Spring 2015

Zero-Inflated Models to Identify Transcription Factor Binding Sites in ChIP-seq Experiments

Sameera Dhananjaya Viswakula
Old Dominion University

ZERO-INFLATED MODELS TO IDENTIFY

TRANSCRIPTION FACTOR BINDING SITES IN

CHIP-SEQ EXPERIMENTS

by

Sameera Dhananjaya Viswakula
B.S. August 2008, University of Colombo, Sri Lanka
M.S. July 2011, University of Texas at El Paso, TX

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
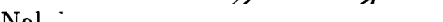Requirements for the Degree of

DOCTOR OF PHILOSOPHY

MATHEMATICS AND STATISTICS

OLD DOMINION UNIVERSITY
May 2015

Approved by:

Norou Diawara (Director)

Nal

N. Rao Chaganty (Member)

N.]

Michael Doviak (Member)

ABSTRACT

ZERO-INFLATED MODELS TO IDENTIFY TRANSCRIPTION FACTOR BINDING SITES IN CHIP-SEQ EXPERIMENTS

Sameera Dhananjaya Viswakula
Old Dominion University, 2015
Director: Dr. Norou Diawara

It is essential to determine the protein-DNA binding sites to understand many biological processes. A transcription factor is a particular type of protein that binds to DNA and controls gene regulation in living organisms. Chromatin immunoprecipitation followed by highthroughput sequencing (ChIP-seq) is considered the gold standard in locating these binding sites and programs use to identify DNA-transcription factor binding sites are known as peak-callers. ChIP-seq data are known to exhibit considerable background noise and other biases. In this study, we propose a negative binomial model (NB), a zero-inflated Poisson model (ZIP) and a zero-inflated negative binomial model (ZINB) for peak-calling. Using real ChIP-seq datasets, we show that ZINB model is the best model for ChIP-seq data. Then we incorporate control data, GC count information, and mappability information into the ZINB regression model as covariates using two link functions. We implemented this approach in C++, and our peak-caller chooses the optimal parameter combination for a given dataset. Performace of our approach is compared with two frequently used peak-callers: QuEST and MACS.

To my parents.

ACKNOWLEDGEMENTS

TABLE OF CONTENTS

APPENDICES

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Deoxyribonucleic acid (DNA) and protein are two of the most important biomolecules in any living organism. DNA carries genetic information, and proteins execute and regulate the life processes. As a result, protein-DNA interactions play a crucial role in central biological processes such as transcription of genes into ribonucleic acid (RNA) and repair of damaged DNA. It is essential to determine the way proteins interact with DNA molecules to better understand biological processes. Specifically, proteins can promote the transcription of genes near the binding site by binding to certain DNA segments [1]. These kinds of proteins are referred to as transcription factors (TFs). Transcription factors control the transcription of genetic information in living cells from DNA to mRNA, and any malformations in this process can cause serious diseases such as cancer [2]. For this reason, it is of vital importance to identify the positions and intensities of transcription factor binding sites (TFBS) across the genome. Chromatin immunoprecipitation followed by massively parallel sequencing (ChIP-seq) is the most successful approach to identify such protein-DNA interactions *in vivo* on a genome-wide scale [3]. In this study, we statistically model ChIP-seq data and determine the actual transcription factor binding sites and their intensities using a novel computer algorithm.

Protein-DNA interactions are studied both computationally and in wet labs. Traditional methods are very slow and laborious. While wet lab experiments provide data and problems for computational methods to solve, computational methods test hypotheses and guide additional lab experiments. As the sequencing cost has continued to drop, several new high-throughput methods have been introduced. These Next Generation Sequencing (NGS) techniques can provide comprehensive binding information much more rapidly and generate several gigabytes (GBs) of data from a single experiment.

1.1 EVOLUTION OF NUCLEIC ACID SEQUENCING

The method of determining the order of nucleotides in a given DNA or RNA molecule is called nucleic acid sequencing [4]. After the completion of the Human Genome Project in 2003, the use of nucleic acid sequencing was dramatically increased. The Human Genome Project was an international achievement to sequence and map all of the chromosomes of human beings, and it took 13 years and cost 3 billion dollars. It was accomplished with first-generation sequencing, known as "Sanger sequencing" [5]. Sanger sequencing was developed by Edward Sanger in 1975. For the subsequent two and a half decades, it was considered the gold standard for nucleic acid sequencing.

However, due to the limited throughput, high cost, and the huge amount of time involved with Sanger-sequencing, demand for cheaper and faster methods has increased substantially. This demand has driven the development of second generation or Next Generation Sequencing (NGS) methods. NGS platforms perform massively parallel sequencing which enables researchers to sequence the entire genome in less than one day [4]. Over the past few years, several NGS platforms have been developed, and these platforms provide low-cost and high-throughput sequencing. Amongst these platforms, four commonly used platforms are Illumina MiSeq/ HiSeq, Roche 454, Ion Torrent, and SOLiD Technologies. For more detailed discussion of each technique, one can refer to the review article by Mardis [6].

The Illumina Genome Analyzer is widely used, and it generates mainly three types of data: ChIP-seq, DNA-seq, and RNA-seq [7]. DNA-seq is used to measure the Single Nucleotide Polymorphism (SNPs) or DNA copy number variation and RNA-seq is used to examine gene expression. ChIP-seq is used to study protein binding patterns. In this study, our goal is to investigate DNA-protein binding using ChIP-seq data.

1.2 CHIP-SEQ DATA ANALYSIS: WORKFLOW/PROTOCOL

Methods for mapping TF binding occupancy across the genome by chromatin immunoprecipitation (ChIP) were developed more than a decade ago [8]. In ChIP assays, a transcription factor or other chromatin protein of interest is enriched by immunoprecipitation from cross-linked cells along with its associated DNA. In earlier

days, genomic DNA sites enriched in this manner were identified by DNA hybridization to a microarray. This technique is known as Chip-chip or ChIP-on-chip [9]. More recently, chromatin immunoprecipitation followed by high-throughput sequencing (ChIP-seq) was introduced to determine the location of DNA binding sites on the genome for a particular protein of interest [10–13]. ChIP-seq has become the most widely used procedure for genome-wide assays of protein-DNA interaction [14]. ChIP-seq reports the count of reads that map to a particular location of interest. It uses only short reads to align to the genome and, as a result, it requires millions of them to provide meaningful data [15]. Due to this requirement, most platforms generate an enormous amount of data within a short period of time, and the cost involved with the techniques are drastically decreasing. For example, Illumina's Solexa 1G NGS produces up to 30 million 21-35 base-pairs (bp) reads per run.

A typical ChIP-seq experimental protocol is as follows. Cells or tissues are treated with a chemical agent, usually formaldehyde, to cross-link proteins covalently to DNA. Formaldehyde is often used because it is heat-reversible. Cross linking is followed by cell disruption and sonication or, in some cases, enzymatic digestion, to shear the chromatin to millions of 100-300 bp fragments [9]. Then the TF with its bound DNA is enriched using a specific antibody by purification (Figure 1).

After immuno-enrichment, the DNA fragments are separated from the protein by reverse cross-linking. Then these fragments are amplified by polymerase chain reaction (PCR). These DNA fragments are sequenced in a series of 20-80 bp on one end (in single-end tag sequencing/ SET data) or on both ends (in paired-end tag sequencing/ PET data) of the fragment, producing millions of short read sequences. These short sequences are usually known as "tags." ChIP-seq was one of the first methods to make use of the power of massively parallel or NGS, using significantly advance real-time PCR and array-based methods. The use of NGS provides relatively high resolution, low noise, and high genomic coverage compared with ChIP-chip assays [16].

Millions of reads generated in each ChIP experiment need to be analyzed, and this analysis begins with alignment to a reference genome. This alignment is done computationally using specially designed algorithms. Bowtie [17], ELANDv2 [18], and MAQ [19] have been the most widely used mapping software for NGS data. Each has its unique set of capabilities, parameters, and options, but these aligners are similar in many ways [20]. The objective of this alignment step is to determine all the
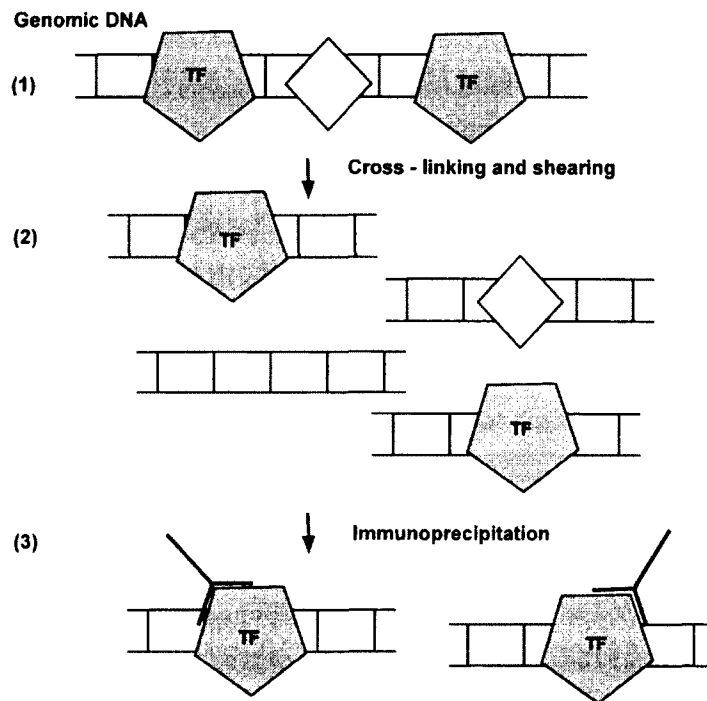
Figure 1 ChIP-seq protocol (a): (1) Transcription factor (TF) is cross-linked to its binding site in genomic DNA. **(2)** DNA is sheared into small fragments by ultrasound sonication. **(3)** A protein specific antibody (shown as symbol Y) is used to separate only the cross-linked fragments.

locations in a reference genome that show perfect or near perfect matches to a given read. This process has to be done for each short read in the dataset [21]. Usually, mapping is done while allowing for a small number (1-3) of sequence mismatches. Different alignment algorithms trade speed for quality of the final alignment, and usually they report the quality of each map in their output.

ChIP-seq analysis begins with mapping of sequence reads to a reference genome. Then peak-calling algorithms are used to find peaks. Differential binding or motif analyses are conducted during the last stages of the ChIP-seq workflows [15]. In this study, our focus is on the peak-calling process.
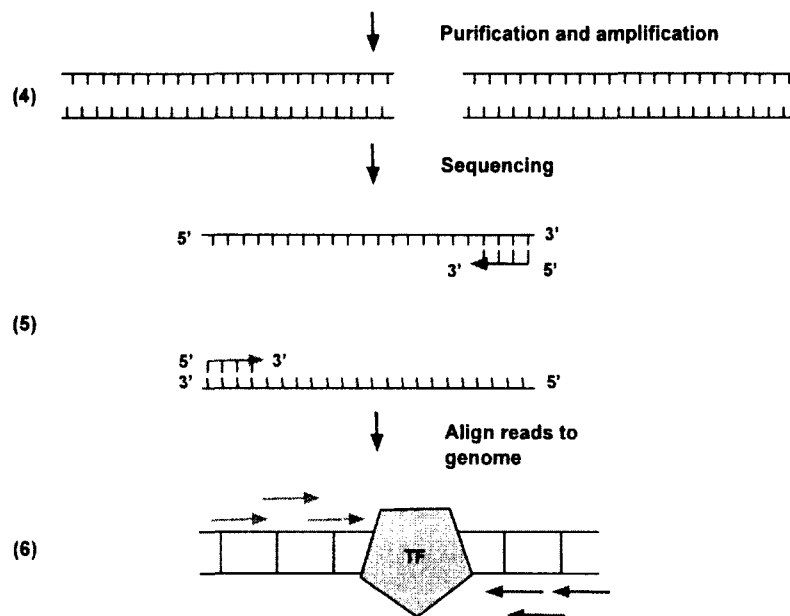
Figure 2 ChIP-seq protocol (b): **(4)** Fragments are purified by reverse cross-linking the TFs and amplified by polymerase chain reaction (PCR). **(5)** Then the the first few bases of fragment ends are sequenced (usually 20-80 bp). **(6)** These short sequences, "tags," are mapped back to a reference genome.

1.3 PEAK-CALLERS

The raw ChIP-seq data are not readily interpretable. ChIP-seq data are a set of alignments of reads across the whole genome. The potential binding sites are identified on the basis of the significant tag accumulation at particular genomic loci. High concentration of tags at a particular location forms a peak as shown in Figure 3. The programs dedicated to identifying potential DNA-protein binding sites identify these tag accumulations one way or the other. Since they identify "peaks," these programs are known as **peak-callers**.
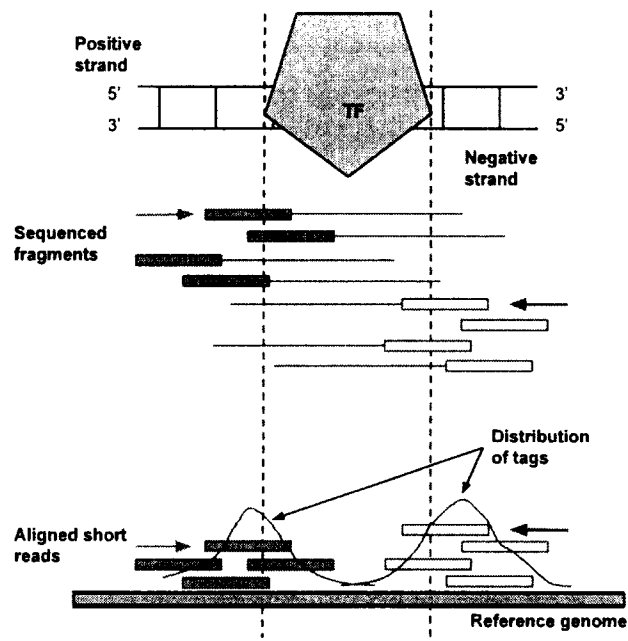
Figure 3 Tag accumulation and peak-calling. Arrow represents the directionality of the binding. The sequenced fragments are mapped back to a reference genome. In a given location, there can be one or more mapped tags. These tag accumulations are the basis of peak-calling.

1.4 CHALLENGES IN CHIP-SEQ DATA AND POTENTIAL IMPROVEMENTS OF EXISTING PEAK-CALLERS

1.4.1 OMITTING STRAND INFORMATION

In 1953, James Watson and Francis Crick proposed a structure for the DNA molecule. Their structure suggested the underlying mechanism of DNA replication, and they proposed that DNA is composed of two side-by-side chains of DNA running anti-parallel to each other [22]. These chains are known as "strands" and are twisted into the shape of a double helix. These two strands are fastened by weak associations between bases of each strand, forming a structure like a spiral staircase. These associations are weak hydrogen bonds, and they are base specific. That means A

(adenine) can only form hydrogen bonds with T (thymine) and C (cytosine) binds only with G (guanine). In other words, if you know the sequence of bases in one strand, you know the respective sequence of bases on the other strand.

The backbone of each strand is a repeating phosphate-sugar polymer. Based on the numbering convention of the carbons of sugar groups, each sugar-phosphate backbone is said to have a 5'-to-3' polarity [23]. Since the DNA is double-stranded and two backbones are in opposite directions, when one strand is oriented 5' → 3', the other strand is 3' → 5'. The DNA strand with the 5' → 3' is referred to as the left (forward/positive/Watson) strand and the other one is called the right (backward/negative/Crick) strand.

If a transcription factor has a sharply focused binding site, in a successful ChIP-seq experiment, one should be able to observe two peaks, one on each strand [24]. The reason for these two peaks (shown in Figure 4 using a typical region of STAT1 ChIP-seq dataset) is that a fragment is always sequenced from its ends toward its midpoint. Therefore, the actual transcription factor binding site is located in between the peak on the left strand (as shown by the left peak in Figure 3) and the peak on the right strand (as shown by the right peak in Figure 3) [24]. For this reason, a challenge of peak calling in ChIP-Seq data is how to combine the tag counts from the two strands to increase the power of detecting real protein-DNA interaction sites [25].

Chromosome 1: Start position: 191592850
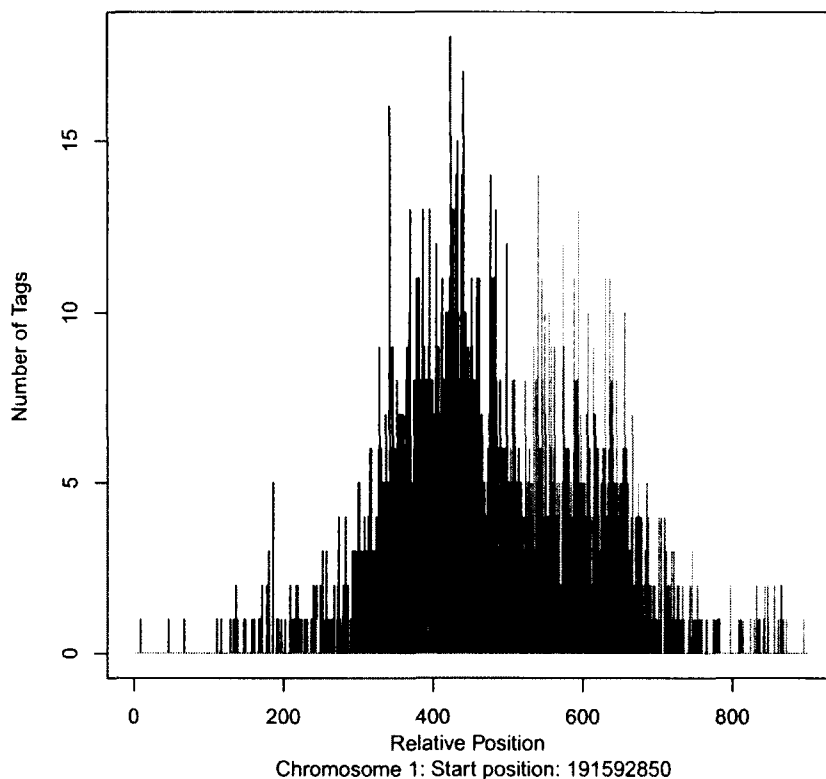
Figure 4 Two peak structure of a successful ChIP-seq experiment. If the experiment is successful, one peak on each strand is observed.

Several peak calling programs have been published since 2008, including MACS [26], SPP [27], and QuEST [28]. MACS [26] estimates a global peak shift size from regions with significant fold changes. After estimating the global peak shift size, MACS then shifts and combines all forward and reverse strand tags toward the center by the estimated shift size and calls peaks on the combined tags using a Poisson model through sliding windows. SPP [27] first selects a global peak shift size by maximizing a linear Pearson correlation of the tag counts between forward and reverse strands. SPP then chooses a window size based on the estimated peak shift size, and it utilizes a sliding window and calls peaks using a score based method. QuEST [28] also determines a global peak shift size. It calls peaks based on a combined score profile created by incorporating the estimated peak shift.

All of the peak-callers mentioned above, estimate a global peak shift size for all potential binding regions. They simply merge the forward and the reverse strand tags and then call the peaks. Therefore, they may lose spatial resolution. Also, when the two strands are combined, some valuable information that can be used to distinguish real binding locations from spurious peaks is also lost.

1.4.2 MODELING OBSERVED TAG COUNTS

As explained in section 1.2, ChIP-seq experiments generate millions of mapped tags. A typical peak-caller would retain only uniquely mapped tags and summarize total tag counts in each small nonoverlapping interval of the genome (referred to as a bin or a window). Poisson distribution is a simple choice for modeling count data, and it assumes the equality of mean and variance. If the variability of tag counts far exceeds the mean (over-dispersion), then a Poisson model is not the best fit for the data. The negative binomial distribution can be treated as an extension of Poisson distribution to handle overdispersion. Also, ChIP-seq data are usually zero-inflated. Therefore, there is still room for improvement.

1.4.3 BACKGROUND NOISE

In the ChIP-seq protocol, most of the unbound DNA fragments are washed out in the immuno-precipitation procedure. However, considerable "non-useful" fragments remain in the library due to the random protein-DNA or antibody samples that are not position-specific. Reads sequenced from these fragments are spread throughout the genome and are considered "background" noise. Therefore, the reads in a ChIP sample can be regarded as a mixture of enrichment "signal" reads and "background" noise reads [29]. Therefore, the peak-calling problem is a process of separating noise from the background. In early ChIP-seq applications without a control, the distribution of the noise was assumed to be uniform [11]. However, recent studies demonstrated that the uniform model is too unrealistic due to the various factors such as sequencing and mapping biases, chromatin structure and genome copy number variations [26,30]. For this reason, the adjustment of these intrinsic biases requires a negative control, which could be generated using non-specific antibody or input DNA [29].

For example, when shearing the DNA, open chromatin regions tend to be fragmented more efficiently than the closed areas. This phenomenon can cause an uneven

distribution of sequence tags across the genome. These open chromatin regions may be associated with higher background signals [31]. Also, repetitive sequences might seem to be enriched because of inaccuracies in the number of copies of the repeats in the assembled genome. Therefore, it is important to compare a peak in the ChIP-seq profile with the corresponding region in a negative control sample [24].

There are three types of commonly used negative control samples. The first type is total DNA input control (*Input DNA*), where non-immunoprecipitated DNA is sheared and sequenced. The second type is *Mock IP control*, where a nonspecific antibody like the immunoglobulin G (*IgG*) is used most of the time. The third type includes all specially designed controls. For instance, if the ChIP is performed on stimulated cells, then using the same antibody on unstimulated cells is a good negative control [21, 32]. Input DNA is the most commonly used control in two-sample ChIP-seq data analysis [12, 13].

Some of the commonly used tools for ChIP-seq analysis require a matched control sample for significant peak detection. Most of them use the fold enrichment, calculated as the ratio of IP and control counts over a region, to identify significant peaks and screen out weak peaks [33, 34]. However, Ho et al. [35] carried out a systematic analysis of ChIP-seq and ChIP-chip datasets and revealed that the input data have variable effects on peak finding. They pointed out the urgency of high-quality input samples for peak-calling. Incorporating control data in our model is discussed in detail in section 3.2.

1.4.4 ADJUSTING FOR GC BIAS

GC content bias is the dependence between G and C bases in a region and the count of ChIP-seq reads in it [7]. This is mainly due to PCR amplification bias in the sample preparation step [36]. Theoretically, this kind of bias can result in an over-representation of GC-rich regions in peaks [3]. GC bias is a well-documented problem, and when modeling ChIP-seq data, this bias should be corrected. Dohm et al. [36] first described this GC bias and stated that ChIP-seq tags are correlated with GC content. In particular, they showed that regions with higher GC content exhibit a greater number of tags. Until recently, GC effect is assumed to be positively correlated with the tag count. Kuan et al. [37] and Benjamini et al. [7] showed that, this GC effect is unimodal, and it is not consistent with repeated experiments or their own reports.

Most current correction methods use the following scheme. They bin both tag counts and GC counts. This bin size is arbitrary. Then a curve describing the conditional mean tag count per GC value is estimated either by binning or smoothing. This estimated curve is used to determine the predicted count for each bin based on its GC count. These predicted values are used to normalize the original signal. While these methods remove most of the GC effect, they do not take the prior information such as the unimodality of the GC curve into account [7]. However, those methods consider the entire genome or the overall distribution of the ChIP-seq data when estimating the GC curve [3]. When finding potential peaks, our focus is only on the ChIP-enriched regions. For this reason, it is important to focus only on the GC effect of those areas of the genome and the tags on them.

1.4.5 MAPPABILITY BIAS

In the ChIP-seq experiment, millions of sequence reads are produced. Then, these reads should be mapped to a reference genome. During this process, a read can be mapped to a unique location in the genome or it can be mapped to multiple locations. As a result, mapping is not uniform across the genome. This mappability bias should be corrected before finding peaks [33,37]. First, the locations that can be mapped uniquely should be identified. A *mappable* location on a genome is a position that a tag can be mapped uniquely. If a tag cannot be mapped uniquely to a location on the genome, those loci are referred to as *unmappable* locations. Mapping on a genome depends on the length of the sequence tag. For example, a tag with 25 bp would map differently across the genome than a tag with 35 bp. In this study, a mappability map generator program Peak-Seq Suite, published by Rozowsky et al. [33] was used to identify the locations with and without unique maps across the genome. This program was run on an HPC cluster and took about seven days to output a mappability map for a given tag length. It outputs a set of binary files, and the `Python` script that comes with Peak-Seq Suite can be used to read them. This `Python` script was modified to output 1 if the location is a mappable location and 0 otherwise. This mappability information is used in our program to adjust for the mappability bias.

Although numerous algorithms have been proposed recently for analyzing the large ChIP-seq datasets, their relative dominances and constraints remain unclear in practical applications [38–41]. Most of the peak-callers use a window/bin based

method, and all the nucleotides within each genomic window are assumed to have the same characteristic score (i.e. tag counts, GC, mappability, control count, etc.). There are several disadvantages of these window-based methods. These window sizes are decided based on a couple of datasets; therefore, their applicability for new datasets is questionable. Most of the peak-callers have an option to change the default window size. However, it can be time-consuming to tune these parameters for each dataset. Moreover, GC bias should be considered for each fragment instead of a bin, which is generally much shorter than a fragment.

Kim et al. [42] proposed a "per-base model" for predicting TFBS using a normal-exponential density, and their peak-caller, "NEXT-Peak" is available at http://www.people.vcu.edu/~nkkim/nextpeak.html. This study is a further development of their work. The NEXT-Peak algorithm is explained in section 2.2. In the next section, methods of three commonly used peak-callers are discussed.

1.5 EXISTING MODELS/ PROGRAMS

In this study, we compare our technique with two commonly used peak-callers: QuEST [28] and MACS [26]. An overview of these programs are given below.

1.5.1 QUANTITATIVE ENRICHMENT OF SEQUENCE TAGS: QUEST

Quantitative enrichment of sequence tags (QuEST) is a statistical method based on the kernel density estimation approach [28]. It first constructs two separate profiles: one for forward tags and the other for reverse tags. These profiles are characterized by areas of substantial enrichment where tags abound. The distance between the forward and reverse profiles is not known and varies from experiment to experiment. QuEST estimates this distance from a robust subset of the data. Then the forward and reverse profiles are shifted and summed to produce the combined density profile (CDP). QuEST then identifies candidates for CDP as positions in the reference genome corresponding to local maxima of the CDP with sufficient enrichment in the control data. The CDP threshold value can vary from experiment to experiment. For this reason, QuEST identifies the false positives by separating the control data into two sets and then identifies peaks, treating one set as a pseudo-IP and the other as a control sample. Then the false discovery rate is calculated as the ratio of the number of peaks predicted by the pseudo-IP analysis to the number of peaks identified in the real ChIP experiment. By doing this, QuEST lets users set a specific threshold and

determine a false discovery rate.

1.5.2 MODEL-BASED ANALYSIS OF CHIP-SEQ: MACS

MACS uses a dynamic Poisson distribution to capture local biases in the genome. It takes advantage of the bimodal pattern of ChIP-seq data to empirically model the shifting size [26]. One of the main differences in MACS and other commonly used peak-callers is that MACS removes duplicate tags based on a binomial distribution p-value calculated using the sequencing depth of the dataset. It captures both mean and variance of the distribution using a single parameter, λ_{BG} (Poisson rate). MACS shifts all tags by $d/2$ toward the 3' ends, where "d" is the distance between the modes of the two strands. Then it slides $2d$ windows across the genome to find candidate peaks with a significant tag enrichment. MACS merges the overlapping enriched peaks, and each tag position is extended d bases from its center. The location with the highest fragment pileup is predicted as the binding location.

When control data are available, MACS uses a dynamic parameter, λ_{local}, defined for each candidate peak as:

$$\lambda_{local} = max(\lambda_{BG}, \lambda_{1k}, \lambda_{5k}, \lambda_{10k}),$$

where λ_{1k}, λ_{5k} and λ_{10k} are λ estimates from 1 kb, 5kb or 10 kb peak centered window in the control sample. If a control sample is not available, λ_{local} is calculated the same way without λ_{1k} using only an IP sample. MACS uses λ_{local} to calculate the p-value of each candidate peak and removes the potential false positives based on a threshold p-value. MACS defines the ratio between the ChIP-seq tag count and λ_{local} as the *fold_enrichment* and reports in the final output.

1.6 USE OF MOTIF SITES FOR VALIDATION

Since peak-calling algorithms predict different numbers of binding sites for a given dataset, it is of foremost importance to compare the performances of peak-callers by assessing the validity of these predicted sites. However, actual binding sites are not available. A DNA-TF interaction depends on the DNA sequence. Many of these TFs recognize a specific DNA sequence pattern, known as a motif, that binds to these locations. A motif is a DNA sequence pattern with a biological meaning, which repeatedly occurs in DNA sequences. Typically, motifs are 5-15 bp long. As a result, performances of peak-callers are often compared using the sites with a high similarity with a motif [40].

However, these motif sites are statistically or computationally predicted entities, and transcription factor binding sites are real biological entities. Finding a motif is a well-known problem in bioinformatics, and there are many motif finding algorithms available in the literature including web tools with user-friendly interfaces [43]. JAS-PAR [44] and TRANSFAC [45] are among the most frequently used motif databases. Motif sites from the JASPAR database were downloaded for this analysis.

1.7 CHIP-SEQ DATA

For the first part of the study, we used three ChIP-seq datasets corresponding to transcription factors (TFs) NRSF, ZNF143 and STAT1. The NRSF [SRA:SRR577995] and ZNF143 [SRA:SRR243553] datasets were downloaded from the SRA database at http://www.ncbi.nlm.nih.gov/sra. The STAT1 dataset [11] was downloaded at http://www.bcgsc.ca/downloads/chiptf/human/STAT1/stimulated/july_23_2008/.

For the second part of the analysis, we used NRSF, STAT1, GABP and STAT6 datasets. The NRSF IP and input [GEO:GSE49570] datasets, STAT1 IP and input [GEO:GSE12782] datasets and STAT6 IP and input [GEO:GSE41317] datasets were downloaded from the GEO database at http://www.ncbi.nlm.nih.gov/geo/. GABP IP and input datasets were downloaded from http://mendel.stanford.edu/sidowlab/downloads/quest/. STAT6 is a paired-end tag dataset and all other datasets are single-end tag datasets. Table 1 gives the number of tags and the sequenced tag length in each dataset.

TABLE 1 Description of datasets used in this analysis

Dataset	Number of Tags	Tag Length	Genome
NRSF	33,075,991	36	hg19
STAT1	15,126,001	27	hg18
ZNF143	25,155,931	36	hg18
NRSF - IP	124,107,966	51	hg19
NRSF - Control	182,058,733	51	hg19
STAT1 - IP	9,538,797	28	hg18
STAT1 - Control	15,202,326	28	hg18
GABP - IP	7,857,864	24	hg18
GABP - Control	17,249,435	24	hg18
STAT6 - IP	42,164,376	50	mm8
STAT6 - Control	38,436,442	50	mm8

1.7.1 PREPROCESSING DATA

Raw sequence reads for NRSF, ZNF143 and STAT6 were in the SRA binary format, and the "fastq-dump" tool of the SRA Toolkit was used to convert them into the "fastq" format. This conversion generates a "fastq" file for each raw file. For example, for the NRSF.sra file, the NRSF.fastq file is created. Figure 5 shows three sequences from the NRSF.fastq file. Each sequence begins with an "@" sign and usually has four lines per sequence.

These raw reads were then mapped to their corresponding genomes (hg18, hg19 or mm8) using Bowtie2 [17] software. Human and mouse genome reference sequences were downloaded from the University of California Santa Cruz (UCSC) Genome Browser at http://genome.ucsc.edu/ [46]. More information about datasets is given in Table 1. These ChIP-seq datasets are typically 2-5GBs in size, and high-performance computers are needed to process and manipulate data. High-performance computing (HPC) clusters at Old Dominion University and the Department of Biostatistics of Virginia Commonwealth University in Virginia were used to run Bowtie2 [17] and all other programs used in this study. A typical dataset with 10

```
@SRR948187.7 HISEQ2:205:D07H3ACXX:2:1101:1217:2230 length=51
TCTNNNNNNNNNNNNNNNNNNNNNNNNNNNCNNNNNNNNTGTGTTTTCATAAGCT
+SRR948187.7 HISEQ2:205:D07H3ACXX:2:1101:1217:2230 length=51
####################################################
@SRR948187.8 HISEQ2:205:D07H3ACXX:2:1101:1458:2045 length=51
NTCAGNACTNTGGGAGGCTGAGGCAGGAGAATCGCTTGAACCTGGAAGGTG
+SRR948187.8 HISEQ2:205:D07H3ACXX:2:1101:1458:2045 length=51
#0;=?#2=@#2=?@@@?@??@?@????<???<?>=<?5>???;;?????3=
@SRR948187.9 HISEQ2:205:D07H3ACXX:2:1101:1263:2046 length=51
NGAAGNNNNNNNNNNNNNNNNNNNNNNNNNGNNNNNNNTAAGCAGCTGCCGGAA
+SRR948187.9 HISEQ2:205:D07H3ACXX:2:1101:1263:2046 length=51
#07;?########################################
```

Figure 5 Three sequences in fastq format. SRA binary format was converted to fastq format using the fastq-dump tool of the SRA Toolkit. Each sequence begins with an "@" sign followed by a sequence identifier. The second line is the raw sequence letters. The third line gives optional information followed by a "+" sign, and the fourth line gives the quality value of the sequence specified in line 2.

million sequences would take about 6 hours to map to a genome on an Intel Xeon E5-2660 v2 2.2GHz computer with 128GB memory. Bowtie2 outputs alignment details in "SAM" format (Figure 6). Alignments for the three sequences shown in Figure 5 are presented in Figure 6.

In the SAM format, there are at least 12 fields available. There can be additional fields after the 12th column, depending on the experiment. These fields are tab-delimited. The first field of the SAM format is the name of the read that has been aligned to a reference sequence, i.e. SRR948187.7, SRR948187.8, and SRR948187.9. In Figure 5, the name of the first sequence is SRR948187.7 in the Bowtie2 output in Figure 6; the first line denotes the corresponding mapped tag. The second field gives the sum of the flags for the map. These flags are necessary to identify the strand of the alignment, i.e. forward/left or reverse/right. For example, a flag of zero means alignment occurred in the forward strand. If the flag is 16, the sequence is aligned to the reverse strand. One can refer to the Bowtie2 [17] manual for more details.

The name of the reference sequence is given in the third field. A "*" means the

```
SRR948187.7      4       *       0       0       *       *
0       0
TCTNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNCNNNNNNNNTGTGTTTTCATAAGCT
#########################################################
YT:Z:UU YF:Z:NS
SRR948187.8      0       chr5    175555694       2       51M
*       0       0
NTCAGNACTNTGGGAGGCTGAGGCAGGAGAATCGCTTGAACCTGGAAGGTG
#0;=?#2=@#2=?@@@?@??@?@????<???<?>=<?5>???;;?????3=      AS:i:-16
XS:i:-18        XN:i:0  XM:i:7 XO:i:0   XG:i:0  NM:i:7
MD:Z:0C0C3C0T0A1T0C40    YT:Z:UU
SRR948187.9      4       *       0       0       *       *
0 0 NGAAGNNNNNNNNNNNNNNNNNNNNNNNNGNNNNNNNTAAGCAGCTGCCGGAA
#07;?#####################################################
YT:Z:UU YF:Z:NS
```

Figure 6 Bowtie2 output for three sequences. Fastq format sequences were aligned to a genome using Bowtie2 and SAM output for the three sequences shown in Figure 5 are presented here. Each line consists of at least 12 fields separated by tabs. Here the sequence file name is "SRR948187" in this illustration.

sequence is not mapped to any one of the reference sequences (i.e. to chromosomes in our case). It can be seen that, in Figure 6, the second sequence is mapped to the fifth chromosome (chr5). However, the third field of the first alignment is a "*". This means that, the third sequence read was not mapped to any reference sequences. The fourth field gives the left-most location of the alignment if it is mapped and "*" otherwise. The quality of the map is provided in the fifth field. A higher mapping quality score implies a map with greater uniqueness. Since we only need the location of the alignment, a reference sequence, and the aligned strand, the rest of the fields are not discussed here. For the rest of this study, these mapped sequence reads are called "tags." A `Perl` script is written to extract the required information from the Bowtie2 output. This script is included in Appendix A.1. Figure 7 shows a few lines from the `Perl` output. The first column represents the strand $(+/-)$, and the second column is the aligned reference sequence, i.e. chromosome information. X and Y

chromosomes are renamed chromosome 23 and 24 respectively for the convenience. The leftmost position of the alignment is given in the third column.

+	5	175555694
+	23	153603448
+	14	59931821
−	10	76498794
−	10	69443718
+	2	231460158
+	12	5389085

Figure 7 Format of extracted information from SAM file. A Perl script is used to extract only the strand, the aligned chromosome and the leftmost position of the alignment for each tag from the SAM output.

1.7.2 PAIRED-END TAG SEQUENCING (PET) DATA

So far, we discussed the data from single-end sequencing (SET) where only one end of the DNA fragment is sequenced. In paired-end tag sequencing data (PET), 20-80 bp from both ends of the DNA fragment are sequenced (Figure 8). For this reason, these paired-ends can reduce the alignment ambiguity, resulting in high-quality mapping. Also, it can improve the mapping rates [47]. Since both ends of the fragments are sequenced and mapped to a reference genome, the actual fragment length can be determined by the mapped locations. This is the main advantage of using PET data compared to SET data. Since these fragment lengths are unknown in SET data, peak-callers usually estimate the mean fragment length as an intermediate step. However, PET data are not widely used in peak-calling.
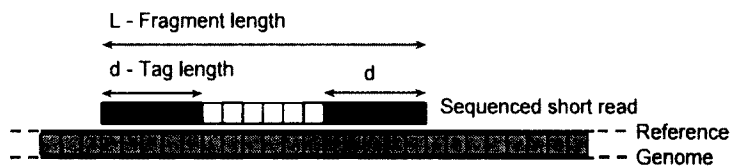
Figure 8 Paired-end tag sequence alignment. 20-80 bp from both ends of the fragments are sequenced and mapped to a reference genome.

Different sets of options should be used with fastq-dump and Bowtie2 programs when preprocessing PET data. In a raw SRA file, information for both mates is available, i.e. sequences of the two ends of the fragment. First, a SRA-formatted PET data file is converted into the fastq format. During the conversion, mates are separated into two different fastq files (Figures 9 and 10).

```
@STAT6.1 HWI-ST183:265:B0AJWABXX:4:1101:1136:2040 length=50
TAGAAAGGACATCAGAATAACAGATTCAAAACACGTCATAAATCAAGTCA
+STAT6.1 HWI-ST183:265:B0AJWABXX:4:1101:1136:2040 length=50
@@@ADDDDFFHFHB<FGGEHGIIGGIGFHH>EGIGDHGGGD*?BDHGDEF
```

Figure 9 Fastq format for PET data - mate 1. Raw PET sequence file is converted using the fastq-dump tool and a fastq formatted file for each mate is generated. This figure shows a sequence of the mate 1.

In Figure 9, the raw sequence is from the STAT6 ChIP-seq dataset. Both mate 1 and mate 2 files should be specified when running the Bowtie2 program. Bowtie2 creates two records for each pair, one for each mate (Figure 11). The first line, starting from STAT6, gives the alignment information for mate 1, and the second line gives it for mate 2. In Figure 11, it can be seen that both mates 1 and 2 are mapped to chromosome 13. Column 4 of the first line gives the alignment location of the first mate, and column 8 of the second line gives that of the second mate. The fragment length is given in the ninth column.

```
@STAT6.1 HWI-ST183:265:BOAJWABXX:4:1101:1136:2040 length=50
TGGTTTGTAAGTTAAGAGCACACGTTTTTTGGCTTTTCTGTGAATAGACT
+STAT6.1 HWI-ST183:265:BOAJWABXX:4:1101:1136:2040 length=50
@@@DDBDFGB>FF@GHFGAB>GBGFFGIJBFDCFGIGGGEDHIEEDGGCG
```

Figure 10 Fastq format for PET data - mate 2. The second fastq formatted file with the mate 2 sequences is shown in this figure. The second line gives the mate 2 sequences in each record.

```
STAT6.1 83       chr13   63348532       42     50M   =    63348472
-110    TGACTTGATTTATGACGTGTTTTGAATCTGTTATTCTGATGTCCTTTCTA
FEDGHDB?*DGGGHDG IGE>HHFGIGGIIGHEGGF<BHFHFFDDDDA@@@
AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0 NM:i:0  MD:Z:50 YS:i:0  YT:Z:CP
STAT6.1 163      chr13   63348472       42     50M   =    63348532
110     TGGTTTGTAAGTTAAGAGCACACGTTTTTTGGCTTTTCTGTGAATAGACT
@@@DDBDFGB>FF@GH FGAB>GBGFFGIJBFDCFGIGGGEDHIEEDGGCG
AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0 NM:i:0  MD:Z:50 YS:i:0  YT:Z:CP
```

Figure 11 Bowtie2 output for PET data. Each sequence tag consists of two alignments. The first line is for the alignment of the first mate, and the second line is for the alignment of the second mate. Column 9 gives the length of the fragment, and a negative fragment length means the second mate is mapped to the left strand.

In a SAM output for a PET dataset, the second column, i.e. the flags, plays a significant role. These bitwise flags indicate setting for a group of parameters. There are eight flags associated with a SAM format, and their values are $1, 2, 4, 8, 16, 32, 64$ and 128. As specified in the Bowtie2 [17] manual, each flag has its definition. The second column of the SAM output gives the sum of these flags, and that value should be decoded first. For example, a flag sum of 83 can be decoded into $83 = 64+16+2+1$. Then, the flag definitions can be used to understand the alignment.

After carefully considering all the possibilities, it was found that 99, 83, 147, and 163 are the only required flag sums for our analysis. Then, the `Perl` script given in Appendix A.2 was used to extract only the required information for our analysis (Figure 12).

–	13	63348532
+	13	63348472
–	15	74063909
+	15	74063833
–	7	5141050
+	7	5140917
–	14	112561353
+	14	112561213
+	1	129202533
–	1	129202661

Figure 12 Format of extracted information from PET Bowtie2 output. Strand information, chromosome number and the starting location of the alignment are extracted using a Perl script.

For the STAT6 data shown in Figure 12, the sequence length is 50 bp, i.e. 50bp from both ends of the fragment were sequenced. The starting position of mate 1 of the first sequence is 633485532 and that of mate 2 is 63348472. In addition, mate 1 is mapped to the forward strand, and mate 2 is mapped to the reverse strand.

The next chapter introduces and describes the normal-exponential model for ChIP-seq data. After explaining the existing Poisson model, a negative binomial, a zero-inflated Poisson model and a zero-inflated negative binomial model are presented, and their performances are compared. Then the best model is selected for the rest of the analysis.

CHAPTER 2

MODELING TAG COUNTS

This chapter explains our approach for identifying the TF binding sites. Since this study is a further development of the NEXT-peak [42] technique, a peak-caller for ChIP-seq based on the normal-exponential two-peak distribution, we first give an overview of the workflow of the NEXT-peak and then explain the basic model for the tag distribution. Then, three new models are proposed, and performances are compared. The optimal model is used for the rest of the implementations and the analysis.

2.1 NEXT-PEAK WORKFLOW

As explained in Kim et al. [42], workflow of the NEXT-peak [42] program is described as follows (Figure 13):

(1) Read the input file with the mapped tag locations.

(2) Define regions using a sliding window. The window length and the minimum tag count for the window can be specified by the user, and default values are 150 and 15 respectively. The number of tags in each sliding window is counted. If the tag count of the neighboring window is more than the minimum count, two windows are combined.

(3) When the locations of the motif sites are available, obtain the maximum likelihood estimates of σ and β (see section 2.2). For a known TF, a publicly available motif pattern is used, e.g., from JASPAR [44]. For an unknown motif, it is recommended to run the NEXT-peak program with default values ($\sigma = 30$ and $\beta = 50$), and identify the strongest motif from a motif search. One can estimate these parameters with selected regions as well.

(4) For each region found in step 2, obtain the maximum likelihood estimates of μ and ν (see section 2.2). Then compute the standard deviation estimates. A goodness-of-fit test is performed for each region.

(5) The region length and the p-value cutoff recommendations are computed.
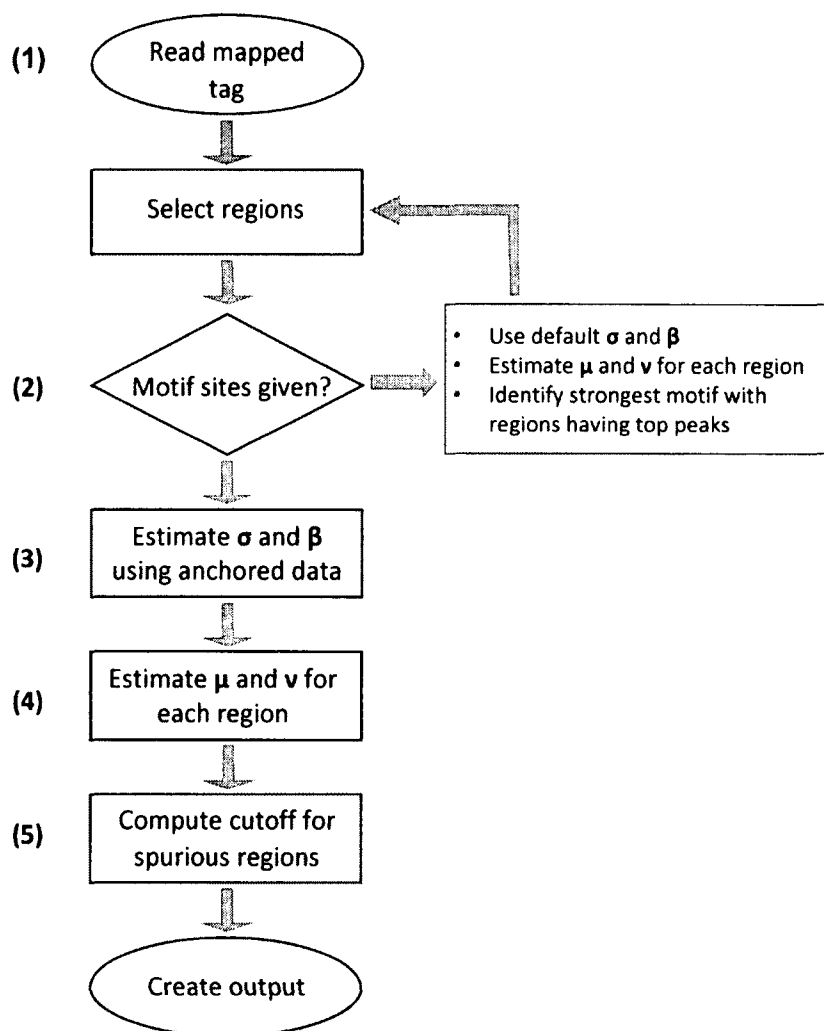
Figure 13 A flowchart for NEXT-peak algorithm. First, the program reads the mapped tags. Then, it selects regions based on the tag count with a user-specified length of the window (default: 150) and a user-specified minimum count (default: 15). If motif site locations are provided, it estimates σ and β using motif site locations. For an unknown motif, run the NEXT-peak program with default values ($\sigma = 30$ and $\beta = 50$), and identify the strongest motif from a motif search. For each region, the program estimates μ and ν. It computes the standard deviation estimates for these estimates. As a post-processing step, the program computes the region length and p-value cut-off recommendations to screen out potential spurious regions when the motif site information is available [42].

This chapter focuses on step 4 above. We propose three new models for count data, and each model has a different set of parameters. Those models are discussed in detail in the following sections. For the rest of the analysis, we assume that mapped tags are read and the regions are identified. A "region" is a segment of DNA that has a significantly higher tag accumulation. Table 2 illustrates a set of regions found using the NEXT-peak algorithm. Therefore, after the second step, we have the starting position, the ending position, the number of observed tags and the corresponding chromosome for each region, in addition to the observed tag count for each location from preprocessed data.

TABLE 2 A part of regions found in step (2) for STAT1 dataset

Region No.	Chromosome	Start	End	Length	Number of Tags
1	1	556608	556905	298	57
2	1	559702	559851	150	36
3	1	703832	703990	159	7
4	1	846298	846447	150	11
5	1	865701	865850	150	6
6	1	865916	866065	150	9
7	1	866087	866236	150	4
8	1	938338	938832	495	37
9	1	1005061	1005210	150	5
10	1	1041388	1041813	426	23
⋮	⋮	⋮	⋮	⋮	⋮

If the ChIP-seq experiment is of good quality, each peak will show a two-peak pattern as shown in Figure 14. However, due to experimental artifacts and various other reasons, ChIP-seq experiments introduce a considerable amount background noise. As a result, there can be regions satisfying the minimum tag number and the minimum length even without the two-peak pattern (Figures 15 and 16).
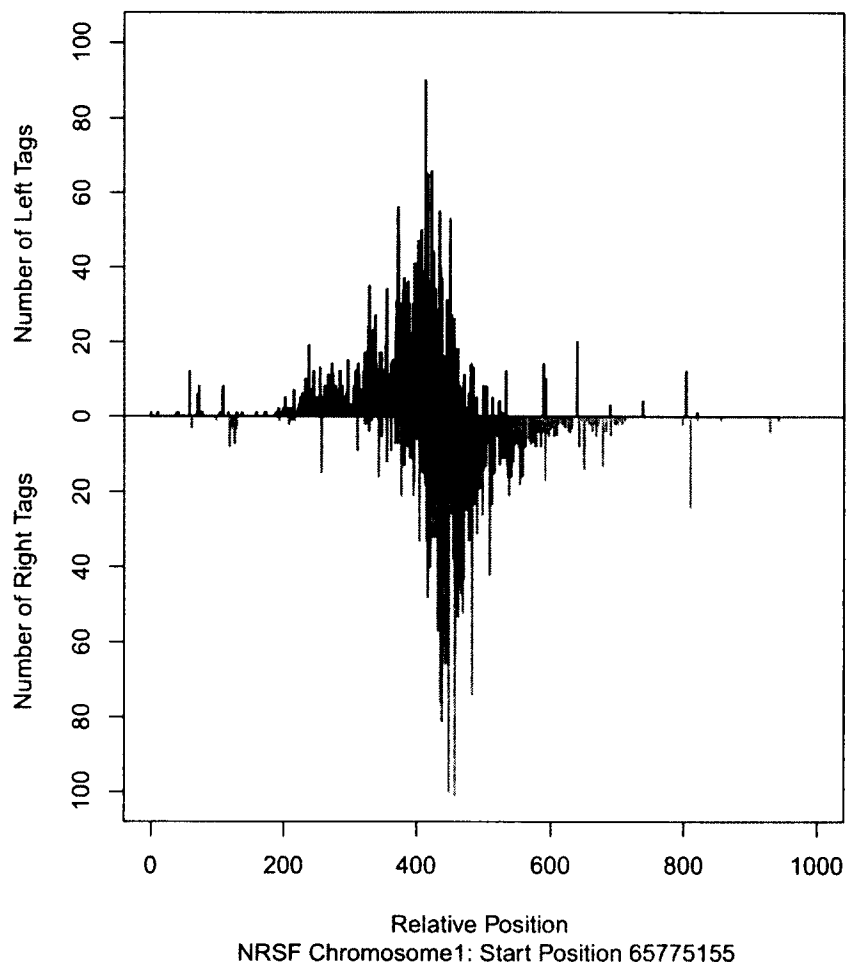
Figure 14 A region with tags primarily due to ChIP signal. When the ChIP-seq experiment is of good quality, each strand shows a peak forming the two-peak structure. This region is found for the NRSF dataset. Tags mapped to the left, and the right strands are represented in the top and the bottom parts of the plot respectively.
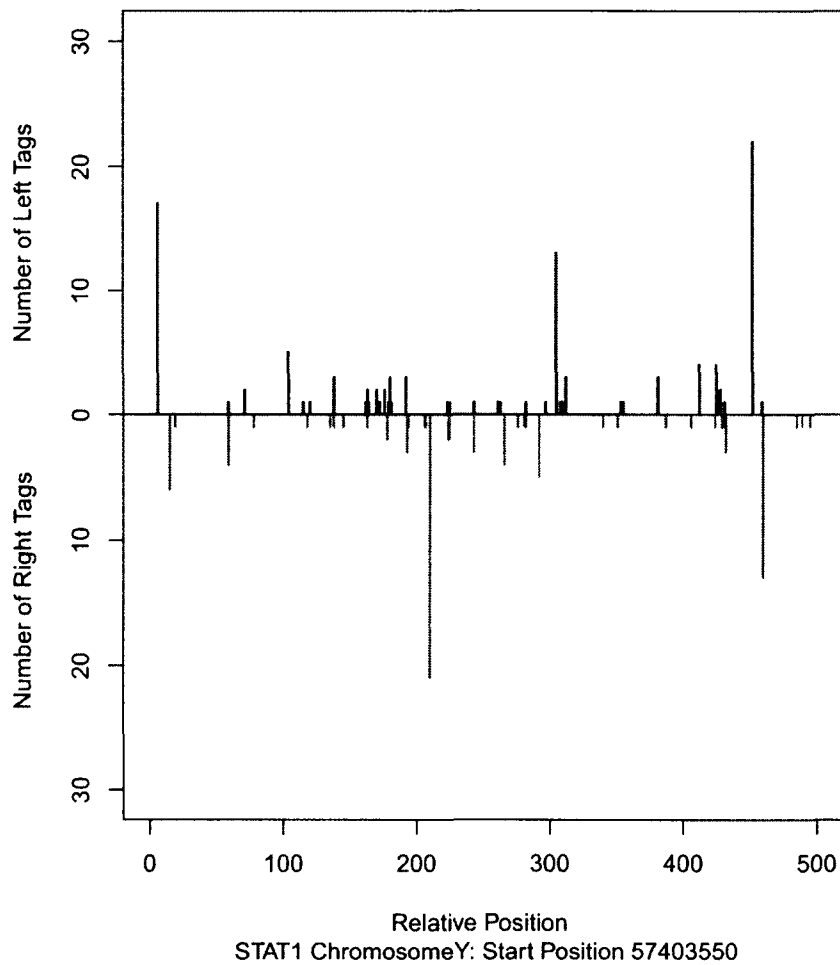
Figure 15 A region without two peak pattern. This region is obtained for the STAT1 dataset. The top and the bottom areas of the plot represent the tags mapped to the left and the right strands respectively. The total observed tag count for the region is greater than the minimum tag count (default = 15). As a result, the region finding algorithm picks this stretch of DNA as a candidate region.
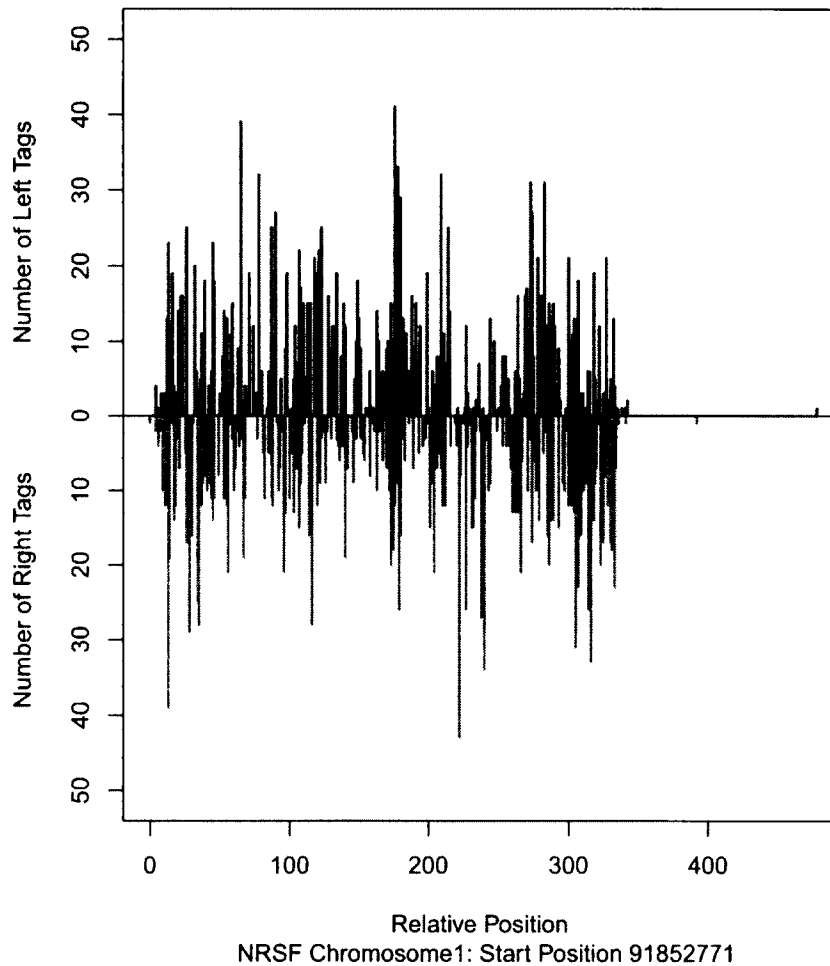
Figure 16 A region with a significantly high background noise. This region is found for the NRSF dataset. The upper and the lower parts of the plot represent the left and the right tags respectively. Although there is a considerably large number of tags (166 left and 310 right tags) observed within this region, the expected two-peak pattern is not visible. Therefore, these tags mainly represent the background noise.

A region length can be 150 bases (default minimum value) to a few thousand bases. Our objective is to identify the exact binding location, i.e. the location of the peak, in each of these regions. In a typical dataset, there can be 10,000-80,000 or even more candidate regions.

In each region, there are at least two parameters to be estimated depending

on the model, and this estimation process is time-consuming. For this reason, the NEXT-peak algorithm was implemented in C++, and our extensions are implemented in the same environment as well. The next section describes the basic model in the NEXT-peak program.

2.2 CONSTRUCTING BASIC MODEL

Assume that a heuristic algorithm identified potential genomic regions, $R_r(r = 1, \ldots, S)$. Consider a specific genomic region R_s, where $s \in \{1, \ldots, S\}$, and let the width of R_s be w_s, with the nucleotide bases having coordinates $1, \ldots, w_s$. Call the forward and backward DNA strands "left" and "right" respectively such that a base at location j on the left strand is the complementary base of the right strand ($A \equiv T$ and $G \equiv C$). Thus, $j = 1, \ldots, w_s$.

Let $x_{ij}^L \in \{1, \ldots, w_i\}$ be the jth, $j = \{1, \ldots, n_i^L\}$, mappable tag location relative to the start location of the ith genomic region. Here, n_i^L is the total number of "left tags" observed on the left strand within the ith genomic region. Subscripts "L" and "R" distinguish quantities relating to the left and right strands, respectively throughout this manuscript. Let $\mathcal{X}^0$ denote the set of locations within R_s where a tag sequence is not unique within the genome, resulting in ambiguous mapping. Therefore, the observed tags for a given genomic region i can be represented by the vector $\mathbf{X} = (x_{i1}^L, \ldots, x_{in^L}^L, x_{i1}^R, \ldots, x_{in^R}^R, \mathcal{X}^0)$ [42,48].

As discussed in section 1.2, in a ChIP-seq experiment, TFs are cross-linked to the DNA, and it is assumed that these unobservable cross-link locations have a random shift from the binding site. Let ξ_{ij} denote this random shift and assume $\xi_{ij} \sim N(\mu_i, \sigma^2)$ for mathematical convenience. Here μ_i is the true binding site location of the ith region and σ^2 is the variance of the shift.

The density of the distribution of ξ_i is

$$\pi(\xi_i | \mu_s, \sigma^2) = \frac{1}{\sigma} \phi(\frac{\xi_i - \mu_s}{\sigma})$$ (1)

where $\phi(\cdot)$ is a standard normal density function.

Unless it's mentioned otherwise, we assume that there's only one binding event on each genomic region. As the next step of the ChIP-seq protocol, the DNA is randomly sheared into millions of fragments. These fragment lengths depend on the experiment and the platform. Usually, average fragment sizes are 100-500 bases. Assume that this shearing process follows a Poisson process over the entire genome.

The fragments are sequenced only on one end, in most of the experiments (Single-End Tags - SET), and only the first dozens of bases are sequenced. In "Paired-End Tags" (PET), dozens of bases on each end of the fragment are sequenced as explained in section 1.7.2. These sequenced sub-fragments are called "reads," and a typical read is about 20-80 bases long. These end reads are usually somewhere near the corresponding cross-link location. However, the shearing process causes randomness in the exact distance between tag ends and cross-link location, and the short reads on the two DNA strands show different systematic biases in their average position relative to the cross-link location [48]. That is, as shown in Figure 17, the short reads mapped to the left strand are expected to demonstrate a shift from the cross-link location to the left. The short reads mapped to the right strand are supposed to demonstrate a shift from the cross-link location to the right (Figure 18).
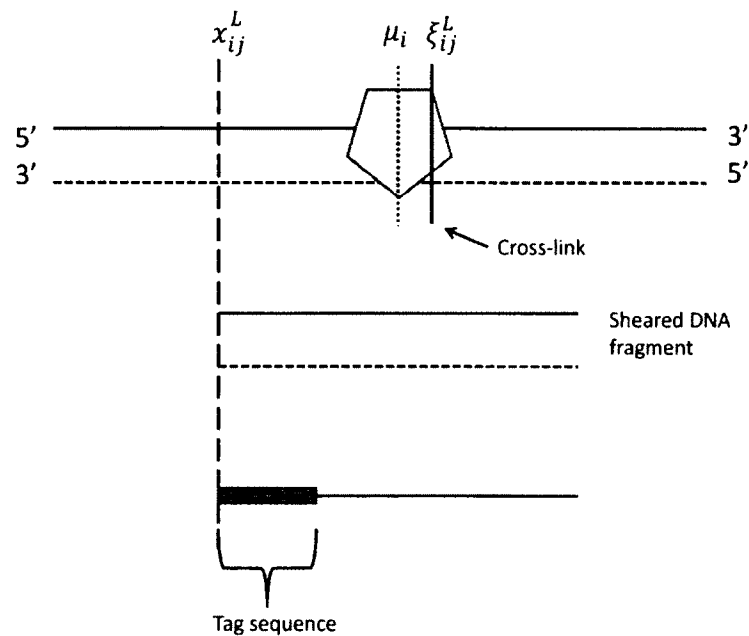
Figure 17 Cross-link for a short read mapped to left strand.

The shifting of the tags ($x_{ij}^L - \xi_{ij}^L$ and $x_{ij}^R - \xi_{ij}^R$) is assumed to have an exponential distribution under the Poisson process assumption. Kim et al. [42] assumed that this
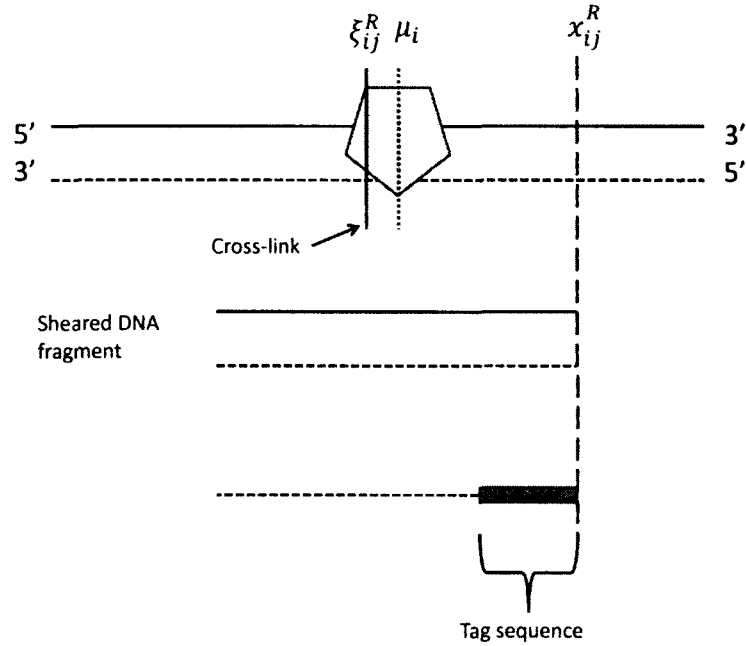
Figure 18 Cross-link for a short read mapped to right strand.

exponential mean, β, is common for all regions. Therefore, for the right tag location x_{ij}^R, with the cross-link location ξ_{ij}^R, the density can be denoted by

$$\pi(x_i|\xi_i, \beta) = \frac{1}{\beta}\exp\left(-\frac{(x_{ij}^R - \xi_{ij}^R)}{\beta}\right) I(x_{ij}^R > \xi_{ij}^R), \qquad (2)$$

where $I(\cdot)$ is an indicator function.

Therefore, the joint density of the (x_{ij}^R, ξ_{ij}^R) is

$$\pi(x_{ij}^R, \xi_{ij}^R) = \pi(x_{ij}^R|\xi_{ij}^R, \beta) \cdot \pi(\xi_{ij}^R|\mu, \beta)$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(\frac{(\xi_{ij}^R - \mu_i)^2}{2\sigma^2}\right)\frac{1}{\beta}\exp\left(-\frac{(x_{ij}^R - \xi_{ij}^R)}{\beta}\right) I(x_{ij}^R > \xi_{ij}{}^R). \qquad (3)$$

Then, the marginal distribution of the joint density is obtained by integrating over ξ_{ij}^R,

$$
\begin{aligned}
f^R(j|\theta) &= \pi(j|\beta, \mu_i, \sigma^2) \\
&= \int \pi(j, \xi_{ij}^R|\beta, \mu_i, \sigma^2)\, d\xi_{ij}^R \\
&= \Phi\left(\frac{j - (\mu_i + \sigma^2/\beta)}{\sigma}\right) \frac{1}{\beta}\exp\left\{-\frac{1}{\beta}(j - (\mu_i + \sigma^2/2\beta))\right\},
\end{aligned}
\tag{4}
$$

where $\Phi(\cdot)$ is the cumulative distribution function (CDF) of the standard normal distribution. Kim et al. [42] named this marginal distribution the "Normal-Exponential" distribution.

Similarly, the left tag location x_{ij}^L can be denoted with a given cross-link location ξ_{ij}^L as,

$$
\pi(\xi_i|\beta) = \frac{1}{\beta}\exp\left(\frac{(x_{ij}^L - \xi_{ij}^L)}{\beta}\right) I(x_{ij}^L < \xi_{ij}^L).
\tag{5}
$$

Hence, after integration over ξ_{ij}^L, the density of x_{ij}^L is

$$
\begin{aligned}
f^L(j|\theta) &= \pi(j|\beta, \mu_i, \sigma^2) \\
&= \int \pi(j \cdot \xi_{ij}^L|\beta, \mu_i, \sigma^2)\, d\xi_{ij}^L \\
&= \left[1 - \Phi\left(\frac{j - (\mu_i - \sigma^2/\beta)}{\sigma}\right)\right] \frac{1}{\beta}\exp\left\{\frac{1}{\beta}(j - (\mu_i - \sigma^2/2\beta))\right\}.
\end{aligned}
\tag{6}
$$

Here, the model parameters μ_i, σ, and β are the same for both left and right tags of the ith region. Therefore, the complete density can be given by,

$$
\begin{aligned}
\pi(x_{ij}^L, \ldots, x_{in_i^L}^L, x_{ij}^R, \ldots, x_{in_i^R}^R|\beta, \mu_i, \sigma^2) &= \prod_{j \in mappable}^{w_i} \pi(x_{ij}^L|\beta, \mu_i, \sigma^2) \\
&\times \prod_{j \in mappable}^{w_i} \pi(x_{ij}^R|\beta, \mu_i, \sigma^2).
\end{aligned}
\tag{7}
$$

The overlaid Normal-Exponential densities for both left and right strands are shown in Figure 19. These asymmetric density curves clearly reflect the duality of the kernel as well as the mirror image feature [48].
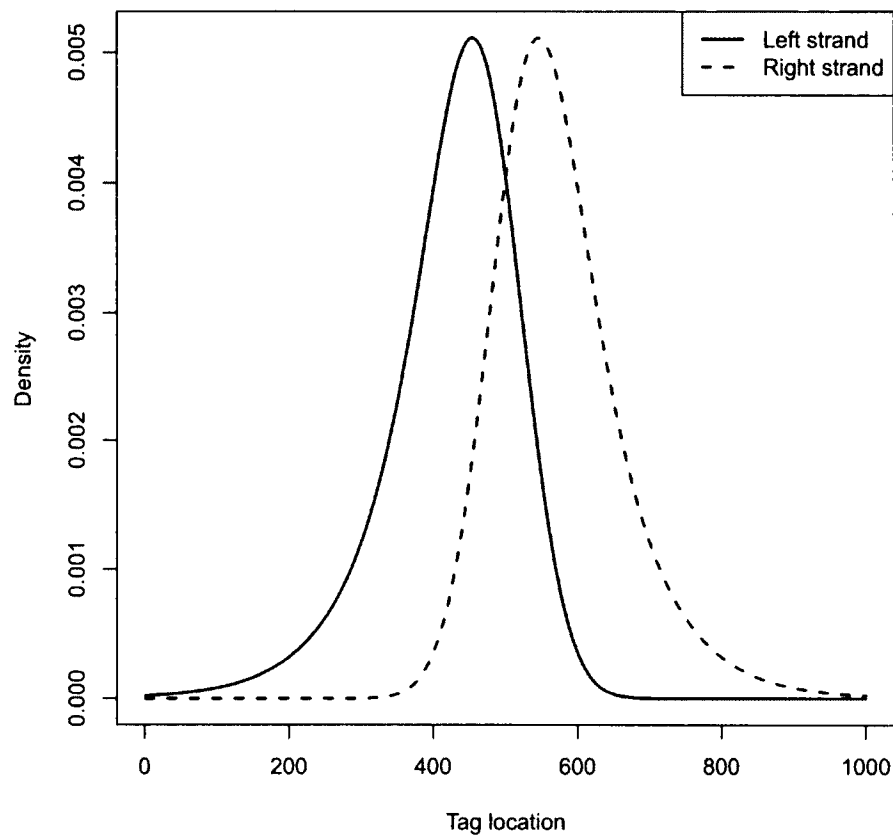
Figure 19 Normal-exponential density. One peak on each strand was observed.

Figure 20 shows the frequency plot of the tags mapped to the left and the right strands around the high-scoring motif sites for STAT1. The overlaid curve is the proposed dual normal-exponential kernel with $\hat{\beta} = 74.1$ and $\hat{\sigma} = 42.3$ [42]. As Park [24] suggested, any peak that does not reflect this mirror image characteristic can be considered as peaks only due to background noise. In fact, extremely high block shaped peaks at repetitive regions or flat signals across the region can be observed.

As Kim et al. [42] proposed, an alternative representation is useful in proposing a model for count data. Let y_{ij}^L and y_{ij}^R be the number of left tags and right tags observed at location j, respectively. Tags cannot be observed at unmappable locations.

Thus, for all practical purposes, $(y_{i1}^L, \ldots, y_{iw_s}^L, y_{i1}^R, \ldots, y_{iw_s}^R, \mathcal{X}^0)$ can be used instead of $(x_{i1}^L, \ldots, x_{in_L}^L, x_{i1}^R, \ldots, x_{in_R}^R, \mathcal{X}^0)$.
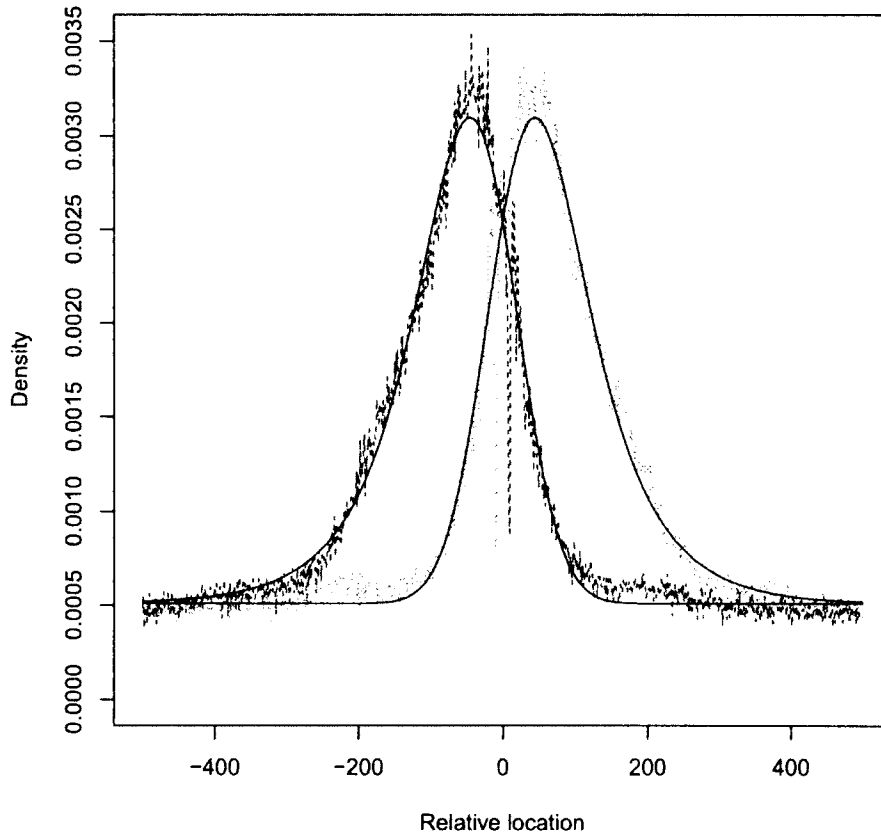
Figure 20 Anchored tag distribution of STAT1.

In high-throughput sequencing experiments, raw data are millions of reads which are aligned to genomic locations. As a result, for each genomic location there can be a zero, one or multiple aligned reads. Thus, a suitable distribution for modeling the observed tag counts in each genomic position should have a support on the non-negative integers. All existing peak-callers use a fixed or a sliding window of genomic locations when modeling the observed tag counts. For example, MACS [26] uses a variable rate Poisson model, where the model mean is determined from control data

by taking the maximum of average tag counts computed on 1kb, 5kb, 10kb, and genome-wide intervals. MOSAiCS [37] divides the genome into small nonoverlapping bins of size 50 bp and uses a negative binomial model for count data.

As mentioned earlier, Kim et al. [42] were the first research group to consider a per base or a location-wise model, and they assumed a Poisson distribution for modeling the observed read counts.

For the rest of the analysis, I use the following notations.

$$f^L(j|\theta) = \left[1 - \Phi\left(\frac{j-(\mu_i-\sigma^2/\beta)}{\sigma}\right)\right] \frac{1}{\beta}\exp\left\{\frac{1}{\beta}(j - (\mu_i - \sigma^2/2\beta))\right\},$$

$$f^R(j|\theta) = \Phi\left(\frac{j-(\mu_i+\sigma^2/\beta)}{\sigma}\right) \frac{1}{\beta}\exp\left\{-\frac{1}{\beta}(j - (\mu_i + \sigma^2/2\beta))\right\},$$

and θ is defined earlier in this section.

In the following sections, the Poisson regression model of the NEXT-peak program is explained. Then a negative binomial regression model, a zero-inflated Poisson regression model, and a zero-inflated negative binomial regression model are proposed to model observed tags. Then, the performances are compared, and the best model is selected as the base model for the rest of the analysis.

2.3 POISSON REGRESSION MODEL FOR TAG COUNTS

The Poisson distribution is widely used in modeling count data. A position and strand dependent mean parameters, λ_j^L and λ_j^R for left and right strands respectively, are considered, and tag counts in each genomic location are assumed to have Poisson distributions with corresponding mean parameter λ_j^L or λ_j^R. Note that tag counts in each region (set of adjacent genomic locations) are a set of independent random variables.

For the region s, i.e. within R_s, let ν_s be the expected number of right tags for TF binding, ρ_s be the uniform background intensity of right tags and λ_j^R be the expected number of right tags at location j. Assume $\lambda_j^L = \nu_s f^L(j|\theta) + \rho_s$ and $\lambda_j^R = \nu_s f^R(j|\theta) + \rho_s$ [42].

Let Y_j^L be the random variable that counts left tags at j, and assume Y_j^L has a Poisson distribution with mean λ_j^L, i.e.,

$$Pr(Y_j^L = y) = exp(-\lambda_j^L)\frac{(\lambda_j^L)^y}{y!}, \tag{8}$$

for $y \in \{0, 1, 2, ...\}$. Similarly, let Y_j^R be the random variable that counts right tags

at j, and assume Y_j^R has a Poisson distribution with mean λ_j^R, i.e.,

$$Pr(Y_j^R = y) = exp(-\lambda_j^R)\frac{(\lambda_j^R)^y}{y!}, \tag{9}$$

for $y \in \{0, 1, 2, ...\}$. Note that for the Poisson distribution, $E[Y_j^L] = \lambda_j^L$, $E[Y_j^R] = \lambda_j^R$, $Var[Y_j^L] = \lambda_j^L$, and $Var[Y_j^R] = \lambda_j^R$.

Let $\theta_s = (\mu_s, \nu_s, \rho_s)$. Under the NEXT-peak model [42], the likelihood of the data $(y_1^L, ..., y_w^L, y_1^R, ..., y_w^R, \mathcal{X}^0)$ in R_s is

$$L(\sigma, \beta, \theta_s) = \Pi_{j \notin \mathcal{X}^0} \left[Pr(Y_j^R = y | \sigma, \beta, \theta_s) \cdot Pr(Y_j^L = y | \sigma, \beta, \theta_s) \right]. \tag{10}$$

Thus, the likelihood for the entire set of regions is

$$L(\sigma, \beta, \theta) = \Pi_{r=1}^S L(\sigma, \beta, \theta_s), \tag{11}$$

where $\theta = (\theta_r : r = 1, .., S)$.

2.4 NEGATIVE BINOMIAL REGRESSION MODEL FOR TAG COUNTS

Although a Poisson model is the simple choice for count data, there are limitations to using a Poisson model in ChIP-seq data. For example, the Poisson model assumes the variance of data to be equal to the mean. However, in ChIP-seq experiments, this is not the case. The negative binomial can be considered as an extension of Poisson distribution to handle overdispersion. That is, the situation where the variance of the counts is larger than expected by a Poisson distribution. Ji et al. [49] illustrated that a negative binomial distribution provides a better fit to count data from ChIP-seq experiments than a Poisson distribution. CisGenome [49], MOSAiCS [37] and QuEST [28] are among the peak-callers which use a negative binomial (NB) model. Furthermore, edgeR [50] uses a NB model to fit RNA-seq in differential expression analysis. RNA-seq is also an NGS technique that uses High Throughput Sequencing (HTS) and is used for differential analysis instead of peak-calling. However, all these techniques are window-based, and our approach is a per-base model.

Hilbe [51] stated that there are about 13 separate types of derivations for the NB distribution. Usually, those who are using the NB distribution have no idea that their parameterization of the NB may differ from the parameterization being utilized by another. The "traditional negative binomial," which is now commonly symbolized

as "NB2," is derived from a Poisson-gamma mixture distribution [51]. One can refer to Hilbe [51] for the derivation of the NB2 model from a Poisson-gamma mixture. In this study, we use the NB2 parameterization of the negative binomial distribution shown in Equation (12).

Let Y_j^L be the random variable that counts right tags at j, and assume Y_j^L has a NB distribution with mean λ_j^L and dispersion α.

$$Pr(Y_j^L = y) = \frac{\Gamma(\alpha^{-1} + y)}{\Gamma(\alpha^{-1})y!} \left(\frac{\alpha \lambda_j^L}{1 + \alpha \lambda_j^L} \right)^y \left(\frac{1}{1 + \alpha \lambda_j^L} \right)^{1/\alpha}, \tag{12}$$

for $y \in \{0, 1, 2, ...\}$.

Similarly, $Pr(Y_j^R = y)$ can be written for $y \in \{0, 1, 2, ...\}$.

Note that $E[Y_j^L] = \lambda_j^L$, $E[Y_j^R] = \lambda_j^R$, $Var[Y_j^L] = \lambda_j^L + \alpha(\lambda_j^L)^2$, and $Var[Y_j^R] = \lambda_j^R + \alpha(\lambda_j^R)^2$.

Let $\theta_s = (\mu_s, \nu_s, \rho_s, \alpha)$. Under the NB model, the likelihood of the data $(y_1^L, ..., y_w^L, y_1^R, ..., y_w^R, \mathcal{X}^0)$ in R_s is

$$L(\sigma, \beta, \theta_s) = \Pi_{j \notin \mathcal{X}^0} \left[Pr(Y_j^R = y | \sigma, \beta, \alpha, \theta_s) \cdot Pr(Y_j^L = y | \sigma, \beta, \theta_s) \right]. \tag{13}$$

Thus, the likelihood for the entire set of regions is

$$L(\sigma, \beta, \theta) = \Pi_{r=1}^{S} L(\sigma, \beta, \alpha, \theta_s), \tag{14}$$

where $\theta = (\theta_r : r = 1, .., S)$.

2.5 ZERO-INFLATED MODELS

In addition to the over-dispersion, NGS data are often zero-inflated as well. In their manuscript, Kuan et al. [37] pointed out that bins with zero mappability always yield zero tag counts under the standard preprocessing protocol, and that gives rise to excess zeros in the observed data. Also, an experiment with insufficient total number of reads (small sequence depth) results in excess zeros [37].

In our proposed method, we do not use the entire genome. We only focus on the areas where tag accumulation is significantly high, i.e. regions, and identify these areas using a unique technique. For NRSF and ZNF143 datasets, regions were found for chromosome 22, and histograms were obtained (Figures 21 and 22).
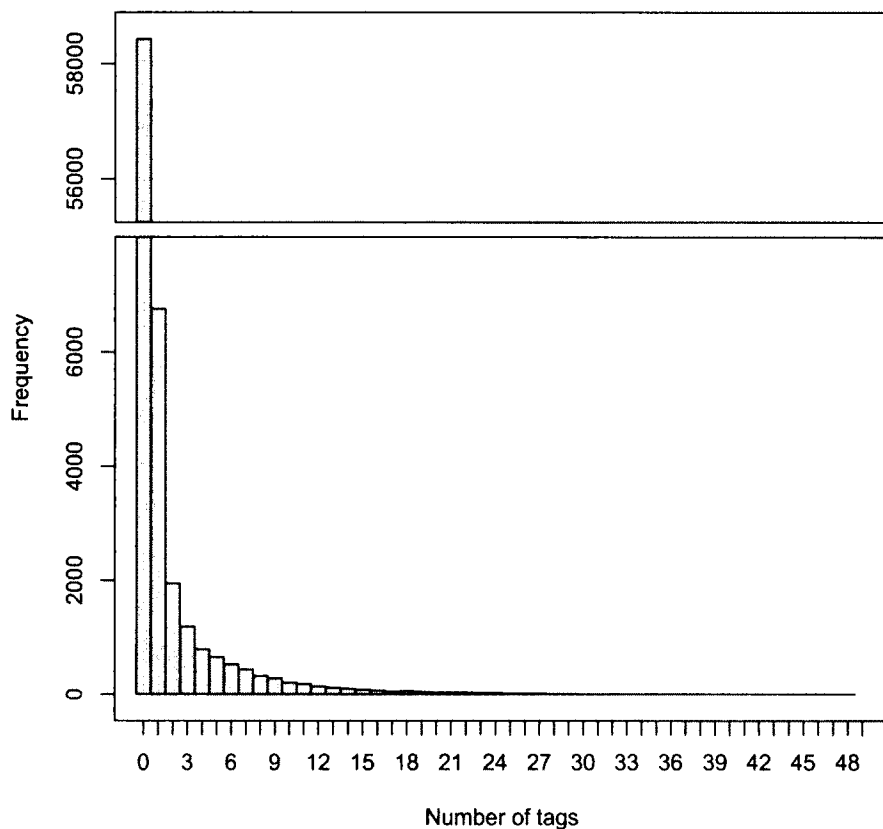
Figure 21 Histogram of tag counts for ZNF143 dataset.
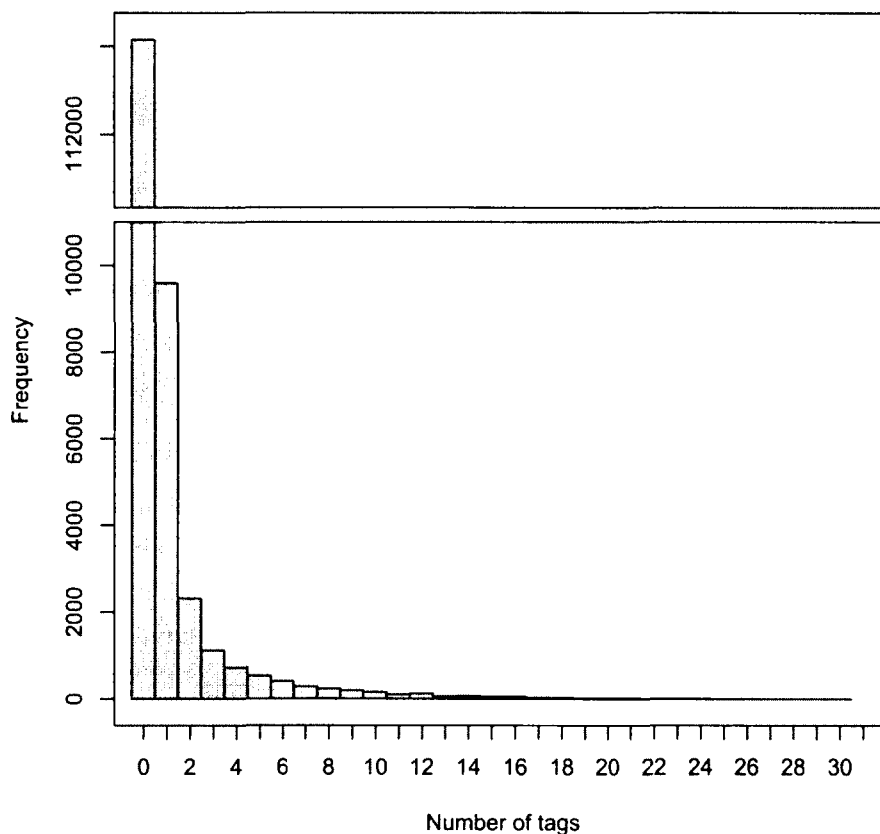
Figure 22 Histogram of tag counts for NRSF dataset.

In the NRSF and ZNF143 datasets, 87.5% and 80.3% of the locations had zero tag counts, respectively. These histograms show the urgency of handling excess zero counts that might not be explained by Poisson and NB models.

Lambert [52] showed a zero-inflated Poisson model is better than a Poisson regression in fitting a dataset with many zeros. Rashid et al. [53] and Dias et al. [54] proposed zero-inflated models to improve model fit in ChIP-seq data. However, they used window-based methods. In zero-inflated models, for each observation, there are two possible data generation processes. It assumes that, with probability π, the only possible observation is 0, and with probability $1 - \pi$, a Poisson or a NB random variable is observed from $g(y_i|x_i)$ [52,55].

In general:

$$y_i = \begin{cases} 0 & \text{with probability } \pi, \\ y_i, \ y_i \sim g(y_i|X_i) & \text{with probability } 1-\pi. \end{cases}$$

The probability of $\{Y_i = y_i|X_i, \pi\}$ is

$$P(Y_i = y_i|X_i, \pi) = \begin{cases} \pi + (1-\pi)g(0|X_i), & y_i = 0, \\ (1-\pi)g(y_i|X_i), & y > 0. \end{cases}$$

In this study, we propose two zero-inflated per-base models, a zero-inflated Poisson model and a zero-inflated negative binomial model for the comparison.

2.5.1 ZERO INFLATED POISSON (ZIP) REGRESSION MODEL FOR TAG COUNTS

In this section, a ZIP model is formulated. For the region s, let ν_s be the expected number of the right tags for TF binding, ρ_s be the uniform background intensity of right tags and λ_j^R be the expected number of the right tags at location j [42].

Let Y_j^L be the random variable that counts right tags at j, and assume Y_j^L has a zero-inflated Poisson distribution with parameters λ_j^L and π.

$$Pr(Y_j^L = y) = \begin{cases} \pi + (1-\pi)e^{-\lambda_j^L}, & \text{if } y = 0 \\ (1-\pi)e^{-\lambda_j^L}\frac{(\lambda_j^L)^y}{y!}, & \text{if } y = 1,2,3.. \end{cases} \tag{15}$$

where $0 \leq \pi \leq 1$ and $\lambda_j \geq 0$.

Similarly, $Pr(Y_j^R = y)$ can be formulated. Note that

$E\left[Y_j^L\right] = (1-\pi)\lambda_j^L$, $E[Y_j^R] = (1-\pi)\lambda_j^R$, $Var[Y_j^L] = (1-\pi)(1+\pi\lambda_j^L)\lambda_j^L$ and $Var[Y_j^R] = (1-\pi)(1+\pi\lambda_j^R)\lambda_j^R$.

Let $\theta_s = (\mu_s, \nu_s, \rho_s)$. Under the ZIP model, the likelihood of the data $(y_1^L, ..., y_w^L, y_1^R, ..., y_w^R, X^0)$ in R_s is

$$L(\sigma, \beta, \theta_s) = \Pi_{j \notin X^0}\left[Pr(Y_j^R = y|\sigma, \beta, \theta_s) \cdot Pr(Y_j^L = y|\sigma, \beta, \theta_s)\right] \tag{16}$$

2.5.2 ZERO INFLATED NEGATIVE BINOMIAL (ZINB) REGRESSION MODEL FOR TAG COUNTS

The formulation of the ZINB model is explained in this section. For the region s, assume ν_s, ρ_s, λ_j^L and λ_j^R are defined as in section 2.5.1. Let Y_j^L be the random variable that counts right tags at j, and assume Y_j^L has a zero-inflated negative binomial distribution with parameters λ_j^L and π.

$$Pr(Y_j^L = y) = \begin{cases} \pi + (1 - \pi)\left(1 + \alpha\lambda_j^L\right)^{-\frac{1}{\alpha}} & \text{if } y = 0 \\ (1 - \pi)\frac{\Gamma(y+1/\alpha)}{\Gamma(1/\alpha)y!}\frac{\left(\alpha\lambda_j^L\right)^y}{\left(1+\alpha\lambda_j^L\right)^{y+1/\alpha}} & \text{if } y = 1,2,3... \end{cases} \tag{17}$$

where α is the negative binomial dispersion parameter, $0 \leq \pi \leq 1$ and $\lambda_j \geq 0$. Similarly, $Pr(Y_j^R = y)$ can be written for y $\epsilon \{0,1,2,...\}$.

Note that

$E[Y_j^L] = (1 - p)\lambda_j^L$, $E[Y_j^R] = (1 - p)\lambda_j^R$, $\text{Var}[Y_j^L] = \lambda_j^L + \left(\frac{\pi}{1-\pi} + \frac{\alpha}{1-\pi}\right)(\lambda_j^L)^2$ and $\text{Var}[Y_j^R] = \lambda_j^R + \left(\frac{\pi}{1-\pi} + \frac{\alpha}{1-\pi}\right)(\lambda_j^R)^2$.

Under the ZINB model, the likelihood of the data $(y_1^L, ..., y_w^L, y_1^R, ..., y_w^R, \mathcal{X}^0)$ in R_s is

$$L(\sigma, \beta, \theta_s) = \Pi_{j \notin \mathcal{X}^0}\left[Pr(Y_j^R = y|\sigma, \beta, \theta_s) \cdot Pr(Y_j^L = y|\sigma, \beta, \theta_s)\right] \tag{18}$$

where $\theta_s = (\mu_s, \nu_s, \rho_s, \alpha, \pi)$.

2.6 PARAMETER ESTIMATION

The parameters of the above four models are estimated by minimizing the negative log-likelihood. The parameters σ and β do not depend on the region R_s. Therefore, they are estimated globally using the motif site locations as discussed in section 1.6. After finding the maximum likelihood estimates (MLEs), $\hat{\sigma}$ and $\hat{\beta}$, the values of the σ and β are fixed, ($\sigma = \hat{\sigma}$ and $\beta = \hat{\beta}$). Then the remaining parameters are estimates as follows. For the Poisson model, the remaining parameters μ_s, ν_s and ρ_s are estimated for each region s by minimizing the corresponding negative log-likelihoods. For the NB model, the dispersion parameter α is estimated globally and fixed before estimating the regional parameters. These global and local estimation processes are iterated until the system of parameters is to stable.

The zero-inflated parameter, π, of the ZIP model is estimated globally, i.e. considering all the regions and with the regional parameters estimated iteratively. For

the ZINB model, both π and α are estimated globally whereas μ_s, ν_s and ρ_s are estimated locally. In a ChIP-seq dataset with 10 million tags, there can be 15,000 - 20,000 regions. Therefore, there are a large number of parameters to be estimated, and the estimation process should be efficient. The bisection method [56], the limited memory BFGS method [57], and Brent's method [58] were used to find the estimates; it was found that Brent's method was more suitable for this study. These methods were implemented in C++ from scratch, and optimization functions were validated using a simulated scheme. Since our final outcome of this study is to release a user friendly peak-calling program, we did not use the GNU Scientific Library (GSL) [59] or any other scientific libraries for the optimization. Therefore, users can download and run the C++ executable file or compile the source files without installing any other libraries. Installing libraries on public domains, for example on HPC clusters, requires administration privileges and often it is an extra burden for users.

2.7 CHOOSING THE BEST MODEL

The previously proposed models were implemented in C++, and three ChIP-seq datasets (ZNF143, NRSF, and STAT1) were used for the comparison. For each dataset, candidate regions were found, and corresponding local and global parameters for each model were estimated. AIC [60] and BIC [61] values were then calculated as follows.

$$
\begin{aligned}
\text{AIC} &= -2 \cdot \ln(\hat{L}) - 2 \cdot k, \\
\text{BIC} &= -2 \cdot \ln(\hat{L}) + k \cdot \ln(n),
\end{aligned}
\tag{19}
$$

where k is the number of free parameters to be estimated, $\hat{L}$ is the maximum value of the likelihood of the model, and n is the number of observations.

TABLE 3 Summary measures for ZNF143 dataset

Model	(-2)Log Likelihood	AIC	BIC	$\hat{\pi}$	$\hat{\alpha}$
Poisson	195 824	196 712	201 102	NA	NA
NB	180 423	181 757	188 352	NA	0.512
ZIP	147 213	148 547	154 967	0.164	NA
ZINB	137 944	139 280	145 709	0.040	0.427

TABLE 4 Summary measures for NRSF dataset

Model	(-2)Log Likelihood	AIC	BIC	$\hat{\pi}$	$\hat{\alpha}$
Poisson	203 326	205 406	216 297	NA	NA
NB	201 916	205 038	221 384	NA	0.156
ZIP	191 313	194 435	210 598	0.128	NA
ZINB	181 990	185 114	201 288	0.010	0.135

TABLE 5 Summary measures for STAT1 dataset

Model	(-2)Log Likelihood	AIC	BIC	$\hat{\pi}$	$\hat{\alpha}$
Poisson	113 380	114 824	122 126	NA	NA
NB	109 794	111 962	122 926	NA	2.226
ZIP	95 135	97 303	108 101	0.577	NA
ZINB	94 751	96 921	107 729	0.488	0.615

From Tables 3, 4 and 5 it can be seen that ZINB has the lowest AIC and BIC values. Therefore, the ZINB model is selected as the basic model in this study. In the next chapter, different covariates are introduced to the ZINB regression model, and their statistical significances are evaluated.

CHAPTER 3

ADDING COVARIATES TO ZERO-INFLATED

NEGATIVE BINOMIAL REGRESSION MODEL

3.1 GLM MODEL FOR COUNT DATA

When the response is a count variable and both over-dispersion and zero inflation are present, the zero-inflated negative binomial model can be considered for such data. As shown in Chapter 2, the ZINB model was the best model for ChIP-seq data among considered. The ZINB regression model is a generalized linear model (GLM) and three main components of a GLM for count data are: a random component, a systematic component, and a link function. The random component is the ZINB distribution which models the count data. In our model, the systematic component is $\nu_s f(j|\theta_s) + \rho_s$. Although, a log link is frequently used for count data, Kim et al. [42] showed that the identity link is a good fit for Poisson regression set up.

Kim et al. [42] assumed

$$E\left[Y_j^L\right] = \lambda_j^L = \nu_s f^L(j|\theta_s) + \rho_s, \tag{20}$$

and

$$E\left[Y_j^R\right] = \lambda_j^R = \nu_s f^R(j|\theta_s) + \rho_s. \tag{21}$$

The main reason for using a log link is that for count data, the mean of the data always should be zero or greater, and a log link guarantees a non-negative mean. In Equations (20) and (21), ν_s is the expected number of tags due to the TF molecule within region s, and ρ_s is the uniform background intensity. Therefore, both ν_s and ρ_s are positive. In addition, normal-exponential densities $f^L(j|\theta_s)$ and $f^R(j|\theta_s)$ are also non-negative for $j \in \{1, \ldots, w_s\}$ from the basic properties of a density function. Thus, the mean number of tags at any given location j is non-negative, and the identity link is appropriate.

The ZINB regression model from Chapter 2 is

$$Pr(Y_j^L = y) = \begin{cases} \pi + (1 - \pi)\left(1 + \alpha\lambda_j^L\right)^{-\frac{1}{\alpha}} & \text{if } y = 0 \\ (1 - \pi)\frac{\Gamma(y+1/\alpha)}{\Gamma(1/\alpha)y!}\frac{\left(\alpha\lambda_j^L\right)^y}{\left(1+\alpha\lambda_j^L\right)^{y+1/\alpha}} & \text{if } y = 1,2,3..., \end{cases} \qquad (22)$$

and

$$Pr(Y_j^R = y) = \begin{cases} \pi + (1 - \pi)\left(1 + \alpha\lambda_j^R\right)^{-\frac{1}{\alpha}} & \text{if } y = 0 \\ (1 - \pi)\frac{\Gamma(y+1/\alpha)}{\Gamma(1/\alpha)y!}\frac{\left(\alpha\lambda_j^R\right)^y}{\left(1+\alpha\lambda_j^R\right)^{y+1/\alpha}} & \text{if } y = 1,2,3.... \end{cases} \qquad (23)$$

Here, Y_j^L and Y_j^R are the random variables that count left and right tags at location j respectively, α is the negative binomial dispersion parameter, $0 \le \pi \le 1$ and $\lambda_j \ge 0$, and $\theta_s = (\mu_s, \nu_s, \rho_s, \alpha, \pi)$. The expected number of left and right tags at location j are as given in Equations (20) and (21), respectively. The covariates are introduced to this ZINB regression model in the following sections.

3.2 INCORPORATING CONTROL DATA TO THE MODEL

As discussed in section 1.4.3, control data might help to identify the true signal in ChIP-seq experiments. When using a control dataset, one of the main problems researchers have to deal with is the different sequence depths of the ChIP (IP) and the control samples. That is, the total number of reads in the IP sample is different from the control sample. Therefore, in order to make the correct use of a control sample, one would have to make the samples comparable before doing any analysis. A common strategy is to linearly scale the sequencing depth ratio. This factor is known as the "normalization factor." It is crucial to estimate the normalizing factor correctly because identification of the weak enrichment sites solely depends on the value of the normalizing factor. Commonly used peak-callers use a normalizing factor to calculate p-values under their hypothesized distribution or to calculate false discovery rates (FDRs) using a sample swapping method [62].

Rozowsky [33] showed that background noise of the ChIP-seq sample and the control sample are approximately linearly related. As summarized by Liang and Keles [62], a standard approach of estimating the normalization factor is as follows. The reference genome is divided into non-overlapping bins of width w. Let n_{1i} and n_{2i} denote the total number of reads in the ith bin in the IP sample and the control sample, respectively. Let $n_i = n_{1i} + n_{2i}$ denote the total number of IP and control reads for bin i. Then the IP/control ratio is estimated using Equation (24).

$$\hat{r} = \frac{\sum_{i \in B} n_{1i}}{\sum_{i \in B} n_{2i}}, \tag{24}$$

where $\mathbf{B}$ is the set of indexes of the bins coming from the background region.

Each normalization factor estimation method utilizes a slightly different approach for estimating $\mathbf{B}$. However, all of those methods first divide the entire genome into non-overlapping bins, while our model is a per-base model. For this reason, we incorporate control data into our model at base level. Observed IP and control tag distributions across a region in the NRSF dataset are shown in Figure 23. The main reason for using the control data is to account for the background noise present in the ChIP-seq data. Background noise can result in a considerable tag accumulation without exhibiting the expected two-peak profile.

Therefore, if a region shows an unexpected pattern from the two-peak profile or if a couple of large IP tag counts are present in a region, control data might be useful to better estimate the peak. It can be seen in Figure 23 that a few outlying tag counts in the IP sample are explained by the control tag count in respective locations. This plot is obtained before normalizing the control tag counts. For the NRSF dataset,

$$\text{Normalization Factor} = \frac{\text{Total IP tag counts}}{\text{Total control tag counts}} = 0.682.$$

This ratio suggests that the observed control tag counts in each location should be multiplied at least by this global IP/control ratio to get the correct picture. For example, an observed control count of 15 is scaled down to $15 \times 0.682 = 10.23$. This process of multiplying the individual or binned control counts by a factor is known as normalization of control data. This method is a naive normalizing technique, and we propose a better method to incorporate control data.
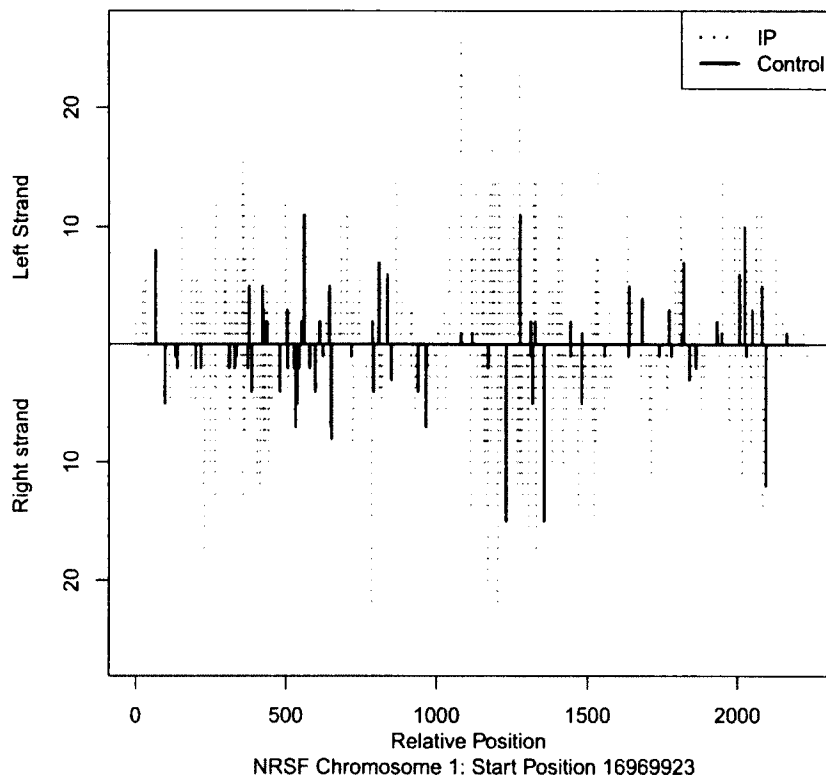
Figure 23 IP and control tags across a region. IP and control tags for both left and right strands are plotted. The top and the bottom sections of the plot are for the observed tags in the left and the right strands respectively.

Let W_j^L and W_j^R be the left and right tags at location j in the control sample. Furthermore, assume that ρ_s is the uniform background intensity in the IP sample as before. Therefore, the signal and the background in the IP sample are already accounted for in our model. In order to take the control data into account, we introduce a new parameter κ, and we call it a "scaling factor" in our model. Although this parameter normalizes the control data, this is different from the "normalization factors" available in the literature. The main difference is that this parameter is estimated at a base level, i.e. no binning is done.

Assume

$$\lambda_j^L = \nu_s f^L(j|\theta_s) + \rho_s + \kappa_s W_j^L, \tag{25}$$

and

$$\lambda_j^R = \nu_s f^R(j|\theta_s) + \rho_s + \kappa_s W_j^R \tag{26}$$

where W_j^L and W_j^R are the observed number of left and right control tags at location j, respectively.

Note that newly added terms in equations (25) and (26) (κ_s, W_j^L, and W_j^R) are all non-negative. Therefore, the non-negative mean requirement is still preserved under this model. It was found that estimating κ using all the regions (i.e., globally) is efficient, and the loss of the estimated value of the likelihood function is negligible when compared to the models with regional scaling factors. Therefore, the scaling factor is globally estimated in this study (i.e. $\kappa_s = \kappa$). In addition to the above identity link model, we propose another model with a logit link.

Assume

$$\lambda_j^L = \nu_s f^L(j|\theta_s) + \rho_s + \kappa W_j^L, \tag{27}$$

$$\lambda_j^R = \nu_s f^R(j|\theta_s) + \rho_s + \kappa W_j^R, \tag{28}$$

and

$$logit(\pi_j) = log\left(\frac{\pi_j}{1 - \pi_j}\right) = \tau_s + \delta_s(W_j^L + W_j^R). \tag{29}$$

The model parameters τ and δ will be estimated globally (i.e. τ_s, δ_s) or regionally (i.e. τ, δ). Furthermore, we introduce a new region finding algorithm to the NEXT-peak [42] program. When the control data are available, the NEXT-peak program finds the regions as follows.

First, the genome is binned into non-overlapping bins of 50 bases. Then, in the IP sample, the number of left tags and right tags are counted in each bin. Then, the total number of tags (left strand + right strand) for each bin is recorded. This technique is repeated to find the total number of control tags in each bin. Then, the total number of tags (left strand + right strand) for each bin is recorded. Fold change (FC) of a given bin is defined as the ratio of IP and control tags in that particular bin. For each bin, FC is calculated, and for convenience, these are converted into a log scale. That is, the log fold change (LFC) = log(IP/control). Then, LFC's are standardized, and these standardized LFC values are used to combine the neighboring bins.

The rest of the process has two main steps: finding an initial bin and combining its neighboring bins (Figure 24). For each chromosome, we start from the first bin. If the ith bin passes the cutoff LFC (default value is 3), we check for the LFC of the $(i+1)^{st}$ bin. If the LFC of the $(i+1)^{st}$ is greater than a cutoff (default value of the second cutoff is 1.5), we combine i^{th} and $(i+1)^{st}$ bins (we call it the joint bin). This process is continued until the adjacent bin fails to pass the second cutoff.
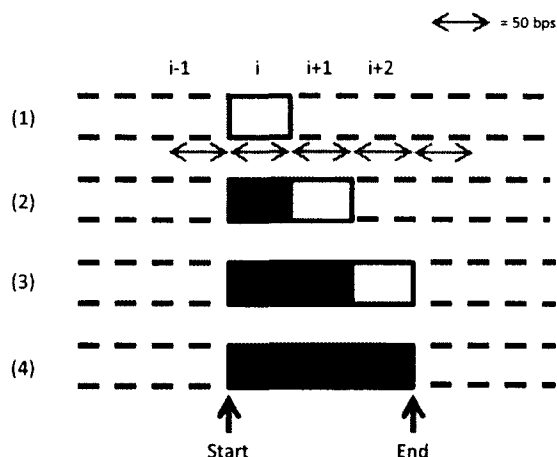
Figure 24 An example of combining bins to form a candidate region. **(1)** Check the LFC of the ith bin. If the LFC is larger than the cutoff value, continue to the $(i+1)^{st}$ bin. **(2)** If the LFC is greater than the second cutoff value, proceed to the $(i+2)^{nd}$ bin. **(3)** If the LFC of the $(i+2)^{nd}$ is larger than the second cutoff value, proceed to the $(i+3)^{rd}$ bin. **(4)** Assume that LFC of the $(i+3)^{rd}$ bin is not greater than the required second cutoff. Therefore, join the $i - (i+3)$ bins to form a region and output the start and the end locations.

When the check for the combining process is done, the starting and the ending coordinates of the joint bin are reported as a candidate region. Then the next initial bin is sought, and the process continues until all the bins are exhausted. This is the basic idea of the region finding algorithm when control data are available.

For STAT1 dataset, peaks were found using both existing and new algorithms.

Then those peaks were ranked based on the expected number of tags due to binding (i.e. ν_s), and 50 peak intervals were considered in increasing fashion, i.e. the top 50 peaks, top 100 peaks, top 150 peaks, and so on. Then the number of true peaks were computed for each interval and percentages of true peaks were plotted (Figure 25). A peak was considered a true peak if the location of the predicted binding site, μ_s, is within 100 bps from a motif site. The motif sites for STAT1 were obtained as discussed in section 1.6. The existing algorithm does not take the control data into account whereas the new algorithm considers the control tag distribution when defining the regions.
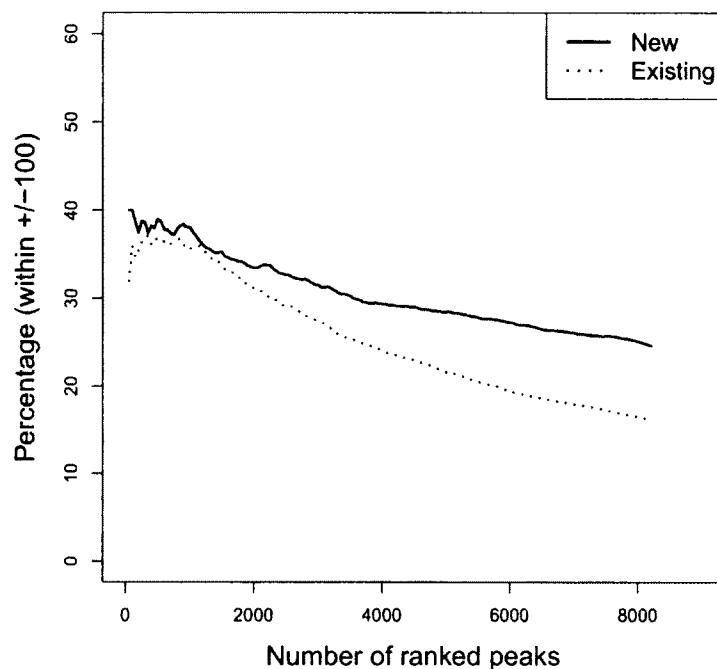
Figure 25 STAT1: Existing vs. new region finding algorithms. Peaks were found from existing and new algorithms and were ranked based on the corresponding ν_s values. Next, they were examined in increasing 50 peak intervals such as top 50 peaks, top 100 peaks, etc. A predicted peak was considered a true positive if it was within 100 bps from a motif site. Then for each interval, the percentage of peaks containing a motif site was computed.

It can be seen in Figure 25 that regions found using the new algorithm captured more true binding sites than the existing one. Performances of these algorithms were also compared for the GABP and STAT1 datasets, and better performance was observed for the algorithm using the control tag distribution to define regions. Therefore, for the rest of the analysis, control data were used to define regions when available.

3.2.1 COMPARING IDENTITY AND LOGIT LINK MODELS

The following models are used for the comparison, and the model with the lowest AIC value is selected.

Model A: Identity link with no control data

$$\lambda_j^L = \nu_s f^L(j|\theta_s) + \rho_s,$$

and

$$\lambda_j^R = \nu_s f^R(j|\theta_s) + \rho_s.$$

Model B: Identity link with control data

$$\lambda_j^L = \nu_s f^L(j|\theta_s) + \rho_s + \kappa W_j^L,$$

and

$$\lambda_j^R = \nu_s f^R(j|\theta_s) + \rho_s + \kappa W_j^R.$$

Model C: Logit link with no control data in the identity link

$$\lambda_j^L = \nu_s f^L(j|\theta_s) + \rho_s,$$

$$\lambda_j^R = \nu_s f^R(j|\theta_s) + \rho_s,$$

and

$$logit(\pi_j) = log\left(\frac{\pi_j}{1 - \pi_j}\right) = \tau_s + \delta_s(W_j^L + W_j^R).$$

Model D: Logit link with control data in the identity link

$$\lambda_j^L = \nu_s f^L(j|\theta_s) + \rho_s + \kappa W_j^L,$$

$$\lambda_j^R = \nu_s f^R(j|\theta_s) + \rho_s + \kappa W_j^R,$$

and

$$logit(\pi_j) = log\left(\frac{\pi_j}{1-\pi_j}\right) = \tau_s + \delta_s(W_j^L + W_j^R).$$

Model E: Logit link with global τ, δ and no control data in the identity link

$$\lambda_j^L = \nu_s f^L(j|\theta_s) + \rho_s,$$

$$\lambda_j^R = \nu_s f^R(j|\theta_s) + \rho_s,$$

and

$$logit(\pi_j) = log\left(\frac{\pi_j}{1-\pi_j}\right) = \tau + \delta(W_j^L + W_j^R).$$

Model F: Logit link with global τ, δ and with control data in the identity link

$$\lambda_j^L = \nu_s f^L(j|\theta_s) + \rho_s + \kappa W_j^L,$$

$$\lambda_j^R = \nu_s f^R(j|\theta_s) + \rho_s + \kappa W_j^R,$$

and

$$logit(\pi_j) = log\left(\frac{\pi_j}{1-\pi_j}\right) = \tau + \delta(W_j^L + W_j^R).$$

The six models above were fitted for NRSF, STAT1 and GABP datasets. Chromosome 1 is the largest chromosome, and it is about 9% of the entire human genome. Therefore, we used tags mapped only to the first chromosome during the model selection stage for a computational consideration.

1. NRSF dataset

For NRSF data, 1471 regions were found in chromosome 1, and a model summary is shown in Table 6.

TABLE 6 Model comparison for NRSF dataset

Model	LogLike	AIC	BIC
ZINB Model F	-313,262	635,354	685,287
ZINB Model E	-313,218	635,266	685,199
ZINB Model A	-308,712	626,254	673,126
ZINB Model B	-308,675	626,181	673,054
ZINB Model D	-272,927	560,565	599,506
ZINB Model C	-272,912	560,536	599,477

It can be seen from Table 6 that ZINB model C, the logit model with no control data in the identity link, achieved the lowest AIC and BIC values. Therefore, ZINB model C was selected as the best model for the NRSF dataset.

2. STAT1 dataset

For the STAT1 dataset, 757 regions were found in chromosome 1. A summary is shown in Table 7.

TABLE 7 Model comparison for STAT1 dataset

Model	LogLike	AIC	BIC
ZINB Model A	-132,835	270,216	294,026
ZINB Model B	-132,650	269,846	293,656
ZINB Model E	-132,443	269,433	294,818
ZINB Model F	-132,096	268,738	294,124
ZINB Model C	-128,862	265,296	282,823
ZINB Model D	-126,622	260,816	278,342

ZINB model D, the logit model with the control data in the identity link, achieved the lowest AIC and BIC values. Therefore, ZINB model D was selected for further analysis.

3. GABP Dataset

For GABP data, 1090 regions were found in chromosome 1, and from Table 8 it can be seen that ZINB model C performed better than the other five models.

**TABLE 8 Model comparison for GABP dataset**

Model	LogLike	AIC	BIC
ZINB Model F	-482,219	970,981	1,008,860
ZINB Model B	-451,697	909,938	945,553
ZINB Model A	-447,671	901,887	937,501
ZINB Model D	-444,722	900,346	927,568
ZINB Model E	-444,610	895,764	933,646
ZINB Model C	-442,402	895,705	922,927

Our program fits these six models for a given dataset using only the largest chromosome of the genome. For example, observed tags from chromosome 1 are selected for the model selection step in both human and mouse genomes. Then, the model with the lowest AIC is selected as the best model. Next, GC and mappability covariates are introduced to the model. Once the best model is selected, peaks are estimated for the entire dataset using the selected model. Models with the GC count and the mappability information are discussed in the next two sections.

3.3 INTRODUCING GC COUNT TO ZINB MODEL

As described in section 1.4.4, the GC content bias can cause problems when calling peaks. Literature in the past used bin-based methods and estimated the GC curve on a global scale, i.e. using the entire genome. Our approach is a per-base method, and we focus only on the areas with higher tag accumulations. First, we identify the regions as described in 3.2.1. The NEXT-peak program estimates the fragment

size, L. The average distance between a fragment end and a cross-link is β under the NEXT-peak model. As a result, the average distance between the fragment ends is $2\beta + d - 1$. Here, d is the tag length for the dataset.

L - Fragment length

d - Tag length

Aligned
short read

- - Reference
- - Genome
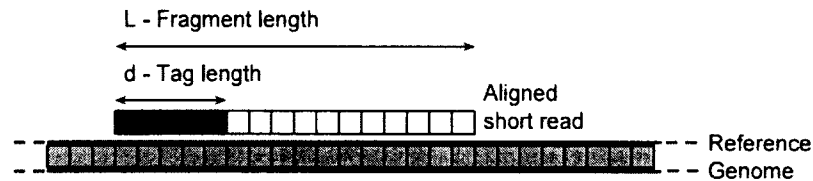
Figure 26 Fragment length vs. tag length. The fragment length is the length of the fragmented DNA sequence produced during the sonication process. In the NGS techniques, only several dozens of bases of the fragments are sequenced. The length of the sequenced portion of the fragment is called the tag length.

We consider this fragment size, L as the width of the sliding window. Starting from the first location of the region, the number of C and G bases within the sliding window is counted and stored in the first location of the window. This window is slid until the entire region is exhausted. Similarly, corresponding GC counts are stored for all the regions. Then, the GC counts are converted to a ratio and rounded to the nearest tenths, i.e. we convert GC counts to $0.0, 0.1, \ldots, 1.0$. For example, if $L = 80$ and the GC count is 35, the GC ratio is $35/80 = 0.4375$, and when it is rounded to the nearest tenth, the GC ratio becomes 0.4. GC ratios were found for the first chromosome of the NRSF dataset. For NRSF data, $\beta = 46.69$ and d = 51. Therefore, $2\times\beta + d - 1 = 143.38$. Therefore, L was set to 144. The histogram of GC ratios is shown in Figure 27.
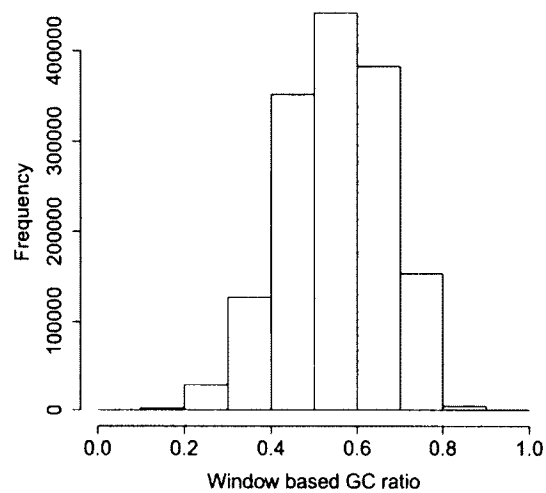
Figure 27 Histogram of GC ratios. For the regions found in the first chromosome of the the NRSF dataset, a sliding window of 144 bases was used to find the number of G and C bases. For each sliding window, the ratio of GC bases were recorded. Then the histogram of GC ratios was obtained.

It can be seen from the histograms that the effects on the observed tags due to the GC percentages are unimodal. The shapes of the histograms of GC ratios for STAT1 and GABP datasets were also similar. Next, the effect of the GC ratios on the observed tag counts was considered, and a smoothed density plot was obtained. Figure 28 shows the marginal effects of the GC ratios on the observed counts.
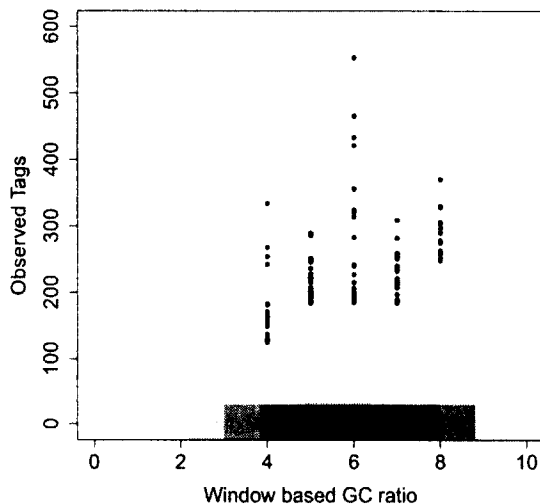
Figure 28 Smoothed scatter plot of GC ratios. The smoothed scatter plot for the regions found in the first chromosome of the the NRSF dataset. Number of observed tags are plotted against the corresponding GC ratios. Darker areas represent a higher number of cases.

It can be seen that a lower number of tags were observed in locations with relatively high or low GC ratios, so the marginal GC effect is unimodal. Therefore, to estimate the GC effect, a vector of GC parameters is introduced to the systematic component of the GLM as in Equations 30 and 31:

$$E\left[Y_j^L\right] = \lambda_j^L = \left[\nu_s f^L(j|\theta_s) + \rho_s\right] \exp\left[\gamma_{GC_j^L}\right], \tag{30}$$

and

$$E\left[Y_j^R\right] = \lambda_j^R = \left[\nu_s f^R(j|\theta_s) + \rho_s\right] \exp\left[\gamma_{GC_j^R}\right], \tag{31}$$

where γ is a vector such that, $\gamma = \{\gamma_{0.0}, \gamma_{0.1}, \ldots, \gamma_{1.0}\}$ and GC_j^L and GC_j^R refer to the GC ratio at location j on the left and right strands respectively. Note that there are 11 parameters to be estimated. γ parameter was estimated along with the other parameters in the model by minimizing the negative log likelihood and their values were plotted against the corresponding parameters. Figures 29 and 30 show the shapes of the gamma estimates for the NRSF and GABP datasets respectively.
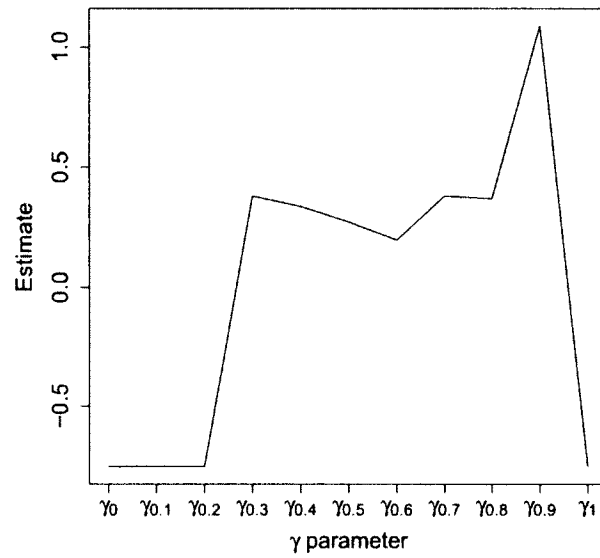
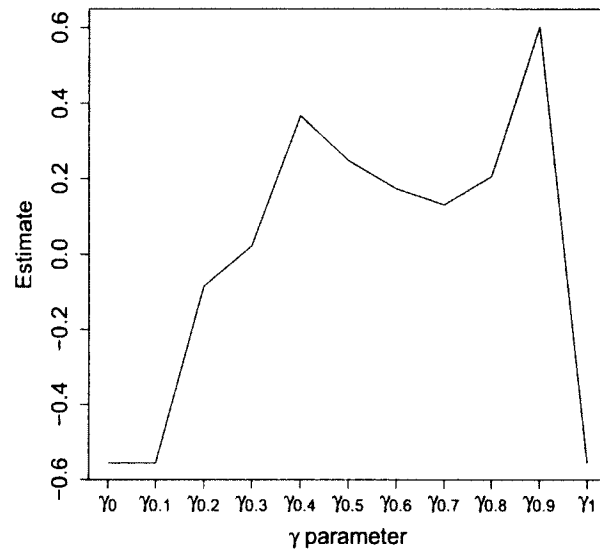**Figure 29 Plot of gamma parameter estimates for NRSF dataset.**

Figure 30 Plot of gamma parameter estimates for GABP dataset.

The GC covariate was introduced to the previously selected models for each dataset, and parameters were estimated minimizing the log likelihood values. A summary for each dataset is shown in Table 9.

TABLE 9 Comparison of models with and without GC covariates

	Model	LogLike	AIC	BIC
NRSF	ZINB Model C	-272,912	560,536	599,477
	ZINB Model C + GC	-277,263	569,260	608,259
STAT1	Model D	-126,622	260,816	278,342
	ZINB Model D + GC	-151,846	311,287	328,865
GABP	Model C	-442,402	895,705	922,927
	ZINB Model C + GC	-506,433	1,023,790	1,051,070

It can be seen that when the GC covariate was introduced, both AIC and BIC values were increased. This suggests that the GC covariate should not be included in the models. Therefore, the selected models in the previous section are the best models for each dataset, i.e. ZINB model C for the NRSF and GABP datasets and ZINB model D for STAT1 dataset.

3.4 INCORPORATING MAPPABILITY INFORMATION

Mappability information is found as described in section 1.4.5 for each genomic location. When calculating the profile likelihood value in each parameter estimation, mappability information for each chromosome is read and stored in a boolean vector. Then, the boolean value is checked for each location before calculating the profile likelihood. In the second line of the following code, $mask\_p[chr][m]$ holds a boolean value for mappability information, and the negative log-likelihood value was calculated only if that location is mappable.

```
for ( size_t m = regLeftEnds [ i ]; m < regRightEnds [ i ]; m++) {
    if ( mask_p [ chr ] [m] )
        negloglik −= ( cntMappedTags_p [0] [ chr ] [m] *
            log ( lambda1 [m−regionLeftEnds [ i ] ] )
            + cntMappedTags_p [1] [ chr ] [m] *
            log ( lambda2 [m−regionLeftEnds [ i ] ] ) );
}
```
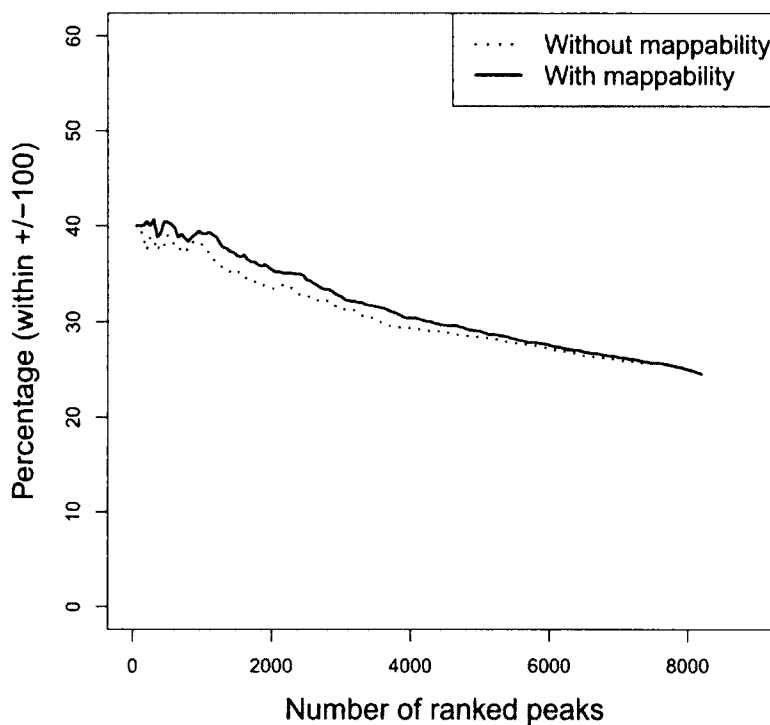

Figure 31 STAT1 dataset with and without mappability information. Number of true binding sites were determined based on the motif site locations and cumulative plots were obtained.

From Figure 31 it can be seen that, model with the mappability information performed better than the model without the mappability information.

3.5 PAIRED END DATA

So far, we analyzed three single-end tag (SET) datasets and found the best model. In this section, a paired-end tag (PET) dataset, STAT6, is considered. ChIP-seq data for STAT6 was preprocessed as explained in section 1.7.2 and Poisson, NB, ZIP, and the previously discussed six ZINB models were fitted. Table 10 gives a summary of the models.

TABLE 10 Summary measures for STAT6 dataset

Model	LogLike	AIC	BIC
Poisson	-454,144.0	912,885.0	937,723.0
ZIP	-197,622.0	402,143.0	437,015.0
ZINB Model F	-102,802.0	212,502.0	249,765.0
ZINB Model E	-102,797.0	212,493.0	249,756.0
ZINB Model B	-102,290.0	211,478.0	246,351.0
ZINB Model A	-102,289.0	211,476.0	246,349.0
NB	-102,037.0	210,970.0	245,833.0
ZINB Model D	-96,663.1	204,818.0	233,816.0
ZINB Model C	-96,660.6	204,813.0	233,811.0

Table 10 is ordered from largest to smallest AIC values. It can be seen that ZINB model C achieved the lowest AIC and BIC values. Figure 32 is a percentage plot of the true peaks found for the fitted models. From the percentage plot it is also evident that ZINB model C found the highest number of true positives based on the motif site locations.
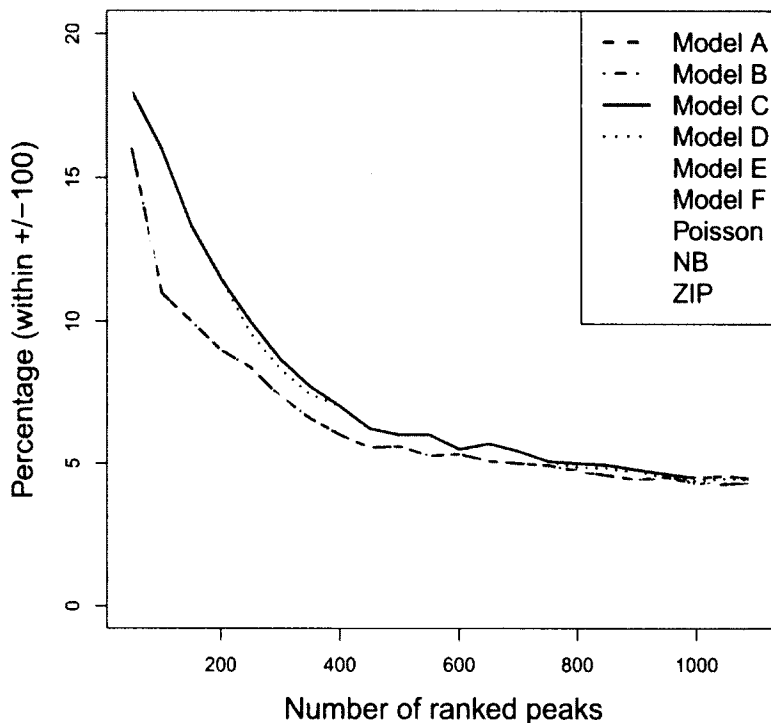
Figure 32 Model comparison for STAT6 dataset. A percentage plot was obtained for all the models. Poisson, NB and ZIP models performed worse than all the ZINB models.

The GC covariate was introduced to the model as in the SET data model, and it was found that the GC covariate is not significant in the model. Therefore, ZINB model C is selected as the best model for the STAT6 dataset. Our program finds the best ZINB regression model using only the regions in the largest chromosome of the corresponding genome. Next, the GC covariate is added to see if there is an improvement in the AIC value. The model with the lowest AIC value is assumed to be the best model for that specific dataset and μ_s, ν_s and other respective parameters of that model are estimated for the entire set of regions.

Likelihood, AIC and BIC values are indicators of the overall fit of the model. In the next section, model fit is compared for selected regions, and it will help us understand how some models outperform others.
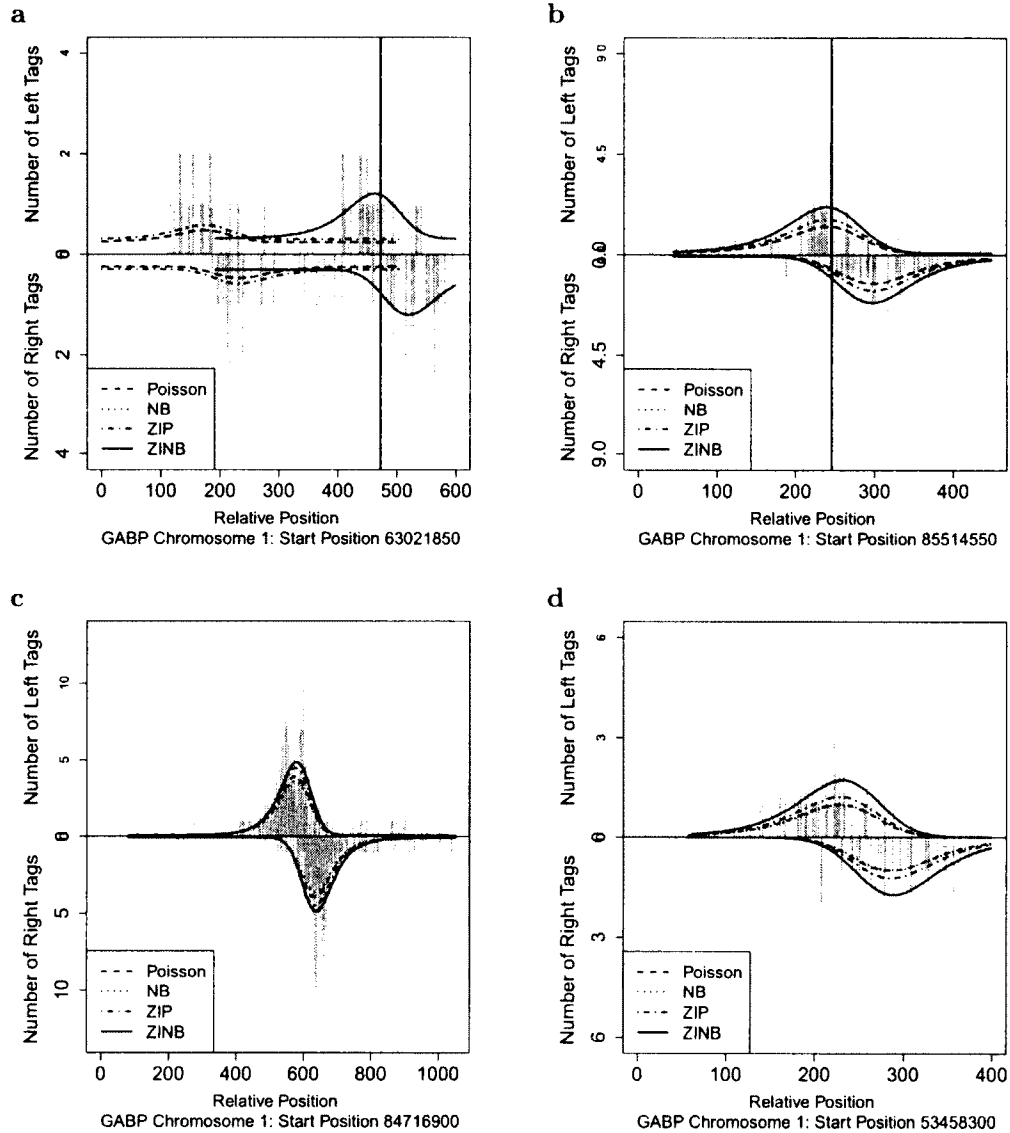
3.6 PERFORMANCE COMPARISON

In section 2.7, based on AIC and BIC values, it was shown that the ZINB model performed better than the Poisson, NB and ZIP models. To further demonstrate the advantages of using the ZINB model, predicted mean values were overlaid for several regions selected in the NRSF, STAT1 and GABP datasets. The expected number of tags due to binding in a given region, ν, measures the strength of the TF binding; therefore, it is used to rank the peaks. Thus, the reliable ν estimates are essential. Heights of the peak in the overlaid plots are proportional to the value of the ν estimates plus the ρ estimates.

Figure 33a is a region with a relatively higher number of locations with zero counts, and it has two high tag accumulations. The location of the motif site is denoted by a thick vertical line. As explained in chapter 1, the estimated binding site lies between the left and the right peaks. Figure 34a shows a region found for the NRSF dataset. It can be seen that the NB model showed left shift in its estimated peak location. The ZINB model estimated the closest binding site to the motif site location. Also, its ν estimate is the largest. In Figures 34b-34d it is evident that the ZINB model fitted observed data better than the other three models.

Figures 35a-35d show four regions found for STAT1 dataset. The ZINB model fitted observed data better than the Poisson, NB and ZIP models.

Figure 33 Performance comparison using GABP regions. For the GABP dataset, observed tag counts were modeled using Poisson, NB, ZIP and ZINB models. Panel (a) shows two high tag accumulation areas in each strand. The thick vertical line denotes the observed motif site location within the region. Poisson, NB and ZIP models estimated the wrong peak.
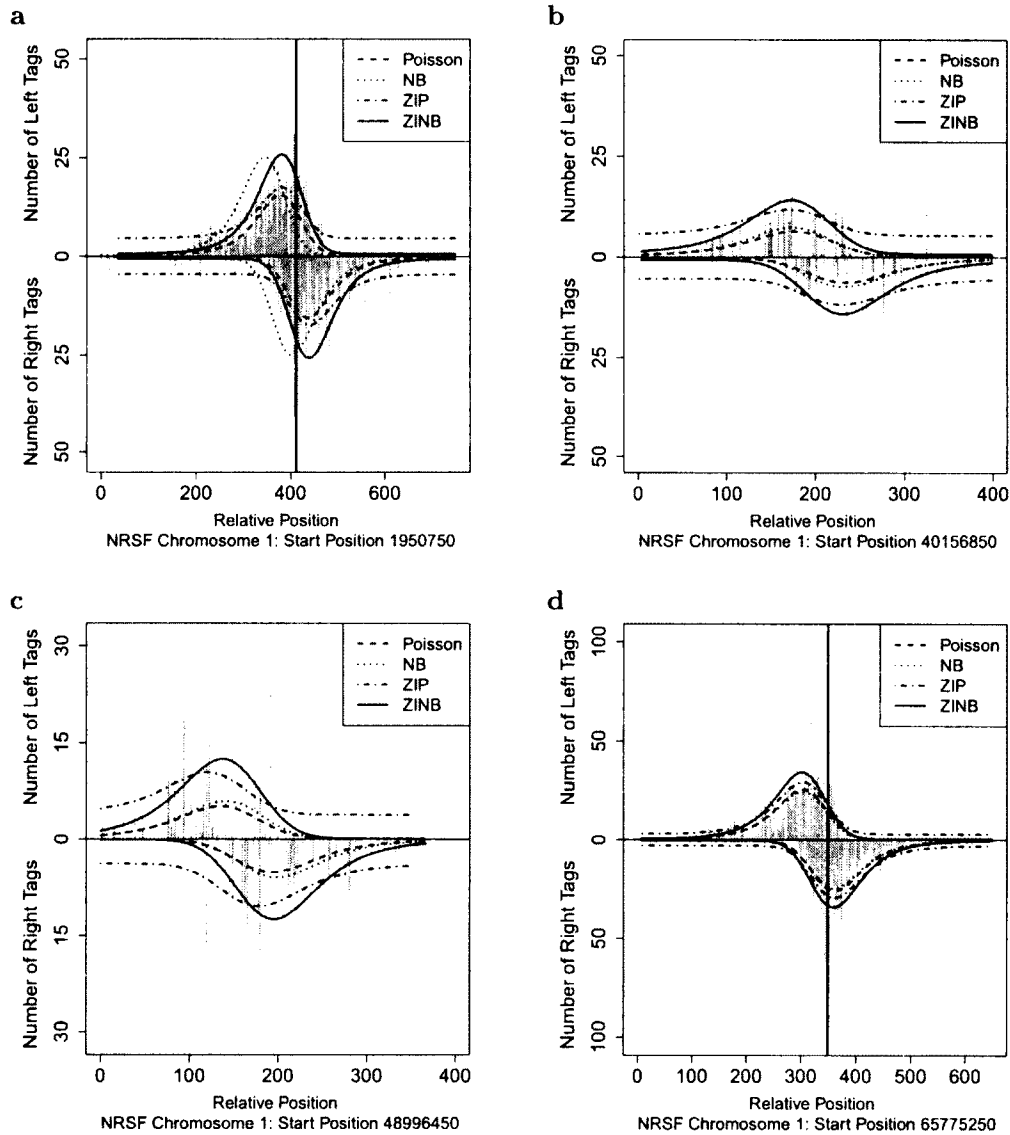
Figure 34 Performance comparison using NRSF regions. Regions were found in the first chromosome, and the Poisson, NB, ZIP, and ZINB models were fitted. Then the estimated mean values were plotted against the observed tags. Thick vertical lines in (a) and (d) denote the location of the motif sites within the region. There were no motif sites observed in regions (b) and (c). The height of the peak is proportional to the estimated ν value plus the estimated ρ value. In all four regions, the ZINB model fits better than the other three models.
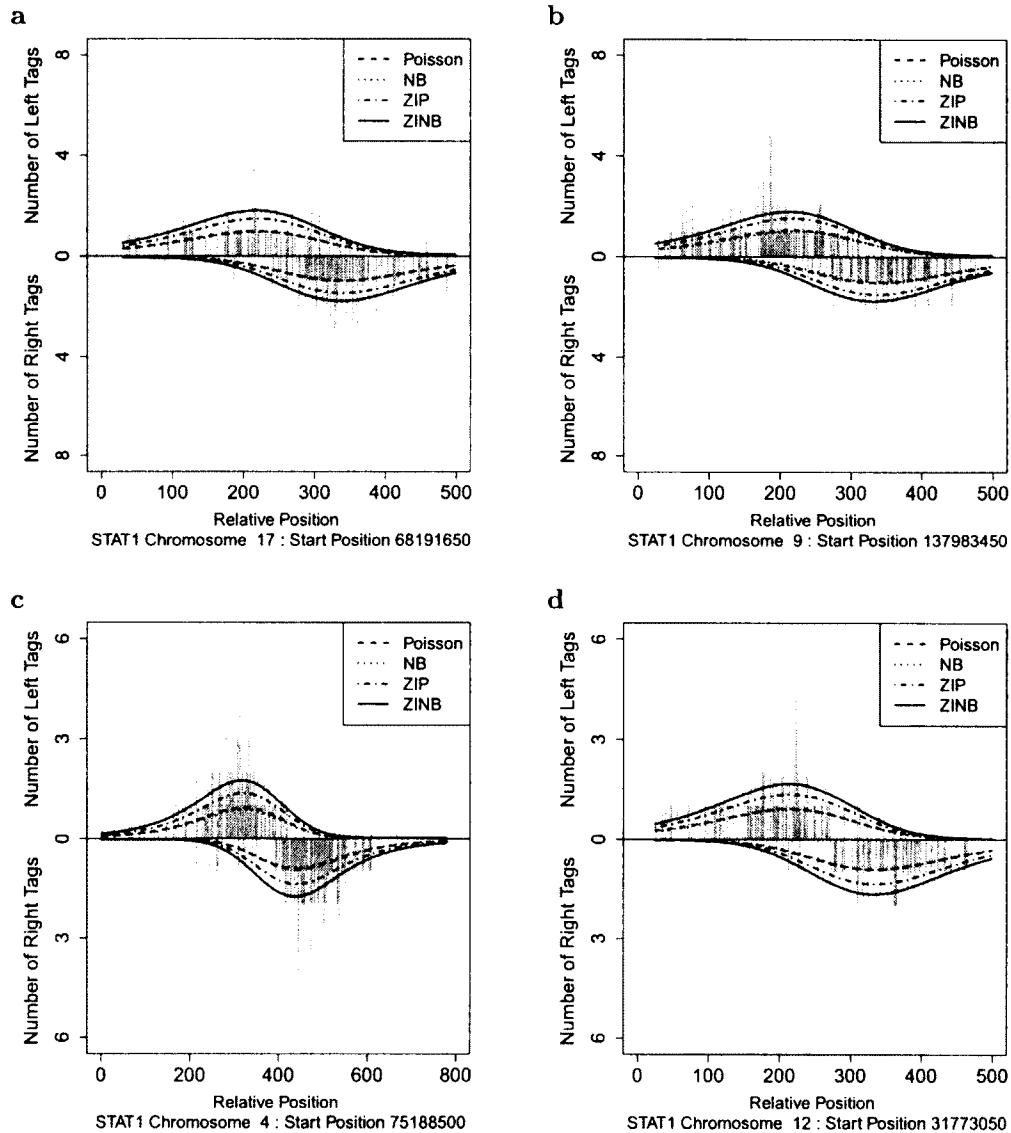
Figure 35 Performance comparison using STAT1 regions. This plot shows four regions with relatively smaller tag frequencies (< 8). Higher peaks indicate a larger ν value. Since regions are ranked based on the value of the ν parameter, it is important to have reliable ν estimates.

From all these plots, it is evident that the ZINB model performed better, capturing the higher variability and the zero counts in the ChIP-seq data. Next, the performance of the ZINB model is compared with the two existing peak-callers.
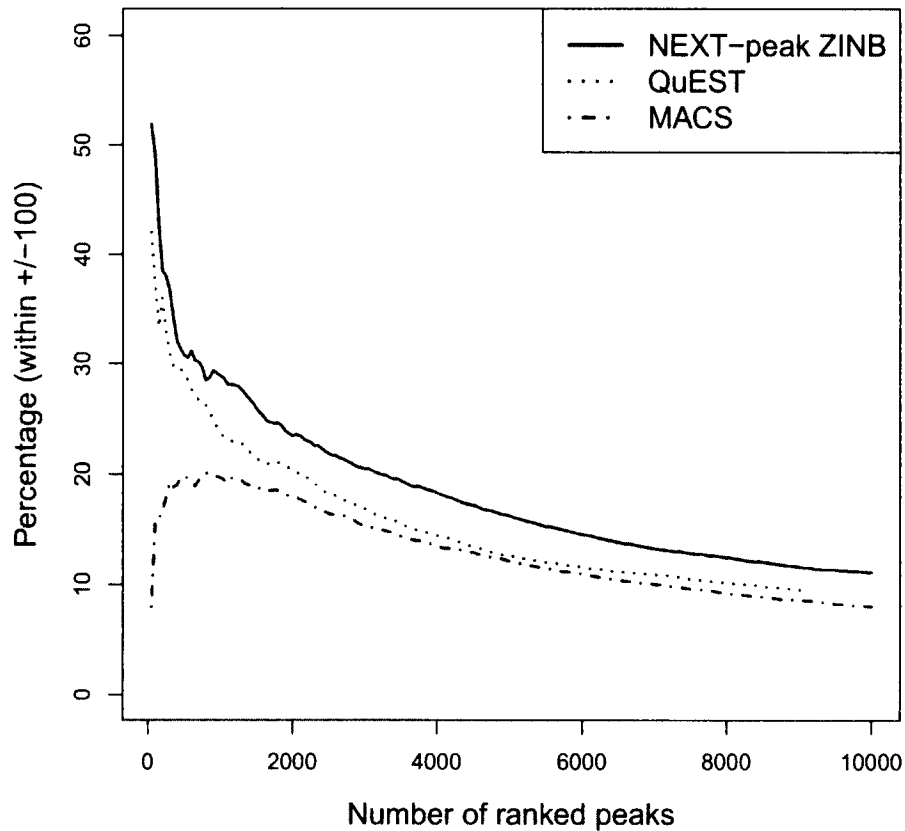
Figure 36 Percentage plot for GABP dataset. Peaks were estimated from the ZINB model and were ranked based on the corresponding ν_s values. Then, QuEST and MACS peak-callers were also used to find peaks for GABP data. Next, they were examined in increasing 50 peak intervals such as the top 50 peaks, top 100 peaks, and so on. A predicted peak was considered a true positive if it was within 100 bps from a motif site. Then for each interval, the percentage of peaks containing a motif site was computed.
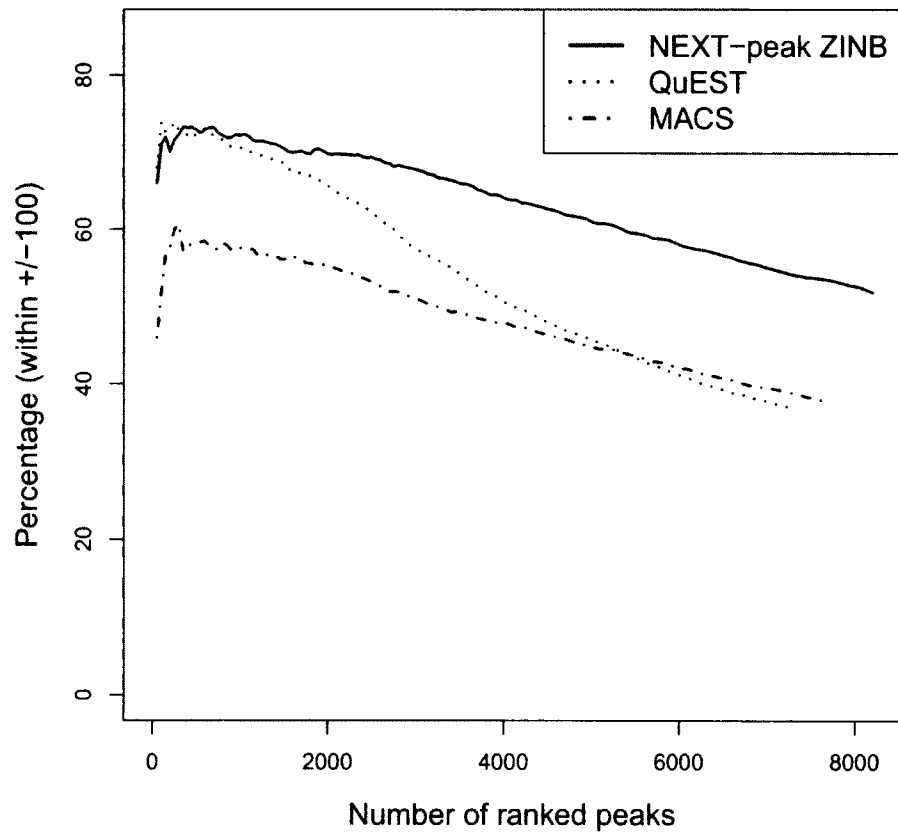
Figure 37 Percentage plot for STAT1 dataset. Cumulative percentages were calculated for STAT1 data and plotted. The ZINB model performed better than QuEST and MACS.

From Figures 36 and 37, it can be seen that our ZINB model performed better than QuEST and MACS. These cumulative percentage plots were obtained by treating the motif site locations as the true binding sites. However, true binding site locations can be different from the motif site locations. Also, there can be true binding site locations that were not captured during the motif search. For example, in Figures 33-35, there are many regions with the two peak pattern. Nevertheless, only a few regions reported having a motif site nearby(shown using a thick vertical line). Thus, a well formulated simulation scheme is needed to account for the variability, zero-inflation and other biases such as GC bias and mappability bias of ChIP-seq experiments. This would be a future extension of this study.

CHAPTER 4

DISCUSSION

In this dissertation we have developed a per-base, zero-inflated negative bino-mial (ZINB) regression model to identify transcription factor binding sites in chro-matin immunoprecipitation followed by massively parallel sequencing (ChIP-seq) data. First, we proposed three new per-base regression models: negative binomial (NB), zero-inflated Poisson (ZIP), and zero-inflated negative binomial (ZINB). Then, using NRSF, ZNF143, and STAT1 datasets, it was shown that the ZINB regression model fit the ChIP-seq data best. A new algorithm was implemented to find the high tag accumulated domains (a.k.a. regions) across the genome. For all three testing datasets (NRSF, STAT1 and GABP), the new algorithm performed better than the existing one. Using an identity link and a logit link, we introduced the GC covariates and the control data into the ZINB regression model. Three single-end tag (SET) IP and control datasets– NRSF, STAT1 and GABP– and a paired-end tag (PET) IP and control dataset (STAT6) were used for the model comparison.

For the NRSF and GABP datasets, the ZINB regression model with the logit link and the identity link with no control data achieved the lowest AIC values. Logit model with the control data in the identity link showed the lowest AIC value for the STAT1 dataset. The GC covariate was not significant in all three datasets. For the PET dataset, the logit model with no control data in the identity link was significant. Furthermore, the GC covariate had no significant effect on the peak prediction. The estimated number of tags due to binding (i.e., ν) is used to rank peaks. The performance of the ZINB models over other models was illustrated using some selected regions. When the mappability information was incorporated, the performances of the ZINB models improved.

Performance of the chosen ZINB models was compared with two existing peak-callers: MACS and QuEST. Known motif site locations were used for the comparison, and any predicted peak within 100 bases from a motif site was considered a true bind-ing site. ZINB models outperformed the MACS and QuEST peak-callers. Although motif sites are being widely used to compare performances of peak-callers, the use

of a well structured simulation scheme would help effectively understand the correct picture.

Our program is coded in C++ for faster computation. Using only the largest chromosome, six ZINB models are fitted, and the model with the lowest AIC value is chosen as the best model for the dataset. The selected model is then fitted using the entire dataset, and peaks are found for each region. A Linux executable and the source code is freely available at http://www.people.vcu.edu/~nkkim/nextpeak.html and https://sites.google.com/site/sameeraviswakula/downloads.

REFERENCES

[1] Kauffman C, Karypis G: **Computational tools for protein-DNA interactions**. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2012, **2**(1):14–28, ISSN 1942-4795.

[2] Schwartzman A, Jaffe A, Gavrilov Y, Meyer CA: **Multiple testing of local maxima for detection of peaks in ChIP-Seq data**. *The Annals of Applied Statistics* 2013, **7**(1):471–494.

[3] de Boer BA, van Duijvenboden K, van den Boogaard M, Christoffels VM, Barnett P, Ruijter JM: **OccuPeak: ChIP-Seq peak calling based on internal background modelling**. *PloS ONE* 2013, **9**(6).

[4] Grada A, Weinbrecht K: **Next-Generation sequencing: Methodology and application**. *Journal of Investigative Dermatology* 2013, **133**(8):e11–e11.

[5] Sanger F, Nicklen S, Coulson AR: **DNA sequencing with chain-terminating inhibitors**. *Proceedings of the National Academy of Sciences* 1977, **74**(12):5463–5467.

[6] Mardis ER: **Next-generation DNA sequencing methods**. *Annual Review of Genomics and Human Genetics* 2007, **9**:387–402.

[7] Benjamini Y, Speed TP: **Summarizing and correcting the GC content bias in high-throughput sequencing**. *Nucleic Acids Research* 2012, **40**(10).

[8] Landt SG, Marinov GK, Kundaje A, Kheradpour P, Pauli F, Batzoglou S, Bernstein BE, Bickel P, Brown JB, Cayting P: **ChIP-seq guidelines and practices of the ENCODE and modENCODE consortia**. *Genome Research* 2012, **22**(9):1813–1831.

[9] Ren B, Robert F, Wyrick JJ, Aparicio O, Jennings EG, Simon I, Zeitlinger J, Schreiber J, Hannett N, Kanin E, Volkert TL, Wilson CJ, Bell SP, Young RA: **Genome-wide location and function of DNA binding proteins**. *Science* 2000, **290**(5500):2306–2309.

[10] Johnson DS, Mortazavi A, Myers RM, Wold B: **Genome-wide mapping of in vivo protein-DNA interactions**. *Science* 2007, **316**(5830):1497–1502.

[11] Robertson G, Hirst M, Bainbridge M, Bilenky M, Yongjun Z, Zeng T, Euskirchen G, Bernier B, Varhol R, Delaney A, Thiessen N, Griffith OL, He A, Marra M, Snyder M, Jones S: **Genome-wide profiles of STAT1 DNA association using chromatin immunoprecipitation and massively parallel sequencing.** *Nature Methods* 2007, 4(8):651 – 657, ISSN 15487091.

[12] Barski A, Cuddapah S, Cui K, Roh TY, Schones DE, Wang Z, Wei G, Chepelev I, Zhao K: **High-resolution profiling of histone methylations in the human genome.** *Cell* 2007, **129**(4):823 – 837, ISSN 0092-8674.

[13] Mikkelsen T, Ku M, Jaffe D, Issac B, Lieberman E: **Genome-wide maps of chromatin state in pluripotent and lineage-committed cells** 2007.

[14] Furey T: **ChIPseq and beyond: new and improved methodologies to detect and characterize proteinDNA interactions** 2012.

[15] Guide: Getting started with chip-seq.

[16] Bailey T, Krajewski P, Ladunga I, Lefebvre C, Li Q, Liu T, Madrigal P, Taslim C, Zhang J: **Practical guidelines for the comprehensive analysis of ChIP-seq data.** *PLoS Computational Biology* 2012, **9**(11).

[17] Langmead B, Trapnell C, Pop M, Salzberg SL: **Ultrafast and memory-efficient alignment of short DNA sequences to the human genome.** *Genome Biology* 2008, **10**(3).

[18] Bauer MJ, Cox AJ, Evers DJ: **ELANDv2 - Fast gapped read mapping for Illumina reads.** In: *ISMB*, ISCB.

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

[20] Rodriguez J, Asimenos G: **Next-generation sequencing technologies.** In: Aerni SJ, Sirota M, eds., *A Bioinformatics Guide for Molecular Biologists*, volume 1, Cold Spring Harbor Laboratory Press, New York, 1 edition, 2014, 155–186.

[21] Ma W, Wong WH: **Chapter three - the analysis of ChIP-Seq data**. In: Voigt C, ed., *Synthetic Biology, Part A*, volume 497 of *Methods in Enzymology*, Academic Press, San Diego, 2011, 51 - 73.

[22] Watson J, Crick F: **Molecular structure of nucleic acids** 1953.

[23] Anthony JF Griffiths DTSRCL Jeffrey H Miller, Gelbart WM: **The nature of DNA**. In: *Modern Genetic Analysis*, W. H. Freeman, New York, 1999.

[24] Park PJ: **ChIPseq: advantages and challenges of a maturing technology**. *Nature Reviews Genetics* 2009, **10**(10):669-680.

[25] Xu J, Zhang Y: **A generalized linear model for peak calling in ChIP-Seq data**. *Journal of Computational Biology : A Journal of Computational Molecular Cell Biology* 2012, **19**(6):826-838.

[26] Zhang Y, Liu T, Meyer CA, Eeckhoute J, Johnson DS, Bernstein BE, Nusbaum C, Myers RM, Brown M, Li W, Liu X: **Model-based analysis of ChIP-Seq (MACS)**. *Genome Biology* 2007, **9**(9).

[27] Kharchenko PV, Tolstorukov MY, Parkguide PJ: **Design and analysis of ChIP-seq experiments for DNA-binding proteins**. *Nature Biotechnology* 2008, **26**(12):1351-1359.

[28] Valouev A, Johnson DS, Sundquist A, Medina C, Anton E, Batzoglou S, Myers RM, Sidow A: **Genome-wide analysis of transcription factor binding sites based on ChIP-Seq data**. *Nature Methods* 2008, **5**(9):829-834.

[29] Xu H, Handoko L, Wei X, Ye C, Sheng J, Wei CL, Lin F, Sung WK: **A signal-noise model for significance analysis of ChIP-seq with negative control**. *Bioinformatics (Oxford, England)* 2010, **26**(9):1199-1204.

[30] Vega VB, Cheung E, Palanisamy N, Sung WK: **Inherent signals in sequencing-based chromatin-immunoprecipitation control libraries**. *PLoS ONE* 2009, **4**(4):e5241.

[31] Teytelman L, zaydn B, Zill O, Lefranois P, Snyder M, Rine J, Eisen MB: **Impact of chromatin structures on DNA processing for genomic analyses**. *PLoS ONE* 2009, **4**(8):e6700.

[32] Zhang Y, Su B: **Peak identification for ChIP-seq data with no controls.** *Zoological Research* 2013, **33**(6).

[33] Rozowsky J, Euskirchen G, Auerbach RK, Zhang ZD, Gibson T, Bjornson R, Carriero N, Snyder M, Gerstein MB: **PeakSeq enables systematic scoring of ChIP-seq experiments relative to controls.** *Nature Biotechnology* 2008, **27**(1):66–75.

[34] Valouev A, Johnson DS, Sundquist A, Medina C, Anton E, Batzoglou S, Myers RM, Sidow A: **Genome-wide analysis of transcription factor binding sites based on ChIP-Seq data.** *Nature Methods* 2008, **5**(9):829–834.

[35] Ho JW, Bishop E, Karchenko PV, Nègre N, White KP, Park PJ: **ChIP-chip versus ChIP-seq: lessons for experimental design and data analysis.** *BMC Genomics* 2010, **12**.

[36] Dohm JC, Lottaz C, Borodina T, Himmelbauer H: **Substantial biases in ultra-short read data sets from high-throughput DNA sequencing.** *Nucleic Acids Research* 2008, **36**(16).

[37] Kuan PF, Chung D, Pan G, Thomson JA, Stewart R, Kele S: **A statistical framework for the analysis of ChIP-Seq data.** *Journal of the American Statistical Association* 2011, **106**(495).

[38] Pepke S, Wold B, Mortazavi A: **Computation for ChIP- seq and RNA-seq studies.** *Nature Methods* 2009, **6**(11 Suppl):S22–S32.

[39] Laajala T, Raghav S, Tuomela S, Lahesmaa R, Aittokallio T, Elo L: **A practical comparison of methods for detecting transcription factor binding sites in ChIP-seq experiments.** *BMC Genomics* 2009, **10**(1):618, ISSN 1471-2164.

[40] Wilbanks EG, Facciotti MT: **Evaluation of algorithm performance in ChIP-Seq peak detection.** *PLoS ONE* 2010, **5**(7):e11471.

[41] Kim H, Kim J, Selby H, Gao D, Tong T, Lip Phang T, Choon Tan A: **A short survey of computational analysis methods in analysing ChIP-seq data.** *Human Genomics* 2011, **5**(2):117–123, ISSN 1479-7364.

[42] Kim NK, Jayatillake RV, Spouge JL: **NEXT-peak: a normal-exponential two-peak model for peak-calling in ChIP-seq data.** *BMC Genomics* 2013, **14**.

[43] Tran NT, Huang CH: **A survey of motif finding web tools for detecting binding site motifs in ChIP-Seq data.** *Biology Direct* 2014, 9(1).

[44] Mathelier A, Zhao X, Zhang AW, Parcy F, Rebecca W, Arenillas DJ, Buchman S, Chen CyY, Chou A, Ienasescu H, Lim J, Shyr C, Tan G, Zhou M, Lenhard B, Sandelin A, Wasserman WW: **JASPAR 2014: an extensively expanded and updated open-access database of transcription factor binding profiles.** *Nucleic Acids Research* 2013, **42**(Database issue):D142–D147.

[45] Matys V, OV K, Fricke E, Liebich I, Land S, A B, Reuter I, Chekmenev D, Krull M, Hornischer K, Voss N, Stegmaier P, B L, Saxel H, Kel A, Wingender E: **TRANSFAC and its module TRANSCompel: transcriptional gene regulation in eukaryotes.** *Nucleic Acids Research* 2005, **34**(Database issue):D108–D110.

[46] Lander E, Linton L, Birren B, Nusbaum C, Zody M, Baldwin J, Devon K, Dewar K, Doyle M, W F, Funke R, Gage D, Harris K, Heaford A, Howland J, Kann L, Lehoczky J, R L, P M, K M, Meldrim J, Mesirov J, Miranda C, Morris W, Naylor J, Raymond C, Rosetti M, Santos R, Sheridan A, Sougnez C, N S, Stojanovic N, Subramanian A, Wyman D, Rogers J, Sulston J, Ainscough R, Beck S, Bentley D, Burton J, Clee C, Carter N, Coulson A, Deadman R, Deloukas P, Dunham A, Dunham I, Durbin R, French L, Grafham D, Gregory S, Hubbard T, Humphray S, Hunt A, Jones M, Lloyd C, A M, Matthews L, Mercer S, Milne S, Mullikin J, Mungall A, Plumb R, Ross M, Shownkeen R, Sims S, Waterston R, Wilson R, Hillier L, JD M, Marra M, Mardis E, Fulton L, Chinwalla A, Pepin K, Gish W, Chissoe S, Wendl M, Delehaunty K, Miner T, Delehaunty A, Kramer J, Cook L, Fulton R, Johnson D, Minx P, Clifton S, Hawkins T, Branscomb E, Predki P, Richardson P, Wenning S, Slezak T, Doggett N, Cheng J, Olsen A, Lucas S, Elkin C, Uberbacher E, Frazier M: **Initial sequencing and analysis of the human genome.** *Nature* 2001, **409**(6822):860–921.

[47] Chung D, Park D, Myers K, Grass J, Kiley P, Landick R, Keles S: **dPeak: high resolution identification of transcription factor binding sites from PET and SET ChIP-Seq data**. *PLoS Computational Biology* 2013, **9**(10).

[48] Jayatillake R: *A Statistical Model to Determine Multiple Binding Sites of a Transcription Factor on DNA Using ChIP-Seq Data*. Ph.D. thesis, Old Dominion University, Mathematics and Statistics Department, 2012.

[49] Ji H, Jiang H, Ma W, Johnson DS, Myers RM, Wong WH: **An integrated software system for analyzing ChIP-chip and ChIP-seq data**. *Nature Biotechnology* 2008, **26**(11):1293–1300.

[50] Robinson MD, McCarthy DJ, Smyth GK: **edgeR: a bioconductor package for differential expression analysis of digital gene expression data**. *Bioinformatics (Oxford, England)* 2010, **26**(1):139–140.

[51] Hilbe JM: *Negative Binomial Regression*. Cambridge University Press, New Haven, 2011.

[52] Lambert D: **Zero-inflated poisson regression, with an application to defects in manufacturing**. *Technometrics* 1992, **34**(1):1–14.

[53] Rashid NU, Giresi PG, Ibrahim JG, Sun W, Lieb JD: **ZINBA integrates local covariates with DNA-seq data to identify broad and narrow regions of enrichment, even within amplified genomic regions**. *Genome Biology* 2011, **12**(7).

[54] Diaz A, Park K, Lim DA, Song JS: **Normalization, bias correction, and peak calling for ChIP-seq**. *Statistical Applications in Genetics and Molecular Biology* 2012, **11**(3).

[55] Erdman D, Jackson L, Sinko A: **Zero-inated poisson and zero-inated negative binomial models using the COUNTREG procedure**. In: *SAS Global Forum; Cary*, SAS Institute Inc.

[56] Burden RL, Faires JD: **2.1 the bisection algorithm**. In: *Numerical Analysis*, PWS Publishers, 3 edition, 1985, 155–186.

[57] Liu DC, Nocedal J: **On the limited memory BFGS method for large scale optimization**. *Mathematical Programming* 1989, **45**(3):503–528, ISSN 0025-5610.

[58] Brent RP: **Chapter 4: An algorithm with guaranteed convergence for finding a zero of a function**. In: *Algorithms for Minimization without Derivatives*, Prentice-Hall, Englewood Cliffs, NJ, 1973.

[59] Gough B: *GNU Scientific Library Reference Manual - Third Edition*. Network Theory Ltd., 3rd edition, 2009, ISBN 0954612078, 9780954612078.

[60] Akaike H: **A new look at the statistical model identification**. *IEEE Transactions on Automatic Control* 1974, **19**(6):716–723.

[61] Schwarz G: **Estimating the dimension of a model**. *The Annals of Statistics* 1978, **6**(2):461–464.

[62] Liang K, Keles S: **Normalization of ChIP-seq data with control**. *BMC Bioinformatics* 2011, **13**.

APPENDIX A

PERL CODES USED TO PREPROCESS DATA

A.1 PERL CODE TO EXTRACT FIELDS FROM BOWTIE2 OUTPUT

```perl
#!/usr/bin/perl -w
while(<>){
        if(/\S+\s+(\d+)\s+chr(\d+)\s+(\d+).*/){
                if($1 == 0){
                        print "+","\t",$2,"\t",$3,"\n";
                }
                elsif($1 == 16){
                        print "-","\t",$2,"\t",$3,"\n";
                }
        }elsif(/\S+\s+(\d+)\s+chrX\s+(\d+).*/){
                if($1 == 0){
                        print "+","\t","23","\t",$2,"\n";
                }
                elsif($1 == 16){
                        print "-","\t","23","\t",$2,"\n";
                }
        }elsif(/\S+\s+(\d+)\s+chrY\s+(\d+).*/){
                if($1 == 0){
                        print "+","\t","24","\t",$2,"\n";
                }
                elsif($1 == 16){
                        print "-","\t","24","\t",$2,"\n";
                }
        }
}
exit;
```

A.2 PERL CODE TO EXTRACT FIELDS FROM BOWTIE2 PET OUTPUT

```perl
#!/usr/bin/perl -w
while(<>){
        if(/\S+\s+(\d+)\s+chr(\d+)\s+(\d+).*/){
                if($1 == 99){
                        print "+","\t",$2,"\t",$3,"\n";
                }
                elseif($1 == 147){
                        print "-","\t",$2,"\t",$3,"\n";
                }
        }
        if(/\S+\s+(\d+)\s+chr(\d+)\s+(\d+).*/){
                if($1 == 83){
                        print "-","\t",$2,"\t",$3,"\n";
                }
                elseif($1 == 163){
                        print "+","\t",$2,"\t",$3,"\n";
                }
        }
        if(/\S+\s+(\d+)\s+chrX(\d+)\s+(\d+).*/){
                if($1 == 99){
                        print "+","\t",$2,"\t",$3,"\n";
                }
                elseif($1 == 147){
                        print "-","\t",$2,"\t",$3,"\n";
                }
        }
        if(/\S+\s+(\d+)\s+chrX(\d+)\s+(\d+).*/){
                if($1 == 83){
                        print "-","\t",$2,"\t",$3,"\n";
                }
                elseif($1 == 163){
                        print "+","\t",$2,"\t",$3,"\n";
```

```
                }
        }
        if(/\S+\s+(\d+)\s+chrY(\d+)\s+(\d+).*/){
                if($1 == 99){
                        print "+","\t",$2,"\t",$3,"\n";
                }
                elseif($1 == 147){
                        print "-","\t",$2,"\t",$3,"\n";
                }
        }
        if(/\S+\s+(\d+)\s+chrY(\d+)\s+(\d+).*/){
                if($1 == 83){
                        print "-","\t",$2,"\t",$3,"\n";
                }
                elseif($1 == 163){
                        print "+","\t",$2,"\t",$3,"\n";
                }
        }
}
exit;
```

VITA

Sameera Dhananjaya Viswakula

Department of Mathematics and Statistics

Old Dominion University

Norfolk, VA 23529

Education

Ph.D.	Computational and Applied Mathematics (Statistics) Old Dominion University, Norfolk, VA (May 2015)
M.S.	Statistics University of Texas at El Paso, El Paso, TX (July 2011)
B.Sc.	Special Degree in Statistics University of Colombo, Sri Lanka (September 2008)

Experience

08/2014 - 04/2015	Biostatistical Programmer Department of Biostatistics, Virginia Commonwealth University, Richmond, VA
08/2011 - 07/2014	Graduate Teaching Assistant Department of Mathematics and Statistics, Old Dominion University, Norfolk, VA
08/2009 - 07/2011	Graduate Teaching Assistant Department of Mathematics and Statistics, University of Texas at El Paso, El Paso, TX
09/2008 - 07/2009	Assistant Lecturer Department of Statistics, University of Colombo, Sri Lanka

Typeset using LaTeX.