild cases, this may be addressed by non-linear normalisation (e.g. quantile normalisation).
Outlier detection was performed by computing Hoeffding's statistic D_a on the joint distribution of A and M for each array. Shown are first the 4 arrays with the highest values of D_a, then the 4 arrays with the lowest values. The value of D_a is shown in the panel headings. 0 arrays had D_a>0.15 and were marked as outliers. For more information on Hoeffing's *D*-statistic, please see the manual page of the function `hoeffd` in the `Hmisc` package.

- Figure 9: Outlier detection for MA plots.

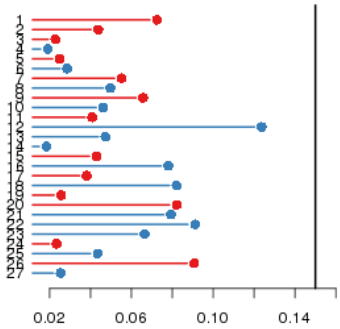


Figure 9 (PDF file) shows a bar chart of the D_a, the outlier detection criterion from the previous figure. The bars are shown in the original order of the arrays. A threshold of 0.15 was used, which is indicated by the vertical line. None of the arrays exceeded the threshold and was considered an outlier.

This report has been created with arrayQualityMetrics 3.16.0 under R version 3.0.0 (2013-04-03).

(Page generated on Fri Jul 12 11:18:25 2013 by hwriter)

Theses

Yang Du, Functional characterization and annotation of trait-associated genomic regions by transcriptome analysis.

Major contributions

- Define and evaluate the penalized uniqueness score, which shows higher sensitivity and specificity in discriminating unique sequence. The tiling probe selection pipeline, incorporating the penalized uniqueness score, could assist in the design of various types and scales of genome tiling experiment.
- Adapt and implement a novel hidden semi-Markov model designed specifically for genomic data segmentation. Through simulation benchmarking with other published tools, the efficient implementation achieves comparable or better sensitivity and specificity in genomic segmentation.

Findings and Insights

- Exploration of public datasets has shown that microarray probes with low penalized uniqueness score could interfere with data quality, and the penalized uniqueness score could serve as a better measurement for sequence heterogeneity.
- By incorporating previous knowledge in the genomic segmentation models, together with data type specific parametric settings, the package provides an unified interface for various segmentation applications, and the results are more biologically and statistically sensible.
- Through an integrative case study, using genomic data from various experiment platforms, functional candidate genes and novel transcriptional units with differential regulation status between phenotypically different groups can be detected. The largely consistent and yet complementary results from different technologies provide multiple evidences for their functional involvement in the related biological processes.

Refereed Journal Publication

- **Yang Du**, Eduard Murani, Siriluck Ponsuksili, and Klaus Wimmers. biomvRhsmm: Genomic segmentation with hidden semi-Markov model. *BioMed Research International*, 2014, 2014. ISSN 2314-6133. doi: 10.1155/2014/910390

  Yang Du concieved the idea and implemented the package. Yang Du designed the evaluation study and analyzed the data. Yang Du wrote the manuscript.

- Siriluck Ponsuksili, **Yang Du**, Frieder Hadlich, Puntita Siengdee, Eduard Murani, Manfred Schwerin, and Klaus Wimmers. Correlated mRNAs and miRNAs from co-expression and regulatory networks affect porcine muscle and finally meat properties. *BMC Genomics*, 14(1):533, 2013. ISSN 1471-2164. doi: 10.1186/1471-2164-14-533

  Yang Du conducted the weighted gene co-expression network analysis. Yang Du edited the manuscript.

- **Yang Du**, Eduard Murani, Siriluck Ponsuksili, and Klaus Wimmers. Flexible and efficient genome tiling design with penalized uniqueness score. *BMC Bioinformatics*, 13(1):323, 2012. ISSN 1471-2105. doi: 10.1186/1471-2105-13-323

  Yang Du developed the methodology and implemented the algorithm. Yang Du designed the evaluation study and analyzed the data. Yang Du wrote the manuscript.

- Siriluck Ponsuksili, **Yang Du**, Eduard Murani, Manfred Schwerin, and Klaus Wimmers. Elucidating molecular networks that either affect or respond to plasma cortisol concentration in target tissues of liver and muscle. *Genetics*, 192(3):1109–1122, 2012. ISSN 0016-6731. doi: 10.1534/genetics.112.143081

  Yang Du conducted the network edge orienting analysis. Yang Du edited the manuscript.

Conference Abstracts & Posters

Klaus Wimmers, **Yang Du**, Nares Trakooljul, Eduard Murani, and Siriluck Ponsuksili. Addressing trait-dependent expression of genes in QTL-regions for WHC by tiling array. In *International Plant and Animal Genome XXII Conference, San Diego, USA*, 2014.

Yang Du, Eduard Murani, Siriluck Ponsuksili, and Klaus Wimmers. biomvRhsmm: Genomic segmentation and copy number variation analysis with Hidden semi-Markov model. In *The joint 21st annual meeting of Intelligent Systems for Molecular Biology (ISMB) and 12th European Conference on Computational Biology (ECCB), Berlin, Germany*, 2013.

Yang Du, Eduard Murani, Siriluck Ponsuksili, and Klaus Wimmers. Characterization of QTL regions by transcriptome profiling with genome tiling arrays. In *DGfZ-Jahrestagung und DGfZ-/GfT-Gemeinschaftstagung 2012, Halle/Saale, Germany*, 2012.

Siriluck Ponsuksili, **Yang Du**, Eduard Murani, Bodo Brand, Manfred Schwerin, and Klaus Wimmers. MicroRNAs and functionally linked mRNAs affecting meat and carcass traits in pigs. In *The 33rd Conference of The International Society for Animal Genetics (ISAG), Cairns, Australia*, 2012.

Klaus Wimmers, Eduard Murani, **Yang Du**, and Siriluck Ponsuksili. Genome-wide expression and association analyses to identify genes either affecting or responding to plasma cortisol in pigs. In *The 33rd Conference of The International Society for Animal Genetics (ISAG), Cairns, Australia*, 2012.

Marcel Adler, **Yang Du**, and Klaus Wimmers. GeneDialog: Funktionelle und epistatische Netzwerke von Genen immunologischer und metabolischer Funktionswege sowie QTL für Immun- und Produktionsmerkmale. In *FUGATO-Statusseminar am 09./10. Februar 2011, Kassel, Germany*, 2011.

Anders Nordgaard and **Yang Du**. Change-point detection in environmental time series - A Bayesian approach. In *The 21th Annual Conference of The International Environmetrics Society (TIES), Margarita Island, Venezuela*, 2010.

biomvRhsmm:
Genomic segmentation with Hidden semi-Markov model

FBN / Wilhelm-Stahl-Allee 2 / 18196 Dummerstorf / www.fbn-dummerstorf.de

Yang Du[1], Eduard Murani[1], Siriluck Ponsuksili[2], Klaus Wimmers[1]

[1] Institute for Genome Biology, Leibniz Institute for Farm Animal Biology, Dummerstorf, Germany

[2] Research Group Functional Genomics, Leibniz Institute for Farm Animal Biology, Dummerstorf, Germany

Background and Introduction

With high throughput experiments like tiling array and Next-generation sequencing (NGS), researchers are looking for continuous homogeneous segments or signal peaks, which would represent transcripts and transcript variants, genome regions of deletion and amplification or genomic regions characterized by particular common features like chromatin states or DNA methylation difference. In the R/Bioconductor package ***biomvRCNS***, we implement a novel hidden semi-Markov model (HSMM) ***biomvRhsmm***, which is specially designed to handle genomic data and tailored to serve as a general segmentation tool for various types of genomic profiles, arising from both microarray and NGS platforms, with native support for modeling spatial patterns carried by genomic position and optional prior learning using annotation or previous studies.

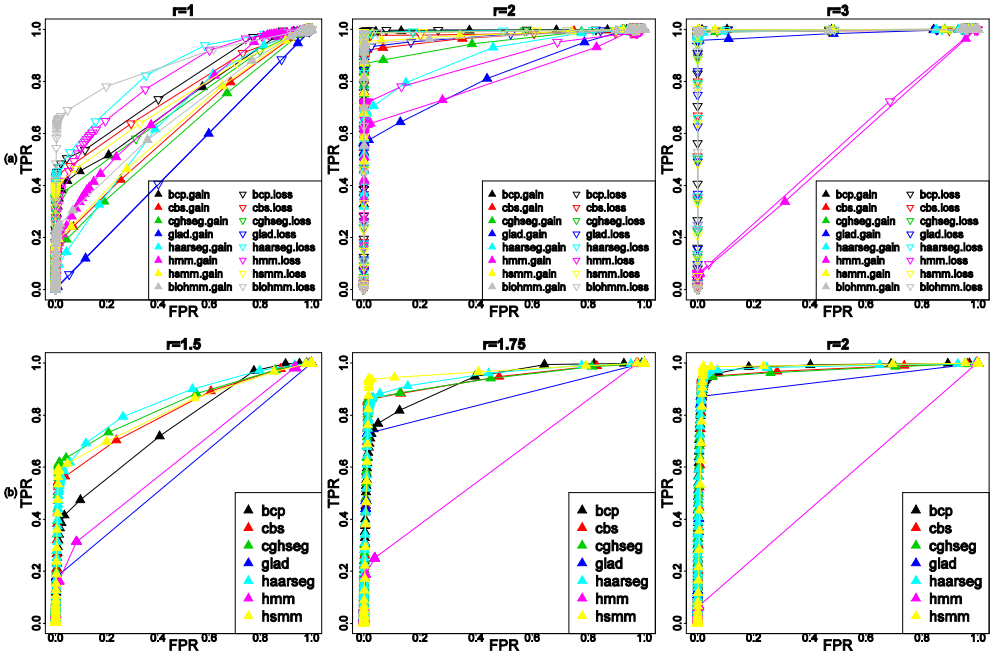
Performance comparison

We compared our model with several other state-of-the-art segmentation algorithms by calculating the Receiver Operating Characteristic (ROC) curves with simulated data (Normal and Poisson). Our model consistently ranks among the top 3 performing models, with respect to area under the ROC curves (AUC) and computing time.

	Time†	AUC rank‡
biomvRhsmm	0.2565	2.9 / 1.8
bcp	1.4630	1.3 / 3.1
bioHMM	6.9681	3.9 / NA
CBS	0.1217	4.7 / 3.6
cghseg	0.2894	4.3 / 3.1
HaarSeg	0.0027	2.9 / 1.9

† Run times are calculated as the mean run time of 2000 simulation iterations

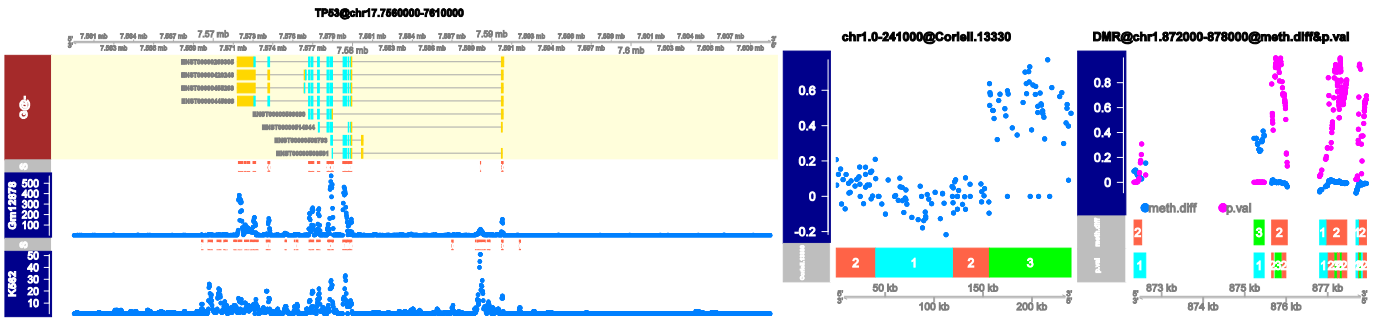
‡ AUC ranks are calculated as the weighted average rank for each model of the two simulation runs (normal / count)



Receiver operating characteristic (ROC) curves for segmentation algorithms comparison under different signal to noise settings (r). The upper panel (a) shows the simulation 1 using normal data, and the lower panel (b) shows the simulation 2 using count data. Compared algorithms are color coded as indicated in the figure legend, while the up-triangle represents segment of gain in simulation 1 and peak in simulation 2, and hollow down-triangle represents segment of loss in simulation 1. Models are labeled using lower case letters of their names. Our proposed model is coded as '*hsmm*' for simplicity, and the hidden Makov model in package *aCGH* is labeled as '*hmm*'.

Applications

We have successfully applied our model on experiments like copy number variation using well studied aCGH dataset of Coriell cell lines from Snijders *et al.* [2001] (bottom center) and transcriptome mapping using RNA-seq data generated by ENCODE project [2004, 2011] (bottom left). Also possibilities of using this model to detect differentially methylated regions (DMRs) in downstream analysis of targeted bisulfite sequencing data has been illustrated, using data from a recent study of leukemia development from Schoofs *et al.* [2013] (bottom right).



Gentleman *et al. Genome Biology*, **5**:R80 (2004)
Snijders *et al. Nature genetics*, **29**:263-264 (2001)
Consortium. *Science*, **306**, 636-640 (2004)
Consortium. *PLoS Biol*, **9**:e1001046, 04 (2011)
Schoofs *et al. Blood*, **121**:178-187 (2013)

Get Me ! In R/BioC,
biocLite('biomvRCNS')

Also available,
@GitHub
@R-Forge

LEIBNIZ INSTITUTE FOR FARM ANIMAL BIOLOGY

CV

Yang Du

Ernst-Heydemann-Str. 8
18057 Rostock, Germany

Phone: (+49) 381 494 7308
Email: yang.du@uni-rostock.de

Education & Training

04/2013 - now	**Ph.D. Candidate** in Bioinformatics Faculty of Computer Science and Electrical Engineering University of Rostock, Rostock, Germany
12/2012	Bioconductor European Developers' Workshop University of Zürich, Zürich, Switzerland
09/2011	European Summer Institute in Statistical Genetics University of Liége, Liége, Belguim
08/2008 - 07/2010	**M.Sc.** Statistics Department of Computer and Information Science Linköping University, Linköping, Sweden
09/2001 - 08/2005	**B.Sc.** Mathematics and applied mathematics School of Mathematical Sciences Fudan University, Shanghai, China

Research & Professional Experience

07/2014 - Now	**Research Assistant** Institute for Biostatistics and Informatics in Medicine and Ageing Research, Rostock, Germany
08/2010 - 07/2014	**Research Assistant** Institute for Genome Biology Leibniz Institute for Farm Animal Biology, Dummerstorf, Germany
09/2005 - 10/2006	**Analyst** Logistic Department, Evergreen Marine Corp., Tianjin, China
08/2005	**Software Engineering Internship** Jinya Electronics Co. Ltd., Tianjin, China
05/2005	**IT Internship** OneWave Technologies Inc., Shanghai, China

Declearation

I hereby declare that this dissertation is my original work and written by myself without using sources and aids other than those already cited and acknowledged. This dissertation, or its other form, has never been used as an examination paper, or been submitted to other faculties and reviewed as a dissertation.

Selbstständigkeitserklärung

Ich erkläre hiermit, dass ich diese vorgelegte Dissertation selbst verfasst und mich dabei keiner anderen als den von mir ausdrücklich bezeichneten Quellen und Hilfen bedient habe, und die den benutzten Werken woertlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht hab. Diese Dissertation wurde in dieser or andere Form weder bereits als Prüfungsarbeit verwendet, noch einer anderen Fakultät als Dissertation vorgelegt. An keiner anderen Stelle ist ein Prüfungsverfahren beantragt.

Rostock, Yang Du