Statistical methods for variant discovery and functional genomic analysis using next-generation sequencing data

Man Tang

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Statistics

Xiaowei Wu, Chair
Inyoung Kim
Christopher Franck
Liqing Zhang

December 6, 2019
Blacksburg, Virginia

Keywords: next-generation sequencing, hidden Markov model, variant calling, transcription factor, nonparametric Bayesian, Poisson processes, Dirichlet process mixture, gene expression, wavelet-based functional model.

Statistical methods for variant discovery and functional genomic analysis using next-generation sequencing data

Man Tang

(ABSTRACT)

The development of high-throughput next-generation sequencing (NGS) techniques produces massive amount of data, allowing the identification of biomarkers in early disease diagnosis and driving the transformation of most disciplines in biology and medicine. A greater concentration is needed in developing novel, powerful, and efficient tools for NGS data analysis. This dissertation focuses on modeling "omics" data in various NGS applications with a primary goal of developing novel statistical methods to identify sequence variants, find transcription factor (TF) binding patterns, and decode the relationship between TF and gene expression levels. Accurate and reliable identification of sequence variants, including single nucleotide polymorphisms (SNPs) and insertion-deletion polymorphisms (INDELs), plays a fundamental role in NGS applications. Existing methods for calling these variants often make simplified assumption of positional independence and fail to leverage the dependence of genotypes at nearby loci induced by linkage disequilibrium. We propose vi-HMM, a hidden Markov model (HMM)-based method for calling SNPs and INDELs in mapped short read data. Simulation experiments show that, under various sequencing depths, vi-HMM outperforms existing methods in terms of sensitivity and F_1 score. When applied to the human whole genome sequencing data, vi-HMM demonstrates higher accuracy in calling SNPs and INDELs. One important NGS application is chromatin immunoprecipitation followed by sequencing (ChIP-seq), which characterizes protein-DNA relations through genome-wide mapping of TF binding sites. Multiple TFs, binding to DNA sequences, often show complex binding patterns, which indicate how TFs with similar functionalities work together to regulate the expression of target genes. To help uncover the transcriptional regulation mech-

anism, we propose a novel nonparametric Bayesian method to detect the clustering pattern of multiple-TF bindings from ChIP-seq datasets. Simulation study demonstrates that our method performs best with regard to precision, recall, and F_1 score, in comparison to traditional methods. We also apply the method on real data and observe several TF clusters that have been recognized previously in mouse embryonic stem cells. Recent advances in ChIP-seq and RNA sequencing (RNA-Seq) technologies provides more reliable and accurate characterization of TF binding sites and gene expression measurements, which serves as a basis to study the regulatory functions of TFs on gene expression. We propose a log Gaussian cox process with wavelet-based functional model to quantify the relationship between TF binding site locations and gene expression levels. Through the simulation study, we demonstrate that our method performs well, especially with large sample size and small variance. It also shows a remarkable ability to distinguish real local feature in the function estimates.

Statistical methods for variant discovery and functional genomic analysis using next-generation sequencing data

Man Tang

(GENERAL AUDIENCE ABSTRACT)

The development of high-throughput next-generation sequencing (NGS) techniques produces massive amount of data and bring out innovations in biology and medicine. A greater concentration is needed in developing novel, powerful, and efficient tools for NGS data analysis. In this dissertation, we mainly focus on three problems closely related to NGS and its applications: (1) how to improve variant calling accuracy, (2) how to model transcription factor (TF) binding patterns, and (3) how to quantify of the contribution of TF binding on gene expression. We develop novel statistical methods to identify sequence variants, find TF binding patterns, and explore the relationship between TF binding and gene expressions. We expect our findings will be helpful in promoting a better understanding of disease causality and facilitating the design of personalized treatments.

Acknowledgments

I am sincerely grateful to my advisor Dr. Xiaowei Wu for his guidance, patience, timely responses to my questions, and careful editing of my dissertation draft. His guidance and mentoring provides me with the foundation and framework on which to build, and improve my research and writing skills to better serve me in my profession. Besides my advisor, I especially want to thank my committee members: Dr. Chris Franck, Dr. Inyoung Kim, and Dr. Liqing Zhang, for their encouragement and insightful comments, which helped me improved this dissertation a lot. I also want to thank Mohammad Shabbir Hasan and Sharmi Banerjee for their immense support in my graduate projects. I wish to thank the graduate communities in the Department of Statistics for their friendship, encouragement and inspiration during my studies. Finally, and most importantly, I want to thank my husband, Yafei, my daughter, Clara, and my parents for all the love, happiness and support they brought into my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivations

Starting from the discovery of DNA structure, considerable scientific and technical advancements have been made in understanding the complexity and diversity of genomes in health and disease. Sequencing the genome is an important step in revealing the underlying biological mechanisms. Developed in 1977, the first-generation sequencing method—Sanger sequencing by the dideoxy chain termination has greatly simplified DNA sequencing [84] so that scientists could be able to sequence DNA efficiently and accurately by using this technique. In the following decades, Sanger sequencing became the most widely used sequencing method [59], leading to the Human Genome Project which requires a 15-year international effort to sequence the 3 billion DNA letters in the human genome [12]. It was the increase in demands for efficient and rapid DNA sequencing for a large number of human individuals and other organisms that triggers the development of new techniques to parallelize the sequencing process and provide thousands or millions of sequences very rapidly at a modest cost [85]. These techniques replaced Sanger sequencing in recent years and are referred to as the second-generation sequencing [78]. The cost of sequencing declined dramatically from the beginning of the first-generation sequencing ($5292.39 per megabase) to the end of the second generation sequencing ($0.014 per megabase) [77]. The third-generation sequencing technologies start a new era of sequencing, making it possible to interrogate single molecules of DNA without amplifying them hence simplifying the sequencing procedures and further reducing the price [61]. With the development of high-throughput next generation sequencing (NGS), millions of genomic data have been produced at a much reduced cost, allowing the identification of biomarkers in early disease diagnosis as well as for personal treatment and driving the transformation of most disciplines in biology and medicine [77, 78, 90].

The NGS technologies produce massive amount of DNA sequence data in the form of ACGT. Such data are usually converted to FASTQ format files and can be downloaded from the

NGS platforms for further analysis. After the generation of FASTQ data, downstream analyses can be conducted, which can be broadly classified into several categories depending on the purpose, such as investigating the genome, the transcriptome, or the epigenome [12]. However, the data explosion from NGS presents great challenges for data modeling and knowledge extraction. As a result, there is a strong motivation to develop novel computational methods to discover the potential of large genomic and epigenomic datasets. In this dissertation, we mainly focus on three problems closely related to NGS and its applications: how to improve variant calling accuracy, how to model transcription factor (TF) bindings and how to quantify of the contribution of TF bindings on gene expression. Correspondingly, we develop new statistical methods to solve these problems: (1) we propose vi-HMM, an HMM-based model, to call variants in whole genome sequencing (WGS) data by taking into account the dependence of genotypes at nearby loci on the genome; (2) we use a Dirichlet process mixture of log Gaussian Cox process (DPM-LGCP) model to identify TF binding patterns; (3) we utilize a log Gaussian Cox process (LGCP) with wavelet-based functional model (WFM) to explore the relationship between TF binding and gene expression levels. Knowledge gained from such variant analysis in WGS, TF pattern analysis, and quantitative analysis of TF and gene expression can be used to facilitate understanding in disease causation and enable the design of personalized treatments based upon a personal genetic profile [12].

1.2 Identifying sequence variants

A breakthrough in NGS during the past decade provides unprecedented opportunities for investigating the contribution of genetic variation to health and disease [4, 97]. Whole genome sequencing is a popular and powerful method for discovering the association between

genotype and phenotype and learning how genetic variations or genes play an role in disease causation and progression. With NGS, the reference genome sequences have been made available in public databases for many organisms. Researchers are now able to perform comparative sequencing or resequencing to detect polymorphisms, mutations, and structural variations between organisms, and understand the biological consequences. There are several steps in processing NGS data towards the final goal of identifying genomic variants with high accuracy (Figure 1.1).

In this dissertation, we focus on the last step of NGS data processing, variant calling. Most current variant calling tools have an underlying assumption that mutational changes, e.g., single-nucleotide polymorphisms (SNPs) or insertion and deletion polymorphisms (INDELs), on nearby loci in the genome are not related. However, investigations of the linkage disequilibrium characteristics between SNPs and INDELs demonstrated nonnegligible SNP-SNP and SNP-INDEL associations [56]. Thus, incorporating the dependence between nearby loci has the potential to improve the accuracy of variant calling. For this reason, we assume Markov property for the positional dependence along the genome, and accordingly propose an HMM-based calling method.

1.2.1 Read alignment

The massive clusters of small DNA templates from NGS are called "reads". Many mapping tools have been developed to align several hundred or thousand millions of short reads generated from NGS platforms to an existing reference genome. The commonly used mapping tools are BFAST[35], BWA[49], SPAP3[55], Bowtie[44], Bowtie2[43], etc. Most of these mappers import FASTQ/CFASTQ files and export SAM/BAM files [26]. Some mappers, such as BWA, Bowtie, and Bowtie2, explore the base quality scores in FASTQ profiles during the

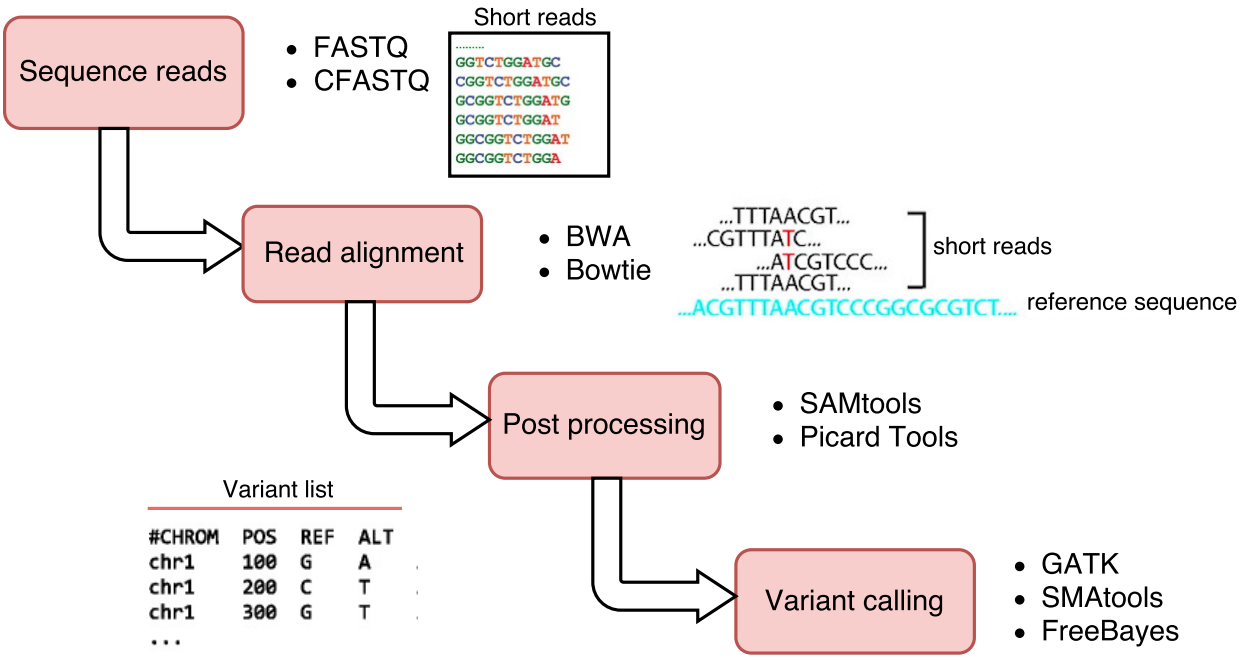

Figure 1.1: WGS workflow. Short sequence reads are mapped onto a reference genome. After post-processing, genetic variants can be identified.

alignment to reduce alignment errors [26, 51]. Accurate and efficient read mapping is a vital prerequisite for variant identification.

1.2.2 Post-processing

Post-processing is essential prior to variant calling. This step includes removal of PCR duplicates, realignment around INDELs, and recalibration of base quality scores [12]. INDEL realignment is performed to realign the reads locally to correct misalignments due to the presence of INDELs and eliminate false positive SNPs near INDELs. Since the base qualities (probability of introducing sequencing error at each base) are subject to various sources of systematic technical error, a recalibration of base quality scores allows us to make adjustment to over- or under-estimated base quality scores caused by various sources of systematic

technical error. Through post processing, one could reduce the number of false positive variants in variant calling.

1.2.3 Variant analysis

After read alignment and post-processing, the last step is to identify genetic variants. There are numerous open source tools available for variant calling, including GATK [21], SAMtools [50], VarScan [40], Platypus [81], FreeBayes [28], VarDict [42], etc. Among these tools, GATK, Platypus, FreeBayes, and SAMtools use Bayesian approaches to identify variants, VarScan proposes a heuristic/statistical method, and VarDict performs local realignment to improve the calling of INDELs. Usually these tools have a set of parameters to control the variant calling process. The majority of these calling tools report a VCF file as output.

1.3 Modeling transcription factor binding patterns

With the help of NGS technology, researchers could profile epigenetic modification for the genomes of many species [62]. Currently, the most popular method to study the epigenome is chromatin immunoprecipitation followed by sequencing (ChIP-seq), which uses chromatin immunoprecipitation and massively parallel DNA sequencing to identify the DNA binding sites of proteins, particularly transcription factors (TFs). ChIP-seq comprises a few basic steps, including: crosslinking proteinDNA complexes, shearing the chromatin, immunoprecipitation, DNA purification, and sequencing the resulting ChIP-DNA fragments simultaneously using a genome sequencer (Figure 1.2) [6, 24, 37, 64]. A number of peak-calling methods have been developed to detect TF binding peaks generated by ChIP-seq. Peak-finding methods (e.g. MACS [96]) typically either shift the read in the 3' direction toward

the peak center or computationally extend the length of tags so that the original DNA fragment can be better represented. Tags from opposite strands are merged to construct an unstranded tag density landscapes, and binding event locations are predicted from the locations with maximum tag coverage within each region that contains a significant enrichment of ChIP-seq tags (i.e. the peak summit)(Figure 1.2) [57].

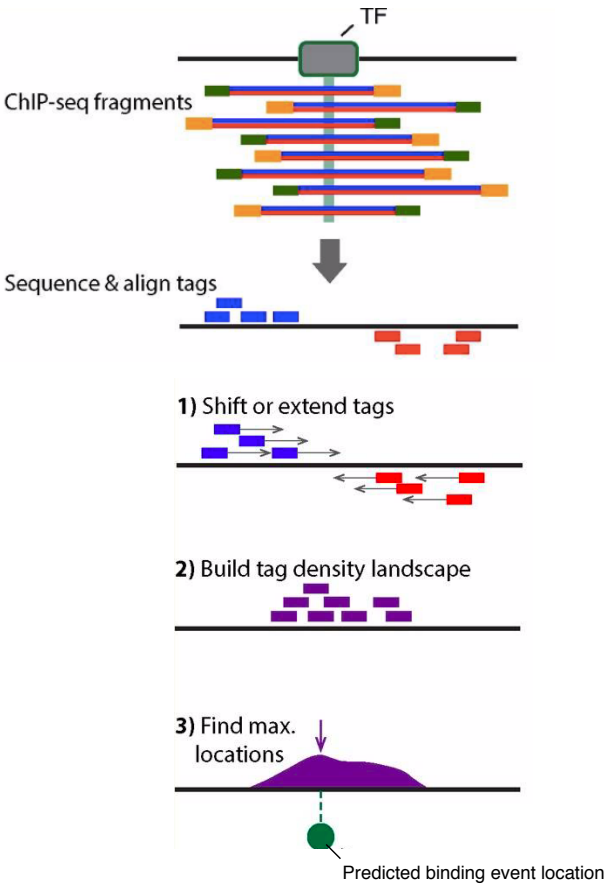

Figure 1.2: Outline of ChIP-seq peak calling based on MACS. Figure is adapted from Mahony and Pugh (2015).

TFs are proteins binding to particular DNA sequences to regulate transcriptional activity. TFs do not act alone, whereas, they interact with each other and with the genome. Therefore, multiple TFs usually exhibit complex binding patterns among their binding sites along

the genome. The characterization of these combinatorial binding patterns is essential to understand their functions underlying the gene regulatory network. Several studies have been conducted to investigate the TF binding patterns by evaluating the ChIP-seq data in the same type of tissue or cell. Lee and Zhou [47] used liquid association (LA) developed by Li [53] to measure the correlation between TFs to determine co-regulators. A computational framework developed by Oh et al. [74] utilized Z-scores to find TF binding patterns. Recently, Cha and Zhou (2014) proposed a statistical method to model TF binding site locations by inhomogeneous Poisson processes and apply Ripley's K-function to detect pairwise TF binding patterns from ChIP-seq data. In another study, a probabilistic model, SignalSpider, was developed to decipher the combinatorial binding sites of TFs [91]. However, current methods rely heavily on subjective distance thresholds or simple empirical tests. In addition, the number of clusters needs to be prespecified for most of the model-based methods, which may impact the robustness of clustering. To fill this gap, we model the TF binding site locations by inhomogeneous Poisson processes whose intensity is characterized by a log Gaussian Cox process, and propose a nonparametric Bayesian model to detect TF clusters without specifying the number of clusters in advance.

1.3.1 Inhomogeneous Poisson processes

After peak calling in ChIP-seq data processing, a set of TF binding regions on the genome are obtained, whose center can be treated as binary binding events. These binding events along the genome characterize the binding behavior of a specific TF, and can be modeled by an inhomogeneous Poisson process because of the following reasons: (1) the event of each binding site falling into a minuscule interval is a rare event and the number of events in non-overlapping regions follow the independent increment rule, (2) the nonuniform distribution of TF peaks along the genome can be well represented by the intensity function of an

inhomogeneous Poisson process. Following the inhomogeneous Poisson process setting, if we define the genome sequence to be the real line $[0, \infty)$, then for any genomic region $[a, b]$ on the real line, the number of binding sites of a specific TF, $N(a, b]$, follows an inhomogeneous Poisson process with intensity function $\lambda(s)$. The probability of n binding sites falling into the interval $(a, b]$ is then given by:

$$P[N(a, b] = n] = \frac{m(a, b)^n}{n!} e^{-m(a,b)} \tag{1.1}$$

where the intensity measure is

$$m(a, b) = \int_a^b \lambda(s) ds \tag{1.2}$$

The inhomogeneous Poisson process has the following properties:

(1) $N(0) = 0$

(2) $N(s)$ has independent increments.

(3) for any $s \in [0, \infty)$,

$$\begin{aligned} P[N(s + \Delta) - N(s) = 1] &= \lambda(s)\Delta + o(\Delta) \\ P[N(s + \Delta) - N(s) \geq 2] &= o(\Delta) \end{aligned} \tag{1.3}$$

1.3.2 Log Gaussian Cox processes

In order to further model the intensity function of the inhomogeneous Poisson process, we use a log-Gaussian Cox process (LGCP). LGCP was first introduced in Statistics by Møller [66]. This model is a combination of a Poisson process at the first level and a Gaussian process at the second level. In LGCP, the Cox process is a point process defined by the following two postulates:

(1) $\lambda = \lambda(s) : s \in \Omega$, where Ω is a non-negative random field, and in this study $\Omega = \mathbb{R}$.

(2) Conditional on λ, the point process is an inhomogeneous Poisson process on Ω with intensity function λ.

The point process is said to be a log Gaussian Cox process where $Z = \log \lambda$ is a Gaussian field. LGCP is a flexible model for aggregation because of the richness of possible mean and covariance functions in Gaussian fields [66].

1.4 Investigating the relationship between TF binding and gene expression

As mentioned before in Section 1.3, TF plays a significant role in the control of gene expression. It can active or inhibit the initiation of gene transcription through binding to particular DNA sequences along a genome. Therefore, understanding the relationship between TF and gene expression at specific spatial locations becomes an important process for decoding biological functions, providing insights into transcriptional misregulation in disease, and further facilitating the development of innovative regenerative medicine. Recent development in high-throughput next-generation sequencing has offered a new approach for mapping and quantifying transcriptomes, which is called RNA-seq (RNA sequencing). The RNA-seq technology has yield large amounts of information about gene expression and makes it possible to study the regulatory of gene expression by combining RNA-seq data and ChIP-seq data [18].

The estimation of gene expression is one of the most important application of RNA-seq, which is primarily based on the number of reads mapping to each transcript sequence. The most frequently used measure is called RPKM, short for reads per kilobase per million mapped

reads [69]. The calculation formula is as follows:

$$RPKM = \frac{10^9 C}{NL} \tag{1.4}$$

where C is the number of reads mapped to a particular gene region, N is the total number of mappable reads in the experiment, and L is the feature length. This measure normalizes away biases in the sequencing approach, such as sample sequencing depth and gene length.

Studies exploring the relationship between TF and gene expressions has been ongoing for many years. Das et al. [20] developed multivariate adaptive regression splines to learn human transcriptional subnetworks; Boulesteix and Strimmer [7] employed the method of partial least squares regression to discover the true TF activities from gene expression and ChIPmicroarray (ChIP-chip) data; Ouyang et al. [75] proposed a log-linear regression model to make predictions on gene expression from ChIP-seq data, in which TF principal components extracted from TF association strength vectors by principal component analysis are served as covariates; Cheng and Gerstein [16] applied the support vector regression to predict gene expression from the TF binding signals. Inputs in these models are based on the featured related to TFs, such as position weight matrix scores, counts of transcription factor binding sites within specific regions, and weighted sum of the corresponding ChIP-seq signal strength. However, extracting features from the ChIP-seq data omits information and thus may lead to improper statistical inferences. Therefore, it is essential to use the complete TF binding sites information instead of the extracted features. In chapter 4, we suggest an alternative statistical framework to incorporate the TF binding sites information in the modeling of gene expression. We combine the log Gaussian Cox process with a wavelet-based functional model to study the relationship between TF binding site locations and expression levels of target genes.

1.4.1 Wavelet series and the discrete wavelet transform

The wavelet basis is defined as:

$$\psi_{jk}(t) = 2^{\frac{j}{2}}\psi(2^j t - k) \tag{1.5}$$

where ψ is a mother wavelet function, j is a scale index and k is a location index. Both j and k are integers that scale and dilate ψ to generate wavelets. The mother wavelet function is constructed so that basis is orthogonal. Then a function $f(t)$ can be represented by the wavelet series:

$$f(t) = \sum_{j,k} d_{jk}\psi_{jk}(t) \tag{1.6}$$

where d_{jk} is the wavelet coefficient and defined as $d_{jk} = \int f(t)\psi_{jk}(t)dt$. In this way, a function can be decomposed into a set of wavelet and scaling coefficients, obtained from projecting the data into the wavelet space by using the discrete wavelet transform (DWT). Similarly, the wavelet coefficients can be converted back to the data space by applying an inverse discrete wavelet transform (IDWT).

1.5 Outline of Dissertation

In this dissertation, we focus on modeling "omics" data in various NGS applications with a primary goal of developing novel statistical methods to identify sequence variants, finding transcription factor binding patterns, and decoding the relationship between TF binding and gene expression levels.

This dissertation is organized as follows:

(1) Chapter 2 describes vi-HMM, a novel HMM-based method to identify small-scale sequence variants (i.e., SNPs and INDELs) in short read data. A unique feature is that this method allows transitions between hidden states (hereafter defined as SNP, insertion, deletion, and match) of adjacent genomic bases. The Viterbi algorithm is used to determine an optimal hidden state sequence. The inferred hidden state sequence provides a direct solution to the identification of SNPs and INDELs.

(2) Chapter 3 describes a Dirichlet process mixture of log Gaussian Cox process (DPM-LGCP) model to detect the clustering pattern of multiple TF bindings from ChIP-seq data. The Integrated Nested Laplace Approximations (INLA) package is used to construct an approximation of the true LGCP likelihood to reduce computational complexity. To enable nonparametric Bayesian inference, Neal's Gibbs sampling is implemented to estimate the clusters of TFs.

(3) Chapter 4 presents a framework to explore the relationship between TF binding and gene expressions. We first model the TF binding site locations by a log Gaussian Cox process so that the TF features can be represented by a intensity function. Then a wavelet-based functional model is applied to quantify the relationship between intensities and the expression levels for target genes. Bayesian method is applied to yield estimates and posterior samples for profiles.

Chapter 2

vi-HMM: A novel HMM-based method for sequence variant identification in short-read data

2.1 Introduction

Rapid evolution of NGS technologies in recent years enables various genetic applications in a fast, efficient, and cost-effective way [59, 63]. One fundamental procedure in NGS data analysis is variant calling, i.e., to identify the existence of genetic variants from short-read data. Accurate and reliable identification of SNPs and INDELs plays an important role in all NGS applications as these common sequence variants are highly abundant in the human genome and have been found to likely influence human traits and disease [14, 65, 70].

The process of variant calling starts with aligning a set of short reads to the reference genome. After reads are correctly mapped, statistical models or heuristics may be used to predict the likelihood of variation at each locus based on available information such as quality scores and allele counts of aligned reads at the locus [3]. Most statistical models used for variant calling are built on the Bayes' theorem, with an ultimate goal to predict genotypes from aligned reads by using the maximum a posteriori (MAP) estimate. Following this Bayesian approach, a number of variant calling tools have been developed, include SAMtools [50], GATK [21], FreeBayes [28], and Platypus [81]. Heuristic based tools, such as VarScan [39], call variants based on a variety of heuristic factors, e.g., minimum allele counts, read quality cut-offs, and bounds on read depth. Though heuristic methods could be robust to outlier data that do not follow probabilistic model assumptions, the selection of cut-offs and bounds is highly empirical which largely restricts their practical usage. Other alternatives such as machine learning tools [60] are also applicable for variant calling, but they appear to be relatively unpopular in practice. Due to divergence of the model assumptions, these variant calling tools perform quite differently on NGS data [32, 76]. It should be noted that, although Bayesian statistical models are highly prevalent in variant calling, existing tools developed using this approach often make simplified assumptions of positional independence and fail to leverage the dependence between genotypes at nearby loci that is caused by linkage

disequilibrium (LD). A statistical model that appropriately incorporates such dependence information has the potential to improve the accuracy of variant detection, especially in regions of high LD in the human genome.

Hidden Markov models (HMMs) can effectively model dependence between adjacent symbols or regions, thus have been extensively used in various disciplines [79]. Since its first application in computational biology in the late 1980s [17], HMMs become popular in biological sequence analysis [93]. Generally speaking, the occurrences of genetic variants (SNPs and INDELs) on the genome are not independent events because of the existence of LD between SNPs or between INDELs and SNPs [56, 65]. For this reason, one may use Markov models to better characterize the dependence between genotypes at nearby loci in order to improve the analysis of NGS data. Several HMM-based programs have been developed for read mapping and variant calling in sequencing data, including Dindel [2], PyroHMMsnp [94], and PyroHMMvar [95]. All these programs call SNPs and INDELs by estimating top candidate (most likely) haplotypes/genotypes using the Bayesian approach. In particular, Dindel [2] constructs a two-layer HMM by treating both the insertion status and its position index as hidden variables, and PyroHMMsnp [94] and PyroHMMvar [95] use HMM to formulate homopolymer errors and employ a weighted alignment graph to reconstruct the consensus sequences. However, these programs are designed for specific applications: Dindel is for INDEL calling, and PyroHMMsnp and PyroHMMvar emphasize the modeling of homopolymer. And the design of these programs complicates its variants detection process and slows down the program.

In this chapter, we propose vi-HMM, a novel HMM-based method for identifying small-scale sequence variants in short-read data. This method allows transitions between hidden states (hereafter defined as "SNP", "Ins", "Del", and "Match") of adjacent genomic loci, and determines an optimal hidden state sequence by using the Viterbi algorithm. The

inferred hidden state sequence provides a direct solution to the identification of SNPs and INDELs. Through simulations, we show that vi-HMM represents an improvement over five other variant callers—GATK HaplotypeCaller, FreeBayes, Platypus, SAMtools, and VarScan in terms of sensitivity, precision, and F_1 score. When applied to a real short-read dataset (NA12878) generated by the Genome in a Bottle (GIAB) project [100], vi-HMM demonstrates its major advantage in identifying INDEL variants as compared to four other variant callers—FreeBayes, Platypus, SAMtools, and VarScan, while still maintaining good performance in SNP-calling at different read coverage depths.

2.2 Methods

Along the genome, the states of genomic bases, i.e., whether or not and which type of sequence variants exist on the bases, often exhibit dependence. Incorporating such dependence information helps improve the accuracy of variant calling but poses challenges in calculating the joint likelihood of the entire sequence. In this study, we assume Markov property for the dependence and accordingly propose a new method for *variant identification on the basis of HMM*, acronymized by vi-HMM.

vi-HMM performs variant calling for SNPs and INDELs after short reads are mapped to a reference genome (an example is shown by IGV viewer in Appendix A.1). Its input includes a reference genome sequence and a file with mapped reads (a SAM/BAM file). The core of this method lies in the construction of an HMM that models state transitions among the bases on the genome as well as emissions from the hidden states to the observed pileup read data. From the HMM, we can uncover the optimal hidden state sequence (i.e., the Viterbi path), which is then used to call variants or infer the underlying genotypes. The workflow of the vi-HMM algorithm is shown in Figure 2.1, including three major steps:

(1) Define the states (Match, SNP, Insertion, and Deletion), and identify the transition probabilities among the states to build the transition matrix.

(2) Compute the likelihood (emission probability) of observing the pileup of reads under different states.

(3) Given a reference genome sequence, find the optimal hidden state sequence by using the Viterbi algorithm and based on which infer variants/genotypes.

Details of these steps are explained in the following subsections.

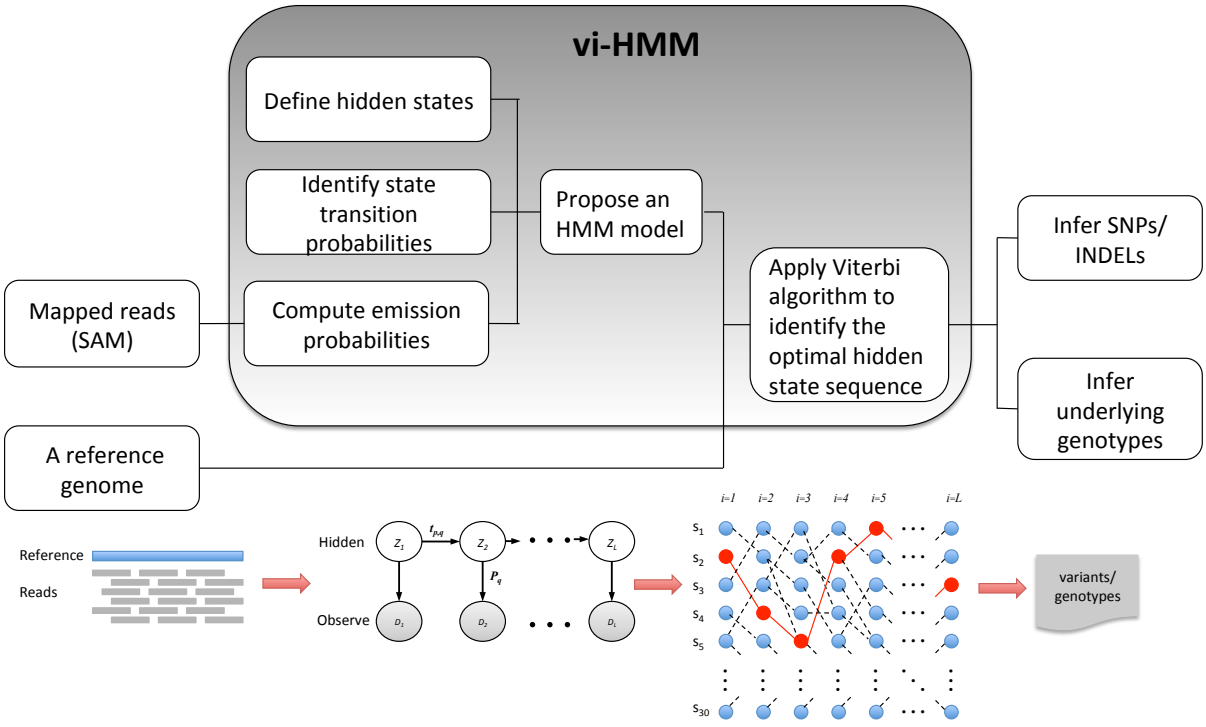

Figure 2.1: Workflow of vi-HMM.

2.2.1 State and transition matrix

We assume that all reads have been mapped to the reference genome by a standard mapping tool, such as Bowtie2 [43] or BWA-MEM [48], resulting in a SAM file. Using the CIGAR strings from the SAM file, detailed alignment information for each base can be obtained, which indicates the relation between the reference genome and the genotype sequence that underlies the mapped reads. We consider a genomic region with length L, that is, a total of L adjacent bases including the ones in the reference but not in the genotype and vice versa. We define an alphabetic set $\Omega = \{A, C, G, T, -\}$ to include the symbolic elements in this genomic region with A, C, G, T denoting the nucleotides and "$-$" denoting a missing nucleotide caused by deletions or insertions. Let R_i and G_i, $1 \leq i \leq L$ denote the symbol on base i for the reference sequence and for the genotype sequence, respectively. Then $R_i \in \Omega$, and G_i can take 15 possible diploid genotypes, enumerated as AA, AC, AG, AT, $A-$, CC, CG, CT, $C-$, GG, GT, $G-$, TT, $T-$, and $--$. In general, the relation between R_i and G_i, i.e., the state of base i, can be defined by "Match", "SNP", "Ins", and "Del" and we use a latent variable Z_i to describe this hidden state on base i. Depending on the value that R_i takes, two cases should be considered for Z_i:

(1) If $R_i \neq -$, the state can be a "Match", "SNP", or "Del" and correspondingly the hidden state variable Z_i can take 15 possible values in accordance with the 15 diploid genotypes, denoted by $s_j, j = 1, \ldots, 15$. For example, suppose the reference base $R_i = A$, then $s_1 = AA$ representing the state "Match", $s_{15} = --$ representing the state "Del" and other states may be considered as "SNP"s.

(2) If $R_i = -$, the state can be either an "Ins" or a not valid state and the hidden state variable Z_i can also take 15 values, denoted by $s_j, j = 16, \ldots, 30$. For example, suppose the reference base $R_i = -$, then $s_{30} = --$ is a not valid state and all other states are

considered as "Ins"s.

It is worth noting that the inference of the most likely genotype G_i is equivalent to finding the most likely Z_i, which directly indicates the occurrence of the variant—SNP or INDEL, on base i.

After defining the hidden states, we characterize transitions among the states by a transition matrix $T = \{t_{mn}\}, 1 \leq m \leq 30, 1 \leq n \leq 30$. Each component of the matrix is defined by $t_{mn} = P(Z_{i+1} = n | Z_i = m), 1 \leq i < L$, representing the probability of being in state n at the current base given the observed state m at the previous base. In our simulation studies, these transition probabilities are set by empirical values. In the analysis of real data, the transition matrix can be obtained by calculating the conditional frequencies of the variants from the NCBI dbSNP database (version 136) [86].

2.2.2 The emission probabilities

Emission probabilities govern the distribution of the observed data (a pileup of reads) at each base given the hidden state at that base. In vi-HMM, we first identify the bases on which the pileup of reads have size ≥ 5. Denoting these read data on base i by $D_i, 1 \leq i \leq L$, we write the probability (likelihood) of observing D_i given the hidden state Z_i as

$$P_i = L(Z_i | D_i) = \prod_{k=1}^{n_i} p(d_{ik} | Z_i), \quad Z_i \in \{s_1, \ldots, s_{30}\} \tag{2.1}$$

where d_{ik} represents the nucleotide on the kth read covering base i and n_i represents the size of the pileup on that base. Since each value taken by the hidden state Z_i corresponds to a specific underlying genotype G_i which contains two alleles A_1 and A_2, we further write the

probability of observing each $d_{ik}, 1 \leq i \leq L, 1 \leq k \leq n_i$ given Z_i as

$$\begin{aligned} p\left(d_{ik}|Z_i\right) &= p\left(d_{ik}|\left\{A_1, A_2\right\}\right) \\ &= \frac{1}{2}p\left(d_{ik}|A_1\right) + \frac{1}{2}p\left(d_{ik}|A_2\right) \end{aligned} \tag{2.2}$$

where the probability of observing d_{ik} given one allele $A \in \{A_1, A_2\}$ is

$$p\left(d_{ik}|A\right) = \begin{cases} \frac{e_{ik}}{4} & \text{if } d_{ik} \neq A \\ 1 - \frac{e_{ik}}{4} & \text{if } d_{ik} = A \end{cases}. \tag{2.3}$$

In the above expression (2.3), e_{ik} represents the sequencing error rate on base i for read k, which can be calculated from the reversed Phred scaled quality score in the SAM file. For simplicity, here we assume that the sequencing error on base i is caused by four possible point mutations (from allele A of G_i to the nucleotide of d_{ik} which may take four other symbols in set Ω) with equal probability. In particular, when "$-$" appears in a read (meaning a deletion in the CIGAR string of the SAM file) so the corresponding Phred quality score on that read base is missing, we take the average of all other reads' Phred quality scores on that base to impute the missing value.

We note that, the emission distribution can vary for different bases. Given n_i pileup reads on base i, for each possible combination of R_i and G_i, i.e., for $Z_i = s_j, 1 \leq s_j \leq 30$, the emission distribution at this base will be a discrete distribution which categorizes the pileup read data D_i into 15 groups corresponding to the possible diploid genotypes. Nevertheless, the probability of observing D_i given Z_i can be easily calculated through a multinomial probability mass function (PMF) by incorporating the sequencing error rates $e_{ik}, 1 \leq k \leq n_i$.

2.2.3 The optimal state sequence

With the HMM parameters identified, we use the Viterbi algorithm to find the optimal hidden state path $\mathbf{Z} = (Z_1, Z_2, \ldots, Z_n)$, which not only indicates the most likely genotypes but also can be used to call SNPs and INDELs directly.

2.3 Data sets

2.3.1 Simulated data

To evaluate the performance of vi-HMM, we use simulated datasets by two different processes: one introducing the position dependence, and one assuming position independence. In the first process, the simulation starts with generating a $50,000$ base-pairs (bp) genomic segment as the reference sequence. In order to take into account the spatial dependence in the genotype, we first generate a haplotype sequence based on an HMM with four states: "Match", "SNP", "Del", and "Ins". The transition matrix of this HMM is pre-specified (for details, please refer to https://github.com/tangmanhd/vi-HMM). The observed haplotype sequence (which takes value from the alphabetic set Ω) is determined by the emission distribution, which, for simplicity, is set to be discrete uniform conditioning on the hidden states. That is, for each base of the haplotype, the probability vector of observing a nucleotide symbol other than the corresponding nucleotide shown in the reference is $[1/3, 1/3, 1/3]$ if the hidden state is "SNP" and is $[1/4, 1/4, 1/4, 1/4]$ if the hidden state is "Ins" (note that the emissions for the other two hidden states are deterministic). Based on the generated haplotype sequence, the second haplotype can be generated by incorporating a pre-specified heterozygous rate. Once the haplotype pair is generated, we then generate the short-read data, half from each haplotype, by specifying the length, number of reads, and base quality. In the

second process, we randomly select a $50,000$-bp section of chromosome 21 as the reference sequence and then simulate paired end reads with wgsim (https://github.com/lh3/wgsim). In both processes, base-calling errors are considered to be stochastic and are generated from a uniform distribution. Then the base-calling error probabilities are transformed into Phred quality scores.

In each process, four datasets were generated at the 15×, 20×, 25×, and 30× sequencing depths, respectively. The simulated short reads are, on average, 100 bp long. All simulated reads are mapped to the reference sequence by using sequence alignment tools Bowtie2 (version 2.2.5) and BWA-MEM (version 0.7.12). After read alignment, we apply vi-HMM, GATK HaplotypeCaller (version 4.0), FreeBayes (version 1.1.0), Platypus (version 0.8.1), SAMtools (version 1.3), and VarScan (version 2.3.9) to these datasets for variant calling. Evaluation of the calling accuracy is based on the following criteria. For SNP calling, if the locus of a called SNP is exactly the same as the truth, this SNP is recorded as a true positive (TP); otherwise it is a false positive (FP). On the other hand, if a true SNP is not identified by the caller, it is a false negative (FN). For INDEL calling, if the called locus is the same as the the simulated truth, this INDEL is regarded as a TP, and the definitions of FP and FN are the same as those for calling SNPs. With these concepts, we calculate the sensitivity, precision, and F_1 score by:

$$\begin{aligned} sensitivity &= \frac{TP}{TP+FN} \\ precision &= \frac{TP}{TP+FP} \\ F_1 &= \frac{2TP}{2TP+FP+FN} \end{aligned} \tag{2.4}$$

This simulation procedure is repeated for 1,000 times to summarize the averages on sensitivity, precision, and F_1 score.

2.3.2 Real data

We further run vi-HMM to call SNPs and INDELs on a dataset from the GIAB project (NA12878, chr21: 1–48129895). This dataset consists of 19,020,457 mapped reads with the hs37d5 genome. The average length of reads is 100.9 bp and the average coverage is 54×. In order to evaluate the performance of variant calling at lower coverage, we downsample this real dataset to 15× and 30× sequencing depths and then apply several variant callers to the datasets accordingly. These three datasets are denoted as low (15×), medium (30×), and high (54×) coverage depths, respectively. A validation dataset by Zook et al. [100] is treated as the "ground truth" to evaluate the calling accuracy [100]. It should be noted that Zook et al. [100] has applied GATK in the process of obtaining these high-confidence variants. Therefore, to avoid biased comparison, we choose to not include GATK but only compare the vi-HMM calling results to those generated by the other four popular variant callers—FreeBayes, Platypus, SAMtools, and VarScan (version numbers of these callers are the same as those in simulations). The transition matrix of vi-HMM is pre-specified according to the conditional frequencies estimated from the NCBI dbSNP database (version 136) [86].

2.4 Results

2.4.1 Performance evaluation based on data simulated by HMM

The SNP calling results by the six variant callers (vi-HMM, GATK HaplotypeCaller, FreeBayes, Platypus, SAMtools, and VarScan) for the simulated data are shown in Figure 2.2. We observe that, when Bowtie2 is used for read mapping, vi-HMM achieves the highest sensitivity and F_1 score at every read coverage depth, indicating its good accuracy in detecting SNPs as compared to the other variant callers, especially at the low-coverage (15× depth)

setting (Figure 2.2a and 2e). The sensitivity of SNP calling by vi-HMM reaches 93.83%, whereas the second highest sensitivity by SAMtools is only 81.45% at the 15× depth; the F_1 score by vi-HMM (95.29%) is also much higher than that by SAMtools (89.27%). All six variant callers show high precision (above 95%) on this simulated data (Figure 2.2c). When BWA-MEM is used for read mapping, the sensitivities and F_1 scores by vi-HMM are also the highest across all read coverage depths. The sensitivity and F_1 score by SAMtools are comparable to those by vi-HMM. Again, high precision is observed for all six variant callers (Figure 2.2d).

For INDEL calling, with Bowtie2 mapping, the sensitivity and F_1 score by vi-HMM are the highest at every read coverage depth (Figure 2.3a and 3e) and the precision by vi-HMM is the second highest (Figure 2.3c), indicating the superiority of vi-HMM in detecting INDELs than the other variant callers. As the coverage depth increases, the INDEL calling accuracy of vi-HMM becomes higher. With BWA-MEM mapping, the sensitivity by vi-HMM is only slightly lower than those by Platypus and GATK HaplotypeCaller (Figure 2.3b). The precisions by vi-HMM are much higher than those by GATK HaplotypeCaller, FreeBayes, and Platypus and slightly lower than those by the other methods (Figure 2.3d). Overall, the F_1 score by vi-HMM reaches the highest at every read coverage depth (Figure 2.3f).

2.4.2 Performance evaluation based on wgsim simulated data

In general, vi-HMM also performs well at calling SNPs and INDELs on the data simulated by wgsim. For SNP calling, when Bowtie2 is used for read mapping, the sensitivity by vi-HMM is slightly lower than that by FreeBayes at the low-coverage (15× depth) setting but becomes the highest when the read coverage depth increases (Figure 2.4a). F_1 scores by vi-HMM and GATK HaplotypeCaller are the highest across all read coverage depths, with

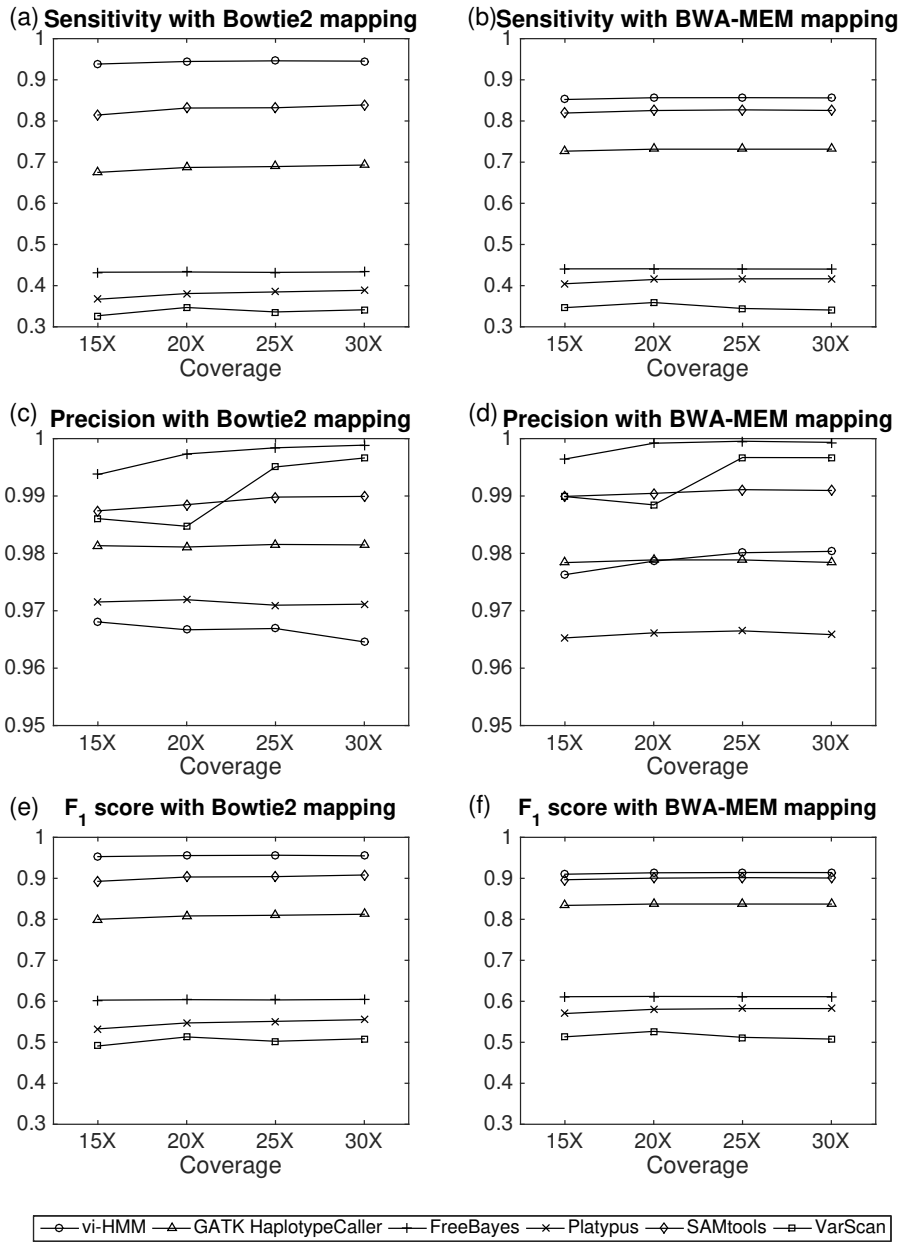

Figure 2.2: Comparison of SNP calling by different variant callers using data simulated by HMM at various sequencing depths.(a) sensitivity with Bowtie2 mapping, (b) sensitivity with BWA-MEM mapping, (c) precision with Bowtie2 mapping, (d) precision with BWA-MEM mapping, (e) F_1 score with Bowtie2 mapping, (f) F_1 score with BWA-MEM mapping.

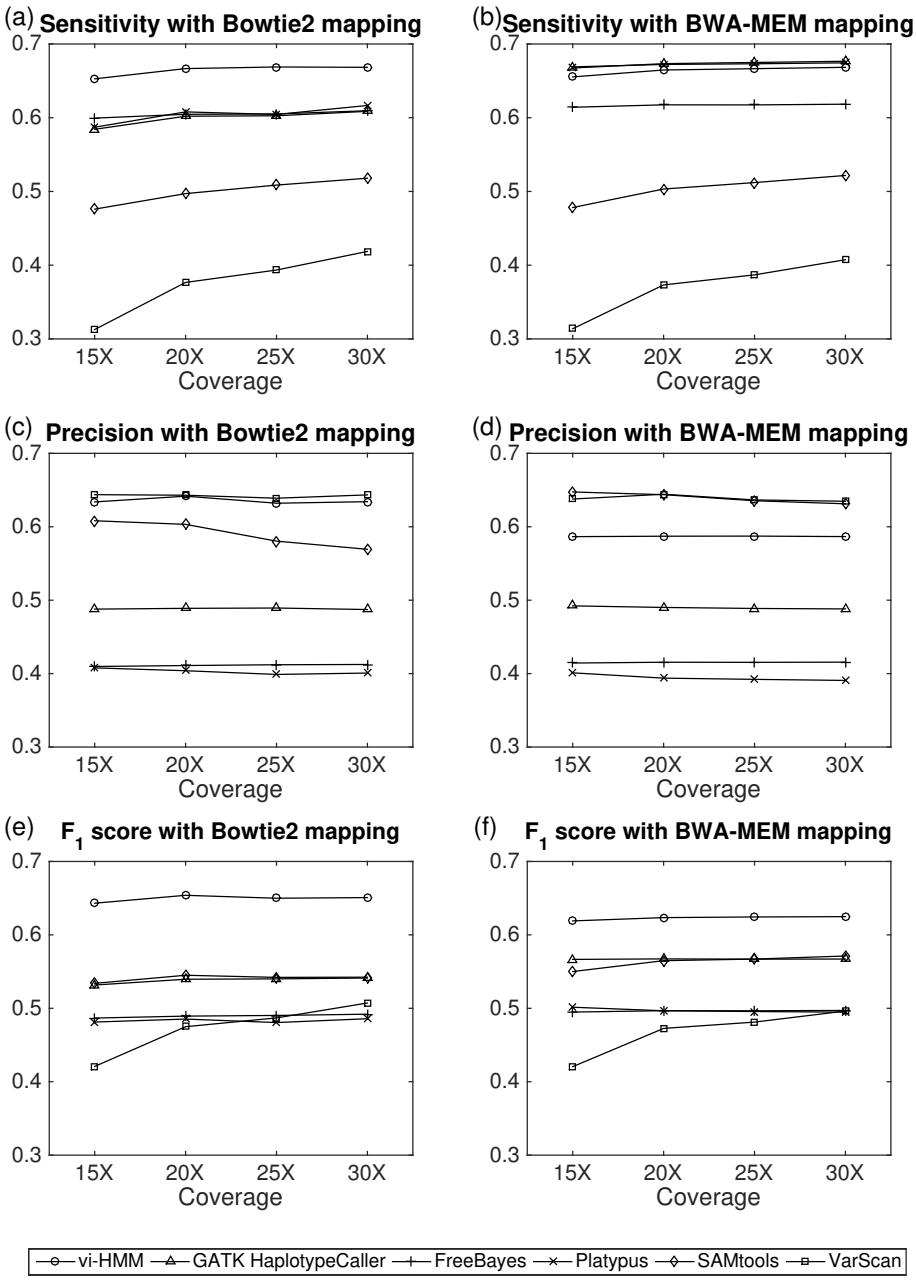

Figure 2.3: Comparison of INDEL calling by different variant callers using data simulated by HMM at various sequencing depths.(a) sensitivity with Bowtie2 mapping, (b) sensitivity with BWA-MEM mapping, (c) precision with Bowtie2 mapping, (d) precision with BWA-MEM mapping, (e) F_1 score with Bowtie2 mapping; (f) F_1 score with BWA-MEM mapping.

only subtle differences between the two (Figure 2.4e). When BWA-MEM is used for read mapping, the sensitivity by vi-HMM is the highest at the median- to high-coverage (20×, 25×, 30× depths) settings (Figure 2.4b). The F_1 scores by vi-HMM and SAMtools achieve the highest at every read coverage depth (Figure 2.4f). For INDEL calling, the sensitivity by vi-HMM is the highest at 15× and 20× depths with Bowtie2 mapping (Figure 2.5a). The F_1 scores by vi-HMM and GATK HaplotypeCaller reach the highest when the read coverage depth increases with both mapping methods (Figure 2.5e and 5f).

2.4.3 Application to whole-genome data for NA12878

The results of comparing the sensitivity, precision, and F_1 score between the five variant callers are shown in Table 2.1, by using real data at the 15×, 30× and 54× sequencing depths. For SNP calling, we observe that all five callers except FreeBayes achieve very high precision ($> 99\%$) at the three depths. Thus the differences in F_1 score are mainly driven by sensitivity, for which vi-HMM and SAMtools outperform the others especially at low (15×: both $> 95\%$) to medium (30×: both $> 99\%$) depths. For INDEL calling, it is obvious that vi-HMM produces the highest F_1 score over all other callers, and the superiority in F_1 score becomes more apparent at low (15×: vi-HMM $> 91\%$ whereas others $< 90\%$) to medium depths (30×: vi-HMM $> 95\%$ whereas other callers' F_1 scores range from 80.54% to 93.67%). We also note that among all five variant callers, vi-HMM is able to control the false positives and false negatives in a balanced way (i.e., achieve $> 90\%$ sensitivity and precision simultaneously) for both SNP and INDEL calling at all three depths, whereas others cannot (for example, FreeBayes and Platypus have low precision in INDEL calling, the sensitivity of SAMtools for calling INDELs is less competitive, and the sensitivity of VarScan for calling both SNPs and INDELs drops too fast at lower depths). We also apply the whole process to chromosome 22 and find vi-HMM and SAMtools achieve very high F_1 score at low (15×:

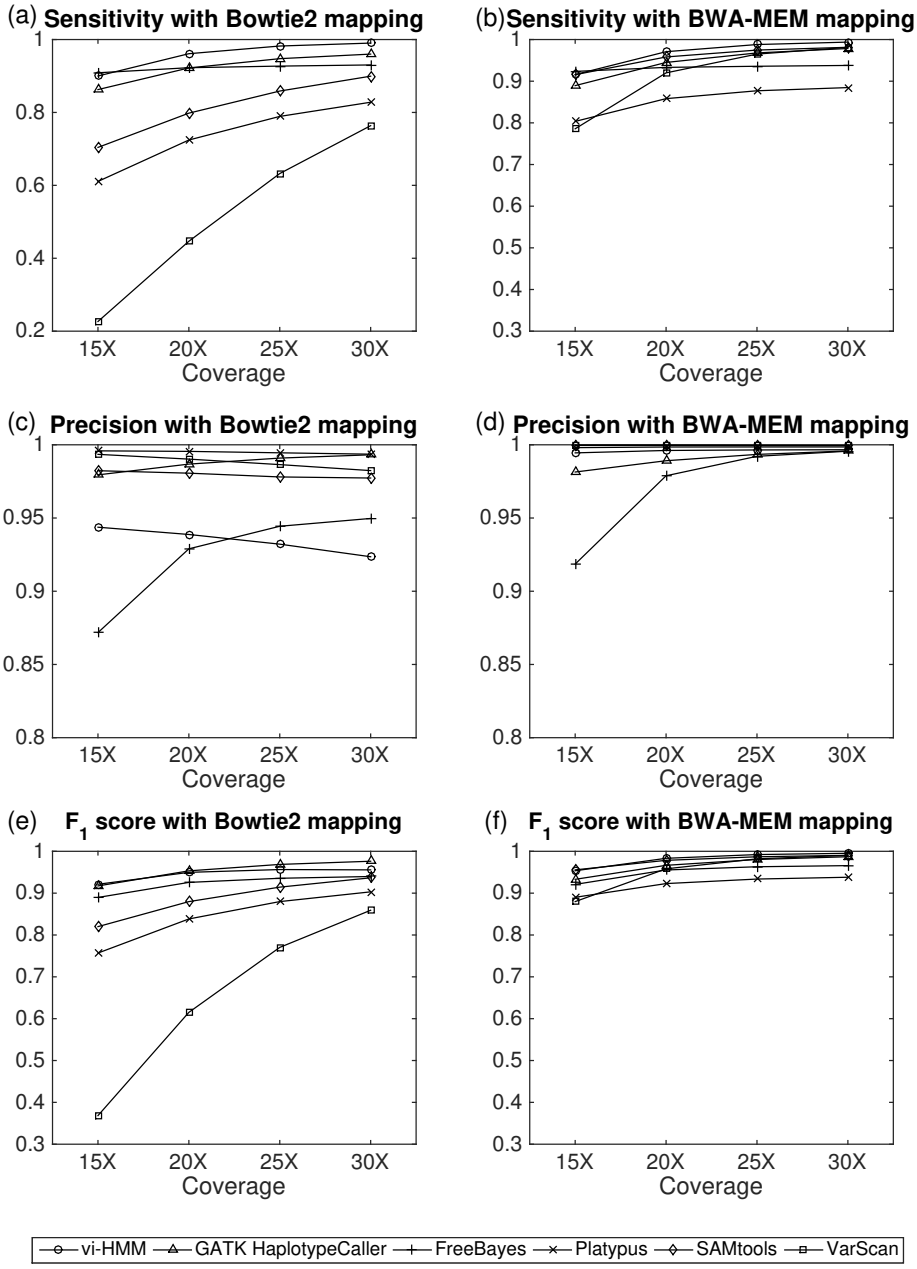

Figure 2.4: Comparison of SNP calling by different variant callers using wgsim simulated data at various sequencing depths.(a) sensitivity with Bowtie2 mapping, (b) sensitivity with BWA-MEM mapping, (c) precision with Bowtie2 mapping, (d) precision with BWA-MEM mapping, (e) F_1 score with Bowtie2 mapping, (f) F_1 score with BWA-MEM mapping.

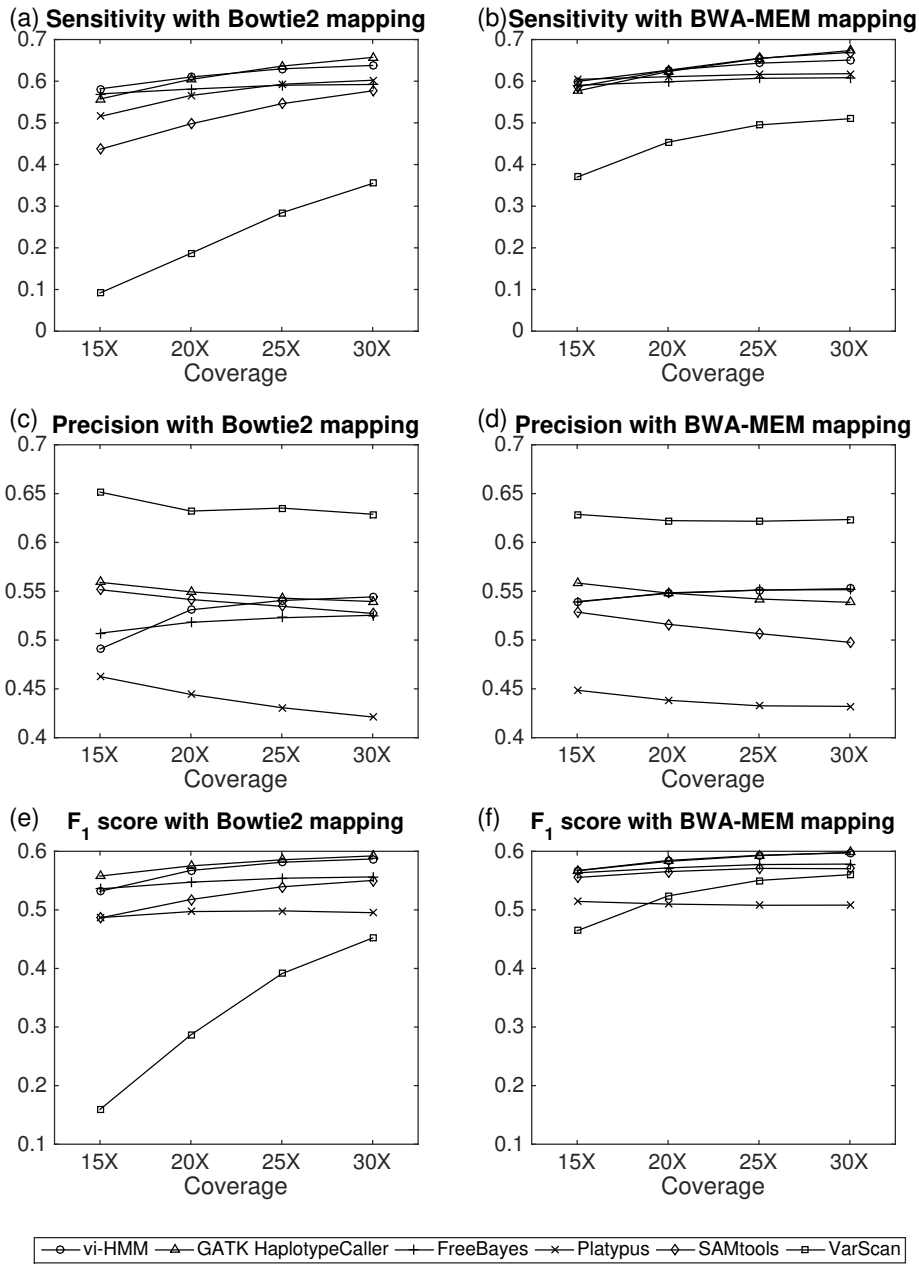

Figure 2.5: Comparison of INDEL calling by different variant callers using wgsim simulated data at various sequencing depths.(a) sensitivity with Bowtie2 mapping, (b) sensitivity with BWA-MEM mapping, (c) precision with Bowtie2 mapping, (d) precision with BWA-MEM mapping, (e) F_1 score with Bowtie2 mapping; (f) F_1 score with BWA-MEM mapping.

both $> 96\%$) to medium ($30\times$: both $> 99\%$) depths for SNP calling and vi-HMM has the highest F_1 score at low depth for INDEL calling (Appendix A.2). These comparisons provide us evidence that on the real datasets, vi-HMM represents an improvement over the other four variant callers in terms of calling SNPs and INDELs, as its performance gets closer to the recognized "ground truth"—which was obtained by GATK variant callers in practice.

Table 2.1: Comparison of different variant callers using real data.

	SNP			INDEL		
Caller	Sensitivity	Precision	F_1 score	Sensitivity	Precision	F_1 score
15X						
vi-HMM	95.11%	99.62%	97.31%	91.95%	90.18%	91.06%
FreeBayes	94.82%	91.61%	93.18%	88.93%	74.79%	81.25%
Platypus	90.97%	99.84%	95.20%	93.74%	70.03%	80.17%
SAMtools	98.66%	99.56%	99.11%	83.79%	95.45%	89.24%
VarScan	76.31%	99.87%	86.51%	74.00%	99.44%	84.85%
30X						
vi-HMM	99.81%	99.44%	99.63%	95.22%	95.62%	95.42%
FreeBayes	95.80%	95.48%	95.64%	90.36%	76.41%	82.80%
Platypus	92.92%	99.73%	96.21%	95.67%	69.54%	80.54%
SAMtools	99.64%	99.62%	99.62%	87.84%	93.23%	90.46%
VarScan	97.93%	99.82%	98.86%	88.59%	99.37%	93.67%
54X						
vi-HMM	99.95%	99.18%	99.56%	95.61%	96.09%	95.85%
FreeBayes	95.88%	96.90%	96.39%	90.77%	77.27%	83.48%
Platypus	92.97%	99.63%	96.18%	96.06%	69.11%	80.38%
SAMtools	99.70%	99.61%	99.65%	88.99%	90.53%	89.75%
VarScan	99.53%	99.77%	99.65%	91.67%	99.24%	95.31%

2.5 Discussion

In this chapter, we describe a new HMM-based method, vi-HMM, for accurate calling of SNP and INDEL variants in mapped reads. By taking advantage of the HMM features, vi-HMM allows us to detect variants directly through inferring an optimal hidden state path from

the observed pileup read data and the reference genome. Both simulation studies and real data analysis have confirmed that vi-HMM is able to improve the accuracy of SNP/INDEL identification as compared to other variant callers, especially at low and medium depths.

As an important step in NGS data analysis, variant calling has received much attention in bioinformatics research. Although a number of variant calling methods have been developed, it remains unclear how different model assumptions used in these methods affect their practical performance. In general, the performance of a variant caller can be evaluated through either real data analysis or simulations. Real data analysis is able to reveal features of the variant caller under different settings (sequencing platforms, coverage depths, etc), however, due to lack of "ground truth" on experimentally validated variant sets in real data, the results of false positives and false negatives in variant identification are often arguable. Simulation studies, on the other hand, provide strong evidences for evaluation of a variant caller or comparison among variant callers. However, the simulated data need to be justified to have similar characteristics as real data in order to guarantee that the conclusions still remain meaningful in real data scenarios.

In the present work, we have performed both simulations and real data analysis to evaluate the proposed variant caller vi-HMM and compare it with other commonly used callers. Interestingly, we found something in common in the two sets of calling results (at 15× and 30× depths using both simulated and real data): (1) Overall, vi-HMM and SAMtools have higher F_1 score than FreeBayes, Platypus, and VarScan, in both SNP calling and INDEL calling; (2) The precision for most variant callers are very high in SNP calling; (3) When sequencing depth increases from low (15×) to medium (30×), most variant callers have better calling performance; (4) The sensitivity and precision for vi-HMM are balanced and remain high across different depths, whereas for the other variant callers they could be very unbalanced (e.g., Platypus and FreeBayes in INDEL calling) or easily influenced by low

depth of the data (e.g., the fast dropping of VarScan sensitivity in INDEL calling from 30× to 15×). These findings in variant calling performance indicate that our simulated data share some similarities with the real data, and both demonstrate that our proposed method, vi-HMM, has a good performance overall and is applicable not only to med-to-high read coverage depth but also to low read coverage, with robust performance.

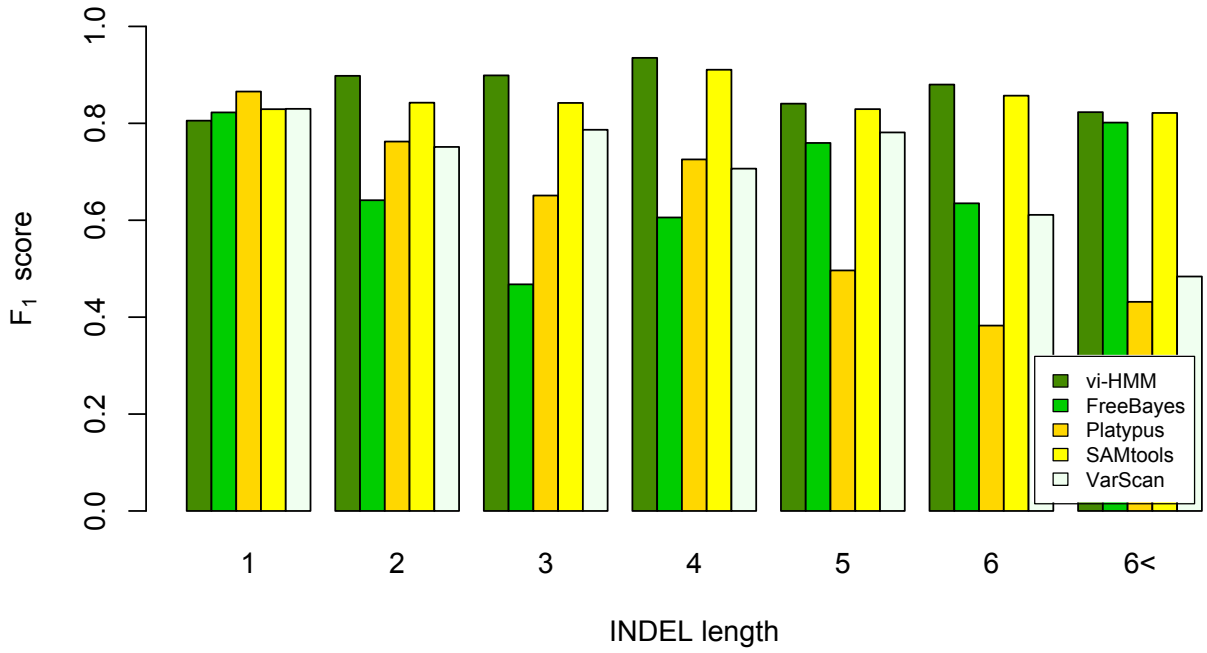

Figure 2.6: F_1 scores by vi-HMM, FreeBayes, Platypus, SAMtools, and VarScan at different INDEL lengths on real data with 15× depth on chromosome 21

Particularly, for the two "better performers" vi-HMM and SAMtools, we also see the differences between their SNP calling and INDEL calling. While they both have high sensitivity, precision, and F_1 score in simulations and real data analysis, vi-HMM does not display remarkable superiority in calling SNPs. This may be because the state "SNP" is more likely to move to "Match" (94.99% from dbSNP) rather than to another variant across the genome. Thus the dependence between "SNP" and the adjacent variants becomes negligible and plays a less important role in SNP calling. However, in terms of INDEL calling, vi-HMM certainly

outperforms SAMtools. This could be possibly explained by the fact that the transition probability from the state "Ins" to "Match" is only 28.80% (data from dbSNP), indicating that there exists strong dependence between "Ins" and the adjacent variants and therefore vi-HMM should have a better performance in calling INDELs by considering such state dependence between adjacent genomic bases.

Another observation in real data analysis is that, when using vi-HMM, the F_1 score for calling INDELs decreases with the increase of INDEL length in general. Figure 2.6 shows the F_1 scores by vi-HMM, FreeBayes, Platypus, SAMtools, and VarScan at different INDEL lengths on real data with 15× depth on chromosome 21. It shows that the F_1 score by vi-HMM retains larger than 80%. Comparing with the other variant callers, vi-HMM appears the highest F_1 score at INDEL length from 1 to 6, indicating that this HMM-based method appears to be more accurate in detecting short INDELs.

Noteworthy, the accuracy of variant calling also depends on the quality of read alignment. In general, the occurrence of INDELs in reads may shift the alignments and result in mismatch [52], which may impact the subsequent variant calling procedure remarkably. This is especially true for large INDELs. As seen from our simulation studies, vi-HMM produces higher sensitivity and F_1 score with Bowtie2 than it does with BWA-MEM at every read coverage depth on the reads simulated by HMM. One plausible explanation is that Bowtie2 performs better than BWA-MEM in the read alignment (further examination of the two aligners on correct mapping, multiple alignment, second alignment, soft/hard clipped reads is included in Appendix A.3). Such a phenomenon of variant calling being influenced by read alignment can also be observed in a simulated dataset with homopolymers (Appendix A.4). It is thus important to choose an alignment tool that produces high quality mapping prior to variant calling.

2.6 Conclusion

In conclusion, we have developed a novel HMM-based method for sequence variant identification in short-read data. This variant caller provides an effective solution to modeling the dependence of adjacent genomic loci, which is expected to be useful for accurate calling of variants but is often overlooked in existing tools. To evaluate the performance of calling SNPs and INDELs in synthetic and real sequencing data, we compared the new variant calling method, vi-HMM, with six prevalent methods (GATK UnifiedGenotyper, GATK HaplotypeCaller, FreeBayes, Platypus, SAMtools, and VarScan) in simulation studies and with four (FreeBayes, Platypus, SAMtools, and VarScan) in real data analysis. Both comparison results demonstrate that vi-HMM is able to identify SNP and INDEL variants in a more accurate (overall high F_1 score), reliable (smaller fluctuations across different read coverage depths), and balanced (both good sensitivity and good precision) way, as compared to the other variant callers.

Chapter 3

Identifying transcriptional regulatory patterns from multiple ChIP-seq profiles

3.1 Introduction

Gene expression is mainly controlled by thousands of TFs that bind to specific DNA sequences. A transcription factor protein can bind and regulate target genes by recognizing a short (6-32 bp) sequence of nucleotides called transcription factor binding site (TFBS) [25]. This binding is able to active or inhibit the transcription of a certain gene. When multiple TFs bind to the genome, the binding patterns become more complex due to the interactions among TFs and between TFs and the genome. Thus, uncovering the protein-protein interactions, or the clustering of multiple-TF bindings, is of great importance in understanding the transcriptional regulation of a particular gene.

Currently, the ChIP-seq technology has been applied to determine the regulatory actions of a TF by generating millions of short reads covering the TFBSs across the genome in a given tissue or cell. Some software (e.g., MACS [96]) have been developed to call transcription factor binding regions from a ChIP-seq profile. With the accumulation of such binding data for multiple TFs, computational methods have been developed to identify binding patterns formed by set of TFs in a particular cell type. Recently, Cha and Zhou (2014) utilized inhomogeneous Poisson process to estimate TFBS distributions and then applied Ripley's K-function to detect pairwise TF binding patterns [11]. In another study, a hierarchical model, SignalSpider, was used to reveal higher-order combinatorial gene transcription patterns from multiple ChIP-seq datasets [91]. Nonetheless, these approaches can be sensitive to the *ad hoc* choice of the number of TF clusters.

In ChIP-seq data processing, peak calling algorithms can be used to find sets of locations at which a transcription factor binds to the genome. The occurrence of such binding events along the genome is often modeled by spatial point processes. Among the point process models, the Poisson process is fundamental to analyze point pattern data. Due to the

structural features of the TFBSs data, a more flexible process, the log Gaussian Cox process (LGCP), is considered to be more appropriate for modeling TF bindings and hence is used in this study. Following the LGCP setting, the binding site locations of a TF are assumed to follow an inhomogeneous Poisson process with an intensity, which, after log transformation, is assumed to follow a Gaussian process. Despite its flexibility, the fitting of such an LGCP model is computationally challenging [72]. Numerous approaches have been developed to estimate the LGCP model and the most commonly used methods are based on Markov chain Monte Carlo (MCMC) algorithms for Bayesian inference [1, 22, 66]. However, the main disadvantage of these methods lies in the expensive computations. Recently, a fast and flexible framework, named INLA (an acronym for integrated nested Laplace approximation), has been developed to fit LGCP [83]. INLA uses Laplace approximation and numerical integration to construct an approximation of the true LGCP likelihood to infer posterior marginal distributions and thereby reducing the computational complexity.

Unlike the previous models which require the number of clusters to be specified in advance, we propose a nonparametric Bayesian approach for clustering multiple-TF bindings in ChIP-seq data. The proposed method assumes an infinite number of latent clusters, whereas only finite number of them are used to generate the observed data. Therefore, it can be used to discover the latent cluster structure of a dataset and allows future data to be assigned to previously unseen clusters. The Dirichlet process mixture model is a commonly used nonparametric Bayesian model that has countable, infinite number of components [34, 41]. In our model, the Dirichlet process is considered as the nonparametric prior of the LGCP with infinitely many components.

To this end, we propose a Dirichlet process mixture of log Gaussian Cox process (DPM-LGCP) model to detect the clustering pattern of multiple-TF bindings from their ChIP-seq data. Details of this model, as well as the Bayesian inference procedure will be introduced

in Section **3.2**. In Section **3.3**, we perform simulation studies and based on the simulated datasets compare the performance of five clustering methods, namely DPM-LGCP, the K-function method by Cha and Zhou [11],Gaussian mixture model (GMM), k-means, and k-means with adjusted center. In Section **3.4**, we apply these methods to ChIP-seq data from mouse embryonic stem cells (ESC) available for 12 TFs and detect transcriptional regulatory modules in the promoter regions. Finally, Section **3.5** finishes this Chapter with discussions.

3.2 Methods

Our method is designed for discovering TF binding patterns from ChIP-seq data. We first use a log Gaussian Cox process (LGCP) to model the binding pattern of a single TF. Since our purpose is to group TFs into a number of latent clusters, a nonparametric Bayesian mixture model—the Dirichlet process mixture is applied to infer the TF clusters.

3.2.1 Log Gaussian Cox processes

Consider a bounded region $\Omega \subset \mathbb{R}$. Let $S = \{s_1, ..., s_n\}$ denote a set of binding site (BS) locations of a TF in the region Ω. We assume that these BS locations follow an inhomogeneous Poisson process with intensity function $\lambda(s), s \in \Omega$, then the likelihood of observing S can be written as:

$$f(S|\lambda(s)) = \exp\left\{|\Omega| - \int_{\Omega} \lambda(s)ds\right\} \prod_{i=1}^{n} \lambda(s_i) \tag{3.1}$$

where $|\Omega|$ is the length of the bounded region. We further use LGCP to model the intensity function $\lambda(s)$:

$$\begin{aligned} z(s) &= \log(\lambda(s)) \\ z(s) &\sim GP(0, C_\theta) \end{aligned} \tag{3.2}$$

where the log intensity $z(s)$ is a Gaussian random field with mean 0 and covariance kernel C_θ, which satisfies $Cov(z(s_l), z(s_m)) = C_\theta(s_l, s_m)$. Here, θ contains parameters that control the shape of the covariance kernel.

3.2.2 Dirichlet process mixture

Suppose we obtain the BS locations for N TFs, denoted by $S_i = \{s_{i1}, ..., s_{in_i}\}$ $(1 \leqslant i \leqslant N)$. Our purpose is to find the grouping structure of these TFs, that is, to reveal the latent clusters that these TFs belong to. We use a finite mixture model and assume that the N TFs come from K clusters. Since we do not know the number of clusters a priori, we propose a nonparametric Bayesian approach which enables automatic clustering of the TFs without prespecifying the number of latent clusters.

The proposed approach is called a Dirichlet process mixture of log Gaussian Cox process (DPM-LGCP), described as follows:

$$\begin{aligned} S_i|\lambda_i(s) &\sim inhomo - Poisson(\lambda_i(s)) \quad s \in \Omega, \quad i = 1, \dots N \\ z_i(s) &= \log(\lambda_i(s)) \\ z_i(s) &\sim G \\ G &\sim DP(m, G_0) \\ G_0 &= GP(0, C_\theta) \end{aligned} \tag{3.3}$$

where m and G_0 are the parameters of the Dirichlet process, and m represents the concentration parameter and G_0 represents the base measure of the Dirichlet process.

3.2.3 Approximating the likelihood using INLA

The main difficulty of the proposed DPM-LGCP model is the calculation of the marginal likelihood of the point process S_i. Furthermore, the integral in the likelihood is complicated by the stochastic nature of $\lambda_i(s)$. To improve the efficiency of posterior sampling, we employ the INLA packages in R to approximate the likelihood of the DPM-LGCP model [83]. INLA adopts an explicit link between the latent Gaussian process $z(s)$ and a discrete spatial Gaussian Markov random field (GMRF) $\mathbf{z} = (z_1, \ldots, z_p)$ [54], allowing us to approximate $z(s)$ with weighted sums of simple basis functions. In particular, it assumes that the covariance kernel of $z(s)$ follows Matérn covariances,

$$C_\theta(x, y) = \frac{\sigma^2}{\Gamma(\nu)\, 2^{\nu-1}} \left(\kappa |x - y|\right)^\nu K_v \left(\kappa |x - y|\right) \tag{3.4}$$

where $K_v(\cdot)$ is the modified Bessel function of the second kind, $\nu > 0$ is the smoothing parameter, $\kappa > 0$ is the range parameter, and σ^2 is the marginal variance. Then $z(s)$ is the solution of the stochastic partial differential equation (SPDE):

$$\left(\kappa^2 - \Delta\right)^{\alpha/2} z(s) = W(s) \tag{3.5}$$

where $\alpha = \nu - d/2$ is an integer (d is the dimension of the space and $d = 1$ in this case), $\Delta = \partial^2/\partial s^2$ is the Laplacian operator and $W(s)$ is spatial white noise. Based on spectral theory, the integer value of α gives continuous domain Markov fields [82] and thus the default value is $\alpha = 2$ in INLA. The solution of $z(s)$ to SPDE implemented in INLA is approximated on a basis representation:

$$z(s) = \sum_{j=1}^{P} \phi_j(s)\, z_j \tag{3.6}$$

where $\phi_j(\cdot)$ are the basis functions of B-spline of degree 1 (piecewise linear) and $\mathbf{z} = (z_1, \ldots, z_P)^T$ is the Gaussian distributed weights. Let Q be the precision matrix for the Gaussian weights $\mathbf{z}$. For the default case $\alpha = 2$, the precision matrix is given by

$$Q = K_\kappa C^{-1} K_\kappa \tag{3.7}$$

where $C_{i,j} = \langle \phi_i, \phi_j \rangle$, $G_{i,j} = \langle \bigtriangledown \phi_i, \bigtriangledown \phi_j \rangle$, and $K_\kappa (K_\kappa)_{i,j} = \kappa^2 C_{i,j} + G_{i,j}$. The weight vector is chosen from Gaussian distribution $x \sim N(\mathbf{0}, \boldsymbol{Q}^{-1})$ so that the distribution of $z(s)$ approximates the distribution of solutions to the SPDE.

In our LGCP, we implement INLA by constructing a regular grid $\mathbf{t} = (t_1, \ldots, t_P)$ in Ω and construct a 1-D B-spline of degree 1 on the grid. Then the intensity function can be expressed as:

$$\lambda(s) = \exp(z(s)) \approx exp\left(\sum_{j=1}^{P} \phi_j(s) z_j\right) \tag{3.8}$$

We can further approximate the integral in equation 3.1 by

$$\int_\Omega \lambda(s)\, ds \approx \sum_{k=1}^{P} \lambda(t_k) \tilde{\alpha}_k \approx \sum_{k=1}^{P} \exp\left\{\sum_{j=1}^{P} \phi_j(t_k) z_j\right\} \tilde{\alpha}_k \tag{3.9}$$

where $\tilde{\alpha}_k$ is the weight for numerical integration that takes the value $|\Omega_k|$ where Ω_k is the support of $\phi(\cdot)$.

The exponential part of equation 3.1 is approximated by

$$\begin{aligned} \exp\left\{|\Omega| - \int_\Omega \lambda(s)\, ds\right\} &\approx \exp\{|\Omega|\} \exp\left\{-\sum_{k=1}^{P} \tilde{\alpha}_k \exp\left\{\sum_{j=1}^{P} \phi_j(t_k) z_j\right\}\right\} \\ &= \exp\{|\Omega|\} \exp\left\{-\tilde{\boldsymbol{\alpha}}^T \exp(\mathbf{A}_1 \mathbf{z})\right\} \end{aligned} \tag{3.10}$$

where $\tilde{\boldsymbol{\alpha}} = (\alpha_1, \ldots, \alpha_P)^T$ and $[\mathbf{A}_1]_{k,j} = \phi_j(t_k), k = 1, \ldots, P, j = 1, \ldots, P$. The product part

of equation 3.1 is approximated by

$$\begin{aligned}\prod_{i=1}^{n}\lambda\left(s_i\right) &\approx \prod_{i=1}^{n}\exp\left\{\sum_{j=1}^{P}\phi_j\left(s_i\right)z_j\right\}\\ &= \exp\left\{\sum_{i=1}^{n}\sum_{j=1}^{P}\phi_j\left(s_i\right)z_j\right\}\\ &= \exp\left\{\mathbf{1}^T\mathbf{A}_2\mathbf{z}\right\}\end{aligned} \tag{3.11}$$

where $[\mathbf{A}_2]_{i,j} = \phi_j(s_i), i = 1,\ldots,n, j = 1,\ldots,P$. Then the log likelihood based on equation 3.1 can be approximated by

$$\log f(S|\lambda(s)) \approx |\Omega| - \tilde{\boldsymbol{\alpha}}^T exp\left(\mathbf{A}_1\mathbf{z}\right) + \mathbf{1}^T\mathbf{A}_2\mathbf{z} \tag{3.12}$$

Let $\boldsymbol{\eta} = exp\left(\mathbf{z}^T\mathbf{A}_1^T, \mathbf{z}^T\mathbf{A}_2^T\right), \boldsymbol{\alpha} = \left(\tilde{\boldsymbol{\alpha}}, \mathbf{0}_{n\times 1}^T\right)$ and construct some pseudo-observations $\boldsymbol{y} = \left(\mathbf{0}_{P\times 1}^T, \mathbf{1}_{n\times 1}^T\right)$. The likelihood can be approximated by

$$f(\boldsymbol{y}|\boldsymbol{z}) \approx C\prod_{i=1}^{P+n}\left(\eta_i\right)^{y_i} exp\left(-\alpha_i\eta_i\right) \tag{3.13}$$

where C is a constant.

INLA approximates the posterior distribution of $\pi(\mathbf{z}|\mathbf{y})$ by numerical integration:

$$\tilde{\pi}(z_j|\mathbf{y}) = \int \tilde{\pi}(z_j|\boldsymbol{\theta},\mathbf{y})\tilde{\pi}(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta} \tag{3.14}$$

The marginal posterior $\tilde{\pi}(\boldsymbol{\theta}|\mathbf{y})$ is an approximation to $\pi(\boldsymbol{\theta}|\mathbf{y})$ by the following equation:

$$\tilde{\pi}(\boldsymbol{\theta}|\mathbf{y}) \propto \left.\frac{\pi(\mathbf{z},\boldsymbol{\theta},\mathbf{y})}{\tilde{\pi}_G\left(\mathbf{z}|\mathbf{y},\boldsymbol{\theta}\right)}\right|_{\mathbf{z}=\mathbf{z}^*(\boldsymbol{\theta})} \tag{3.15}$$

where $\mathbf{z}^*(\boldsymbol{\theta})$ is the posterior mode of $\mathbf{z}$ at a given $\boldsymbol{\theta}$ and $\tilde{\pi}_G(\mathbf{z}|\mathbf{y},\boldsymbol{\theta})$ is a Gaussian approximation to $\pi(\mathbf{z}|\mathbf{y},\boldsymbol{\theta})$.

INLA approximates the marginal likelihoods by

$$\tilde{\pi}(\mathbf{y}) = \int \frac{\pi(\mathbf{y},\boldsymbol{\theta},\mathbf{z})}{\tilde{\pi}_G(\mathbf{z}|\mathbf{y},\boldsymbol{\theta})}\Big|_{\mathbf{z}=\mathbf{z}^*(\boldsymbol{\theta})} d\boldsymbol{\theta} \tag{3.16}$$

The integration of $\boldsymbol{\theta}$ is performed by using numerical integration [36]. We denote the marginal likelihood of S_i as $H(S_i) \approx \tilde{\pi}(\mathbf{y})$. Since we assume that given the intensity function the BS locations for each TFs are independent, the joint likelihood can be expressed by $f(S_i, S_j|\mathbf{z}) = f(S_i|\mathbf{z})f(S_j|\mathbf{z})$. Similarly, the marginal likelihood $H(S_i, S_j)$ can be calculated by INLA.

3.2.4 Algorithm for sampling from Dirichlet Process mixture model

Gibbs sampling is commonly used for drawing samples from the posterior distribution of the Dirichlet Process mixture model. This approach is straightforward, easily implemented and converges fast. Historically, several Gibbs sampling approaches have been proposed to sample from the Dirichlet Process mixture model[23, 73]. We consider a procedure following Neal [73, Algorithm 3]. The $\boldsymbol{z}$ is integrated out from the algorithm and the state of Markov chain consists only c, the latent cluster states.

The Markov chain is initialized by setting the latent cluster states $c = (c_1, \ldots, c_N)$, for example, all the c_i set to K different clusters. Let K^- be the resulting number of clusters and n_k^- be the cluster sizes. We update c_i by repeatedly drawing a new value for each c_i using the following conditional probability:

$$(c_i \mid c_{-i}, \{S_i\}) = \begin{cases} k & \text{with prob.} \propto n_k^- H(S_i|S_l, c_l = k, l \neq i), k = 1, \ldots, K^- \\ K^- + 1 & \text{with prob.} \propto mH(S_i) \end{cases},$$

where c_{-i} represents c_l for $l \neq i$ and $H(S_i|S_l, c_l = k, l \neq i) = H(S_i, S_l, c_l = k, l \neq i)/H(S_l, c_l = k, l \neq i)$. All components are calculated using INLA.

3.3 Simulation study

To evaluate the performance of the DPM-LGCP model, we use simulated datasets by two different processes: one assuming the TFBSs are from inhomogeneous Poisson processes, and the other assuming the TFBSs are from finite Gaussian mixture models.

In the first process, there are several steps to generate the simulation data:

(1) Generate intensity functions from a log Gaussian process with mean zero and squared exponential covariance function $K(x, x') = \exp\left\{-\frac{\|d\|^2}{2l^2}\right\}$.

(2) Prespecify the clusters that the TFs belong to (the number of unique clusters is the sames as the number of intensity functions), for example, we generate BS locations for 20 TFs and set seven of them from one cluster, seven from the second cluster and the remaining six from the third cluster.

(3) Generate homogeneous Poisson process events on a finite interval.

(4) Reject an appropriate fraction of the generated events based on corresponding intensity function and the remaining events form the BS locations for each TF.

In the second process, we first specify three Gaussian mixture distributions and assume that each of them has three normal components with mean $\{6, 10, 14\}$ and variance $\{1, 1, 1\}$. The weights of these components in the three Gaussian mixture distributions are $\{0.6, 0.2, 0.2\}$, $\{0.2, 0.6, 0.2\}$, and $\{0.2, 0.2, 0.6\}$, respectively. Then we sample from the Gaussian mixture distribution and these samples are considered as the BS locations for one TF. In total, we generate BS locations for 20 TFs and set seven of them from the first Gaussian mixture distribution, seven from the second distribution and the remaining six from the third distribution.

In each process, we simulate 1000 data sets and apply the DPM-LGCP model to do clustering. We also apply four other methods, i.e. the Ripley's K-function method [11], GMM, k-means and k-means with adjusted center, to analyze the simulated data for comparison.

The overall idea behind the Ripley's K-function method is to vary distant t over a wide range, obtain a K-function to summarize the clustering pattern on all distance thresholds and pick the pattern with the largest value of K-function. The K-function is defined by:

$$\widehat{K}_{\mathbf{S_i S_j}}(t) = \frac{1}{L} \sum_{a=1}^{n_i} \sum_{b=1}^{n_j} \frac{w(s_a, s_b)^{-1} I(d_{s_a,s_b} \leq t)}{\lambda_{\mathbf{S_i}}(s_a) \lambda_{\mathbf{S_j}}(s_b)} \tag{3.17}$$

where $\mathbf{S_i} = (s_1, \ldots, s_{n_i})$ and $\mathbf{S_j} = (s_1, \ldots, s_{n_j})$ are the BS locations for TF i and TF j, $I(\cdot)$ is the indicator function, $w(s_a, s_b)$ is a weight function for edge correction, d_{s_a,s_b} is the distance between s_a and s_b and $\lambda_{\mathbf{S_i}}$, $\lambda_{\mathbf{S_j}}$ are the intensity functions for TFs i and j. From this function, if the BSs of two TFs bind in a same cluster, one would expect a large value of $\hat{K}_{\mathbf{S_i S_j}}(t)$.

To implement GMM, k-means and k-means with adjusted center to cluster TFs, we need to first discretize the intensity function for each TF by INLA so that the representation of intensity function becomes a vector. Therefore, these methods can be used to partition the

N vectors into k (prespecified) sets. For the k-means with adjusted center, the new center of cluster k is the intensity function calculated by INLA given BS locations of TFs belonging to cluster k.

In the simulation study, we know the 'ground truth' labels for TFs and thus external measures, such as the precision, recall, and F_1 score can be used to evaluate the clustering results. If the estimated pair of TFs is truly in the same cluster, this pair is recorded as a true positive (TP); otherwise, it is a false positive (FP). If the true pair of TFs is not identified, it is recorded as a false negative (FN). We define the precision, recall and F_1 score as follows:

$$P = \frac{\#\{\text{TP}\}}{\#\{\text{TP}\} + \#\{\text{FP}\}}$$
$$R = \frac{\#\{\text{TP}\}}{\#\{\text{TP}\} + \#\{\text{FN}\}}$$

$$F_1 = \frac{2 \cdot P \cdot R}{P + R}$$

where P is the precision rate and R is recall rate.

Table 3.1 shows the average precision, recall and F_1 score over 1000 datasets for DPM-LGCP, K-function, GMM, k-means and k-means with adjusted center using the data simulated from inhomogeneous Poisson processes. We observe that DPM-LGCP achieves the highest values of precision, recall and F_1 score, followed by GMM. The method of K-function yields the lowest values of precision, recall and F_1 score. Although the F_1 scores of GMM, k-means and k-means with adjusted center are higher than 90% , the number of clusters need to be specified in advance in these algorithms, making them less attractive. Thus, DPM-LGCP performs best among these methods.

Table 3.2 shows the average precision, recall and F_1 score over 1000 datasets for DPM-LGCP, K-function, GMM, k-means and k-means with adjusted center using the data simulated from

Table 3.1: Performance of TF clustering by DPM-LGCP, Ripley's K-function, Gaussian mixture model (GMM), k-means, and k-means with adjusted center using the data simulated from inhomogeneous Poisson processes.

Method	Precision	Recall	F_1 score
DPM-LGCP	99.15%	99.62%	99.55%
K-function	77.24%	88.47%	81.94%
GMM	95.92%	96.61%	96.25%
k-means	93.76%	95.64%	94.65%
adj.k-means	91.67%	94.26%	92.86%

GMMs. We observe that DPM-LGCP, k-means and k-means with adjusted center achieves very high precision, recall and F_1 score values ($> 99\%$), followed by GMM, which indicate that DPM-LGCP, k-means and k-means with adjusted center perform better than the baseline in this setting. Still the method of K-function yields the lowest values of precision, recall and F_1 score.

Table 3.2: Performance of TF clustering by DPM-LGCP, Ripley's K-function, Gaussian mixture model (GMM), k-means, and k-means with adjusted center using the data simulated from GMMs.

Method	Precision	Recall	F_1 score
DPM-LGCP	99.30%	99.73%	99.45%
K-function	31.26%	52.74%	38.62%
GMM	86.43%	93.381%	89.20%
k-means	99.84%	99.86%	99.85%
adj.k-means	99.63%	99.67%	99.65%

To test whether the clustering results of DPM-LGCP are sensitive to the value of m, the concentration parameter of the DP prior, we assume $m \sim Gamma(1, 1)$ in the DP and re-run the simulation analysis. Again, DPM-LGCP achieve very high values of precision (99.48%), recall (99.57%) and F_1 score (99.52%). We cannot see any significant difference between this result and that of DPM-LGCP with pre-specified m. Since it is relatively computational expensive to do clustering by assuming hyper-prior for parameter m, the

DPM-LGCP model with $m = 1$ is preferred.

3.4 Real data analysis

ChIP-seq data for 12 TFs (E2f1, Esrrb, Klf4, Nanog, Oct4, Stat3, Smad1, Sox2, Tcfcp2l1, Zfx, c-Myc and n-Myc) in mouse embryonic stem cells (ESCs) are obtained from NCBI GEO with the accession number GSE11431 [15]. Considering the regulatory function features in different genome regions, here we choose the promoter regions (defined as 2500 bp upstream and 500 bp downstream from the transcription start sites (TSSs) [15, 33]) on chromosome 18 as an example region and apply the DPM-LGCP model to discover TF binding patterns on this region.

For the real data, we do not have the 'ground truth' labels and thus the SD index, an internal measure, is used to evaluate model performance. The SD index considers the average scattering of clusters and the total separation of clusters [30]. Similar criteria for evaluating the goodness of clustering include the DaviesBouldin index, Dunn index, and Silhouette coefficient, etc. The details of SD index calculation can be found in Appendix B.1. The smaller the index, the better the clustering result.

Figure 3.1 shows the clustering results of TFs in the promoter regions. Many TF co-occupancies in the clustering results have been well recognized previously, such as Nanog–Oct4–Sox2 and n-Myc–c-Myc–Zfx (Figure 3.1) [15]. The MCMC algorithm converges fast and DPM-LGCP achieves the smallest value of SD index in the promoter for the real data analysis as seen in Table 3.3.

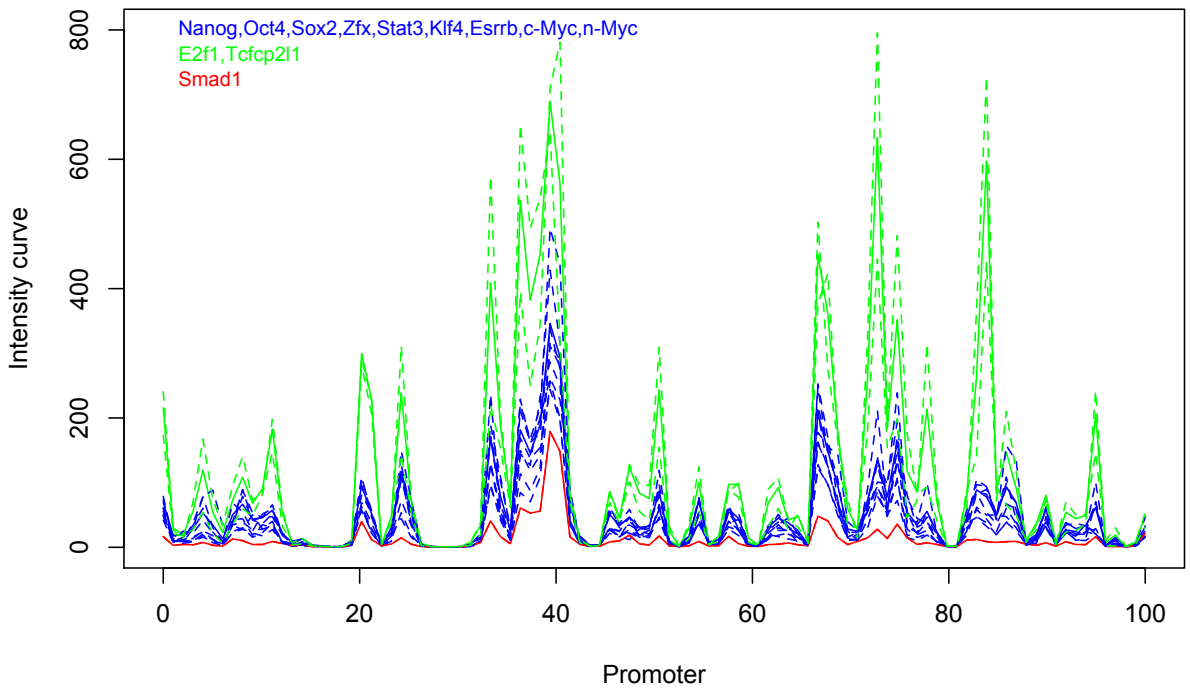

Figure 3.1: The estimated clusters and intensities in the promoter regions. Solid lines: the estimated binding intensities of the clusters; dotted lines: the estimated binding intensities of the individual TFs. The x-axis represents the genomic BS locations mapped on the real line between 0 and 100.

Table 3.3: SD index in the promoter for the real data analysis by DPM-LGCP, Ripley's K-function, Gaussian mixture model (GMM), k-means, and k-means with adjusted center.

Method	SD index
DPM-LGCP	0.00594
K-function	0.01014
GMM	0.01078
k-means	0.00982
adj.k-means	0.00982

3.5 Discussion

We apply a novel DPM-LGCP model to investigate the combinatorial binding patterns of TFs in promoters. Specifically, we employ LGCP to model TF binding site locations and employ a Dirichlet process prior to facilitate TF clustering. To improve computational efficiency, Laplace approximation and numerical integration in INLA package are adopted to construct an approximation to the true LGCP likelihood. In reality, the number of TF clusters is unknown and thus can not be specified in advance. However, the introduction of DPM model allows us to find the best setting of the number of clusters by letting the data speak for themselves.

The simulation study in the first process shows that our method, DPM-LGCP, outperforms other methods such as Ripley's K-function [11], GMM, k-means and k-means with adjusted center in TF clustering; in the second process where data are generated from a "wrong" model for our method, DPM-LGCP is still very competitive as it achieves similar performance to k-means and k-means with adjusted center, better than the baseline (GMM). As a nonparametric modeling approach, our method has remarkable advantages in avoiding restrictive parametric assumptions and creating substantial flexibility and robustness [71]. We also notice that in both processes, the performance of k-means is better than k-means with adjusted center. In k-means, the centroid of a cluster is the mean of the observations in that cluster; whereas in k-means with adjusted center, the centroid is calculated by INLA. The squared Euclidean distance between a TF and centroid in k-mean is always smaller than that in k-means with adjusted center, therefore, the latter one is more likely to remove that TF from a correctly assigned cluster. As for the other method introduced by Cha and Zhou (2014), the calculation of Ripley's K-function depends on a distance variable t and it is computational intense to select the most appropriate value of t, especially for clustering TFs on whole genome. We apply the method on ChIP-seq data for ESC and achieve some

interesting findings. TFs generally bind to specific genome regions and interact with other TFs or the basal transcription apparatus to activate or inhibit target gene expression. Here, we choose the promoter regions as an example and observe several TF clusters known in ESC, such as Nanog–Oct4–Sox2 and n-Myc–c-Myc–Zfx.

In summary, DPM-LGCP allows us to model the distribution of TFs on genome and identify the TF patterns without prespecified number of clusters. In general, it performs the best in all five clustering methods. This method can be applied to cluster TFs in different regulatory regions, other cells or data taking the form of a point pattern. Importantly, the results of TF clustering can be combined with gene expression or motif analysis to help researchers better understand the underlying gene regulatory mechanism.

Chapter 4

Modeling the association between transcription factor binding and gene expression levels in mouse embryonic stem cells

4.1 Introduction

Transcription factors are sequence-specific DNA-binding protein that play an important role in the transcriptional regulation of gene expression. In recent years, the development of technology allows rapid, high-throughput characterization of TF binding sites through chromatin immunoprecipitation sequencing technologies. In addition, the widely used ultra-high-throughput RNA sequencing technology provides more reliable and accurate gene expression measurements. These allow a powerful first step in the study of the regulatory functions of TF on gene expression.

Many previous studies, including computational analysis or experiments, have been carried out to study the relationship between gene expression and TF binding. In the study of transcriptional regulation, predictive model is commonly used in which gene expression levels are treated as response variable and various features related to TFs as the predictors. These features may include the abundance of motifs recognized by the TFs [10, 38], motif scores based on position-specific weight matrices [19], ChIP log-ratios for TFs [27], etc. A number of different methods have been employed to study how gene expressions are affected by TF binding, including linear regression [19, 27, 38, 75] , Bayesian error analysis [88], multivariate adaptive regression splines [20], partial least squares regression [7], and deep integrative approach [16]. However, most of these approaches could not make full use of the functional features of TF binding, that is, how the TFBS vary over a continuum as a function along the genome.

The main assumption in previous studies is that the feature of a TF, such as the motif and TF binding sites, is positional independent in a region or across the genome. This assumption is disputed and TF position interdependence has been observed in many studies [5, 9, 45, 58]. The inaccurate characterizing of TF binding data may reduce the statistical

power or even lead to biased inferences and wrong decisions. Therefore, when exploring the relationship between TF and gene expression, it is important to use the complete TF binding sites information instead of a consensus sequence, or even mononucleotide frequency weight matrices to describe the feature of a TF.

In our study, we model the TF binding sites by an inhomogeneous Poisson process and thus the intensity function can be considered as a representative of the complete feature of a TF binding to a region. Several previous studies have shown that TF binding signal around the TSSs of genes are predictive of gene expression levels with fairly high accuracy [16, 75]. Thus, the TF binding sites around gene can be represented by several intensity functions and then existing functional data analysis methods can be employed to study the relationship between a TF and gene expression. Since it is difficult to represent intensity functions by simple parametric forms, non-parametric modeled are more preferred to model the functions. Most of previous literature adopt kernel or smooth splines to model the functional data [8, 29, 80, 87, 92]. However, these approaches are using global bandwidths and penalties and not well suited for irregular functional data. In this study, we adopt a Bayesian wavelet-based approach, developed by Morris and Carrol [68], which accommodates multiple fixed-effect and random-effect predictors with flexible between-curve correlation assumptions suggested by various experimental designs.

The chapter is structured as follows: Section **4.2** presents the method, including a description of a log Gaussian Cox process with wavelet-based functional model (LGCP-WFM) in Section **4.2.1**, the detailed posterior sampling procedure in Section **4.2.2** and **4.2.3** and the model fitting evaluation criteria in Section **4.2.4**. In Section **4.3**, we perform simulation studies to evaluate the performance of the method and also apply the method to a real data set. Section **4.4** closes the chapter with a discussion and some conclusions.

4.2 Method

4.2.1 Model

As mentioned in Chapter 3, a Log Gaussian Cox process can be used to describe the binding pattern of a single TF. Denote $S = \{s_1, s_2, ..., s_n\}$ the set of BSs of TF on a particular region of a genome sequence. We assume $s_i \in \Omega$, where Ω is a closed interval of the real line. We further assume that S follows an inhomogeneous Poisson process with intensity function $\lambda(s)$, $s \in \Omega$. The likelihood of S can thus be written as:

$$f(S|\lambda(s)) = \exp\left\{|\Omega| - \int_{\Omega} \lambda(s)ds\right\} \prod_{i=1}^{n} \lambda(s_i) \tag{4.1}$$

where $|\Omega|$ is the length of the interval. In an LGCP, the intensity function $\lambda(s)$ is assumed to follow a log-Gaussian process, i.e. $\lambda(s) = \exp(Z(s))$, and $Z(s) \sim GP(m, C_\theta)$. Here m is a mean function and $C_\theta(\cdot, \cdot)$ is the covariance kernel of $Z(s)$, where θ contains parameters that control the shape of the covariance kernel.

Now suppose that the gene expression level measurements come from N different genes. The BS's of TF surrounding the TSS of the ith gene are denoted by $S_i = \{s_{i1}, s_{i2}, ..., s_{in_i}\}$, $i = 1, ..., N$. To understand the association between BS's and gene expression, we propose a functional model, described as follows:

$$\begin{aligned} S_i|\lambda_i(s) &\sim \text{inhomo-Poisson}(\lambda_i(s)), \quad s \in \Omega, \quad i = 1, ..., N \\ \log(\lambda_i(s)) &= Z_i(s) \\ Z_i(s) &= X_i\mathbf{B}(s) + E_i(s) \end{aligned} \tag{4.2}$$

where $\lambda_i(s)$ and $Z_i(s)$ represent the intensity function and the log of intensity function for the ith gene, $i = 1, ..., N$, respectively. X_i, a vector of p elements (p is the number of gene expression levels), represents the ith row of a design matrix and indicates which gene expression level the ith gene belongs to, $i = 1, ..., N$. $\mathbf{B}(s) = (B_1(s), ..., B_p(s))^T$ is a vector of fixed effect functions. The function $B_i(s)$ represents the partial effect of covariate i on the function at position s. Using vector representation, the last formula in equation 4.2 can be written as:

$$\mathbf{Z}(s) = \mathbf{XB}(s) + \mathbf{E}(s) \tag{4.3}$$

where $\mathbf{Z}(s) = (Z_1(s), ..., Z_N(s))^T$ is a vector of log transformed intensity functions, stacked by rows and $\mathbf{E}(s) = (E_1(s), ..., E_N(s))^T$ is a vector of residual error functions. $\mathbf{X}$ is an $N \times p$ design matrix that includes the gene expression levels or other covariates of interest.

Since it is generally difficult to fit the continuous curves directly, we assume that all functions are measured on a common equally-spaced fine grid $\mathbf{t} = (t_1, ..., t_T)^T$. The discrete version of equation 4.3 can be written as:

$$\mathbf{Z} = \mathbf{XB} + \mathbf{E} \tag{4.4}$$

where $\mathbf{Z}$ is an $N \times T$ matrix of the log-transformed intensities of TFs on the grid $\mathbf{t}$, $\mathbf{B}$ is a $p \times T$ matrix of fixed effects and $\mathbf{E}$ is an $N \times T$ matrix of residual errors. The residual error matrix is assumed to follow $\mathcal{MN}(\mathbf{P}, \mathbf{Q})$, where $\mathbf{P}$ is an $N \times N$ covariance matrix and $\mathbf{Q}$ is a $T \times T$ covariance matrix. A special case of this model, $\mathbf{P} = \mathbf{I}$, is considered in this study.

To reduce the correlation between nearby binding sites, equation 4.4 is transformed by wavelet-based approach. In particular, we first project the observed curves from the data space to the wavelet space by applying the discrete wavelet transformation (DWT) to each row of $\mathbf{Z}$, represented as $\mathbf{D} = \mathbf{Z}\mathbf{W}^T$, where $\mathbf{W}$ is an orthogonal DWT matrix. By right multiplying the DWT matrix $\mathbf{W}^T$ on both side of model 4.4, we get a wavelet-space version of the model:

$$\mathbf{D} = \mathbf{X}\mathbf{B}^* + \mathbf{E}^* \tag{4.5}$$

where $\mathbf{B}^* = \mathbf{B}\mathbf{W}^T$ is a $p \times T$ matrix containing the wavelet coefficients, and $\mathbf{E}^* = \mathbf{E}\mathbf{W}^T$ is an $N \times T$ matrix containing the residual errors in the wavelet space. The columns of $\mathbf{D}$, $\mathbf{B}^*$, and $\mathbf{E}^*$ are all double-indexed by wavelet scale j and location k. $\mathbf{E}^*$ follows a $\mathcal{MN}(\mathbf{P}, \mathbf{Q}^*)$, with $\mathbf{Q}^* = \mathbf{W}\mathbf{Q}\mathbf{W}^T$. Due to the whitening property of wavelet transforms, we assume that columns of $\mathbf{E}^*$ are independent, making $\mathbf{Q}^*$ diagonal, represented by $\mathbf{Q}^*$ =diag(ψ^*_{jk}). We denote a set of variance components by Ψ^*_Q.

Working with the wavelet-space model 4.5, we denote the (j,k)th column as:

$$\mathbf{d}_{jk} = \mathbf{X}\mathbf{b}^*_{jk} + \mathbf{e}^*_{jk} \tag{4.6}$$

where $\mathbf{d}_{jk} = \{D_{ijk}\}_{i=1}^N$, $\mathbf{b}^*_{jk} = \left\{B^*_{ajk}\right\}_{a=1}^p$, and $\mathbf{e}^*_{jk} = \left\{E^*_{ijk}\right\}_{i=1}^N$. We apply a 'g-prior' on $\mathbf{b}^*_{jk}$ and specify the following priors on the other parameters:

$$
\begin{aligned}
\mathbf{b}^*_{jk} &\sim \mathcal{MVN}(\mathbf{0}, g\psi^*_{jk}(\mathbf{X}^T\mathbf{X})^{-1}) \\
\mathbf{e}^*_{jk} &\sim \mathcal{MVN}(\mathbf{0}, \psi^*_{jk}\mathbf{I}) \\
\psi^*_{jk} &\sim \mathcal{IG}(a, b) \quad a, b > 0
\end{aligned} \tag{4.7}
$$

where the hyperparameters g, a, and b are fixed.

4.2.2 Posterior sampling by Markov chain Monte Carlo method

After specifying the priors, we apply a MCMC algorithm to draw the posterior samples of the parameters in the above model. A block Gibbs sampler is used to obtain the posterior samples of the parameters in the wavelet-space model. Then inverse DWT is applied to obtain the posterior samples of $\mathbf{B}^*$ in the data-space model. INLA packages is employed to obtain the posterior samples of $\mathbf{Z}$ in the data-space model. The details of sampling process are as follows:

- step 0: Initialize $\mathbf{Z}$, $\left\{B^*_{ajk}\right\}_{a=1}^{p}$, $\left\{E^*_{ijk}\right\}_{i=1}^{N}$ based on INLA package approximation and MLE estimation.
- Step 1: Project the curves from the data space into the wavelet spaces by applying the DWT to obtain $\mathbf{D}$.
- Step 2: For each j,k, update the fixed effects from $f(\mathbf{b}^*_{jk}|\psi^*_{jk}, \mathbf{d}_{jk}, \mathbf{X}, \mathbf{S})$, which is available in a closed form as a multivariate normal distribution.
- Step 3: For each j,k, update the variance component from $f(\psi^*_{jk}|\mathbf{d}_{jk}, \mathbf{X}, \mathbf{S})$, which is also a closed form as an inverse gamma distribution.
- Step 4: Project the samples of parameters back into the data space by using the IDWT.

- Step 5: For each i, update the log transformed intensity function from $f(Z_i|\mathbf{B}^*, \mathbf{X}, \mathbf{S})$, which can be approximated by INLA package.

These steps are repeated until convergence is reached. After a burn-in period, we collect posterior samples of the parameters. For those in the wavelet-space, i.e. $\mathbf{B}^*$, the IDWT can be applied to the posterior samples to obtain posterior samples of $\mathbf{B}$ based on which Bayesian inference can be performed in the original data space.

4.2.3 Conditional distributions for $\mathbf{b}^*_{jk}$, ψ^*_{jk} and Z_i

With the model and priors in Section 4.2.1, we can calculate the full posterior distribution:

$$f(\mathbf{Z}, \mathbf{B}^*, \Psi^*_Q|\mathbf{X}, \{S_i\}) \propto f(\{S_i\}\,|\mathbf{Z}, \mathbf{X}, \mathbf{B}^*, \Psi^*_Q) f(\mathbf{Z}|\mathbf{X}, \mathbf{B}^*, \Psi^*_Q) f(\mathbf{B}^*|\mathbf{X}, \Psi^*_Q) f(\Psi^*_Q) \tag{4.8}$$

where $\mathbf{Z}$ and $\mathbf{D}$ represent the log-transformed intensity functions in the data space and the wavelet space, respectively. Thus, we use $\mathbf{D}$ to calculate $\mathbf{b}^*_{jk}$ and ψ^*_{jk} in the wavelet space.

$$\begin{aligned} f(\mathbf{b}^*_{jk}, \psi^*_{jk}|\mathbf{d}_{jk}, \mathbf{X}, \{S_i\}) &\propto \mathcal{L}(\mathbf{b}^*_{jk}, \psi^*_{jk}|\mathbf{d}_{jk}, \mathbf{X}, \{S_i\}) \pi(\mathbf{b}^*_{jk}|\mathbf{X}, \psi^*_{jk}) \pi(\psi^*_{jk}) \\ &\propto {\psi^*_{jk}}^{-\frac{N}{2}-\frac{p}{2}-a-1} \exp\left\{ -\frac{1}{2\psi^*_{jk}} (\mathbf{d}_{jk} - \mathbf{X}\hat{\mathbf{b}}^*_{jk})^T (\mathbf{d}_{jk} - \mathbf{X}\hat{\mathbf{b}}^*_{jk}) \right. \\ &\quad - \frac{1}{2\psi^*_{jk}} (\mathbf{b}^*_{jk} - \hat{\mathbf{b}}^*_{jk})^T \mathbf{X}^T \mathbf{X} (\mathbf{b}^*_{jk} - \hat{\mathbf{b}}^*_{jk}) \\ &\quad \left. - \frac{1}{2g\psi^*_{jk}} (\mathbf{b}^*_{jk})^T \mathbf{X}^T \mathbf{X} \mathbf{b}^*_{jk} - \frac{b}{\psi^*_{jk}} \right\} \end{aligned} \tag{4.9}$$

where $\hat{\mathbf{b}}^*_{jk}$ is the maximum likelihood estimator of $\mathbf{b}^*_{jk}$, specified by $\hat{\mathbf{b}}^*_{jk} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T d_{jk}$. The conditional distribution $f(\mathbf{b}^*_{jk}|\psi^*_{jk}, \mathbf{d}_{jk}, \mathbf{X}, \{S_i\})$ is

$$\mathcal{MVN}\left(\frac{g}{g+1}\hat{\mathbf{b}}^*_{jk}, \frac{g\psi^*_{jk}}{g+1}(\mathbf{X}^T\mathbf{X})^{-1}\right), \tag{4.10}$$

and the posterior distribution $f(\psi^*_{jk}|\mathbf{d}_{jk}, \mathbf{X}, \mathbf{S})$ is

$$\mathcal{IG}\left(\frac{N}{2} + a, b + \frac{\hat{\sigma}^2_{jk}}{2} + \frac{(\hat{\mathbf{b}}^*_{jk})^T\mathbf{X}^T\mathbf{X}\hat{\mathbf{b}}^*_{jk}}{2(g+1)}\right), \tag{4.11}$$

where $\hat{\sigma}^2_{jk} = (\mathbf{d}_{jk} - \mathbf{X}\hat{\mathbf{b}}^*_{jk})^T(\mathbf{d}_{jk} - \mathbf{X}\hat{\mathbf{b}}^*_{jk})$.

INLA package is used to calculate the posterior distribution of Z_i in the data space. One of the advantage of using INLA is that it approximates $Z_i(s)$ by constructing a regular grid $\mathbf{t}$ on Ω, which is consistent with our assumptions in fitting functional data. INLA uses numerical integration to approximate the posterior distribution of $f(Z_i|\mathbf{B}^*, \mathbf{X}, \mathbf{S})$. For more details, please refer to Chapter 3.

4.2.4 Evaluation Criteria

Three measures are used to summarize the model performance in estimating the fixed effect functions $B_a(t)$, including the integrated mean squared error (IMSE), the integrated posterior variability (IPVar), and the integrated total variability (ITVar). These three measures summarize the deviation of the posterior mean from the truth, the variability about the posterior mean, and posterior variability about the truth, respectively. The equations are

expressed as following:

$$
\begin{aligned}
IMSE &= \int_T \left\{\hat{\theta}(t) - \theta_0(t)\right\}^2 dt \\
IPVar &= G^{-1} \sum_{g=1}^{G} \int_T \left\{\theta^{(g)}(t) - \hat{\theta}(t)\right\}^2 dt \\
ITVar &= G^{-1} \sum_{g=1}^{G} \int_T \left\{\theta^{(g)}(t) - \theta_0(t)\right\}^2 dt
\end{aligned}
\tag{4.12}
$$

where $\theta_0(t)$ is the truth for a functional parameter $\theta(t)$, $\hat{\theta}(t)$ is the posterior mean, $\theta^{(g)}(t)$ is the posterior samples, $g = 1, ..., G$. Note that $ITVar = IMSE + IPVar$.

To identify binding regions that significantly influence different gene expression levels, linear transformation can be applied to the posterior samples of $\mathbf{B}$ to obtain the posterior samples of the contrast effects, denoted by $C_m(t), m = 1, ..., M$ (M depends on the number of contrasts one plans to test). Then the Bayesian false discovery rate (FDR) approach introduced by Morris et al. [67] is used to evaluate the model inferential performance and identify significant binding regions [46, 89, 99]. In our model, $\mathbf{Z}$ is a vector of log-transformed intensity functions and thus both $B_a(t)$ and $C_m(t)$ are in log scale. Suppose we are interested in identifying the TF binding regions that have at least a δ-fold difference in the contrast. Based on the MCMC procedure, a contrast should have G posterior samples, denoted by $C_m^g(t), g = 1, ..., G$. The point-wise posterior probabilities of at least δ-fold difference at each grid location can be computed by $p_m(t_l) = Pr\{|C_m(t_l)| \geqslant log(\delta))|Z\} \approx G^{-1} \sum_{g=1}^{G} I\left\{\left|C_m^{(g)}(t_l)\right| \geqslant log(\delta)\right\}$ for $t_l, l = 1, ..., T$. Then $1 - p_m(t_l)$ can be interpreted as the estimate of local FDR at t_l. We can identify the set of locations with $p_m(t_l) \geqslant \phi_\alpha$ as significant, where ϕ_α is a threshhold determined based on a desired global FDR bound α ($0 < \alpha < 1$).

To obtain ϕ_α, first, sort $p_m(t_l), l = 1, ..., T$ in descending order to obtain $p_{m(l)}, l = 1, ..., T$. Second, find $\upsilon = max\left\{l^* : (l^*)^{-1} \sum_{l=1}^{l^*} (1 - p_{m(l)}) \leqslant \alpha\right\}$ and set $\phi_\alpha = p_{m\upsilon}$. Let the set of

locations $\gamma = \{t_l : p_m(t_l) > \phi_\alpha\}$ be the set of discoveries and then we can further compute the model-based estimates of FDR, false omission rate (FOR), sensitivity and specificity. The formulas are expressed as following:

$$\begin{aligned} FDR &= \mathbb{N}(\gamma)^{-1} \sum_{t_l \in \gamma} 1 - p_m(t_l) \\ FOR &= \mathbb{N}(\gamma')^{-1} \sum_{t_l \in \gamma'} p_m(t_l) \\ Sensitivity &= \left\{ \sum_{l=1}^{T} p_m(t_l) \right\}^{-1} \sum_{t_l \in \gamma} p_m(t_l) \\ Specificity &= \left\{ \sum_{l=1}^{T} \{1 - p_m(t_l)\} \right\}^{-1} \sum_{t_l \in \gamma'} \{1 - p_m(t_l)\} \end{aligned} \tag{4.13}$$

where $\gamma \cup \gamma' = \mathcal{T}$ and $\mathbb{N}(\gamma) = \sum_{l=1}^{T} I(t_l \in \gamma)$.

4.3 Numerical analysis

4.3.1 TF binding and gene expression data

TF binding and gene expression data are collected to investigate the association between TF binding sites and gene expression. Chen et al.[15] provides the ChIP-seq binding peak data of 12 TFs in mouse ESCs, which can be downloaded from NCBI GEO with the accession number GSE11431. Among these 12 TFs, Nanog is chosen as an example in this study since it plays a fundamental role in the self-renewal and pluripotency of ESCs [13, 98]. The gene expression data are obtained from Ouyang et al. [75], in which Ouyang et al. applied the singular value decomposition (SVD) to analyze gene expression patterns upon ESC differentiation and grouped the genes into the Uniform High or Uniform Low genes

based on the value of GPC1. We randomly select 72 genes, half from the Uniform High expressed genes and the other half from the Uniform Low expressed genes, and obtain the binding sites around the TSSs of these genes based on the ChIP-seq data for Nanog.

4.3.2 Simulation study

To have a better understanding of the spatial effect of TF binding on gene expression, we mimic the real data collected from Chen et al. [15] and Ouyang et al. [75]. LGCP model is applied to the real data to estimate the log-transformed intensity functions using INLA package. Then WFM is applied to the estimated functional data and gene expression levels, by which we get the estimates of fixed effect $\mathbf{B}^*$ and variance parameter Ψ_Q^*. These parameters are treated as the underlying truth during simulation.

In the simulation study, the residuals are simulated from Gaussian distributions in the wavelet space. The data generated in the wavelet space are then inverse-transformed to the data space and the TF binding sites are generated from inhomogeneous Poisson processes with the intensity functions $e^{\mathbf{Z}}$. To assess the performance of our model under different scenarios, we change the sample sizes for each expression groups to 20, 36 and 50 hence the totally sample sizes become 40, 72, and 100. We also reduce the variance parameters to $\frac{1}{10}\Psi_Q^*$ to examine how the variances affect model performance. Therefore, we have six scenarios in total and for each scenario we simulate 100 data sets. For each data set, using a burn-in of 200 and a thinning of 2, we obtain 400 posterior samples. Trace plots suggest good mixing. Three measures, IMSE, IPVar and ITVar, are used to assess the model performance under different scenarios. Based on the posterior samples of the fixed effect functions, we construct a contrast: $C(t) = B_1(t) - B_2(t)$ and obtain the threshold ϕ_α of at least 1.5-fold different with $\alpha = 0.1$ described in Morris et al. [67]. Then we compute the FDR, FOR, sensitivity

and specificity based on the threshold $\phi_{0.1}$.

The simulation results are shown in Tables 4.1 and 4.2. For the fixed effect function $B_1(t)$, the IMSE, IPVar and ITVar become smaller as the sample size increases; for a specific sample size, the IMSE, IPVar and ITVar get larger with the increase of variance (Table 4.1). The IMSE, IPVar and ITVar for $B_2(t)$ have similar patterns. On the other hand, all scenarios achieve very low FDRs and high specificities (Table 4.2). The values of FOR become lower with the increase of sample size and decrease of variance and the values of sensitivity become larger with the increase of sample size and decrease of variance. This indicates that our model perform increasingly better as the variances become smaller and the sample sizes become larger.

Table 4.1: IMSE, IPVar and ITVar for $B_a(t), a = 1, 2$ across all simulated data sets under different scenarios.

Scenario		$B_1(t)$			$B_2(t)$		
Variance	Sample size	IMSE	IPVar	ITVar	IMSE	IPVa	ITVar
$\frac{1}{10}\Psi_Q^*$	40	0.0580	0.0241	0.0821	0.0635	0.0289	0.0924
$\frac{1}{10}\Psi_Q^*$	72	0.0331	0.0107	0.0438	0.0366	0.0133	0.0499
$\frac{1}{10}\Psi_Q^*$	100	0.0211	0.0102	0.0323	0.0312	0.0103	0.0414
Ψ_Q^*	40	0.0833	0.0404	0.1237	0.0994	0.0406	0.1400
Ψ_Q^*	72	0.0632	0.0234	0.0866	0.0752	0.0235	0.0987
Ψ_Q^*	100	0.0594	0.0170	0.0764	0.0576	0.0171	0.0747

We plot the posterior means for the fixed effect functions $B_1(t)$ and $B_2(t)$ with the corresponding 95% point-wise credible intervals for all scenarios from one of the 100 simulation runs (Figure 4.1 and 4.2). As seen from these figures, the posterior mean estimates are closer to the truth with a smaller variance and larger sample size and the credible intervals become relatively narrow as the variance decreases and sample size increases.

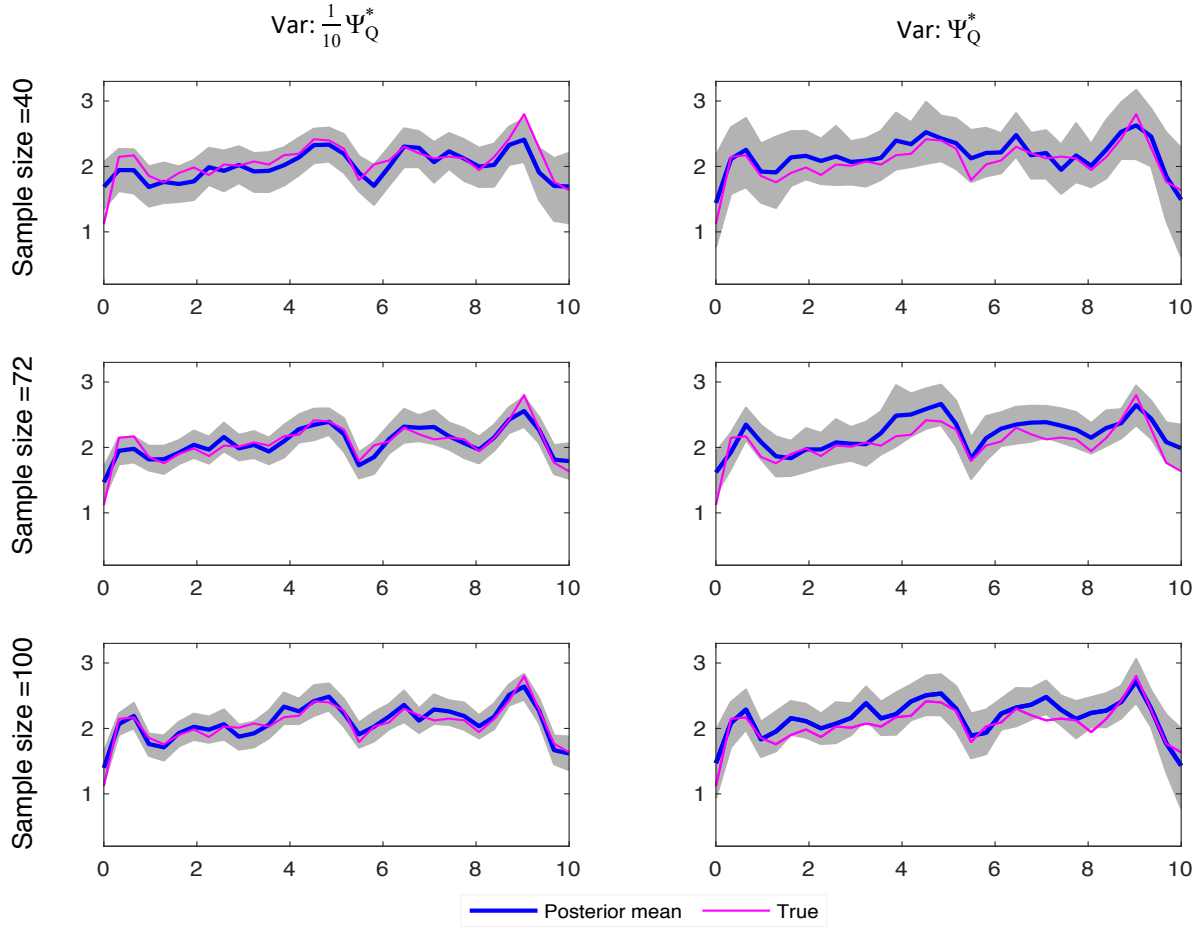

Figure 4.1: Estimation of $B_1(t)$ in simulation. This plot presents posterior means (blue line) and 95% credible intervals (grey bands) under all six scenarios, along with the true $B_1(t)$ (pink). This plot is for one of the 100 simulations.

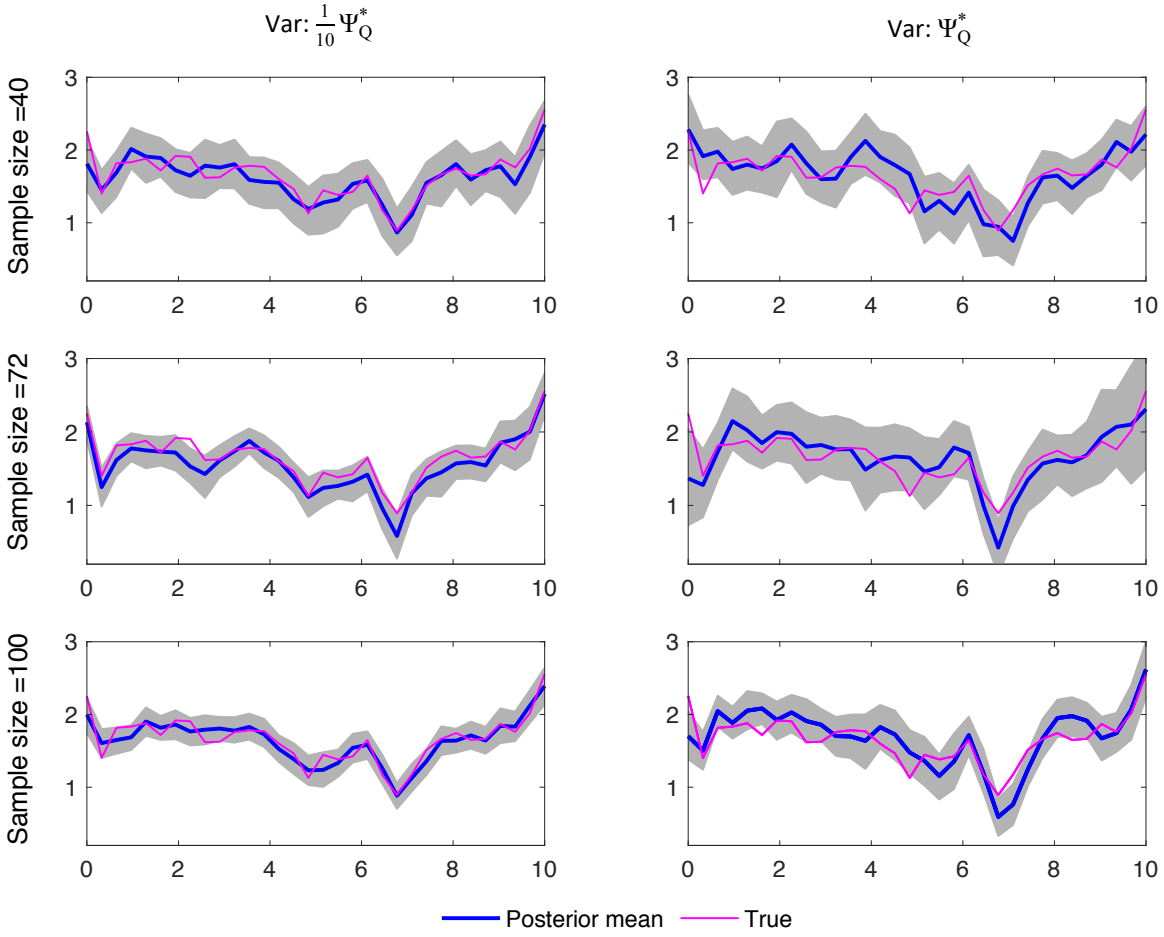

Figure 4.2: Estimation of $B_2(t)$ in simulation. This plot presents posterior means (blue line) and 95% credible intervals (grey bands) under all six scenarios, along with the true $B_2(t)$ (pink). This plot is for one of the 100 simulations.

Table 4.2: FDR, FOR, sensitivity and specificity for $C(t)$ in terms of 1.5-fold difference with $\alpha = 0.1$ across all simulated data sets under different scenarios.

Scenario		$C(t)$			
Variance	Sample size	FDR	FOR	Sensitivity	Specificity
$\frac{1}{10}\Psi_Q^*$	40	0.0648	0.2364	0.5636	0.9716
$\frac{1}{10}\Psi_Q^*$	72	0.0620	0.1700	0.7240	0.9640
$\frac{1}{10}\Psi_Q^*$	100	0.0565	0.1344	0.8029	0.9621
Ψ_Q^*	40	0.0708	0.2838	0.5445	0.9632
Ψ_Q^*	72	0.0689	0.2235	0.6377	0.9620
Ψ_Q^*	100	0.0634	0.1940	0.7014	0.9613

4.3.3 Real data analysis

Our model is applied on the real data, i.e. expression levels from 72 genes (half from the Uniform High expressed genes and the other half from the Uniform Low expressed genes) and binding sites of Nanog around the TSS of these genes. Thus the design matrix $\mathbf{X}$ can be written as:

$$X = \begin{bmatrix} 1 & 0 \\ \vdots & \vdots \\ 1 & 0 \\ 0 & 1 \\ \vdots & \vdots \\ 0 & 1 \end{bmatrix} \tag{4.14}$$

We compute the contrast $C(t) = B_1(t) - B_2(t)$ from the posterior samples of the fixed effect functions and calculate the FOR, sensitivity and specificity based on the threshold $\phi_{0.1}$ for a 1.5-fold difference.

The results of real data analysis are shown in Figure 4.3. The first panel contains the mean

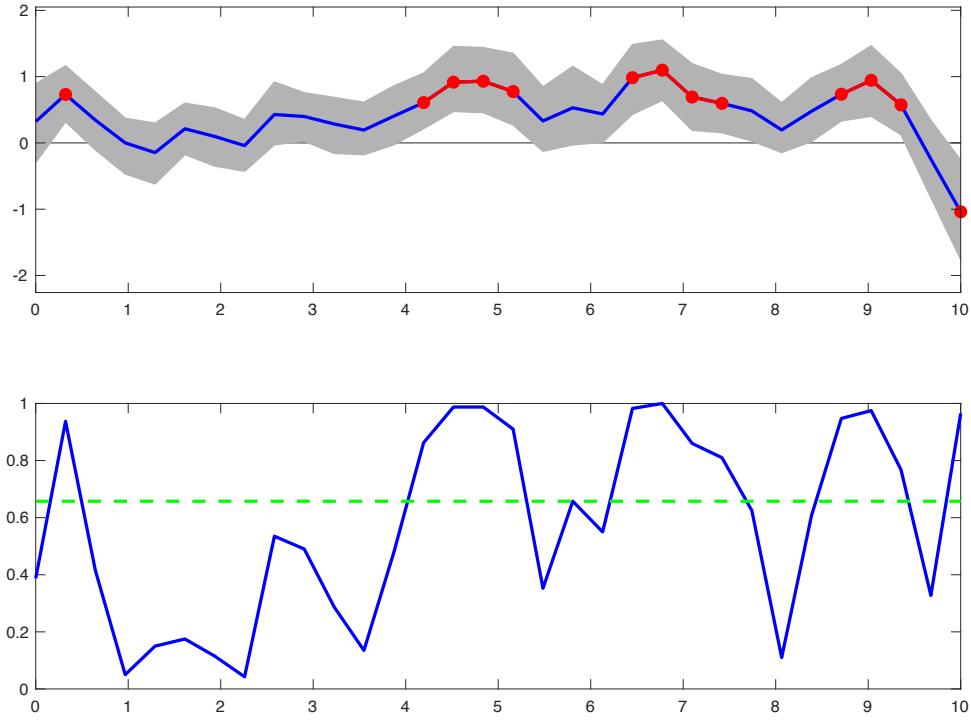

Figure 4.3: Regions flagged for 1.5-fold difference in real data. (a) the significant regions flagged on the posterior mean function $C(t)$ with 95% credible intervals. (b) The corresponding posterior probability estimates and the thresholds obtained using Bayesian FDR-based inference with $\alpha = 0.1$.

and 95% credible intervals for contrast function $C(t)$ and the red color indicates the regions flagged as significant in terms of a 1.5 -fold difference with a global FDR of $\alpha = 0.1$. The second panel contains the corresponding posterior probability for a 1.5-fold difference (i.e. $p(t) = Pr\{|C(t)| \geqslant log(1.5)|Z\}$) and the green line represents the threshold $\phi_{0.1}$. In this analysis, there are two points (i.e. 0.32 and 10) and three contiguous regions flagged as significant, [4.19, 5.16], [6.45,7.42], and [8.71,9.35]. This indicates that the Nanog's binding pattern in the second half of the TSS region influence the gene expression levels significantly. Based on the posterior samples for $C(t)$, we also compute the empirical estimates of the FOR, Sens and Spec for 1.5-fold difference. Our model leads to a moderately low FOR (0.3415), a moderately high Sens (0.6489) and a very high Spec (0.9255).

4.4 Discussion

With the rapid development of high-throughput sequencing technology, ChIP-seq data are increasingly collected, which provides us with great opportunity to decipher the roles of TFs in gene expression regulation. There is a need for methods to analyze these data sets and uncover the underlying mechanisms. In this chapter, we have introduced a novel method, LGCP-WFM, to study the regulatory associations between TFs and target genes. Through simulation study, we demonstrate that our method performs well, especially with large sample size and small variance. It also shows a remarkable ability to distinguish real local feature in estimating the regression coefficient function.

Our model is able to utilize the complete TF binding sites information around TSSs of genes by LGCP and INLA is adopted to approximate the marginal posterior distribution of intensity functions by constructing a discrete grid in the data space. To reduce inherent correlation within binding intensity functions, these functional data can be easily projected from

the data space into the wavelet space by applying the DWT in the WFM model. Through the MCMC algorithm, we obtain the posterior sample of the fixed effect functions that can be used to perform further Bayesian estimation, inference or prediction. We have assumed Gaussian distributions for the residuals in the proposed model. Extensions facilitating other distributions with heavier tails, such as t_1, t_2, t_3 and DE, can be performed following the work of Zhu et al. [99]. In addition, the random effects can be incorporated in this model that allows the mean functions and covariance surfaces to vary over strata and makes the model more flexible.

In summary, our method provide a novel approach to addressing the association between point process data and scalar features. Besides the Nanog TF investigated in this study, we expect to apply the method to discover the binding effect of other TFs on target genes, such as STAT3, OCT4 and SMAD1. This method is also sufficiently flexible to be applied to a wide range of NGS sequencing data from various experimental designs.

Appendices

Appendix A

First Appendix

A.1 SNPs in IGV viewer

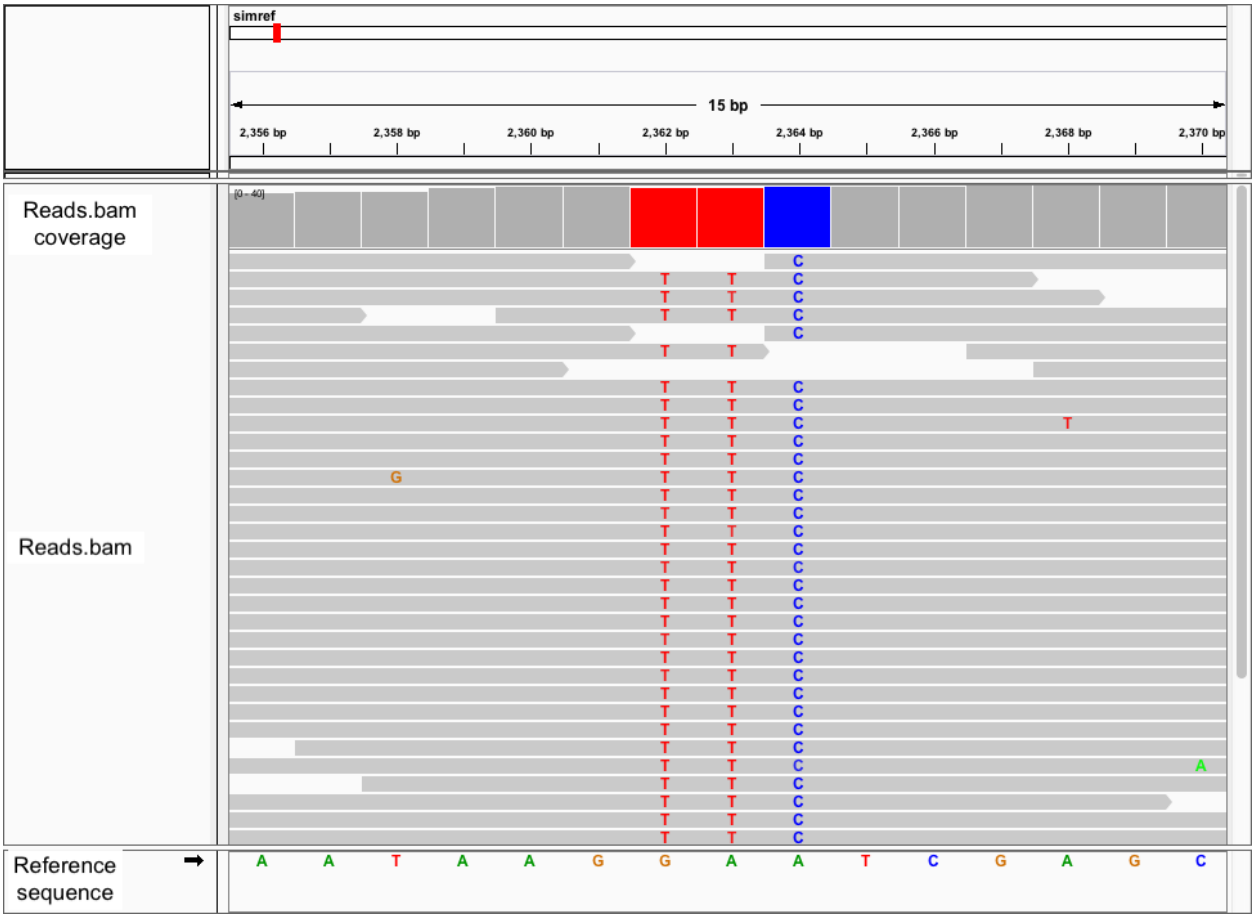

Figure A.1: This figure illustrates a view of a section of mapped reads in the dataset simulated by HMM. It shows three SNPs at base 2362 (G / T), 2363 (A / T) and 2364 (A / C).

A.2 Comparison of different variant callers using real data on chromosome 22

	SNP			INDEL		
Caller	Sensitivity	Precision	F_1 score	Sensitivity	Precision	F_1 score
15X						
vi-HMM	94.19%	99.36%	96.71%	89.79%	77.72%	83.32%
FreeBayes	94.75%	82.48%	88.19%	87.83%	63.83%	73.93%
Platypus	90.87%	99.86%	95.15%	92.84%	62.41%	74.64%
SAMtools	98.53%	99.58%	99.05%	77.99%	90.62%	83.83%
VarScan	72.47%	99.94%	84.02%	64.60%	98.50%	78.03%
30X						
vi-HMM	99.80%	99.24%	99.52%	94.01%	89.89%	91.90%
FreeBayes	95.86%	90.15%	92.92%	89.66%	67.88%	77.27%
Platypus	93.26%	99.59%	96.32%	95.78%	61.21%	74.69%
SAMtools	99.74%	99.64%	99.69%	82.15%	88.24%	85.09%
VarScan	97.69%	99.87%	98.77%	84.29%	98.42%	90.81%
50X						
vi-HMM	99.96%	98.55%	99.25%	93.88%	89.98%	91.90%
FreeBayes	95.94%	92.98%	94.44%	89.54%	70.12%	78.65%
Platypus	93.37%	99.44%	96.31%	96.33%	60.49%	74.32%
SAMtools	99.80%	99.66%	99.73%	83.67%	85.34%	84.50%
VarScan	99.58%	99.84%	99.71%	88.93%	99.17%	93.32%

A.3 The alignment information by Bowtie2 and BWA-MEM at different coverage depths

Bowtie2 and BWA-MEM are compared on correct mapping, multiple alignment, second alignment, and soft/hard clipped reads. We do not list the average percentages of hard clipped reads because all of them are zeros for both aligners.

Table A.3.1: The average percentage of correct mapping by Bowtie2 and BWA-MEM using simulated data at various sequencing depths.

Mapper	15X	20X	25X	30X
Bowtie2	96.31%	95.79%	96.35%	95.90%
BWA-MEM	94.69%	94.62%	94.65%	94.61%

Table A.3.2: The average percentage of multiple alignment by Bowtie2 and BWA-MEM using simulated data at various sequencing depths.

Mapper	15X	20X	25X	30X
Bowtie2	3.15%	3.21%	3.15%	3.23%
BWA-MEM	0	0	0	0

Table A.3.3: The average percentage of soft clipped reads by Bowtie2 and BWA-MEM using simulated data at various sequencing depths.

Mapper	15X	20X	25X	30X
Bowtie2	0	0	0	0
BWA-MEM	8.68%	8.74%	8.71%	8.76%

A.4 Performance of vi-HMM on simulated data with homopolymers

We test the performance of vi-HMM with homopolymers on a simulated dataset with 30× coverage depth. vi-HMM produces high calling accuracies of SNPs and INDELs, which are only slightly smaller than those in the general situation. One potential explanation for the lower INDEL calling accuracy is that the INDELs are redundant in the output[31].

SNP and INDEL callings by vi-HMM using simulated data with homopolymers at 30× depths.

	SNP			INDEL		
Caller	Sensitivity	Precision	F_1 score	Sensitivity	Precision	F_1 score
Bowtie2	88.39%	87.46%	87.92%	59.16%	57.47%	58.30%
BWA-MEM	79.76%	88.52%	83.91%	58.13%	53.14%	55.52%

Appendix B

Second Appendix

B.1 SD index

The SD index is an internal measure to evaluate clustering results. It considers the average scattering of clusters and the total separation of clusters [30]. The average scattering for the clusters, noted S, is calculated by variance of the clusters and variance of the dataset. The variance of dataset is defined as:

$$\sigma_x^p = \frac{1}{N}\sum_{k=1}^{N}(x_k^p - \bar{x}^p)^2$$
$$\sigma(x) = \begin{bmatrix} \sigma_x^1 \\ \vdots \\ \sigma_x^d \end{bmatrix} \tag{B.1}$$

where N is the number of observations, d is the number of dimension and $\bar{x}^p$ is the expected value in the pth dimension.

The variance of cluster is defined as:

$$\sigma_{\nu_i}^p = \frac{1}{\|c_i\|} \sum_{k \in c_i} (x_k^p - \nu_i^p)^2$$
$$\sigma(\nu_i) = \begin{bmatrix} \sigma_{\nu_i}^1 \\ \vdots \\ \sigma_{\nu_i}^d \end{bmatrix} \tag{B.2}$$

where c_i is the ith cluster, $\|c_i\|$ is the number of elements in the ith cluster, ν_i^p is the center point of the ith cluster in the pth dimension.

The average scattering for clusters is defined as:

$$Scatt = \frac{1}{n_c} \sum_{i=1}^{n_c} \frac{\|\sigma(\nu_i)\|}{\|\sigma(x)\|} \tag{B.3}$$

where n_c is the number of clusters, $\|\sigma(\nu_i)\| = \sqrt{\sigma(\nu_i)^T \sigma(\nu_i)}$.

evaluates separation difference based on distances between cluster centers.

The total separation of clusters evaluates the separation of clusters and it is computed based on distances between cluster centers. Let D_{max} and D_{min} denote the largest and the smallest distance between the center of clusters:

$$D_{max} = \max_{i \neq j} \|\nu_i - \nu_j\|$$
$$D_{min} = \min_{i \neq j} \|\nu_i - \nu_j\| \tag{B.4}$$

The total separation of clusters, denoted by $\mathcal{D}$, is given as follows:

$$\mathcal{D} = \frac{D_{max}}{D_{min}} \sum_{k=1}^{n_c} \left(\sum_{i=1, i \neq j}^{n_c} \|\nu_i - \nu_j\| \right)^{-1} \tag{B.5}$$

The SD index is finally defined as:

$$SD = \alpha Scatt + \mathcal{D} \tag{B.6}$$

where α is a weight equal to the value of $\mathcal{D}$ obtained for the partition with the greatest number of clusters.

Bibliography

[1] R. P. Adams, I. Murray, and D. J. MacKay. Tractable nonparametric bayesian inference in poisson processes with gaussian process intensities. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 9–16. ACM, 2009.

[2] C. A. Albers, G. Lunter, D. G. MacArthur, G. McVean, W. H. Ouwehand, and R. Durbin. Dindel: accurate indel calls from short-read data. *Genome Research*, 21(6):961–973, 2011.

[3] A. Altmann, P. Weber, D. Bader, M. Preuß, E. B. Binder, and B. Müller-Myhsok. A beginners guide to SNP calling from high-throughput DNA-sequencing data. *Human Genetics*, 131(10):1541–1554, 2012.

[4] R. Artuso, A. Provenzano, B. Mazzinghi, L. Giunti, V. Palazzo, E. Andreucci, A. Blasetti, R. Chiuri, F. Gianiorio, P. Mandich, et al. Therapeutic implications of novel mutations of the rfx6 gene associated with early-onset diabetes. *The pharmacogenomics journal*, 15(1):49, 2015.

[5] Y. Barash, G. Elidan, N. Friedman, and T. Kaplan. Modeling dependencies in protein-dna binding sites. In *Proceedings of the seventh annual international conference on Research in computational molecular biology*, pages 28–37. ACM, 2003.

[6] A. Barski, S. Cuddapah, K. Cui, T.-Y. Roh, D. E. Schones, Z. Wang, G. Wei, I. Chepelev, and K. Zhao. High-resolution profiling of histone methylations in the human genome. *Cell*, 129(4):823–837, 2007.

[7] A.-L. Boulesteix and K. Strimmer. Predicting transcription factor activities from com-

bined analysis of microarray and chip data: a partial least squares approach. *Theoretical Biology and Medical Modelling*, 2(1):23, 2005.

[8] B. A. Brumback and J. A. Rice. Smoothing spline models for the analysis of nested and crossed samples of curves. *Journal of the American Statistical Association*, 93(443):961–976, 1998.

[9] M. L. Bulyk, P. L. Johnson, and G. M. Church. Nucleotides of transcription factor binding sites exert interdependent effects on the binding affinities of transcription factors. *Nucleic acids research*, 30(5):1255–1261, 2002.

[10] H. J. Bussemaker, H. Li, and E. D. Siggia. Regulatory element detection using correlation with expression. *Nature genetics*, 27(2):167, 2001.

[11] M. Cha and Q. Zhou. Detecting clustering and ordering binding patterns among transcription factors via point process models. *Bioinformatics*, 30(16):2263–2271, 2014.

[12] V. Chaitankar, G. Karakülah, R. Ratnapriya, F. O. Giuste, M. J. Brooks, and A. Swaroop. Next generation sequencing technology and genomewide data analysis: Perspectives for retinal research. *Progress in retinal and eye research*, 55:1–31, 2016.

[13] I. Chambers and A. Smith. Self-renewal of teratocarcinoma and embryonic stem cells. *Oncogene*, 23(43):7150, 2004.

[14] K. Chen, M. D. McLellan, L. Ding, M. C. Wendl, Y. Kasai, R. K. Wilson, and E. R. Mardis. Polyscan: an automatic indel and SNP detection approach to the analysis of human resequencing data. *Genome Research*, 17(5):659–666, 2007.

[15] X. Chen, H. Xu, P. Yuan, F. Fang, M. Huss, V. B. Vega, E. Wong, Y. L. Orlov, W. Zhang, J. Jiang, et al. Integration of external signaling pathways with the core transcriptional network in embryonic stem cells. *Cell*, 133(6):1106–1117, 2008.

[16] C. Cheng and M. Gerstein. Modeling the relative relationship of transcription factor binding and histone modifications to gene expression levels in mouse embryonic stem cells. *Nucleic acids research*, 40(2):553–568, 2011.

[17] G. A. Churchill. Stochastic models for heterogeneous DNA sequences. *Bulletin of Mathematical Biology*, 51(1):79–94, 1989.

[18] A. Conesa, P. Madrigal, S. Tarazona, D. Gomez-Cabrero, A. Cervera, A. McPherson, M. W. Szcześniak, D. J. Gaffney, L. L. Elo, X. Zhang, et al. A survey of best practices for rna-seq data analysis. *Genome biology*, 17(1):13, 2016.

[19] E. M. Conlon, X. S. Liu, J. D. Lieb, and J. S. Liu. Integrating regulatory motif discovery and genome-wide expression analysis. *Proceedings of the National Academy of Sciences*, 100(6):3339–3344, 2003.

[20] D. Das, Z. Nahlé, and M. Q. Zhang. Adaptively inferring human transcriptional subnetworks. *Molecular systems biology*, 2(1), 2006.

[21] M. A. DePristo, E. Banks, R. Poplin, K. V. Garimella, J. R. Maguire, C. Hartl, A. A. Philippakis, G. Del Angel, M. A. Rivas, M. Hanna, et al. A framework for variation discovery and genotyping using next-generation dna sequencing data. *Nature genetics*, 43(5):491, 2011.

[22] P. Diggle. A kernel method for smoothing point process data. *Applied statistics*, pages 138–147, 1985.

[23] M. D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the american statistical association*, 90(430):577–588, 1995.

[24] J. Feng, T. Liu, B. Qin, Y. Zhang, and X. S. Liu. Identifying chip-seq enrichment using macs. *Nature protocols*, 7(9):1728, 2012.

[25] G. B. Fogel, D. G. Weekes, G. Varga, E. R. Dow, A. M. Craven, H. B. Harlow, E. W. Su, J. E. Onyia, and C. Su. A statistical analysis of the transfac database. *Biosystems*, 81(2):137–154, 2005.

[26] N. A. Fonseca, J. Rung, A. Brazma, and J. C. Marioni. Tools for mapping high-throughput sequencing data. *Bioinformatics*, 28(24):3169–3177, 2012.

[27] F. Gao, B. C. Foat, and H. J. Bussemaker. Defining transcriptional networks through integrative modeling of mrna expression and transcription factor binding data. *BMC bioinformatics*, 5(1):31, 2004.

[28] E. Garrison and G. Marth. Haplotype-based variant detection from short-read sequencing. *arXiv preprint arXiv:1207.3907*, 2012.

[29] W. Guo. Functional mixed effects models. *Biometrics*, 58(1):121–128, 2002.

[30] M. Halkidi, M. Vazirgiannis, and Y. Batistakis. Quality scheme assessment in the clustering process. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 265–276. Springer, 2000.

[31] M. S. Hasan, X. Wu, L. T. Watson, Z. Li, and L. Zhang. Ups-indel: A better approach for finding indel redundancy. In *2016 IEEE 6th International Conference on Computational Advances in Bio and Medical Sciences (ICCABS)*, pages 1–1. IEEE, 2016.

[32] M. S. Hasan, X. Wu, and L. Zhang. Performance evaluation of indel calling tools using real short-read data. *Human Genomics*, 9(20), 2015.

[33] N. D. Heintzman, R. K. Stuart, G. Hon, Y. Fu, C. W. Ching, R. D. Hawkins, L. O. Barrera, S. Van Calcar, C. Qu, K. A. Ching, et al. Distinct and predictive chromatin

signatures of transcriptional promoters and enhancers in the human genome. *Nature genetics*, 39(3):311, 2007.

[34] N. L. Hjort, C. Holmes, P. Müller, and S. G. Walker. *Bayesian nonparametrics*, volume 28. Cambridge University Press, 2010.

[35] N. Homer, B. Merriman, and S. F. Nelson. Bfast: an alignment tool for large scale genome resequencing. *PloS one*, 4(11):e7767, 2009.

[36] A. Hubin and G. Storvik. Estimating the marginal likelihood with integrated nested laplace approximation (inla). *arXiv preprint arXiv:1611.01450*, 2016.

[37] D. S. Johnson, A. Mortazavi, R. M. Myers, and B. Wold. Genome-wide mapping of in vivo protein-dna interactions. *Science*, 316(5830):1497–1502, 2007.

[38] S. Keleş, M. van der Laan, and M. B. Eisen. Identification of regulatory elements using a feature selection method. *Bioinformatics*, 18(9):1167–1175, 2002.

[39] D. C. Koboldt, K. Chen, T. Wylie, D. E. Larson, M. D. McLellan, E. R. Mardis, G. M. Weinstock, R. K. Wilson, and L. Ding. Varscan: variant detection in massively parallel sequencing of individual and pooled samples. *Bioinformatics*, 25(17):2283–2285, 2009.

[40] D. C. Koboldt, Q. Zhang, D. E. Larson, D. Shen, M. D. McLellan, L. Lin, C. A. Miller, E. R. Mardis, L. Ding, and R. K. Wilson. Varscan 2: somatic mutation and copy number alteration discovery in cancer by exome sequencing. *Genome research*, 22(3):568–576, 2012.

[41] B. Kulis and M. I. Jordan. Revisiting k-means: New algorithms via bayesian nonparametrics. *arXiv preprint arXiv:1111.0352*, 2011.

[42] Z. Lai, A. Markovets, M. Ahdesmaki, B. Chapman, O. Hofmann, R. McEwen, J. Johnson, B. Dougherty, J. C. Barrett, and J. R. Dry. Vardict: a novel and versatile variant caller for next-generation sequencing in cancer research. *Nucleic acids research*, 44(11):e108–e108, 2016.

[43] B. Langmead and S. L. Salzberg. Fast gapped-read alignment with Bowtie 2. *Nature methods*, 9(4):357–359, 2012.

[44] B. Langmead, C. Trapnell, M. Pop, and S. L. Salzberg. Ultrafast and memory-efficient alignment of short dna sequences to the human genome. *Genome biology*, 10(3):R25, 2009.

[45] M.-L. T. Lee, M. L. Bulyk, G. Whitmore, and G. M. Church. A statistical model for investigating binding probabilities of dna nucleotide sequences using microarrays. *Biometrics*, 58(4):981–988, 2002.

[46] W. Lee and J. S. Morris. Identification of differentially methylated loci using wavelet-based functional mixed models. *Bioinformatics*, 32(5):664–672, 2015.

[47] Y. Lee and Q. Zhou. Co-regulation in embryonic stem cells via context-dependent binding of transcription factors. *Bioinformatics*, 29(17):2162–2168, 2013.

[48] H. Li. Aligning sequence reads, clone sequences and assembly contigs with bwa-mem. *arXiv preprint arXiv:1303.3997*, 2013.

[49] H. Li and R. Durbin. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, 25(14):1754–1760, 2009.

[50] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, and R. Durbin. The sequence alignment/map format and SAMtools. *Bioinformatics*, 25(16):2078–2079, 2009.

[51] H. Li and N. Homer. A survey of sequence alignment algorithms for next-generation sequencing. *Briefings in bioinformatics*, 11(5):473–483, 2010.

[52] H. Li, J. Ruan, and R. Durbin. Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Research*, 18(11):1851–1858, 2008.

[53] K.-C. Li. Genome-wide coexpression dynamics: theory and application. *Proceedings of the National Academy of Sciences*, 99(26):16875–16880, 2002.

[54] F. Lindgren, H. Rue, and J. Lindström. An explicit link between gaussian fields and gaussian markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498, 2011.

[55] C.-M. Liu, T. Wong, E. Wu, R. Luo, S.-M. Yiu, Y. Li, B. Wang, C. Yu, X. Chu, K. Zhao, et al. Soap3: ultra-fast gpu-based parallel alignment tool for short reads. *Bioinformatics*, 28(6):878–879, 2012.

[56] J. T. Lu, Y. Wang, R. A. Gibbs, and F. Yu. Characterizing linkage disequilibrium and evaluating imputation power of human genomic insertion-deletion polymorphisms. *Genome biology*, 13(2):R15, 2012.

[57] S. Mahony and B. F. Pugh. Protein–dna binding in high-resolution. *Critical reviews in biochemistry and molecular biology*, 50(4):269–283, 2015.

[58] T.-K. Man and G. D. Stormo. Non-independence of mnt repressor–operator interaction determined by a new quantitative multiple fluorescence relative affinity (qumfra) assay. *Nucleic acids research*, 29(12):2471–2478, 2001.

[59] E. R. Mardis. Next-generation DNA sequencing methods. *Annual Review of Genomics and Human Genetics*, 9:387–402, 2008.

[60] L. K. Matukumalli, J. J. Grefenstette, D. L. Hyten, I. Y. Choi, P. B. Cregan, and C. P. V. Tassell. Application of machine learning in SNP discovery. *BMC Bioinformatics*, 7(1):1, 2006.

[61] B. McNally, A. Singer, Z. Yu, Y. Sun, Z. Weng, and A. Meller. Optical recognition of converted dna nucleotides for single-molecule dna sequencing using nanopore arrays. *Nano letters*, 10(6):2237–2244, 2010.

[62] E. Meaburn and R. Schulz. Next generation sequencing in epigenetics: insights and challenges. In *Seminars in cell & developmental biology*, volume 23, pages 192–199. Elsevier, 2012.

[63] M. L. Metzker. Sequencing technologies—the next generation. *Nature Reviews Genetics*, 11(1):31–46, 2010.

[64] T. S. Mikkelsen, M. Ku, D. B. Jaffe, B. Issac, E. Lieberman, G. Giannoukos, P. Alvarez, W. Brockman, T.-K. Kim, R. P. Koche, et al. Genome-wide maps of chromatin state in pluripotent and lineage-committed cells. *Nature*, 448(7153):553, 2007.

[65] R. E. Mills, W. S. Pittard, J. M. Mullaney, U. Farooq, T. H. Creasy, A. A. Mahurkar, D. M. Kemeza, D. S. Strassler, C. P. Ponting, C. Webber, and S. E. Devine. Natural genetic variation caused by small insertions and deletions in the human genome. *Genome Research*, 21(6):830–839, 2011.

[66] J. Møller, A. R. Syversveen, and R. P. Waagepetersen. Log gaussian cox processes. *Scandinavian journal of statistics*, 25(3):451–482, 1998.

[67] J. S. Morris, P. J. Brown, R. C. Herrick, K. A. Baggerly, and K. R. Coombes. Bayesian analysis of mass spectrometry proteomic data using wavelet-based functional mixed models. *Biometrics*, 64(2):479–489, 2008.

[68] J. S. Morris and R. J. Carroll. Wavelet-based functional mixed models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(2):179–199, 2006.

[69] A. Mortazavi, B. A. Williams, K. McCue, L. Schaeffer, and B. Wold. Mapping and quantifying mammalian transcriptomes by rna-seq. *Nature methods*, 5(7):621, 2008.

[70] J. M. Mullaney, R. E. Mills, W. S. Pittard, and S. E. Devine. Small insertions and deletions (INDELs) in human genomes. *Human Molecular Genetics*, 19(R2):R131–R136, 2010.

[71] P. Müller and F. A. Quintana. Nonparametric bayesian data analysis. *Statistical science*, pages 95–110, 2004.

[72] I. Murray, Z. Ghahramani, and D. MacKay. Mcmc for doubly-intractable distributions. *arXiv preprint arXiv:1206.6848*, 2012.

[73] R. M. Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of computational and graphical statistics*, 9(2):249–265, 2000.

[74] Y. M. Oh, J. K. Kim, S. Choi, and J.-Y. Yoo. Identification of co-occurring transcription factor binding sites from dna sequence using clustered position weight matrices. *Nucleic acids research*, 40(5):e38–e38, 2011.

[75] Z. Ouyang, Q. Zhou, and W. H. Wong. Chip-seq of transcription factors predicts absolute and differential gene expression in embryonic stem cells. *Proceedings of the National Academy of Sciences*, 106(51):21521–21526, 2009.

[76] S. Pabinger, A. Dander, M. Fischer, R. Snajder, M. Sperk, M. Efremova, B. Krabichler, M. R. Speicher, J. Zschocke, and Z. Trajanoski. A survey of tools for variant analysis of next-generation genome sequencing data. *Briefings in Bioinformatics*, 15(2):256–278, 2014.

[77] M. A. Pereira, F. S. V. Malta, M. C. M. Freire, and P. G. P. Couto. Application of next-generation sequencing in the era of precision medicine. In *Applications of RNA-Seq and Omics Strategies-From Microorganisms to Human Health*. InTech, 2017.

[78] B. Rabbani, H. Nakaoka, S. Akhondzadeh, M. Tekin, and N. Mahdieh. Next generation sequencing: implications in personalized medicine and pharmacogenomics. *Molecular BioSystems*, 12(6):1818–1830, 2016.

[79] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[80] J. A. Rice and B. W. Silverman. Estimating the mean and covariance structure non-parametrically when the data are curves. *Journal of the Royal Statistical Society: Series B (Methodological)*, 53(1):233–243, 1991.

[81] A. Rimmer, H. Phan, I. Mathieson, Z. Iqbal, S. Twigg, A. Wilkie, G. McVean, G. Lunter, W. Consortium, et al. Integrating mapping-, assembly-and haplotype-based approaches for calling variants in clinical sequencing applications. *Nature Genetics*, 46(8):912–918, 2014.

[82] Y. A. Rozanov. Markov random fields. In *Markov Random Fields*, pages 55–102. Springer, 1982.

[83] H. Rue, S. Martino, and N. Chopin. Approximate bayesian inference for latent gaussian models by using integrated nested laplace approximations. *Journal of the royal statistical society: Series b (statistical methodology)*, 71(2):319–392, 2009.

[84] F. Sänger, S. Nicklen, and A. Coulson. Dna sequencing with chain-terminating inhibitors. *Proc. Natl. Acad. Sci. USA*, 74:5463–5467, 1977.

[85] J. Shendure and H. Ji. Next-generation dna sequencing. *Nature biotechnology*, 26(10):1135, 2008.

[86] 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.

[87] J. G. Staniswalis and J. J. Lee. Nonparametric regression analysis of longitudinal data. *Journal of the American Statistical Association*, 93(444):1403–1418, 1998.

[88] N. Sun, R. J. Carroll, and H. Zhao. Bayesian error analysis model for reconstructing transcriptional regulatory networks. *Proceedings of the National Academy of Sciences*, 103(21):7988–7993, 2006.

[89] H. Tran, H. Zhu, X. Wu, G. Kim, C. Clarke, H. Larose, D. Haak, S. Askew, J. Barney, J. Westwood, et al. Identification of differentially methylated sites with weak methylation effects. *Genes*, 9(2):75, 2018.

[90] F. R. Vogenberg, C. I. Barash, and M. Pursel. Personalized medicine: part 1: evolution and development into theranostics. *Pharmacy and Therapeutics*, 35(10):560, 2010.

[91] K.-C. Wong, Y. Li, C. Peng, and Z. Zhang. Signalspider: probabilistic pattern discovery on multiple normalized chip-seq signal profiles. *Bioinformatics*, 31(1):17–24, 2014.

[92] H. Wu and J.-T. Zhang. Local polynomial mixed-effects models for longitudinal data. *Journal of the American Statistical Association*, 97(459):883–897, 2002.

[93] B. J. Yoon. Hidden Markov models and their applications in biological sequence analysis. *Current Genomics*, 10(6):402–415, 2009.

[94] F. Zeng, R. Jiang, and T. Chen. PyroHMMsnp: an SNP caller for Ion Torrent and 454 sequencing data. *Nucleic Acids Research*, 41(13):e136–e136, 2013.

[95] F. Zeng, R. Jiang, and T. Chen. PyroHMMvar: a sensitive and accurate method to call short indels and SNPs for Ion Torrent and 454 data. *Bioinformatics*, 29(22):2859–2868, 2013.

[96] Y. Zhang, T. Liu, C. A. Meyer, J. Eeckhoute, D. S. Johnson, B. E. Bernstein, C. Nusbaum, R. M. Myers, M. Brown, W. Li, et al. Model-based analysis of chip-seq (macs). *Genome biology*, 9(9):R137, 2008.

[97] S. Zhao, K. Watrous, C. Zhang, and B. Zhang. Cloud computing for next-generation sequencing data analysis. *Cloud Computing-Architecture and Applications, InTech, Rijeka*, pages 29–51, 2017.

[98] Q. Zhou, H. Chipperfield, D. A. Melton, and W. H. Wong. A gene regulatory network in mouse embryonic stem cells. *Proceedings of the National Academy of Sciences*, 104(42):16438–16443, 2007.

[99] H. Zhu, P. J. Brown, and J. S. Morris. Robust, adaptive functional regression in functional mixed model framework. *Journal of the American Statistical Association*, 106(495):1167–1179, 2011.

[100] J. M. Zook, B. Chapman, J. Wang, D. Mittelman, O. Hofmann, W. Hide, and M. Salit. Integrating human sequence data sets provides a resource of benchmark SNP and indel genotype calls. *Nature Biotechnology*, 32(3):246–251, 2014.