The Pennsylvania State University

The Graduate School

A GENERALIZED LINEAR MODEL FOR PEAK CALLING IN CHIP-SEQ DATA

A Dissertation in

Department of Statistics

by

Jialin Xu

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

December 2012

The thesis of Jialin Xu was reviewed and approved* by the following:

Yu Zhang
Associate Professor of Statistics
Dissertation Advisor, Chair of Committee

Naomi S. Altman
Professor of Statistics

Debashis Ghosh
Professor of Statistics and Public Health Sciences

Ross Hardison
T. Ming Chu Professor of Biochemistry and Molecular Biology

David Hunter
Professor of Statistics
Department Head

*Signatures are on file in the Graduate School.

Abstract

Chromatin immunoprecipitation followed by massively parallel sequencing (ChIP-Seq) has become a routine for detecting genome-wide protein-DNA interaction. The success of ChIP-Seq data analysis highly depends on the quality of peak calling to detect peaks of tag counts at a genomic location and evaluate if the peak corresponds to a real protein-DNA interaction event. The challenges in peak calling include 1) how to combine the forward and the reverse strand tag data to improve the power of peak calling, 2) how to account for the variation of tag data observed across different genomic locations, and 3) how to use the negative control data to reduce false positives caused by regional biases that might be generated by local structure.

I introduce a new peak calling method based on the generalized linear model (GLMNB) that utilizes negative binomial distribution to model tag count data and accounts for the variation of background tags that may randomly bind to the DNA sequence at varying levels due to local genomic structures and sequence contents. I allow local shifting of peaks observed on the forward and the reverse stands, such that at each potential binding site, a binding profile representing the pattern of a real peak signal is fitted to best explain the observed tag data with maximum likelihood. Our method can also detect multiple peaks within a local region if there are multiple binding sites in the region.

I also extend the model to incorporate ChIP-Seq data with multiple tracks in order to answer broader scientific questions. Assuming there are k ChIP replicates and one negative control data under c biological conditions, the extended model with likelihood ratio test can be used to identify 1) binding event under at least one conditions or 2) differential binding events under different biological conditions.

Table of Contents

List of Figures

List of Tables

List of Symbols

GLMNB Peak calling algorithm using generalized linear model using negative binomial distribution, p. 6.

$\mathbf{Y}$ A random variable that follows a certain distribution determined by f(y), p. 22.

μ The expected value of random variable $\mathbf{Y}$, p. 22.

β The coefficients in the linear combination of μ, p. 22.

$\mathbf{X}$ A predictor used to model $\mathbf{Y}$, p. 22.
A binding profile matrix, whose j-th column vector, $\vec{x}_{j,j'}$, contains smoothed profile vector from all k_j ChIP samples and one negative control sample, p. 46.

θ A interested model parameter in general, p. 23.

ϕ A model parameter not of interests, p. 23.

λ The expected value of y_i under Poisson distribution, p. 27.

δ The scale parameter in Γ distribution, p. 27.

μ_i The mean parameter for i-th bin for Γ distribution, which after log transformation is linked to the linear combination of expected tag count in i-th bin, $\mathbf{x_i}$, with β as coefficient, p. 27.
The expected value of y_i in i-th bin from a certain sliding window, p. 28.

α The dispersion parameter in negative binomial distribution(NB-1), p. 28.
The dispersion parameter in negative binomial distribution(NB-2), p. 29.

u_i a random variable following a Γ distribution with mean equal to 1, p. 28.

ν A common constant for shape parameter α and rate paramter β in a Γ distribution, p. 29.

ϵ_i A random error in nonparametric regression, not necessarily normally distributed, p. 30.

$m(x_i)$ A model-free function as a predictor for y, p. 30.

$\hat{m}_h(x)$ A Kernel regression estimator for $m(x)$, p. 31.

$K(t)$ A Gaussian kernel function in Gasser and Muller estimator, p. 31.

h Bandwidth parameter in kernel regression, p. 31.

α_{FWER} Family-wise type I error rate, p. 32.

α_{adj} Type I error rate in each test after multiplicity correction using Bonferroni correction or Sidak correction, p. 32.

FDR False discovery rate, p. 33.

$\vec{y}$ Observed tag count vector in a sliding window, p. 42.

y^S Observed tag count vector from ChIP sample in a sliding window, p. 42.

$y_i^F (y_i^R)$ Observed tag count in i-th bin from forward(reverse) strand in a sliding window, p. 42.

y^C Observed tag count vector from negative control sample in a sliding window, p. 42.

$z_i^F (z_i^R)$ Observed tag count in i-th bin from forward(reverse) strand in a sliding window from negative control sample, p. 42.

$\mathbf{y}$ Observed tag count vector combining all ChIP samples and negative control samples, p. 42.

$\hat{m}^F(t+\theta)$ Estimated profile height at position t for forward strand after shifting to the left by θ, p. 44.

$\hat{m}^R(t-\theta)$ Estimated profile height at position t for reverse strand after shifting to the right by θ, p. 44.

$h_F(h_R)$ bandwidth parameter for forward(reverse) strand, p. 44.

dm_i The bin center coordinate for i-th bin, p. 45.

$x_i^F(x_i^R)$ The expected tag counts for forward(reverse) strands from profiles in i-th bin, p. 45.

c_i The left boundary position in i-th bin, p. 45.

$\vec{z}_j$ A smoothed profile vector from negative control sample under j-th condition, p. 46.

$\beta_{1,j}$ The common coefficient for smoothed signal profile $x_{j,j'}^S$ in all replicates under j-th condition, p. 46.

$\beta_{2,j}$ The coefficient for smoothed signal profile x_j^C from negative control sample under j-th condition, p. 46.

k_j Number of ChIP samples under j-th condition, p. 46.

c Number of biological conditions, p. 46.

ω Model parameter vector $(\beta_1, \theta, \alpha)$ from GLMNB with one ChIP sample and no negative control sample, p. 49.
Model parameter vector $(\beta_1, \theta, \beta_2, \alpha)$ from GLMNB with one ChIP sample and one negative control sample, p. 52.

Chapter 1

Introduction

1.1 ChIP-Seq

Chromatin immunoprecipitation with massively parallel DNA sequencing (ChIP-Seq) is a new powerful high throughput tool for detecting protein-DNA interactions. Compared with traditional technologies, i.e., ChIP-chip, ChIP-Seq has been shown to offer higher specificity, sensitivity, peak resolution and reproductivity with stronger signal to noise ratios but less starting material[1].

While ChIP-Seq experiment offers higher data quality, it also gives several challenges for calling peaks in the subsequent data analysis. First, the mapped ChIP-Seq data are distinguished by single end, unpaired forward and reverse strands, which mark both ends of ChIP fragments, rather than the precise protein-DNA binding sites. It is challenging to combine both forward and reverse strands properly and increase the peak resolution. Second, since the ChIP reactions are enrichments rather than purifications, it is critical to model the background noise using a proper statistical model and distinguish real binding events from background noise. ChIP-Seq data shows local biases in different genomic regions due to chromatin structure, GC content bias in genome sequence as well as sequencing bias and mapping bias from experiments. Third, it is a interesting though challenging topic to increase peak calling power and specificity by combining multiple tracks of ChIP-Seq data. Suppose a protein has different binding preferences under different experimental conditions(for example, different cell life cycles) for the same genome location, we may wonder whether there is a general framework to answer the following questions: 1) find genome locations where there is a binding event under all conditions except negative conditions; 2) find genome locations where there is a binding event under at least one condition.

1.1.1 ChIP-Seq experiment

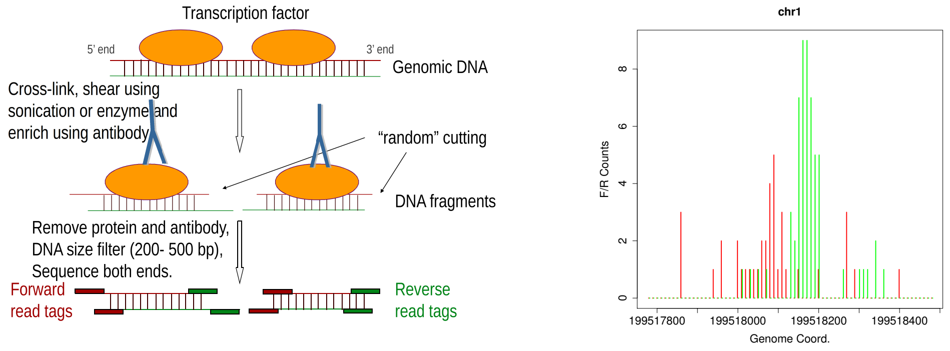
It is worthy to quickly go over the process of ChIP-Seq experiment so that we can understand model setting in this dissertation. There are typically two sets of data, ChIP sample and negative control sample. ChIP sample is the one we apply all procedures to detect the binding tags, whereas negative control sample (or input sample) is the one we process through exactly the same procedures as we do for ChIP sample but without adding the specific protein or transcription factor. Therefore, negative control sample theoretically contains all random background tags without enrichment for the specific transcription factor. There can be false positive signals in the control sample, due to for example high GC content, genome sequence binding affinity bias or sequencing error. As shown in Figure 1.1(a), in the ChIP sample, the target transcription factor (TF) is added into the cell sample. The transcription factor will cross-link with genome sequence after incubation for several hours. Then genome sequence will be sheared by sonication or non-sequence specific enzyme. The cutting positions are considered randomly distributed not far from TF protected DNA region without a specific sequence pattern. But the genome sequence at or close to the binding position has less chance been cut, thanks to the protection by TF. Therefore, the entire genome sequence are chopped into pieces of either TF-DNA complex or DNA only. TF-DNA complex contains a short fragment of DNA sequence and a TF binding to it. A TF-specific antibody (either monoclonal or polyclonal antibody) is added and specifically enriches those TF-DNA complexes. A majority of TF-DNA complexes are caught by the antibody during the enrichment step, whereas most DNA fragment not bound to the transcription factor are washed out. However, since the enrichment is less efficient than purification, the DNA fragment library actually contains a majority of DNA fragments from TF-DNA complexes and a minority of DNA fragments from DNA fragment only. The affinity between TF-DNA complexes and antibody is strong, whereas the affinity between DNA fragment and antibody is weak and therefore considered as background noise. After the enrichment, TF and the antibody are detached from DNA fragment and removed by changing the buffer condition. All enriched DNA fragments pass through a DNA size filter. Only those DNA fragments with a specific length (i.e. 200 base pair (bp) to 500 bp) will pass the filter and proceed to the sequencing step.

Due to the consideration of high throughput sequencing efficiency, only the two short ends (usually 25 to 50 nt) of each DNA fragment are sequenced separately from both 5' and 3' ends. The 5' end sequenced DNA are called forward read tags and the 3' end sequenced DNA are called reverse read tags. We only consider single end unpaired data in this dissertation. There is no knowledge to pair a forward read tag and a reverse tag and map back to exactly the same DNA fragment before sequencing. The dataset produced directly from the sequencing procedures are called raw data. The raw data are mapped to a reference genome using short sequence aligners, for example, Bowtie [2]. As a final step, read tags go through a data cleaning step, where only read tags uniquely mapped to reference genome are kept. The chromosome, start and end genome coordinates and strand direction information is stored as four columns in the dataset. For example, the tag at chromosome 1 starting at 199518200 and ending at 199518236 from forward strand is displayed as follows:

chromosome start end dir chr1 199518200 199518236 F

It is straightforward to extract forward and reverse tags respectively for a coordinate range for a specific chromosome. For illustration purposes, non-overlapping bins of width 10 bp in the desired range are created. All forward (reverse) tags with starting positions in a certain bin will be counted. In other words, the binned data contain counts of forward (reverse) tags starting from this bin. Then the forward (reverse) count data can be plotted as vertical bars in red(green).

Figure 1.1(b) illustrates a data display example for bined data. The X-axis is the genome coordinate in chromosome 1. The Y-axis is the tag counts starting at a certain bin. The red(green) vertical bars represent number of forward(reverse) tags in 10 bp wide bins. Forward and reverse signals form two peaks roughly 70 bp apart. The distance between observed data peaks varies by locations, but reflects the protected region of genome sequence by TF. The heights of forward and reverse peaks are generally similar but varies by locations too. They represent the strength of TF binding signals if there is no background noise.



(a)

(b)

Figure 1.1. (a)ChIP-Seq experiment and (b)ChIP-Seq data display example

According to the ChIP-Seq experiment description above combined with experience on ChIP-Seq data analysis, there are several features of ChIP-seq data that require our attention:

- Forward tags and reverse tags are single end, unpaired count data;

- The distribution of tags, even in the negative control sample, varies a lot along the entire genome, with different levels of average and variation;
- The peak position of forward strand tags and reverse strand tags are shifted in opposite directions around the binding position, and the peak shift distances vary by transcription factor and genome location;
- Due to different non-specific binding affinity in different locations, there are cases where strong signals can be observed in both ChIP sample and negative control data, which are false positives.

A unique feature of ChIP-Seq data is that the tag counts are obtained from the two strands of the genome. Due to the ChIP-Seq technology, at each protein-DNA binding site, the tag counts observed from the forward strands are mostly located on the left hand side of the binding site, and the tag counts observed from the reverse strand are mostly located on the right hand side. Due to the single-end, unpaired feature, a challenge of peak calling in ChIP-Seq data is therefore how to combine the tag counts from the two strands to increase the power of detecting real protein-DNA interaction sites. In particular, the accumulation (peaks) of tag counts at real binding sites often forms specific shapes, named binding profiles for both strands. It will help us to distinguish real protein binding event from random binding event. The binding profile is partially related to the ChIP-Seq technology and the structure of the proteins of interest, and therefore a valuable resource for binding event detection. The distance between forward strand peak and reverse strand one, in other hands, varies by transcription factors, experiments and even genome location. It should be taken into account when designing the binding site prediction algorithm to rule out its impact on signal strength variation as much as possible. How to best estimate such distance and therefore increase the power of detecting real protein-DNA interaction sites is therefore an interesting topic. In addition to real peaks, regions in the genome have varying levels of random binding affinity. Tags are more frequently observed at locations with open chromatin structures, and sequence contents also affect the variability of random tag counts ([3, 4, 5]). How to best account for the variation of tag counts across the genome and distinguish between real protein-DNA interaction and random binding event is an important problem in peak-calling. In the following session, a few previously published ChIP-Seq callers are reviewed in order to identify aspects for the new peak calling method to improve power. After the review of published methods, my peak calling method using Generalized linear model with negative binomial distribution(GLMNB) will be present.

1.1.2 Literature review on previous published methods

Peak calling in ChIP-Seq data analysis has been a very active field since its introduction. It seems almost impossible and not necessary to describe all details from previous methods in this section. One can refer to a comprehensive review on method comparison in ChIP-Seq peak callers written by Pepke et al [6], an algorithm performance review by Wilbanks et al [7], and two ChIP-Seq technology review papers by Park [8] and Metzker [9]. Here I would like to summarize modeling/algorithm features on several selected popular methods published since 2008 and focus on the commons and difference compared to GLMNB. These methods includes MACS by Zhang et al [1], SPP by Kharchenko et al [10], CisGenome by Ji et al [11], QuEST by Valouev et al [12], BayesPeak by Spyrou et al [3, 4], SISSRs by Jothi et al [13], HPeak by Qin et al [14].

Most previous methods utilized four steps: 1) building a background model, 2) estimating a global peak shift, 3) calling peaks, and 4) reporting predicted binding positions with significant p-values or false discovery rate (FDR)[15].

Yong Zhang et al presented MACS [1] in 2008, which became a very popular tool. MACS first empirically estimates a peak shift based on a sample of high confidence windows which contain a large fold-enrichment between ChIP and negative control samples. Such global peak shift size is estimated as half the distance between the modes of the forward and the reverse peaks from high confidence windows. Tags are merged together by shifting forward strand tags to the right and reverse strand tags to the left by this global peak shift size. There are two steps in MACS to prepare data for call peaking. MACS first linearly scales the total control tag count to be the same as the total ChIP tag counts so that FDR are appropriate. MACS also removes duplicate tags in excess of the tag amount with respect to the current sequencing depth (i.e. 1 tag per position for 3.9 million tags for FoxA1 ChIP sample) per genomic position in order to avoid biases during ChIP-DNA PCR amplification. MACS uses a scan window and assumes that merged tag counts in the non-overlapping window in non-enriched regions follows a Poisson distribution, with the Poisson parameter being the maximum value of average tag counts within a 1 thousand base pair(kb), 5 kb and 10 kb neighborhood or across the entire genome. A candidate peak is called if its p-value under the Poisson distribution is below a threshold p-value (1e-5 by default). An empirical FDR is reported only when a control data is available.

There are two key features in MACS. 1) Empirical modeling peak shift size and shifting tags before peak calling. 2) MACS shifts and combines all forward and reverse strand tags toward the center by the estimated shift size. To account for the local variability of tag counts due to genomic features, MACS estimates a local Poisson parameter as the average tag counts from an up to 10 kb neighboring region around each sliding window. The local Poisson parameter is calculated differently with or without control samples.

Even though the empirical peak shift size modeling is good enough for rough determination of forward and reverse strand peak distance, such a peak shift ignores the peak shift variation among different genomic regions. Even though MACS utilizes a local estimation for the Poisson parameter, it does not fully consider the variation of tag count mean and variance in different genomic regions. When there are no control data available, MACS does not compute a FDR. Due to the constraint of mean and variance equality in the Poisson distribution, however, MACS is not able to model peak data if the variability of tag counts far exceeds the mean. Also, MACS reports binding regions with highly variable sizes, ranging from 200 bp to 7 kb. Due to its wide range of sizes of the predicted binding intervals, MACS tends to call only a single peak at regions of clusters of peaks.

SPP [10] was published by Kharchenko et al in 2008. Rather than estimating the global peak shift size based on a large number of aligned signals, SPP first selects a global peak shift size from a cross-correlation analysis, which maximizes the 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. Two methods are recommended by SPP, window tag density method(WTD) and mirror tag correlation(MTC), depending on tag distribution immediately near the center of TF protected region. TF protected region is defined as a region between a major forward tag peak and a major reverse tag peak, where there is barely no ChIP-Seq tags thanks to TF protection from sonication or enzyme cutting in ChIP-Seq experiment. For example, MTC works better if most of TF protected regions in the entire genome are less than 30 bp. In WTD method, SPP utilizes a sliding window and calls a peak if a binding score is locally maximized, which is defined as twice of the difference between two qualities: 1) the geometric mean of the forward upstream tag counts and the reverse downstream tag counts and 2) the arithmetic mean of the forward downstream tag counts and the reverse upstream tag counts. In other words, SPP is looking for a position that maximizes the correlation between forward/reverse strands and upstream/downstream characters similar to Chi-square test in a contingent table. In MTC method, SPP focuses on the mirror similarity of forward and reverse strand peak shapes. In addition to the same binding score as that in WTD method, MTC method also looks at the Pearson's correlation between tag count vector of forward strand in upstream and that of reverse strand in the downstream. A peak position is identified at the location where such mirror similarity and binding scores reach a local maximum. For both methods, window tag counts are adjusted by subtracting the weighted number of negative control tags from the tag counts in ChIP sample within the same window. The weight is calculated as the ratio of overall tag counts from ChIP sample to overall tag counts from negative control sample. SPP returns a FDR[15] for each window, which is estimated as the fraction of the number of binding positions with a certain score or higher found in the negative control sample over the number found in ChIP sample. Such a background sample is either a negative control sample if available or a random background sample generated by randomly reassigning position of ChIP sample tags. SPP assigned a common smallest FDR value to the top peaks until the present of a new false positive appears as binding score increases, which does not reflect the peak strength for top peaks.

CisGenome proposed by Hongkai Ji et al [11] uses a sliding window (window size= 100 bp) strategy with a two-pass algorithm for peak calling. In the first pass, high-quality peaks are detected to estimate peak shift, which is computed as median distance between the modes of coupled forward and reverse peaks. In the second pass, the reads are shifted toward the center by the estimated peak shift size and peaks are called using sliding window strategy again. A peak is called if the observed tag counts within a sliding window(default 100 bp) significantly exceeds the expected tag counts based on a background distribution. When negative control sample is not available, CisGenome models the observed tag counts in a window using a negative binomial distribution, which is claimed to allow the background rate of DNA tag occurrence to vary across the genome and to have a more flexible Γ distribution. The negative binomial parameters from the non-binding regions where there are two or fewer reads per window. When negative control data is available, CisGenome models the difference between ChIP counts and control counts in window using binomial distribution conditional on the total tag counts from the ChIP and negative control samples. It has been demonstrated that the negative binomial model can better fit ChIP-Seq datasets than the Poisson model

[11]. CisGenome also takes advantage of control data and fits it with conditional binomial distribution. However, this method ignores the shape of binding profile or the similarity of forward and reverse strands.

Valouev et al [12] introduced QuEST in 2008. QuEST estimated a global peak shift size as half of the average distance between peaks on the negative and positive strand from regions with a large number of tags (over 600 tags within 300 bp and at least 20-fold scoring maximum than its next local scoring maximum if control sample is not available or at least 20-fold changes between scoring maximum in ChIP sample and that in control sample). Secondly, QuEST creates separate kernel density estimation profiles for both strands and combines both into one height score by shifting toward the center using the estimated peak shift size. In other words, within each window of size 21 bp, a combined kernel density is estimated. Local peak height and ratio of peak height between ChIP sample and control sample are used for peak calling. To be more specific, a candidate peak is called, if a sliding window contains a shifted profile with height greater than the threshold calculated from FDR procedure, and the following artificial criteria meets as well. The criteria include a) the lowest point between the current peak and the adjacent higher one is lower than 0.9 times the height of the higher peak, b) the height of background peak is lower than the background height threshold and c) the height ratio between ChIP and background is greater than a certain threshold. A empirical FDR is used to correct multiple comparison problems. The control data is split into two parts, pseudo-ChIP and control dataset. The same peak calling procedure is applied on pseudo-ChIP. Peaks called from pseudo-ChIP sample are treated as false positive calls and used to calculate FDR. The empirical FDR is calculated as number of false positives from pseudo-ChIP sample divided by number of peaks called in ChIP-Seq experiment. However, QuEST called peaks only based on local peak height and ratio. QuEST does not involve any statistical models to fit the count data and does not convert peak scores into definitive P values.

SISSRs was introduced by Jothi et al[13] in 2008. SISSRs first estimates the average DNA fragment length from the ChIP-Seq reads as the average distance between nearest forward and reverse strand tags. All short read tags are directionally extended from their start position to form a hypothetical DNA fragment by this estimated DNA fragment length, the same method employed by Robertson et al[16]. Such DNA fragment length is used to identify candidate binding sites and estimate false discovery rate(FDR). SISSRs partitions the genome into small windows of equal size(by default of size 20 bp), then counts the number of forward and reverse tags located in each window. SISSRs then calculates a net tag count for each window, as the difference between the number of forward tags and the number of reverse tags in the same window. A candidate binding site is recorded whenever the net tag count transits from positive to negative, and a few arbitrary conditions are satisfied. These conditions include: 1) the number of reverse strand tags upstream of candidate binding sites of size F is above a user defined tag count threshold (2 tags by default); 2) the number of forward strand tags downstream of candidate binding sites is above the same tag count threshold; 3) the sum of these two tag counts is above another tag count threshold estimated based on the user-set FDR.

The FDR is estimated as the ratio of the expected peak number based on background Poisson distribution, to the number observed in the real data for a certain tag counts. Such count value R is estimated from a Poisson background distribution. The only parameter of Poisson background distribution is the expected number of forward and reverse tags, λ, within a window of length the same as DNA fragment length. It can be calculated as the DNA fragment length multiplied by the number of tags divided by the mappable genome length. SISSRs is claimed to discover many more peaks than previous published methods, which is probably due to the over-optimistic Poisson model with constant expected value and therefore over-optimistic FDR calculation. By default extremely small FDR threshold of 10^{-3}, SISSRs discovers over 15,000, 10,000 and 5,500 peaks for GABP, FoxA1 and NRSF ChIP-Seq data, close to the amounts discovered by other popular peak callers such as MACS, QuEST, SPP [7]. Even though SISSRs does not allow users to choose a specific FDR threshold, one can still imagine that if a FDR threshold of 0.05 is used, SISSRs will discover many more peaks with many false positives included.

BayesPeak presented by Spyrou et al [3, 4] uses a fully Bayesian hidden Markov model to call peaks and is available in Bioconductor[17] package in R software. BayesPeak first divides the whole genome into non-overlapping windows of equal sizes. The window size is at least half the estimated average DNA fragment length, which is much larger than the bin size used in previous methods, such as SISSRs[13], QuEST[12], HPeak[14]. Forward and reverse tag counts in each window are recorded. Since the window size is at least half the average fragment length, forward tag counts in a window are highly correlated with the reverse tag counts in the adjacent downstream window. Instead of using two hidden states (ChIP-enriched/ non-enriched) for each window as HPeak[14] did, BayesPeak assumes four hidden states for two adjacent windows along the entire genome. These four hidden states include 1) non-enriched state in both windows; 2) non-enriched state for the first window and ChIP-enriched state for the second window; 3) ChIP-enriched state for the first window and non-enriched state for the second window; and 4) both ChIP-enriched states for both windows. The first hidden state is considered to have no enrichment effect, while the rest three states are considered to have the same enrichment effect. Therefore, the forward tag counts in the current window and the reverse tag counts in the next window in non-enriched state are assumed to follow a Poisson distribution. The expected value equals to the product of a relative fragment abundance parameter for non-enriched state multiplied by a correlation parameter between ChIP sample and input sample with a power of the tag counts in both windows from input data. The forward tag counts in the current window and the reverse strand tag counts in the next window in ChIP-enriched state are assumed to follow a similar Poisson distribution but another relative fragment abundance parameter for ChIP-enriched state added to that for non-enriched state. And both relative fragment abundance parameters for ChIP-enriched state and non-enriched state are assumed to follow a gamma distribution with two different sets of parameters. Therefore, the forward tag counts and reverse tag counts are modeled using negative binomial distribution conditioning on enrichment states. Beta prior is used for the transition probability from ChIP-enriched state to non-enriched state and that from ChIP-enriched state to ChIP-enriched state. Using Markov chain Monte Carlo(MCMC), all these model parameters are estimated. The likelihood expression were evaluated using Baum-Welch algorithm and Gibbs sampling[18]. The nature of the hidden states is then estimated by the marginal posterior probabilities using the estimated model parameters. For instance, if the posterior probability of ChIP-enriched state for a specific window is greater than 0.5, then the hidden state is assigned as ChIP-enriched state. The computing time is also a practical problem for BayesPeak. For example, it takes 11 hours for BayesPeak on Linux laptop with 4GB memory to complete peak calling for chromosome 1 (247MB with 310,000 tags in ChIP sample and 430,000 tags in input sample). With the help of parallel computing technology in 12 cores, it takes roughly 20 hours to complete peak calling for FoxA1 ChIP data with negative control.

HPeak proposed by Qin et al [14] uses a hidden Markov model-based (HMM) Peak-finding algorithm to analyze ChIP-Seq data. HPeak is a model-based approach compared to some model-free methods discussed above, such as SPP. HPeak first extends each short read tag directionally from its start position to form a hypothetical DNA fragment(HDF) by a DNA fragment length, the same method employed by Robertson et al[16] and Jothi et al[13]. HPeak then partitions the whole genome into equal size bins(by default, 25 bp) and counts the numbers of extended HDF that fall in each bin. Then adjacent bins with non-zero tag counts are merged into single wide candidate peaks. The read coverage of each peak is recorded if this coverage exceeds a significance threshold. In the peak calling step, HPeak applied a two-state HMM on the HDF coverage profile to classify bins into either ChIP-enriched(peak) regions or non-enriched(background) regions. For ChIP only data, Generalized Poisson (GP) distribution and zero-inflated Poisson (ZIP) distribution are used to model read counts in ChIP-enriched states and background states, respectively. The generalized Poisson distribution is claimed much more flexible to model ChIP-enriched regions than Poisson distribution, because it contains two parameters, a Poisson mean λ and dispersion parameter, ϕ, and allows the variance different from the mean, similar to the negative binomial distribution. The zero-inflated Poisson distribution is used to model non-enriched regions whose bins contain mostly no tag because of low noise of ChIP-Seq data. A zero-inflated Poisson distribution is a mixture distribution of point mass at zero and a Poisson distribution, which contains two parameters, a Poisson mean μ and the proportion of zeros π in the mixture distribution. For an experiment with both ChIP sample and negative control sample, the HDF count differences are calculated between ChIP sample and control sample for all bins along the whole genome. The ChIP-enriched regions in ChIP sample are modeled using GP distribution, while both the ChIP-enriched and non-enriched regions in control sample, along with the non-enriched regions in ChIP sample, are modeled using ZIP distribution.

The HMM parameters are estimated from summary statistics using Viterbi algorithm[19]. For example, the initial probability of being in a peak and the transition probability from background to peaks, are estimated as the proportion of the genome that is covered by candidate peaks. The transition probability from peaks to background is defined in a way such that the length of peaks is roughly equivalent to the median length of merged candidate peaks. Two parameters in GP distribution and ZIP distribution are estimated with the method of moments. For two-sample cases, parameters of GP and ZIP distribution are estimated for ChIP samples and control samples, respectively.

Two steps of the Viterbi algorithm are iteratively applied until convergence: 1) conditioning on the current estimate of model parameters, hidden states are assigned for each bin; 2) conditioning on the currently assigned hidden states, bins are classified into ChIP-enriched and non-enriched states and therefore model parameters are updated. As a result, HPeak reports genomic location and length of merged peaks, the summit location with highest HDF coverage, the log transformed posterior probability of bins in ChIP-enriched states. There is no FDR reported by HPeak.

After detailed introduction on these previous published ChIP-Seq peak callers, we can now summarize their features.

The first feature is the background modeling using negative binomial distribution rather than Poisson distribution suggested by CisGenome, BayesPeak and HPeak. In addition to the mean parameter in Poisson distribution, negative binomial distribution contains a dispersion parameter which gives more flexibility to model background with variance larger than the mean. Therefore, modeling using negative binomial distribution is claimed to better fit ChIP-Seq data [11]. The second feature is the adjusted local tag background from MACS. MACS utilizes an adjusted local tag average as the estimated mean parameter λ for Poisson distribution in background; it accounts for not only the tag distribution in the current window, but also the tag distribution in the neighboring regions. Such strategy reduces the peak significance when the neighborhood background level is high and prevents false positives. Another widely used feature is sliding window strategy. With proper background modeling and multiple comparison correction, peaks can be called with appropriate FDR or p-values.

Even though these published methods are powerful to call peaks, there are also several aspects that require improvement. First of all, all programs above estimate a global and constant peak shift size for all potential binding regions based on either average distance between forward and reverse tag peaks from high confidence regions or the maximum Pearson correlation between forward and reverse tag vectors. All programs simply merge the forward and reverse tags together before peak calling, and thus may lose power if the estimated peak shift size is inaccurate at some real binding sites. The variation of peak shift distance in different genome locations is simply ignored, which will lose power. For example, Figure 1.2 shows a histogram of half distance between forward and reverse peaks in 654 windows containing the most tags in FoxA1 ChIP sample. The average value of 62 bp is used in MACS to merge forward and reverse strand tags. The standard deviation of 38 bp is simply ignored, not to mention that the sample average is not a good center tendency estimate for skewed data.

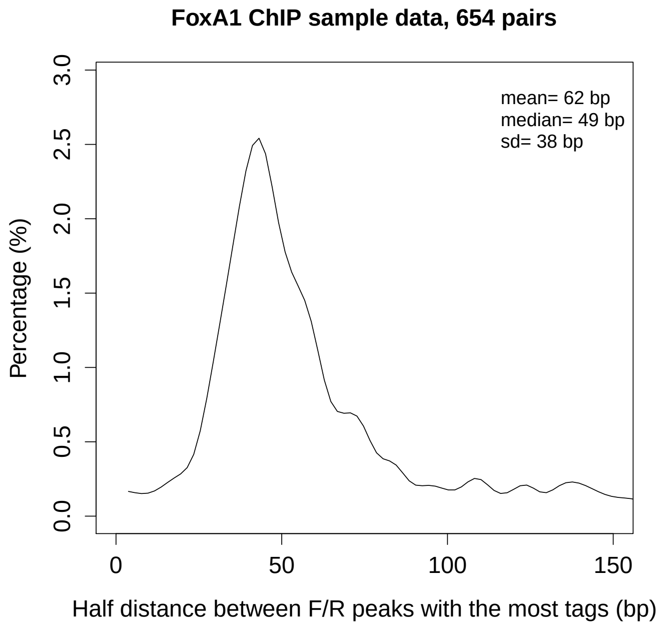


Figure 1.2. Histogram of half distances between forward and reverse peaks from 654 tag rich regions

In addition, most methods except QuEST and SPP(MTC) do not incorporate any binding shape information(including similarity between forward and reverse peaks). QuEST and SPP(MTC) also have their own limits. QuEST only combines forward and reverse tags within an arbitrary fixed range, 90 bp. SPP(MTC) combines forward and reverse tag information with a wider range but does not perform well if the protected region is wider than 30 bp.

Besides these two aspects, no algorithms investigates demonstrate a simulation study to access the correctness of p-value and peak calling resolution. A simulation study is demonstrated in Chapter 3.

Peak callings using MACS, SPP, CisGenome, SISSRs and BayesPeak are carried out using their default setting and their corresponding performances are compared with GLMNB. MACS and SPP are chosen because they are ranked top two with the highest spacial resolution among ChIP-Seq peak calling algorithms by Willbanks and Facciotti[7]. CisGenome is chosen because it is the first algorithm to employ negative binomial distribution to model the background noise and claims that negative binomial is superior to Poisson distribution. SISSRs is chosen because it claims to discover more peaks than most other algorithms. I want to evaluate GLMNB performance with respect to power. Finally, BayesPeak is chosen because it is one of the two algorithms that employ Bayesian framework on peak calling.

1.1.3 Important features in ChIP-Seq data

There are also several important features that utilized the first time in this dissertation.

The first feature in ChIP-Seq data is the strong correlation between forward and reverse strand tags. To show that, I use non-overlapping window of size 10 kb to scan all chromosomes in FoxA1 ChIP sample and negative control sample and record the tag counts. The scatter plots of forward tag counts and reverse tag counts from FoxA1 ChIP sample in Figure 1.3(a) and negative control sample in Figure 1.3(b) are plotted. The correlation between forward and reverse tag counts for ChIP sample($r = 0.959$) and negative control sample($r = 0.914$) are both very strong. Besides, at a binding site, the tag counts often follow a binding profile specific to the target protein, which provides valuable information to best distinguish between a real binding event from spurious peaks caused by events other than the target protein.

A second valuable feature in ChIP-Seq data is the variable peak shift distance depending

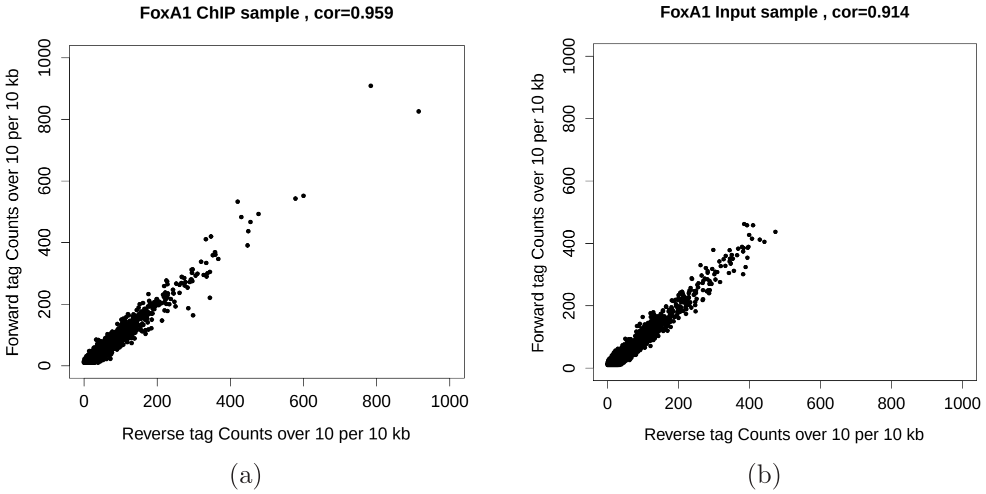


Figure 1.3. Scatter plot of forward and reverse tag counts for FoxA1 ChIP-Seq data in ChIP sample(a) and control sample(b).

on genome locations. As plotted in Figure 1.2 and discussed above, even for the same transcription factor, the forward and reverse peak distance may vary depending on the genome structure or sequence content. It is more flexible to use a local peak shift parameter rather than a constant value to fit the data best.

A third essential feature is the moderate correlation between ChIP sample data and input data. Thanks to the enrichment step in chromatin immunoprecipitation, the tag spatial distribution in the ChIP sample is moderately correlated with that in negative control sample. For example, in Figure 1.4, total tag counts from both strands in 10 kb non-overlapping windows in ChIP sample are plotted against those in negative control sample and have a Pearson's correlation value equal to 0.576. Each black dot represents total tag counts from ChIP sample and negative control sample in a 10 kb window. There are windows where tag counts in ChIP sample are far greater than those in negative control sample shown in top left side of the plot, which suggest their possible binding status. By contrast, there are also windows where tag counts in ChIP sample are very close to those in negative control sample shown along the bottom right edge, which suggest a non-binding status. Without the information from negative control sample, one may easily call those windows shown on the bottom right edge, for example the one with 300 ChIP tags and 490 input tags, a significant binding site according to its large value of tag counts in ChIP sample. However, it should be classified as non-binding position. Therefore, a negative control sample should be included in the peak calling framework in order to reduce false positives caused by this correlation.

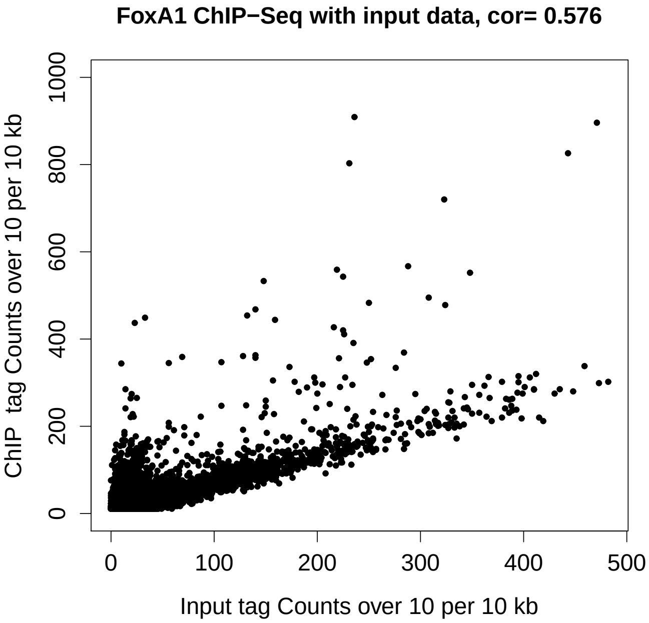


Figure 1.4. Scatter plot of tag counts in 10kb non-overlapping windows between ChIP and input samples.

1.2 Significance of GLM for peak calling

There are several points I need to highlight for this Ph.D dissertation in order to distinguish from previous methods.

First, it has be realized that the over-dispersion and high proportion of zero counts violate the assumption for Poisson distribution. Therefore, we use negative binomial distribution to model non-binding region. Second, we propose to call peaks with the adjustment of control data in GLMNB, rather than artificially set a cutoff of fold enrichment. This strategy should be more efficient in combining control data in peak calling. Third, a few methods that estimate peak shift only estimate a global one and ignore its variation before calling peaks. In other words, they do not allow the peak shift to vary in different regions, which may reduce the power of peak calling. The proposed generalized linear model (GLMNB) includes a adjusted peak shift parameter and let the model select a value that maximizes the likelihood. Fourth, 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. In particular, the accumulation (peaks) of tag counts at real binding sits in specific shapes referred as a binding profile, can help us distinguish real protein binding from random binding. The binding profile is partially determined by the ChIP-Seq technology and by the structure of the proteins of interest. In addition to real peaks, regions in the genome have varying level of random peaks. Tags are more frequently observed at locations with open chromatin structures, and sequence content also affects the variability of random tag counts ([3, 4, 5]). How to best account for the variation of tag counts across the genome and distinguish between real protein-DNA interaction and random peaks is an important problem in peak-calling. The local peak shift estimate along with the forward and reverse binding profiles generated in high confidence regions provides a powerful way to solve the question above. Fifth, since we use a sliding window strategy, we utilize false discovery rate to correct multiple testing for positively dependent tests. Last but not least, along with the increasing amount of ChIP-seq data, it will increase the specificity if we can integrate multiple ChIP-Seq tracks to call peaks. It is straightforward under generalized linear model framework.

1.3 Generalized linear model

In order to model the non-binding ChIP-seq count data using the negative binomial distribution and call peaks by hypothesis testing, we use the generalized linear model (GLM) framework. A benefit of GLM is its capability of combining different levels of information as covariates in the model. In such cases, we can simply add or remove different covariates without changing the framework.

1.3.1 Generalized linear model

The theory of generalized linear models was introduced by Nelder and Wedderburn in 1972 [20] as an extension from ordinary linear regression model. It allows modeling based on random error model besides normal distribution, for example Poisson distribution and Negative Binomial distribution for count data. The purpose of GLM is to specify the relationship between observed response variable and a certain number of covariates. Usually, GLM model the mean of response variable using a linear combination of covariates. Therefore, GLM contain three components.

- A random component for the response, $\mathbf{y}$, with a distribution following an exponential family.
- A linear systematic component (linear predictor) connecting covariates, $\eta = \mathbf{X}\beta$.
- A known monotonic, one-to-one, differentiable link function $g(\centerdot)$ connecting the linear predictor to the fitted values, i.e., $E(\mathbf{y}) = g^{-1}(\eta)$.

where $\mathbf{X}$ are independent variables, β is a coefficient vector for $\mathbf{X}$, and η is the linear combination of $\mathbf{X}$.

To sum up, in GLM, we specify the observed response value with a certain exponential family distribution, f, with parameters $w = (\mu, \ldots)$. We can model the mean μ using a monotonic, one-to-one, differentiable link function $g(\textbf{.})$ relating covariates $\mathbf{X}$ and corresponding coefficients β.

$$\begin{aligned} Y &\sim f(y) & (1.1) \\ \mu &= E(Y) & (1.2) \\ \eta &= g(\mu) = \mathbf{X}'\beta = \sum_{j=1}^{p} x_j\beta_j & (1.3) \end{aligned}$$

1.3.2 Regularity conditions for asymptotic property of maximum likelihood estimate in Generalized Linear model

Some regularity conditions for asymptotically normal distribution of Wald test statistic and asymptotically χ^2 distribution of likelihood ratio test in generalized linear model include[21, 22]:

- The set of data values which has positive probability should not depend on the unknown parameter.

- The observed samples are independent and identically distributed; Or if the assumption of i.i.d. observations does not hold, the amount of information in the data increases indefinitely as the sample size increases;

- The first and second derivatives of the log-likelihood function are defined;

- The Fisher information matrix must be positive semidefinite and continuous as a function of a parameter;

- The maximum likelihood estimator is consistent.

1.3.3 Newton-Raphson method

Even through there are two other methods for finding estimates of GLM, Fisher score method and iteratively re-weighted least squares (IRLS) method, we highlight the Newton-Raphson method in the following description because of its quick convergence and straightforward implementation. We start the Newton-Raphson method from the joint independent and identically distributed (i.i.d.) probability function as

$$f(\mathbf{y};\theta,\phi)=\prod_{i=1}^{n}f(y_i;\theta,\phi)$$

The likelihood function is

$$Ł(\theta,\phi;\mathbf{y})=\prod_{i=1}^{n}f(\theta,\phi;y_i)$$

More specifically, for exponential family,

$$f(y_i;\theta,\phi)=\exp\left\{\frac{y_i\theta_i-b(\theta_i)}{a(\phi)}+c(y_i,\phi)\right\}$$

we have the following likelihood and log likelihood function

$$\begin{aligned} Ł(\mathbf{y};\theta,\phi) &= \prod_{i=1}^{n}\exp\left\{\frac{y_i\theta_i-b(\theta_i)}{a(\phi)}+c(y_i,\phi)\right\} \\ \ell(\mathbf{y};\theta,\phi) &= \sum_{i=1}^{n}\left\{\frac{y_i\theta_i-b(\theta_i)}{a(\phi)}+c(y_i,\phi)\right\} \end{aligned}$$

where θ is the canonical parameter, $b(\theta)$ is the cumulative, ϕ is the dispersion parameter and $c()$ is a normalization term.

To get the first derivatives of the coefficient vector β, we use the chain rule from the log likelihood function,

$$\begin{aligned}
\frac{\partial \ell}{\partial \beta_j} &= \sum_{i=1}^{n} (\frac{\partial \ell_i}{\partial \theta_i})(\frac{\partial \theta_i}{\partial \mu_i})(\frac{\partial \mu_i}{\partial \eta_i})(\frac{\partial \eta_i}{\partial \beta_j}) \\
&= \sum_{i=1}^{n} (\frac{y_i - b'(\theta_i)}{a(\phi)})(\frac{1}{Var(\mu_i)})(\frac{\partial \mu}{\partial \eta})_i(x_{ij}) \\
&= \sum_{i=1}^{n} (\frac{y_i - \mu_i}{a(\phi)Var(\mu_i)})(\frac{\partial \mu}{\partial \eta})_i(x_{ij}) \qquad (1.4)
\end{aligned}$$

where $i = 1, \ldots, n$ indexes the observations and x_{ij} is the i-th observation for the j-th covariate $\mathbf{X_j}, j = 1, \ldots, p$.

We use Newton-Raphson method to find estimates $\hat{\beta}$ by iterating the following formula.

$$\beta^{(\mathbf{r})} = \beta^{(\mathbf{r-1})} - \left\{\ell''(\beta^{(\mathbf{r-1})})\right\}^{-1} \ell'(\beta^{(\mathbf{r-1})}) \qquad (1.5)$$

for $r = 1, 2, \ldots$ with a reasonable vector of starting values $\beta^{(\mathbf{0})}$ until convergence.

The matrix of second derivatives (the observed Hessian matrix) is given by

$$\begin{aligned}
\frac{\partial^2 \ell}{\partial \beta_j \partial \beta_k} &= \sum_{i=1}^{n} \frac{1}{a(\phi)} \frac{\partial}{\partial \beta_k} \left\{ (\frac{y_i - \mu_i}{Var(\mu_i)})(\frac{\partial \mu}{\partial \eta})_i(x_{ij}) \right\} \\
&= \sum_{i=1}^{n} \frac{1}{a(\phi)} [(\frac{\partial \mu}{\partial \eta})_i \left\{ (\frac{\partial}{\partial \mu})_i (\frac{\partial \mu}{\partial \eta})_i (\frac{\partial \eta}{\partial \beta_k})_i \right\} \frac{y_i - \mu_i}{Var(\mu_i)} \\
&\quad + \frac{y_i - \mu_i}{Var(\mu_i)} \left\{ (\frac{\partial}{\partial \eta})_i (\frac{\partial \eta}{\partial \beta_k})_i \right\} (\frac{\partial \mu}{\partial \eta})_i] x_{ij} \\
&= -\sum_{i=1}^{n} \frac{1}{a(\phi)} [\frac{1}{Var(\mu_i)} (\frac{\partial \mu}{\partial \eta})_i^2 \qquad (1.6) \\
&\quad -(\mu_i - y_i) \left\{ \frac{1}{Var^2(\mu_i)} (\frac{\partial \mu}{\partial \eta})_i^2 \frac{\partial Var(\mu_i)}{\partial \mu} - \frac{1}{Var(\mu_i)} (\frac{\partial^2 \mu}{\partial \eta^2})_i \right\}] x_{ij} x_{ik}
\end{aligned}$$

With the first two derivatives from equations (1.4) and (1.6), one can easily implement a Newton-Raphson algorithm to obtain the MLE of β.

The standard error of coefficient estimates can be found from the diagonal elements of variance estimate. In fact, the usual variance estimate in statistical software package is calculated as the inverse matrix of negative second derivatives. For GLM, we calculate the variance estimate using observed Hessian matrix [23].

1.3.4 Poisson GLM

Count data are often fitted using a Poisson GLM. Counts refer to a simple counting of events, i.e., number of read tags within a small range of genome coordinates in ChIP-Seq data. If the following assumptions hold, we can fit count data using Poisson model.

- The probability of observing a single event over a small interval is approximately proportional to the size of that interval.
- The probability of two events occurring in the same narrow interval is negligible.
- The probability of an event within a certain interval does not change over different intervals.
- The probability of an event in one interval is independent of the probability of an event in any other interval.

Since the Poisson probability density function (pdf) can be formulated as

$$f(y;\mu) = e^{-\mu}\frac{\mu^y}{y!}$$

where μ is the expected value of y.

we can write it in exponential-family form as follows

$$f(y;\mu) = exp\{ylog(\mu) - \mu - log\Gamma(y+1)\}$$

Therefore, the canonical link is log link, $\theta = log(\mu)$. In other words, we can model the counts using a Poisson GLM

$$\begin{aligned} Y &\sim Poisson(\mu) \\ log(\mu) &= (X)^T\beta \end{aligned}$$

And the mean and variance from a Poisson GLM are

$$\begin{aligned} E(Y) &= \mu \\ Var(Y) &= \mu \end{aligned}$$

However, there are two disadvantages about modeling count data using Poisson model. First, the mean and variance functions of the Poisson distribution are identical. In practice, it is almost never the fact in ChIP-Seq data. Over-dispersion happens when the real variance is greater than the expected variance in the model. In our Poisson GLM, over-dispersion often occurs because the real variance is higher than the expected variance, the same value as the mean. Hence, we tend to reject less null hypothesis than we should and lose power if over-dispersion happens.

Second, Poisson model assumes that the ratio between variance and mean are constant, 1. It is not the case for count data in ChIP-Seq data. Within a region with a fairly large amount of tags on average, we tend to see large variation of counts. However, within a region with small amount of tags on average, we tend to see small variation. So it suggests that the ratio between variance and mean should vary.

1.3.5 Negative binomial GLM

Negative binomial GLM is a good alternative model that overcomes the over-dispersion problem of Poisson model. The negative binomial GLM uses the same log link function as Poisson model and is almost always constructed based on a Poisson-gamma mixture model. There are two methods for motivating the negative binomial regression model, NB-1(constant over-dispersion) and NB-2(variable over-dispersion) regression models. Since we also need to take into account of variable ratio between variance and mean, we here emphasize NB-2 model.

1.3.5.1 Constant over-dispersion

In constant over-dispersion NB-1 GLM, we consider the following Poisson-gamma mixture,

$$\begin{aligned} y_i|\lambda_i, x_i &\sim Poisson(\lambda_i) \qquad (1.7) \\ \lambda_i &\sim \Gamma(\delta, \mu_i) \\ log(\mu_i) &= \mathbf{x_i}\beta \end{aligned}$$

where y_i is observed tag count in i-th bin in a sliding window, x_i is expected tag count in i-bin. λ_i is the expected value for y_i under Poisson distribution, which at the same time is a random variable following a Γ distribution with two parameters, δ and μ_i. δ is the scale parameter in Γ distribution not dependent on bins and μ_i is the mean parameter for i-th bin for Γ distribution. And μ_i after log transformation is linked to the linear combination of expected tag count in i-th bin with β as coefficient.

Then the mixture distribution is derived as

$$\begin{aligned} f(y_i|x_i) &= \int_0^\infty \frac{e^{-\lambda_i}\lambda_i^{y_i}}{y_i!}\frac{\delta^{\mu_i}}{\Gamma(\mu_i)}\lambda_i^{\mu_i-1}e^{-\lambda_i\delta}d\lambda_i \\ &= \frac{\delta^{\mu_i}}{\Gamma(y_i+1)\Gamma(\mu_i)}\frac{\Gamma(y_i+\mu_i)}{(\delta+1)^{y_i+\mu_i}} \\ &= \frac{\Gamma(y_i+\mu_i)}{\Gamma(y_i+1)\Gamma(\mu_i)}\left(\frac{\delta}{1+\delta}\right)^{\mu_i}\left(\frac{1}{1+\delta}\right)^{y_i} \end{aligned}$$

Let dispersion parameter $\alpha = 1/\delta$, we have

$$f(y_i|x_i) = \frac{\Gamma(y_i+\mu_i)}{\Gamma(y_i+1)\Gamma(\mu_i)}\left(\frac{1}{1+\alpha}\right)^{\mu_i}\left(\frac{\alpha}{1+\alpha}\right)^{y_i}$$

With the first two moments as

$$\begin{aligned} E(y_i) &= e^{\mathbf{x_i}\beta}\alpha \\ Var(y_i) &= e^{\mathbf{x_i}\beta}(1+\alpha)\alpha \end{aligned}$$

Since the variance to mean ratio, also named over-dispersion, is given by $(1+\alpha)\alpha$, which is constant for all observations, we call this setting as constant over-dispersion, or NB-1.

1.3.5.2 Variable over-dispersion

Variable over-dispersion(NB-2) GLM allow a gamma heterogeneity where the gamma noise has a mean of 1. It is a more general situation than constant over-dispersion (NB-1) GLM. As derived in Hardin and Hilbe's book [24], an individual unobserved random variable u_i follows Γ distribution with mean= 1. The product $\mu_i u_i$ is the conditional Poisson mean.

$$y_i|u_i, x_i \sim Poisson(\mu_i u_i) \tag{1.8}$$

$$\begin{aligned} \text{where } u_i &\sim \Gamma(\alpha = \nu, \beta = \nu) && (1.9) \\ \text{That is } f(u_i) &= \frac{\nu^\nu}{\Gamma(\nu)} u_i^{\nu-1} e^{-\nu u_i} \\ \text{link function: } \log \mu_i &= x_i \beta && (1.10) \end{aligned}$$

where y_i and x_i are defined the same as in equation 1.7. μ_i is the expected value of y_i in i-th bin from a certain sliding window. u_i is a random variable following Γ distribution. ν in equation 1.9 is a common constant for shape parameter α and rate paramter β in a Γ distribution, which will be cancelled out after calculating unconditional distribution of y_i. Using log link function, we are able to link μ_i, the expected value of y_i, with the linear combination of expected tag count x_i for i-th bin with β as coefficients in equation 1.10.

The conditional distribution is

$$f(y_i|u_i) = \frac{e^{-\mu_i u_i}(\mu_i u_i)^{y_i}}{y_i!}$$

The unconditional distribution is

$$\begin{aligned} f(y_i|x_i) &= \int_0^\infty \frac{e^{-\mu_i u_i}(\mu_i u_i)^{y_i}}{y_i!} \frac{\nu^\nu}{\Gamma(\nu)} {u_i}^{\nu-1} e^{-\nu u_i} du_i \\ &= \frac{\mu_i^{y_i}}{\Gamma(y_i+1)} \frac{\nu^\nu}{\Gamma(\nu)} \int_0^\infty e^{-(\mu_i+\nu)u_i} u_i^{(y_i+\nu)-1} du_i \\ &= \frac{\mu_i^{y_i}}{\Gamma(y_i+1)} \frac{\nu^\nu}{\Gamma(\nu)} \frac{\Gamma(y_i+\nu)}{(\mu_i+\nu)^{y_i+\nu}} \\ &= \frac{\Gamma(y_i+\nu)}{\Gamma(y_i+1)\Gamma(\nu)} \left(\frac{\mu_i}{\mu_i+\nu}\right)^{y_i} \left(\frac{\nu}{\mu_i+\nu}\right)^{\nu} \\ &= \frac{\Gamma(y_i+\nu)}{\Gamma(y_i+1)\Gamma(\nu)} \left(1 - \frac{1}{\mu_i/\nu+1}\right)^{y_i} \left(\frac{1}{\mu_i/\nu+1}\right)^{\nu} \end{aligned}$$

Let dispersion paramter $\alpha = 1/\nu$

$$f(y_i|x_i) = \frac{\Gamma(y_i+1/\alpha)}{\Gamma(y_i+1)\Gamma(1/\alpha)} \left(1 - \frac{1}{\alpha\mu_i+1}\right)^{y_i} \left(\frac{1}{\alpha\mu_i+1}\right)^{1/\alpha}$$

The moments of this distribution is given by

$$\begin{aligned} E(Y_i) &= \mu_i \\ Var(Y_i) &= \mu_i + \alpha\mu_i^2 \end{aligned}$$

Therefore, the over-dispersion is given by $1 + \alpha\mu_i$. In exponential family notation, we have

$$\begin{aligned} f(y_i; \mu_i, \alpha) &= \exp\{y_i \log\left(\frac{\alpha\mu_i}{1+\alpha\mu_i}\right) + \frac{1}{\alpha}\log\left(\frac{1}{1+\alpha\mu_i}\right) + \log\Gamma(y_i + 1/\alpha) \\ &\quad - \log\Gamma(y_i+1) - \log\Gamma(1/\alpha)\} \end{aligned}$$

And the full log likelihood function is given by

$$\begin{aligned} \ell(\mu; y, \alpha) &= \sum_{i=1}^{n}\{y_i \log(\alpha\exp(\mathbf{x_i}\beta)) - (y_i + \frac{1}{\alpha})\log(1 + \alpha\exp(\mathbf{x_i}\beta)) \\ &\quad + \log\Gamma(y_i + 1/\alpha) - \log\Gamma(y_i+1) - \log\Gamma(1/\alpha)\} \end{aligned} \tag{1.11}$$

1.4 Nonparametric regression

Suppose we have n observations $(x_1, y_1), \ldots, (x_n, y_n)$ which follows the model

$$y_i = m(x_i) + \epsilon_i$$

where ϵ_i is a random error with $E(\epsilon_i|x_i) = 0$ and $Var(\epsilon_i|x_i) = \sigma^2$(homoscedasticity). It is not necessarily a normal random variable.

Since we do not have any information about how x and y are related, we may want to estimate $m(x)$ without specifying a form of $m(x)$.

1.4.1 Kernel regression estimators

One common used method to estimate $m(x)$ is Kernel regression estimator in the following form.

$$\hat{m}_h(x) = \sum_{i=1}^{n} \int_{s_{i-1}}^{s_i} K_h(u-x)duY_i$$
$$\text{where } K_h(\cdot) = \frac{1}{h}K(\frac{\cdot}{h})$$

with $s_i = (X_{(i)} + X_{(i+1)})/2$, $X_{(0)} = -\infty$ and $X_{(n+1)} = +\infty$. We call it Gasser and Muller (GM)-estimator [25]. In order to get a close form of the first and second derivatives of $\hat{m}(x)$, we choose Gaussian kernel as kernel function.

$$K(t) = \frac{1}{\sqrt{2\pi}} exp(-\frac{t^2}{2})$$

So the GM-estimator can be written as

$$\hat{m}_h(x) = \frac{1}{\sqrt{2\pi}h} \sum_{i=1}^{n} \int_{s_{i-1}}^{s_i} exp[-\frac{(u-x)^2}{2h^2}]duY_i \tag{1.12}$$

with $s_i = (X_{(i)} + X_{(i+1)})/2$, $X_{(0)} = -\infty$ and $X_{(n+1)} = +\infty$. where the bandwidth, h, is usually estimated using least-squares cross-validation[26]. Least-squares cross-validation minimizes the following criterion function

$$LSCV(h) = \frac{1}{n} \sum_{i=1}^{n} (\hat{m}_{h,-i}(x_i) - Y_i)^2$$

where $\hat{m}_{h,-i}(x_i)$ denote the leave-one-out estimators with the i-th point dropped.

1.5 Multiple testing correction

When we conduct a set of hypothesis testing simultaneously, we need to adjust the significant level appropriately in order to control family wise type I error rate(FWER). In other words, multiple testing correction is to adjust the significance level α_j being more stringent for j-th test, such that the FWER is controlled at a desired level α_{FWER}.

1.5.1 Multiple testing correction by controlling family wise error rate

Suppose all the hypothesis tests are independent of each other, we can adjust the significance level directly using Bonferroni correction or Sidak correction. With Bonferroni correction, the adjusted significance level for n hypothesis testings is

$$\alpha_{adj} = \frac{\alpha_{FWER}}{n}$$

With Sidak correction, the adjusted significance level for n hypothesis testings is

$$\alpha_{adj} = 1 - (1 - \alpha_{FWER})^{\frac{1}{n}}$$

However, both methods controlling FWER can be too conservative for large-scale testing problems, such as ChIP-Seq data.

1.5.2 Multiple testing correction by controlling false discovery rate

If the testing results are viewed as exploratory and can be re-tested using another independent study, control of false discovery rate(FDR) is preferred. A false discovery rate (FDR) is defined as the expected rate that significant features are truly null, in my case, non-binding status. For example, controlling FDR at 5% means among all features called significant, 5% of these are allowed to be truly non-binding on average. Comparing with the classical approach controlling the FWER in a strong sense, the approach controlling FDR can be viewed as controlling FWER in weak sense [15]. And FDR is a sensible measure of the balance between the number of true positives and false positives in many genome-wide studies [27]. FDR is defined as the expected value of the proportion of number of false positives among all of those called significant as shown below:

$$FDR = E(\frac{\#FalsePositives}{\#Positives})$$

Usually, the FDR is difficult to calculate in practice. So the following calculation is used as an approximation of FDR when the total number of testing is large[27].

$$FDR = \frac{E(\#FalsePositives)}{\#Positives}$$

The Benjamini-Hochberg-Yekutieli procedure [15] is widely used in practice to control the false discovery rate. If the tests are independent or positively correlated, one can find the largest index k from a non-decreasing ordered p-values $P_{(1)} \leq P_{(2)} \leq \cdots \leq P_{(m)}$ from m multiple testing, such that the following inequality holds:

$$P_{(k)} \leq \frac{k}{m}\alpha$$

And the top k hypotheses with the smallest p-values are rejected. It guarantees FDR controlled at level α.

When the number of testing is large, FDR can further be approximated as

$$F\hat{D}R = \frac{m \times P_{(k)}}{k}$$

In this dissertation, this formula is used to estimated FDR from multiple tests on significance of profile coefficient β_1 and likelihood ratio test, since the number of tests is considered large scale.

Chapter 2

Datasets and Methods

2.1 ChIP-Seq datasets

The following real ChIP-Seq data with or without negative control samples are used in Chapter 4 and Chapter 5 to examine the performance of GLMNB and compare with other algorithms. FoxA1 ChIP-Seq data were obtained from Zhang et al [1]. There are two samples available, ChIP sample and negative control sample, with about 3.9 million and 5.2 million uniquely mapped tags, respectively. The ChIP libraries were prepared using PCR pre-amplification step and size selection for DNA fragments between 150 and 400 bp. All tags are of length 36 bp. FoxA1 was reported to bind FoxA2 motif with conservative DNA sequence (CYTGTTACWYW), FoxA1 motif (WAAGTAAACA) and Foxo1 motif (CTGTTAC)[28, 29]. Here Y stands for C/T and W stands for A/T.

Growth-associated binding protein(GABP) ChIP sample, neuron-restrictive silencer factor(NRSF) with monoclonal antibody ChIP sample and NRSF with polyclonal antibody ChIP sample were obtained from Valouev et al [12]. The GABP ChIP sample contains 7.9 million uniquely mapped tags. GABP was reported to bind/interact with GABPA motif with a conservative DNA sequence of (RACCGGAAGT), where R stands for A/G.

NRSF ChIP samples with monoclonal antibody and polyclonal antibody contain 5.4 million and 8.8 million uniquely mapped tags, respectively. NRSF was reported to bind/interact with REST(NRSF) motif with a conservative DNA sequence of (GGMGCTGTCCATGGT-GCTGA) [30].

Ets variant gene 1(ETV1) ChIP data were obtained from Abe et al[31] with GEO numbers GSM558678 and GSM558677. There are a ChIP sample and a negative control sample, with about 10.7 million and 15.0 million uniquely mapped tags, respectively. The conservative DNA motif sequence is (G/C/A)GGA)(A/T)(G/A).

Epstein-Barr virus nuclear antigen 2 (EBNA2) ChIP data and Recombining binding protein suppressor of hairless(RBPJ) ChIP data were obtained from Zhao et al[32]. There are a ChIP sample and a negative control sample for EBNA2 data, with about 6.6 million and 7.0 million uniquely mapped tags with GEO numbers GSM729852 and GSM729855, respectively. There are two biological replicates of ChIP samples and a negative control sample of RBPJ data, with about 7.6 million, 8.9 million and 7.0 million uniquely mapped tags and GEO numbers GSM729853, GSM729854 and GSM729855, respectively. The conservative DNA motif sequences for EBNA2 and RBPJ include EBF(DGTCCCYRGGGA), RUNX(AAACCACARM), ETS (ACAGGAAGTG), NFκB (WGGGGATTTCCC) and PU.1(MGGAAGTG motifs[32, 33], where D stands for A/G/T.

GATA1 multiple time point ChIP data are obtained from Professor Ross Hardison's lab. There are one ChIP sample and one negative control sample for each time point of vehicle cells, 0hr, 3hr, 14hr, 24hr and 30hr. It is a good practical dataset for testing GLMNB in multiple track settings.

2.2 Simulated ChIP sample and negative control sample

Simulated ChIP sample and negative control sample are generated as described below and used to evaluate peak calling performance in Chapter 3. The background ChIP-seq data are simulated from a negative binomial distribution with size parameter 0.0042 and the probability of success parameter 0.57, which roughly correspond to 9 million tags mapped to the whole genome on each strand. We further simulated 500 ChIP-Seq peak positions randomly with at least 5 kb apart between any two. The tags from these 500 peaks are simulated under another negative binomial distribution. The average tag counts are given by the FoxA1 profile shown in Figure 2.1 multiplied by a simulated peak strength, 2^γ, where γ is defined as a signal fold change relative to the profile. The probability of success parameter of the negative binomial distribution is $p = 0.1$. The fold change parameter γ is generated from a standard normal distribution. Most values of 2^γ lie in $[0.125, 8]$, which is slightly wider than the estimated coefficients, $\hat{\beta}_1$, from the FoxA1 peaks. The peak shift for each simulated peak region is also generated from a normal distribution with mean 100 bp and standard deviation 20 bp. Finally we merged the simulated background data into the simulated ChIP-Seq peak data by adding the tag counts in both data together at each genome coordinate. To examine the capability of calling peaks when there are strong signals in both ChIP sample and negative control sample, 95 peak positions are randomly selected from 500 ones. Tags near these 95 positions combined with previously independently generated background tags are treated as a negative control sample. In other words, these randomly selected peaks are considered as non-binding positions, whose strong signals are due to non-specific binding events. Any predicted peaks on these 95 positions are classified as false positives.

2.3 Negative binomial GLM with shifting parameters

We propose this negative binomial generalized linear model(GLMNB) with shifting parameters to call peaks in ChIP-Seq data. This model is built based on generalized linear model described in section 1.3. Before discussing the model, we have the following assumptions:

- Around a predicted binding site, there are both forward(F) and reverse(R) peaks with adequate and comparable tag counts and similar shapes;
- Forward strand peak is on the upstream(left) of binding site and reverse strand peak is on the downstream(right) of binding site, the same distance apart;
- There are common binding profiles/shapes among binding sites along the entire genome for both strands.

There are six components that will be discussed in the following sections.

1. Binding profiles for forward and reverse strands respectively;
2. Sliding overlapping windows;
3. Shifting parameter from the assumed protein binding site to the forward (reverse) peaks;
4. Negative binomial generalized linear model in general;
5. Negative binomial generalized linear model with ChIP sample only;
6. Negative binomial generalized linear model with negative control.

2.3.1 Binding profiles

We want to first construct binding profiles based on the ChIP sample data. We first use a non-overlapping window to accumulate tag counts from high confidence region across the whole genome. The window of size winsize=1,000 bp scanning through the whole genome, and collect all windows that contain at least $profile_{wincount}$=50 forward (reverse) tags, respectively. The reason to use 1,000 bp rather than the same window size as peak calling is that we need a wider window that allows profiles to shift to both sides from the center. After collecting all such windows, we center the windows at the peak positions of forward (reverse) tags. Second, we accumulate tags from all centered windows, and record the average of F/R tags at every base pair. Here, we treat relative coordinates as $X_j, X_j = -500, -499, -498, \ldots, 499, 500$ and the average counts as Y_j. Remember the average counts Y_j is not differentiable yet, and therefore the first and second derivatives do not exist. Finally, we can get a smooth curve $\hat{m}(x)$ using kernel regression estimator with Gaussian kernel mentioned in section 1.4.1. Here x is the relative coordinate in the window, with values between -500 and 500. The bandwidth parameter h for kernel regression is chosen by least square cross validation using non-parametric(np) package in R. Such bandwidth parameter h minimizes the sum of square difference between the kernel estimated average count in each bin and observed average counts using cross validation.

Based on the GM estimator derived from equation (1.12), we can also calculated the first and second derivatives for $\hat{m}_h(x)$,

$$\begin{aligned}
\hat{m}_h(x) &= \frac{1}{\sqrt{2\pi}h} \sum_{i=1}^{n} \int_{s_{i-1}}^{s_j} exp[-\frac{(u-x)^2}{2h^2}] du Y_j && (2.1)\\
\hat{m}_h'(x) &= \frac{1}{\sqrt{2\pi}h^3} \sum_{i=1}^{n} \int_{s_{i-1}}^{s_j} exp[-\frac{(u-x)^2}{2h^2}](u-x) du Y_j && (2.2)\\
\hat{m}_h''(x) &= \frac{1}{\sqrt{2\pi}h^3} \sum_{i=1}^{n} \int_{s_{i-1}}^{s_j} exp[-\frac{(u-x)^2}{2h^2}] \left(\frac{(u-x)^2}{h^2} - 1\right) du Y_j && (2.3)
\end{aligned}$$

with $s_j = (X_{(j)} + X_{(j+1)})/2$, $X_{(0)} = -\infty$ and $X_{(n+1)} = +\infty$.

As a result, we obtain a binding profile for each strand, representing the smoothed and double differentiable shapes of real binding peaks. Figure 2.1 illustrates binding profiles at each relative coordinate in a window before(vertical bars) and after(curves) smoothing for forward and reverse strands. These binding profiles are generated from FoxA1 ChIP sample data. Figure 2.2 shows the first derivative of binding profiles for forward (red) and reverse (green) strands.

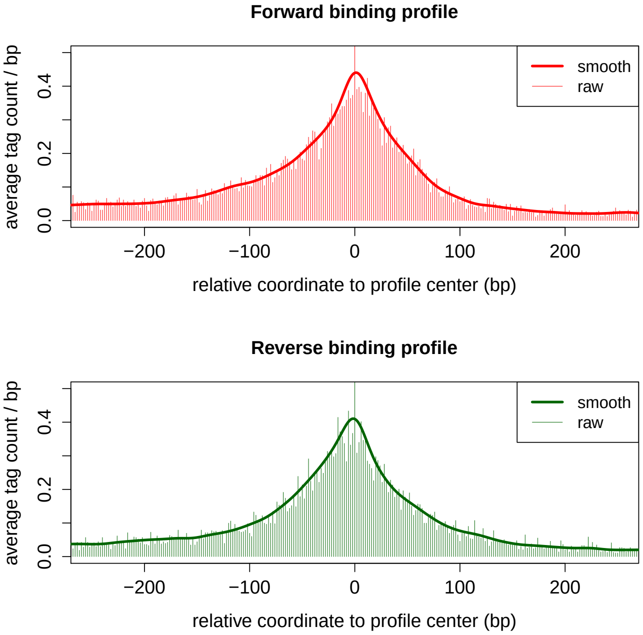
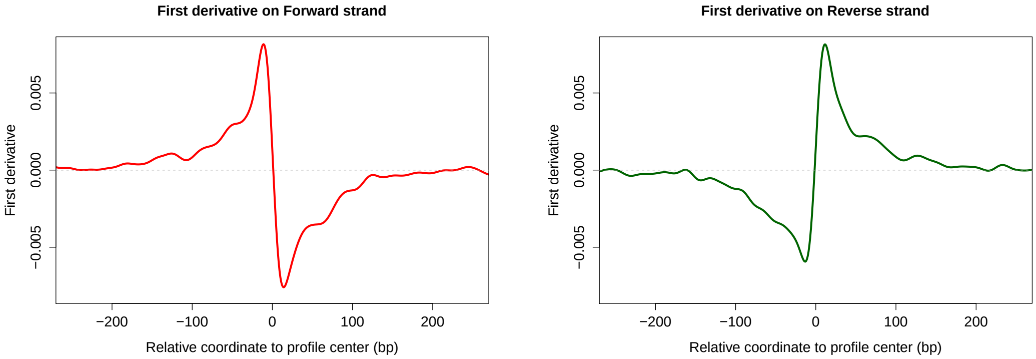


Figure 2.1. Smoothed binding profiles constructed from FoxA1 ChIP-Seq data. Smoothed (curve) and raw (vertical bars) forward and reverse binding profiles are estimated from FoxA1 ChIP-Seq data, shown in red and green, respectively.

2.3.2 Sliding window

We use the sliding window strategy to scan the genome and call peaks. The size of sliding windows is specified by users according to the selection size of DNA fragments in the ChIP-Seq experiment and the protein protected DNA size. A sliding window width should be wide enough to observe all the shifted forward and reverse tags around the binding site. It yet cannot be too wide since including a wide background region will reduce predicted peak significance. Since DNA fragments between 150 bp and 400 bp were selected in most ChIP-



(a) (b)

Figure 2.2. The first derivative plots of non-shifted binding profiles for forward(a) and reverse(b) strands. Adaptive window size is chosen such that the first derivative of forward and reverse profile is not zero.

Seq data and the protein protected DNA size is around 100 bp, we consider winsize=500 bp as appropriate default size of sliding windows. The user can choose a more appropriate window size if desired.

The step size of sliding windows should also be considered as a trade-off between calculation efficiency and peak spatial resolution. Even through we will have high resolution by using a small step size, a lot of unnecessary computing time will be spent on trying to call an identical peak among several sliding windows. If the step size is too large, however, the peak calling resolution is reduced and there is a good chance that some true peaks will be missed between two nearby sliding windows. Stepsize=10 bp is chosen as the default step size to get high resolution while maintaining good computing speed. Users have the flexibility to elect their own step size to achieve the balance between spatial resolution and computing time.

After determining winsize and step size, each sliding window is divided into n bins of binsize=10 bp. The number of forward(F) and reverse(R) tags, whose first nucleotide falls in bins, are recorded. The count data within each bin from ChIP sample are named the observed tag counts by bins and denoted as $y_i^F, i = 1, \dots, n$ for forward strands and $y_i^R, i = 1, \dots, n$ for reverse strands, respectively. Here y_i^F and y_i^R denote observed tag count in $i-$th bin in a sliding window from forward and reverse strands, respectively. The tag counts from both strands were merged into a tag count vector. This procedure is repeated for each sliding window.

$$\vec{y} = y^S = \left(y_1^F, \ldots, y_n^F, y_1^R, \ldots, y_n^R\right)^T \tag{2.4}$$

where y^S denotes the observed tag count vector from ChIP sample data. In a ChIP sample only dataset, for example, we have $n = 50$ and an observed tag count vector of length 100 for each sliding window. If a negative control sample is available, an observed tag count vector at the same sliding window from the negative control data are collected and attached to the observed tag count vector from the ChIP sample data, y as follows.

$$\vec{y} = \begin{pmatrix} y^S \\ y^C \end{pmatrix} \tag{2.5}$$

$$y^C = \left(z_1^F, \ldots, z_n^F, z_1^R, \ldots, z_n^R\right)^T \tag{2.6}$$

where y^C denotes the observed tag count vector from negative control data, and $z_i^F, i = 1, \ldots, n$ and $z_i^R, i = 1, \ldots, n$ are tag counts per bin from negative control data.

In more general, assume that there are k_j $(j = 1, \ldots, c)$ ChIP sample replicates and one negative control sample under each of c biological conditions. We can construct an observed tag count vector $\mathbf{y}$ for each sliding window as following,

$$\mathbf{y} = \begin{pmatrix} \vec{y}_1 \\ \vec{y}_2 \\ \ldots \\ \vec{y}_c \end{pmatrix} \tag{2.7}$$

$$
\begin{aligned}
\vec{y}_j &= \begin{pmatrix} y^S_{j,1} \\ y^S_{j,2} \\ \ldots \\ y^S_{j,k_j} \\ y^C_j \end{pmatrix} \\
\vec{y}^S_{j,j\prime} &= \left(y^F_1, \ldots, y^F_n, y^R_1, \ldots, y^R_n\right)^T_{j,j\prime}, j\prime = 1, \ldots, k_j && (2.8) \\
y^C_j &= \left(z^F_1, \ldots, z^F_n, z^R_1, \ldots, z^R_n\right)^T_j && (2.9)
\end{aligned}
$$

2.3.3 Shifting parameter

Remember in section 2.3.1, we have discussed how to generate the binding profile from high confidence regions in ChIP sample data. However, there is a peak shifting distance from binding position to its upstream peak formed by forward tags and its downstream peak formed by reverse tags. And such a distance varies by genomic locations. In order to account for such variability and combine the information from forward and reverse strands effectively, we add in a shifting parameter, θ, which represents such a peak shift. This parameter measures the distance between forward/reverse strand peaks in two ends and predicted binding sites in the middle. Since we assume the forward and reverse strand departs the same distance from binding sites, the forward and reverse peaks are at the genome coordinates $(x - \theta)$ and $(x + \theta)$, where x is the predicted binding site.

As shown in Figure 2.3, suppose that we have a binding site at the center of a sliding window. In order to fit the observed forward and reverse tags best, we need to shift the profile of the forward strand (red) to the left and the profile of the reversed strand (green) to the right by θ, for example 50 bp in the figure. Since the shifting parameter θ is a model parameter, we can estimate its maximum likelihood estimator (MLE) along with other model parameters. The detailed method on finding MLE is described in section 2.3.4. It makes sense to have forward and reverse smoothed profile in slightly different heights and shapes, which is taken care of by the profile coefficient, β_1, in the model. That is because forward and reverse strand tags are single end, unpaired data and we often observe a slight difference in shape from different windows. Allowing such a difference can fit the real data better.

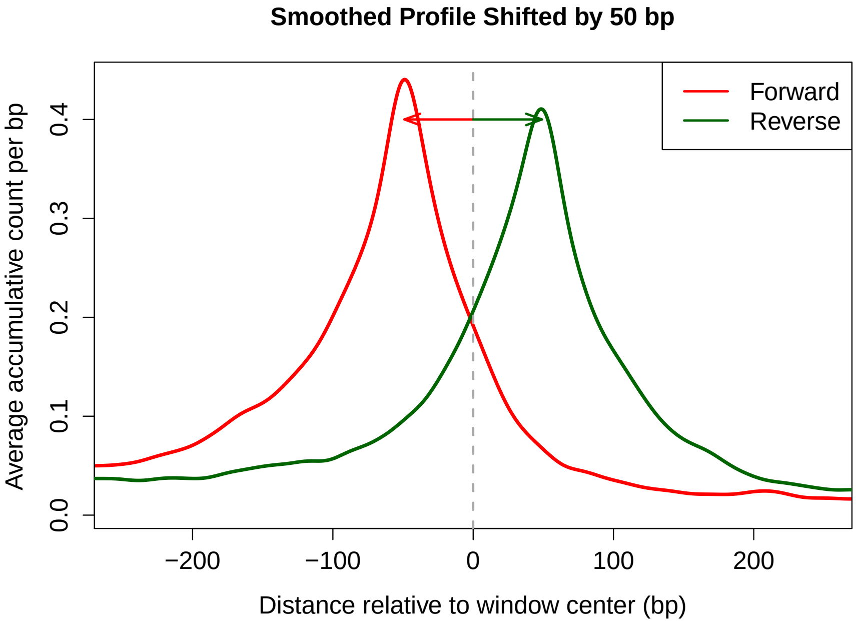


Figure 2.3. Forward(reverse) smoothed Profiles are shifted toward left(right) by a local parameter, θ, in order to fit observed tag counts.

In more details, with a positive shifting parameter, θ the estimated height at any genome position t of the forward strand profile and of the reverse strand profile are

$$\begin{aligned}\hat{m}^F(t+\theta) &= \frac{1}{\sqrt{2\pi}h_F}\sum_{j=1}^{N}\left[\int_{S_{j-1}}^{S_j} exp(-\frac{1}{2}\left(\frac{u-t-\theta}{h_F}\right)^2)du\right]Y_j^F \\ \hat{m}^R(t-\theta) &= \frac{1}{\sqrt{2\pi}h_R}\sum_{j=1}^{N}\left[\int_{S_{j-1}}^{S_j} exp(-\frac{1}{2}\left(\frac{u-t+\theta}{h_R}\right)^2)du\right]Y_j^R\end{aligned}$$

where h_F and h_R are the bandwidths for forward and reverse strand. The bandwidths are estimated to minimize the difference between predicted shape and observed average tag counts using least-squares cross-validation. It is implemented using R function (npregbw) in the non-parametric library, np.

So for the i-th bin in a sliding window, the expected read tag counts in a bin centered at position dm_i after shifting are

$$\begin{aligned} x_i^F &= \int_{c_i}^{c_{i+1}} \hat{m}^F(t+\theta)dt \simeq \hat{m}^F(\frac{c_i+c_{i+1}}{2}+\theta)\times \text{binsize} \\ &\doteq \hat{m}^F(dm_i+\theta)\times \text{binsize} \qquad (2.10)\\ x_i^R &= \int_{c_i}^{c_{i+1}} \hat{m}^R(t-\theta)dt \simeq \hat{m}^R(\frac{c_i+c_{i+1}}{2}-\theta)\times \text{binsize} \\ &\doteq \hat{m}_R(dm_i-\theta)\times \text{binsize} \qquad (2.11)\\ x_i &\doteq (x_i^F, x_i^R) = (\hat{m}^F(dm_i+\theta), \hat{m}^R(dm_i-\theta))\times \text{binsize} \qquad (2.12) \end{aligned}$$

where c_i is the left boundary position of bins with equal binsize. and dm_i is the middle point of bins. Then based on equation (2.2) the first derivative of x_i can be written as

$$\begin{aligned} \frac{\partial x_i}{\partial\theta} &= (\frac{\partial}{\partial\theta}\hat{m}^F(dm_i+\theta), \frac{\partial}{\partial\theta}\hat{m}^R(dm_i-\theta))\times \text{binsize} \\ \frac{\partial}{\partial\theta}\hat{m}^F(dm_i+\theta) &= \frac{1}{h^3\sqrt{2\pi}}\sum_{j=1}^{N}\left[\int_{S_{j-1}}^{S_j} exp(-\frac{1}{2}(\frac{u-dm_i-\theta}{h})^2)(u-dm_i-\theta)du\right]Y_j^F \\ \frac{\partial}{\partial\theta}\hat{m}^R(dm_i-\theta) &= \frac{-1}{h^3\sqrt{2\pi}}\sum_{j=1}^{N}\left[\int_{S_{j-1}}^{S_j} exp(-\frac{1}{2}(\frac{u-dm_i+\theta}{h})^2)(u-dm_i+\theta)du\right]Y_j^R \end{aligned}$$

Now we can also derive the second derivative of x_i from equation (2.3) as follows:

$$\begin{aligned} \frac{\partial^2 x_i}{\partial\theta^2} &= (\frac{\partial^2}{\partial\theta^2}\hat{m}^F(dm_i+\theta), \frac{\partial^2}{\partial\theta^2}\hat{m}^R(dm_i-\theta))\times \text{binsize} \\ \frac{\partial^2}{\partial\theta^2}\hat{m}^F(dm_i+\theta) &= \frac{1}{h^3\sqrt{2\pi}}\sum_{j=1}^{N}\left[\int_{S_{j-1}}^{S_j} exp(-\frac{1}{2}(\frac{u-dm_i-\theta}{h})^2)(\frac{1}{h^2}(u-dm_i-\theta)^2+1)du\right]Y_j^F \end{aligned}$$

$$\frac{\partial^2}{\partial\theta^2}\hat{m}^R(dm_i-\theta) = \frac{1}{h^3\sqrt{2\pi}}\sum_{j=1}^{N}\left[\int_{S_{j-1}}^{S_j} exp(-\frac{1}{2}(\frac{u-dm_i+\theta}{h})^2)(\frac{1}{h^2}(u-dm_i+\theta)^2-1)du\right]Y_j^R$$

From negative control sample, a pair of forward and reverse pseudo binding profiles are generated using the same procedure above, denoted as $\hat{m}_C^F(t+\theta)$ and $\hat{m}_C^R(t-\theta)$, where subscript C denotes that it comes from the negative control data. The term "pseudo" marks that it is false positive signals from negative control data. The pseudo estimated tag counts from negative control data are denoted as x_i^C by plugging $\hat{m}_C^F(t+\theta)$ and $\hat{m}_C^R(t-\theta)$ into equation (2.12).

2.3.4 Negative binomial GLM for general cases

We use negative binomial distribution with a variate overdispersion parameter (NB-2) generalized linear model to fit the relationship between observed count data and binding profiles as well as binding profiles from negative control data. That is the Poisson model with gamma heterogeneity where gamma noise has a mean of 1. As described in section 1.3.5.2, the expected tag count, μ, after logarithm transformation with base e is linked to the linear combination of binding profile from ChIP data, $\mathbf{X}$, binding profile from negative control data, $\mathbf{z}$, and baseline average tag counts, β_0.

$$\begin{aligned}
\log\mu &= \mathbf{X}\beta = \beta_0 + \sum_{j=1}^{c}\left(\sum_{j\prime=1}^{k_j}(\beta_{1,j}\vec{x}_{j,j\prime}) + \beta_{2,j}\vec{z_j}\right)\\
\mu &= E(\vec{y})\\
\vec{x}_{j,j\prime} &= ((\vec{0})^T,\ldots,(\vec{0})^T,\ldots,(\vec{0})^T,\ldots,(x_{j,j\prime}^S)^T,\ldots,(\vec{0})^T,\ldots,(\vec{0})^T,\ldots,(\vec{0})^T)^T\\
\vec{z}_j &= ((x_1^C)^T,\ldots,(x_1^C)^T,\ldots,(x_j^C)^T,\ldots,(x_j^C)^T,\ldots,(x_j^C)^T,\ldots,(x_c^C)^T,\ldots,(x_c^C)^T)^T
\end{aligned}$$

Where $x^S_{j,j\prime}(j\prime = 1, \ldots, k_j)$ is the smoothed profile vector from the $j\prime$-th ChIP sample under the j-th biological condition. x^C_j is the smoothed pseudo profile vector from negative control sample under j-th biological condition. μ_i is expected tag counts, $u_i = exp(\epsilon_i)$ is a gamma distributed noise with mean 1. λ_i is the linear combination of x. $\beta_{1,j}$ is the common coefficient for smoothed signal profile $x^S_{j,j\prime}$ in all replicates under j-th condition. $\beta_{2,j}$ is the coefficient for smoothed pseudo signal profile x^C_j under j-th condition. And β_0 is the baseline parameter. Please note that this is a general model for three scenarios, 1) one ChIP sample only, 2) one ChIP sample and one negative control, and 3) multiple tracks, including k_j $(j = 1, \ldots, c)$ ChIP samples under c biological conditions and one negative control sample. I will discuss the first two scenarios in the following sub-sessions in more details and the third scenario in Chapter 5.

2.3.5 Negative binomial GLMNB with ChIP sample only

Here we discuss the first scenario mentioned in section 2.3.4, when there is only one ChIP sample data. This is the simplest scenario, where $c = 1$ and $k = 1$ and there is no negative control sample. The model written in equation (2.13) can be re-written as follows.

$$\begin{aligned} \log \mu &= \beta_0 + \beta_1 \vec{x} \\ \mu &= E(\mathbf{y}) \\ \vec{x} &= x(\theta)^S \end{aligned}$$

where $\mathbf{y}$ is the one from equation (2.4), β_0 is the log of baseline average tag count from non-specific events, θ is the peak shifting parameter and β_1 is the coefficient of binding profile.

To test whether a binding event from this ChIP sample only, the following hypothesis is tested against a normal distribution.

$$\begin{aligned} H_0 : \beta_1 &= 0 \\ H_A : \beta_1 &> 0 \end{aligned} \tag{2.13}$$

According to Hardin and Hilbe [24],

$$\hat{\beta}_1 \xrightarrow{D} N(\beta_1, Var(\hat{\beta}_1)) \tag{2.14}$$

where standard error of $\hat{\beta}_1$ is estimated from square root of the first diagonal element of expected Hessian matrix shown in equation (1.6). Therefore, Wald test is used to examine the significance of β_1 using the asymptotic normal distribution.

An alternative approach is the likelihood ratio test(LRT). In our constant β_0 strategy, the log of baseline average tag count, β_0, is fixed at the initial values, either estimated from the current window or a neighborhood using adjusted baseline strategy (Refer to section 2.3.8 for more details). There is one variables, α, in the null model. There are three variables, β_1, θ, and α in the alternative model. One can calculate the log of the likelihood ratio between the alternative model and null model as $\Delta\ell = \ell_1 - \ell_0$. Then we have the following asymptotic χ^2 distribution in the non-binding region.

$$2 \times \Delta\ell \xrightarrow{D} \chi^2_{df} \tag{2.15}$$

where the degrees of freedom is estimated as the sample median of $2 \times \Delta\ell$ from non-binding regions, in practice those regions with small amount of tags (for example, ≥ 5 on both strands but ≤ 8 in either strands) per 500 bp window. Even though θ parameter violates the regularity conditions for χ^2 distribution, with a corrected degrees of freedom, one can still use LRT to test the significance of TF binding event[34].

The log likelihood function is as follows:

$$\begin{aligned}\ell(\boldsymbol{\mu};\boldsymbol{y},\alpha) &= \sum_{i=1}^{2n}\{y_i\log(\alpha\exp(x_i\beta_1+\beta_0)) - (y_i+\frac{1}{\alpha})\log(1+\alpha\exp(x_i\beta_1+\beta_0)) \\ &\quad +\log\Gamma(y_i+1/\alpha)-\log\Gamma(y_i+1)-\log\Gamma(1/\alpha)\}\end{aligned} \tag{2.16}$$

There are three parameters need to be estimated using maximum likelihood method for ChIP sample only.

$$\boldsymbol{\omega} = (\beta_1, \theta, \alpha)$$

where β_1 is the coefficient for smoothed binding profile x_i, θ is the shifting parameter discussed in section 2.3.3 and α is the dispersion parameter discussed in section 1.3.5.2.

Then the gradient is

$$\begin{aligned}\frac{\partial\ell}{\partial\boldsymbol{\omega}} &= (\frac{\partial\ell}{\partial\beta_1},\frac{\partial\ell}{\partial\theta},\frac{\partial\ell}{\partial\alpha})^T \\ \frac{\partial\ell}{\partial\beta_1} &= \sum_{i=1}^{2n}\left\{y_ix_i-(y_i+\frac{1}{\alpha})x_i\frac{\alpha exp(x_i\beta_1+\beta_0)}{1+\alpha exp(x_i\beta_1+\beta_0)}\right\} \\ \frac{\partial\ell}{\partial\theta} &= \sum_{i=1}^{2n}\left\{y_i\left(\frac{\partial x_i}{\partial\theta}\right)\beta_1-(y_i+\frac{1}{\alpha})\frac{\alpha exp(x_i\beta_1+\beta_0)}{1+\alpha exp(x_i\beta_1+\beta_0)}\left(\frac{\partial x_i}{\partial\theta}\right)\beta_1\right\} \\ \frac{\partial\ell}{\partial\alpha} &= \sum_{i=1}^{2n}\{\frac{y_i}{\alpha}+\frac{1}{\alpha^2}log(1+\alpha exp(x_i\beta_1+\beta_0)) \\ &\quad -(y_i+\frac{1}{\alpha})\frac{exp(x_i\beta_1+\beta_0)}{1+\alpha exp(x_i\beta_1+\beta_0)}+\psi_0(y_i+\frac{1}{\alpha})(-\alpha^{-2})-\psi_0(\frac{1}{\alpha})(-\alpha^{-2})\} \\ \text{where } \psi_0(t) &= \frac{\Gamma'(t)}{\Gamma(t)} \text{ is digamma function}\end{aligned}$$

And the Hessian matrix is written in the following formula:

$$\frac{\partial^2 \ell}{\partial \boldsymbol{\omega}^2} = \begin{pmatrix} \frac{\partial^2 \ell}{\partial \beta_1^2} & \frac{\partial^2 \ell}{\partial \beta_1 \partial \theta} & \frac{\partial^2 \ell}{\partial \beta_1 \partial \alpha} \\ \frac{\partial^2 \ell}{\partial \theta \partial \beta_1} & \frac{\partial^2 \ell}{\partial \theta^2} & \frac{\partial^2 \ell}{\partial \theta \partial \alpha} \\ \frac{\partial^2 \ell}{\partial \alpha \partial \beta_1} & \frac{\partial^2 \ell}{\partial \alpha \partial \theta} & \frac{\partial^2 \ell}{\partial \alpha^2} \end{pmatrix} \tag{2.17}$$

where

$$\begin{aligned}
\frac{\partial^2 \ell}{\partial \beta_1^2} &= \sum_{i=1}^{2n} \left(-(y_i + \frac{1}{\alpha}) x_i \frac{\alpha exp(x_i\beta_1 + \beta_0) x_i}{(1 + \alpha exp(x_i\beta_1 + \beta_0))^2} \right) \\
\frac{\partial^2 \ell}{\partial \beta_1 \partial \theta} &= \sum_{i=1}^{2n} \left(y_i \frac{\partial x_i}{\partial \theta} - (y_i + \frac{1}{\alpha}) \frac{\partial x_i}{\partial \theta} \frac{\alpha exp(x_i\beta_1 + \beta_0)}{1 + \alpha exp(x_i\beta_1 + \beta_0)} (1 + \frac{x_i \beta_1}{1 + \alpha exp(x_i\beta_1 + \beta_0)}) \right) \\
\frac{\partial^2 \ell}{\partial \beta_1 \partial \alpha} &= \sum_{i=1}^{2n} \left(-x_i \frac{exp(x_i\beta_1 + \beta_0)}{1 + \alpha exp(x_i\beta_1 + \beta_0)} (-\frac{1}{\alpha} + \frac{y_i + \frac{1}{\alpha}}{1 + \alpha exp(x_i\beta_1 + \beta_0)}) \right) \\
\frac{\partial^2 \ell}{\partial \theta^2} &= \sum_{i=1}^{2n} (y_i \beta_1 \frac{\partial^2 x_i}{\partial \theta^2} - (y_i + \frac{1}{\alpha}) \beta_1 (\frac{\alpha exp(x_i\beta_1 + \beta_0) \beta_1 (\frac{\partial x_i}{\partial \theta})^2}{[1 + \alpha exp(x_i\beta_1 + \beta_0)]^2} \\
&+ \frac{\alpha exp(x_i\beta_1 + \beta_0)}{1 + \alpha exp(x_i\beta_1 + \beta_0)} \frac{\partial^2 x_i}{\partial \theta^2})) \\
\frac{\partial^2 \ell}{\partial \theta \partial \alpha} &= \sum_{i=1}^{2n} \left(\frac{exp(x_i\beta_1 + \beta_0) \beta_1 \frac{\partial x_i}{\partial \theta}}{\alpha(1 + \alpha exp(x_i\beta_1 + \beta_0))} - \frac{(y_i + \frac{1}{\alpha}) exp(x_i\beta_1 + \beta_0) \beta_1 \frac{\partial x_i}{\partial \theta}}{(1 + \alpha exp(x_i\beta_1 + \beta_0))^2} \right) \\
\frac{\partial^2 \ell}{\partial \alpha^2} &= \sum_{i=1}^{2n} [\frac{-y_i}{\alpha^2} - \frac{2}{\alpha^3} log(1 + \alpha exp(x_i\beta_1 + \beta_0)) + \frac{2 exp(x_i\beta_1 + \beta_0)}{\alpha^2 (1 + \alpha exp(x_i\beta_1 + \beta_0))} \\
& + \frac{(y_i + \frac{1}{\alpha}) exp^2(x_i\beta_1 + \beta_0)}{(1 + \alpha exp(x_i\beta_1 + \beta_0))^2} + \psi_0'(y_i + \frac{1}{\alpha}) \frac{1}{\alpha^4} + \psi_0(y_i + \frac{1}{\alpha}) \frac{2}{\alpha^3} \\
& - \psi_0'(\frac{1}{\alpha}) \frac{1}{\alpha^4} - \psi_0(\frac{1}{\alpha}) \frac{2}{\alpha^3}]
\end{aligned}$$

In section 2.3.2, we have the close form of $\{x_i\}_{i=1}^{2n}$, $\{\frac{\partial x_i}{\partial \theta}\}_{i=1}^{2n}$ and $\{\frac{\partial^2 x_i}{\partial \theta^2}\}_{i=1}^{2n}$. We are also able to get the close form of gradient vector, Hessian matrix and therefore the covariance matrix and standard error of β_1. The Newton-Raphson method is used to obtain the maximum likelihood estimators (MLE) for parameters $\omega = (\beta_1, \theta, \alpha)$. In principle, starting from a random value of $\omega^{(\mathbf{0})}$, we iteratively update ω by

$$\omega^{(\mathbf{r})} = \omega^{(\mathbf{r-1})} - \left\{\ell''(\omega^{(\mathbf{r-1})})\right\}^{-1} \ell'(\omega^{(\mathbf{r-1})}) \tag{2.18}$$

for $r = 1, 2, \ldots$ until convergence, where r is the iteration index.

2.3.6 GLMNB modeling with negative control data

As stated in section 1.1.1, there are positions where strong signals appear in both the negative control and ChIP samples. These positions are considered as false positives, and should be identified by the algorithm automatically. There are also non-specific binding events, where background tags are widely and evenly distributed in a few kb region in both samples because of genome sequence structure, such as GC content. GLMNB still calls peaks in such regions but takes this noisy background into account by automatically increasing the average baseline tag amounts parameter β_0 according to the average tag count in negative control sample.

The observed tag count vectors are constructed as described in equation (2.5) in section 2.3.2. In this case, we have $c = 1$ and $k = 1$ with one negative control sample from equation (2.3.4).

$$\begin{aligned}
\log \mu &= \mathbf{X}\beta = \beta_0 + \beta_1 \vec{x} + \beta_2 \vec{z} \qquad (2.19)\\
\mu &= E(\mathbf{y})\\
\mathbf{y} &= \left((y^S)^T, (y^C)^T\right)^T\\
\vec{x} &= ((x(\theta)^S)^T, (\vec{0})^T)^T\\
\vec{z} &= ((x(\theta)^C)^T, (x(\theta)^C)^T)^T
\end{aligned}$$

where β_0 is the log of estimated average tag count from negative control sample, θ is the peak shifting parameter, β_1 is the coefficient of smoothed signal profile, β_2 is the coefficient of smoothed pseudo signal profile and α is the dispersion parameter from negative binomial model. There are four parameters need to be estimated using maximum likelihood method.

$$\boldsymbol{\omega} = (\beta_1, \theta, \beta_2, \alpha)$$

To test a binding event in the ChIP sample after considering the negative control data, the following hypothesis is tested against normal distribution.

$$\begin{aligned} H_0 : \beta_1 &= 0 \\ H_A : \beta_1 &> 0 \end{aligned} \tag{2.20}$$

Rather than estimating β_0 from ChIP sample, we estimate β_0 as the log of average baseline tag count from negative control sample. Model specified in equation 2.19 enables us to clarify whether the tag count vector observed in ChIP sample y^S is due to TF binding event or background noise displayed in negative control data. If observed tags from ChIP sample y^S follows the same pattern as those from negative control sample y^C, then β_2 will be significantly different from zero but β_1 will not. If observed tags from negative control data y^C show a noisy and widely spread pattern, β_0 is increased. But β_1 will not be significantly different from zero, either. Only when y^S shows a spatial distribution close to signal profile generated from ChIP sample, but not similar to y^C, β_1 will be significantly different from zero and a peak will be called.

GLMNB makes use of Wald test or likelihood ratio test to make such decisions. In the Wald test, we are testing the same hypothesis as shown above in equation 2.20. By default, GLMNB uses likelihood ratio test and test the significance of likelihood ratio between null model and alternative model from 1. In the null model, there are two free parameters, coefficient of pseudo signal profile β_2 and dispersion parameter α. In the alternative model, there are two additional parameters, coefficient of ChIP binding profile β_1 and shifting parameter θ. And two times of log-likelihood ratio is asymptotically χ^2 distributed. If the asymptotic regularity conditions hold, the degree of freedom is two. However, as discussed in section 2.3.5 and 3.1, θ parameter violates regularity conditions. Even though it is still asymptotically χ^2 distributed, the degree of freedom is no longer 2. It is in practice estimated from regions with small amount tags(for example, ≥ 5 on both strands but ≤ 8 in either strands).

The log likelihood function is as follows:

$$\begin{aligned}\ell(\boldsymbol{\mu};\boldsymbol{y},\alpha) &= \sum_{i=1}^{2n}\{y_i \log(\alpha\exp(x_i\beta_1+z_i\beta_2+\beta_0)) - (y_i+\frac{1}{\alpha})\log(1+\alpha\exp(x_i\beta_1+z_i\beta_2+\beta_0)) \\ &\quad +\log\Gamma(y_i+1/\alpha)-\log\Gamma(y_i+1)-\log\Gamma(1/\alpha)\}\end{aligned} \tag{2.21}$$

Then the gradient is

$$\begin{aligned}\frac{\partial\ell}{\partial\boldsymbol{\omega}} &= (\frac{\partial\ell}{\partial\beta_1},\frac{\partial\ell}{\partial\theta},\frac{\partial\ell}{\partial\alpha},\frac{\partial\ell}{\partial\beta_2})^T \\ \frac{\partial\ell}{\partial\beta_1} &= \sum_{i=1}^{2n}\left\{y_ix_i-(y_i+\frac{1}{\alpha})x_i\frac{\alpha exp(x_i\beta_1+z_i\beta_2+\beta_0)}{1+\alpha exp(x_i\beta_1+z_i\beta_2+\beta_0)}\right\} \\ \frac{\partial\ell}{\partial\theta} &= \sum_{i=1}^{2n}\left\{y_i\left(\frac{\partial x_i}{\partial\theta}\right)\beta_1-(y_i+\frac{1}{\alpha})\frac{\alpha exp(x_i\beta_1+z_i\beta_2+\beta_0)}{1+\alpha exp(x_i\beta_1+z_i\beta_2+\beta_0)}\left(\frac{\partial x_i}{\partial\theta}\right)\beta_1\right\} \\ \frac{\partial\ell}{\partial\alpha} &= \sum_{i=1}^{2n}\{\frac{y_i}{\alpha}+\frac{1}{\alpha^2}log(1+\alpha exp(x_i\beta_1+z_i\beta_2+\beta_0)) \\ &\quad -(y_i+\frac{1}{\alpha})\frac{exp(x_i\beta_1+z_i\beta_2+\beta_0)}{1+\alpha exp(x_i\beta_1+z_i\beta_2+\beta_0)}+\psi_0(y_i+\frac{1}{\alpha})(-\alpha^{-2})-\psi_0(\frac{1}{\alpha})(-\alpha^{-2})\} \\ \frac{\partial\ell}{\partial\beta_2} &= \sum_{i=1}^{2n}\left\{y_iz_i-(y_i+\frac{1}{\alpha})z_i\frac{\alpha exp(x_i\beta_1+z_i\beta_2+\beta_0)}{1+\alpha exp(x_i\beta_1+z_i\beta_2+\beta_0)}\right\} \\ \text{where } \psi_0(t) &= \frac{\partial\Gamma(t)}{\partial t}\end{aligned}$$

And the Hessian matrix is written in the following form:

$$\frac{\partial^2 \ell}{\partial \boldsymbol{\omega}^2} = \begin{pmatrix} \frac{\partial^2 \ell}{\partial \beta_1^2} & \frac{\partial^2 \ell}{\partial \beta_1 \partial \theta} & \frac{\partial^2 \ell}{\partial \beta_1 \partial \alpha} & \frac{\partial^2 \ell}{\partial \beta_1 \partial \beta_2} \\ \frac{\partial^2 \ell}{\partial \theta \partial \beta_1} & \frac{\partial^2 \ell}{\partial \theta^2} & \frac{\partial^2 \ell}{\partial \theta \partial \alpha} & \frac{\partial^2 \ell}{\partial \theta \partial \beta_2} \\ \frac{\partial^2 \ell}{\partial \alpha \partial \beta_1} & \frac{\partial^2 \ell}{\partial \alpha \partial \theta} & \frac{\partial^2 \ell}{\partial \alpha^2} & \frac{\partial^2 \ell}{\partial \alpha \partial \beta_2} \\ \frac{\partial^2 \ell}{\partial \beta_2 \partial \beta_1} & \frac{\partial^2 \ell}{\partial \beta_2 \partial \theta} & \frac{\partial^2 \ell}{\partial \beta_2 \partial \alpha} & \frac{\partial^2 \ell}{\partial \beta_2^2} \end{pmatrix} \tag{2.22}$$

where

$$\begin{aligned}
\frac{\partial^2 \ell}{\partial \beta_1^2} &= \sum_{i=1}^{2n} \left(-(y_i + \frac{1}{\alpha}) x_i \frac{\alpha exp(x_i\beta_1 + z_i\beta_2 + \beta_0) x_i}{(1 + \alpha exp(x_i\beta_1 + z_i\beta_2 + \beta_0))^2} \right) \\
\frac{\partial^2 \ell}{\partial \beta_1 \partial \theta} &= \sum_{i=1}^{2n} \left(y_i \frac{\partial x_i}{\partial \theta} - (y_i + \frac{1}{\alpha}) \frac{\partial x_i}{\partial \theta} \frac{\alpha exp(x_i\beta_1 + z_i\beta_2 + \beta_0)}{1 + \alpha exp(x_i\beta_1 + z_i\beta_2 + \beta_0)} (1 + \frac{x_i\beta_1}{1 + \alpha exp(x_i\beta_1 + z_i\beta_2 + \beta_0)}) \right) \\
\frac{\partial^2 \ell}{\partial \beta_1 \partial \alpha} &= \sum_{i=1}^{2n} \left(-x_i \frac{exp(x_i\beta_1 + z_i\beta_2 + \beta_0)}{1 + \alpha exp(x_i\beta_1 + z_i\beta_2 + \beta_0)} (-\frac{1}{\alpha} + \frac{y_i + \frac{1}{\alpha}}{1 + \alpha exp(x_i\beta_1 + z_i\beta_2 + \beta_0)}) \right) \\
\frac{\partial^2 \ell}{\partial \theta^2} &= \sum_{i=1}^{2n} (y_i \beta_1 \frac{\partial^2 x_i}{\partial \theta^2} - (y_i + \frac{1}{\alpha}) \beta_1 (\frac{\alpha exp(x_i\beta_1 + z_i\beta_2 + \beta_0) \beta_1 (\frac{\partial x_i}{\partial \theta})^2}{[1 + \alpha exp(x_i\beta_1 + z_i\beta_2 + \beta_0)]^2} \\
&+ \frac{\alpha exp(x_i\beta_1 + z_i\beta_2 + \beta_0)}{1 + \alpha exp(x_i\beta_1 + z_i\beta_2 + \beta_0)} \frac{\partial^2 x_i}{\partial \theta^2})) \\
\frac{\partial^2 \ell}{\partial \theta \partial \alpha} &= \sum_{i=1}^{2n} \left(\frac{exp(x_i\beta_1 + z_i\beta_2 + \beta_0) \beta_1 \frac{\partial x_i}{\partial \theta}}{\alpha (1 + \alpha exp(x_i\beta_1 + z_i\beta_2 + \beta_0))} - \frac{(y_i + \frac{1}{\alpha}) exp(x_i\beta_1 + z_i\beta_2 + \beta_0) \beta_1 \frac{\partial x_i}{\partial \theta}}{(1 + \alpha exp(x_i\beta_1 + z_i\beta_2 + \beta_0))^2} \right) \\
\frac{\partial^2 \ell}{\partial \alpha^2} &= \sum_{i=1}^{2n} [\frac{-y_i}{\alpha^2} - \frac{2}{\alpha^3} log(1 + \alpha exp(x_i\beta_1 + z_i\beta_2 + \beta_0)) + \frac{2 exp(x_i\beta_1 + z_i\beta_2 + \beta_0)}{\alpha^2 (1 + \alpha exp(x_i\beta_1 + z_i\beta_2 + \beta_0))} \\
&\quad + \frac{(y_i + \frac{1}{\alpha}) exp^2(x_i\beta_1 + z_i\beta_2 + \beta_0)}{(1 + \alpha exp(x_i\beta_1 + z_i\beta_2 + \beta_0))^2} + \psi_0'(y_i + \frac{1}{\alpha}) \frac{1}{\alpha^4} + \psi_0(y_i + \frac{1}{\alpha}) \frac{2}{\alpha^3} \\
&\quad - \psi_0'(\frac{1}{\alpha}) \frac{1}{\alpha^4} - \psi_0(\frac{1}{\alpha}) \frac{2}{\alpha^3}] \\
\frac{\partial^2 \ell}{\partial \beta_1 \partial \beta_2} &= \sum_{i=1}^{2n} \left(-(y_i + \frac{1}{\alpha}) x_i z_i \frac{\alpha exp(x_i\beta_1 + z_i\beta_2 + \beta_0) z_i}{(1 + \alpha exp(x_i\beta_1 + z_i\beta_2 + \beta_0))^2} \right) \\
\frac{\partial^2 \ell}{\partial \partial \theta \beta_2} &= \sum_{i=1}^{2n} \left(-(y_i + \frac{1}{\alpha}) z_i \beta_1 \frac{\partial x_i}{\partial \theta} \frac{\alpha exp(x_i\beta_1 + z_i\beta_2 + \beta_0)}{(1 + \alpha exp(x_i\beta_1 + z_i\beta_2 + \beta_0))^2} \right)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial^2 \ell}{\partial \alpha \partial \beta_2} &= \sum_{i=1}^{2n} \left(\frac{z_i}{\alpha} \frac{exp(x_i\beta_1 + z_i\beta_2 + \beta_0)}{1 + \alpha exp(x_i\beta_1 + z_i\beta_2 + \beta_0)} - (y_i + \frac{1}{\alpha}) z_i \frac{exp(x_i\beta_1 + z_i\beta_2 + \beta_0)}{(1 + \alpha exp(x_i\beta_1 + z_i\beta_2 + \beta_0))^2} \right) \\
\frac{\partial^2 \ell}{\partial \beta_2^2} &= \sum_{i=1}^{2n} \left(-(y_i + \frac{1}{\alpha}) z_i^2 \frac{\alpha exp(x_i\beta_1 + z_i\beta_2 + \beta_0)}{1 + \alpha exp(x_i\beta_1 + z_i\beta_2 + \beta_0)} \right)
\end{aligned}$$

2.3.7 Model parameter initial values

In GLMNB algorithm, the initial value of $\alpha^{(0)}$ is set as the MLE calculated from all tag counts in the current chromosome under the null hypothesis of no protein binding. $\theta^{(0)}$ is set as one half of the median distance between the two peaks on forward/reverse strands observed in the top 100 windows (ranked by total tag counts). $\beta_1^{(0)}$ and $\beta_2^{(0)}$ equal to zero. Since we use constant β_0 model as default, β_0 is set as 1) logarithm of average tag count in the current window or 2) adjusted baseline using the adjusted baseline strategy described in section 2.3.8.

2.3.8 Adjusted baseline strategy

Despite that the fixed parameter β_0 can be set as logarithm of the average tag count in the current window, an adjusted baseline strategy from MACS is also adopted. The baseline parameter β_0 initial value is set as the logarithm of the maximum tag count average from the current window, 1 kb, 5 kb and 10 kb from ChIP sample if negative control sample is not available. It is set as the logarithm of the maximum tag count average from the current window, 1 kb, 5 kb, and 10 kb from negative control sample when negative control sample is available. Here after in this dissertation, full model refers to the constant baseline model with adjusted baseline strategy.

2.3.9 Sliding window filtration

We apply the above model to test all sliding windows across the genome for peak calling. Since our test statistic is only asymptotically χ^2 distributed, we filtered out all windows with tag counts less than cutoff (=5 tags per 500 bp window by default) on either the forward or the reverse strand. That is, we do not test in windows with very few tag counts. The peaks with very few tags is not scientifically meaningful. So this filtration will not reduce the statistical power. However, it can significantly reduce the number of tests and save computing time.

2.3.10 Constant peak shifting parameter model

In order to examine the importance of the local estimated peak shifting parameter θ, I define and execute the constant θ model, where θ is fixed at the initial value in both null and alternative models. For example, when only ChIP sample data are available, the null model contains one free parameter α. The alternative model contains two free parameters β_1 and α. The degree of freedom in LRT is 1. Similarly, when negative control data are available, the degree of freedom in LRT is also 1.

2.3.11 Constant dispersion parameter model

In order to examine the importance of the additional dispersion parameter α in negative binomial distribution compared to Poisson distribution, I define the constant α model, where α is fixed at the estimated dispersion parameter from the entire genome in both null and alternative models. For example, when only ChIP sample data is available, the null model contains one free parameter β_0. The alternative model contains three free parameters β_1, θ and β_0. The degree of freedom in LRT is 2. Similarly, the degrees of freedom in the LRT is 2 when negative control data are available.

2.3.12 Motif Identification

We used HOMER ([35]) to identify motifs at the predicted binding sites. HOMER searches *de novo* motifs within 150bp around every predicted binding site by GLMNB, SPP, MACS, CisGenome, SISSRs and BayesPeak, separately. If more than one motif is found, the motif closest to the predicted binding site is recorded.

Chapter 3

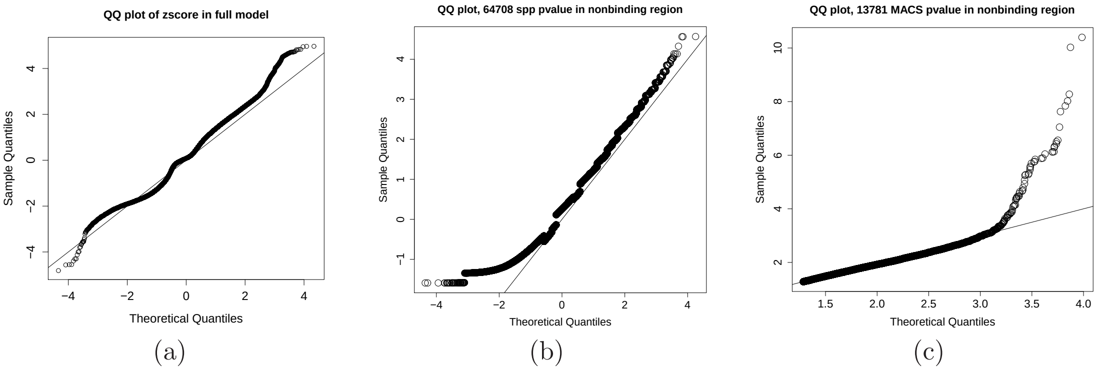
Simulation Study

3.1 Peak calling on simulated ChIP sample in non-binding region

A ChIP-Seq dataset is simulated as described in section 2.2. The simulated data contains 500 peaks distributed in a 300Mb region. Forward and reverse binding profiles are generated using the simulated data as described in the method sections 2.3.1 and 2.3.3. Then GLMNB is used to call peaks using a sliding window with default winsize=500 bp and stepsize=10 bp.

We first examined the p-values in non-binding regions produced by GLMNB. A non-binding region is defined as a region at least 500 bp away from all simulated peaks, so that tags in these regions only come from simulated background. Figure 3.1(a) shows the quantile-quantile (QQ) plot of the GLMNB's z-scores compared to standard normal distribution. There are 69,003 overlapping p-values with more than 5 tags per 500 bp window calculated from non-binding region, in which there is slight inflation in the positive end. We observed that the GLMNB z-scores are approximately normally distributed, with a slight deviation that is likely due to the small tag counts in non-binding regions and also the irregular peak shift parameter θ used in our model. We did not observe extremely strong departure in the QQ plot at the extreme values in the positive end, suggesting that our p-values from normal approximation are appropriate for peak calling. SPP is not based on a statistical model and does not output valid p-values. However, if I convert the FDR from SPP into p-values, many more SPP p-values(64,708) are discovered from non-binding regions. These p-values are converted by multiplying FDR with the peak order and divided by total test number, one can further convert p-values into z-scores and plot the QQ plot as shown in Figure 3.1(b). In the positive extreme end, it appears approximately normal distribution. So FDR estimates in SPP are appropriate. MACS does provide p-value output when there is no negative control sample. Figure 3.1(c) shows a Quantile-Quantile(QQ) plot of MACS z-scores (converted from p-values) compared to a standard normal distribution. This is the most comparable result with GLMNB from MACS that we are able to obtain. Due to the restriction of MACS program, we were only able to obtain p-values < 0.1, rather than all p-values in the full range of $[0, 1]$. Further due to MACS automatic peak region expansion procedure, we were not able to restrict the same peak width as used by GLMNB at 500 bp. We obtained 13,781 MACS p-values < 0.1 from the non-binding regions, and the sizes of MACS peaks ranged from 400 bp to 6 kb. As observed in Figure 3.1(c), MACS z-scores from the non-binding regions significantly deviated from the standard normal distribution at large values. That is, the significance output by MACS is greatly inflated in our simulated data. For instance, at a threshold where we expect 30 false positive peaks, the actual number of false positives called by MACS is 234. The inflation of the significance by MACS is likely due to its Poisson model assumption. Therefore, its FDR values tend to be liberal, or much more significant than it should be.

We further calculated the FDRs from GLMNB and compared such values with other algorithms if a FDR is provided for ChIP sample data only. Scatter plots between FDR



(a) (b) (c)

Figure 3.1. Quantile-quantile plot of z-scores output by (a) GLMNB, (b) SPP and (c) MACS using data from non-binding regions in the simulated data.

in log_{10} scale and simulated peak strength are plotted for GLMNB, SPP, MACS in Figure 3.2(a)-(c). Because there are no FDR provided by CisGenome, SISSRs and BayesPeak, the corresponding predicting strength values are plotted against simulated peak strength in Figure 3.2(d)-(f). FDRs are in practice calculated as the expected number of false positives divided by the total number of positives ([36]). Given that peaks are simulated, observed FDRs are calculated as the observed number of false positives divided by the total number of called peaks. A predicted peak is matched to a true simulated peak if the predicted binding site is within 200 bp of the true binding site. We used 200 bp distance to evaluate software performance because otherwise MACS will miss too many true peaks due to its automatic expansion of peak region and therefore inaccurate prediction of binding locations. At a 5% FDR threshold, GLMNB called 508 (non-overlapping) peaks, among which 492 were true peaks and 16 were false positives, yielding an observed FDR 3.1% in Figure 3.2(a). Out of the 500 simulated true peaks, 8 peaks were missed by GLMNB at 5% FDR, yielding a 98.4% power. GLMNB gives an observed FDR(3.1%) less than the expected FDR at 5%, and therefore conservatively controls the false positives, which is good in practice. As shown in Figure 3.2(a), FDR values from GLMNB are in fact positively correlated with the simulated peak strength. The FDR value is more significant for a stronger simulated peak, for example, the one with peak strength greater than 8 has a GLMNB FDR=10^{-61} on the right bottom in the plot. So the rank of GLMNB FDR value well represents the strength of peaks.

As shown in Figure 3.2(b) SPP called 672 peaks, including 498 true positives and 174 false positives. Even though SPP called 6 more true peaks than GLMNB, the number of false positives is much larger, resulting in an observed FDR of 174/672 = 26%. Its power is higher but still very close to that of GLMNB, 99.6%. No false positives give extremely significant FDR values. Since SPP is not based on any statistical models, it assigns a minimum FDR to all top ranked peaks if their scores are stronger than the maximum scores observed in the negative control data. Such FDR values will not change unless another false positive appears as SPP score decreases. As a result, SPP's FDR values look flat regardless of the simulated peak strength. Its peak ranks do not represent protein binding strength.

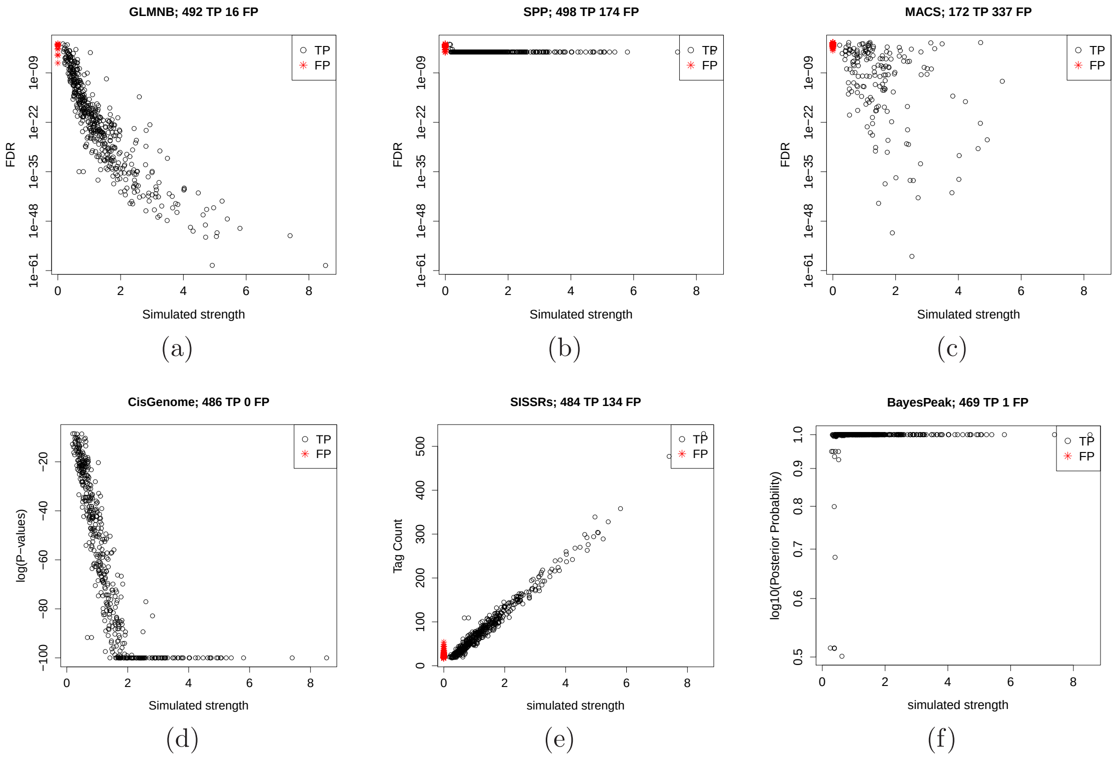
MACS called 509 peaks, including 172 true positives and 337 false positives with FDR and simulated peak strength plotted in Figure 3.2(c). MACS called many fewer true peaks than GLMNB and SPP, yielding a power of 34.4%. There is no FDR values reported by MACS if ChIP sample data are the only input. It however called 337 false positives, resulting a FDR of 66.3%, which is not optimal in performance. If we lose the range of defining true positives from 200 bp to 1,000 bp, MACS is able to increase the true positives to 341 and reduces the false positive amount to 168 (data not shown). That is a 68.2% power and 33% FDR, still not competitive GLMNB and SPP. It is mostly due to MACS automatic extension on peak region and therefore inaccurate peak detection. MACS FDR and the simulated peak strength are not well correlated, suggesting that the peaks ranked by MACS may not correctly represent the protein binding strength. For example, some false positive peaks have much significant p-values than the true positives.

CisGenome called 486 peaks, all of which are true positives. It achieves 97.2% power slightly lower than GLMNB and SPP and 0% FDR under FDR threshold of 5%, which is more conservative than GLMNB and SPP. CisGenome does not output estimated FDR directly for ChIP sample only, so $log_{10}(p-values)$ are plotted against simulated peak strength as shown in Figure 3.2(d). Such p-values are highly correlated with simulated strength when simulated strength is small, for example, less than 2. The p-values are all 10^{-100} when simulated strength are greater than 2, which makes the strongest peaks indistinguishable.

SISSRs called 618 peaks under FDR threshold 0.1% by default, among which 484 are true positives and the rest 134 are false positives. As claimed by Jothi et al [13], SISSRs is able to discover more peaks than other programs. However, in this specific simulated ChIP sample data, it does not identify as many true peaks as GLMNB and SPP but many more false positives. SISSRs achieves 96.8% power but suffers 22% observed FDR, many more liberal than expected at 0.1% default level. SISSRs does not allow users to change FDR threshold setting. But one can image if a less constraint FDR is set, it will suffer worse observed FDR in this dataset. SISSRs provides total tag counts only in called positions rather than p-values or FDR. Therefore, such total tag counts are plotted against simulated peak strength in Figure 3.2(e). The tag counts are highly correlated with peak strength, which could be one good measure to rank peaks. However, one also notices that several false positives even contain roughly 50 tags.

BayesPeak called 470 peaks, among which 469 are true peaks and one is false positive. That concludes 93.8% power slightly less than GLMNB, SPP, CisGenome and SISSRs and 0.2% FDR, more conservative than GLMNB. BayesPeak outputs a posterior probability value of each position being enriched. Posterior probabilities in log scale from BayesPeak output are plotted against simulated peak strength in Figure 3.2(f). Most posterior probabilities are close to value 1.0, forming a flat line on the top regardless the simulated strength, which indicates a good separation between enrichment and non-enrichment status. However, posterior probabilities does not provide rich information to distinguish the strength of ChIP signals.

In summary, GLMNB prediction has the second best power among the six ChIP-Seq peak callers and conservative false discovery rate based on this specific simulated ChIP sample data. Its FDR also represents the peak strength well.



(a) (b) (c)

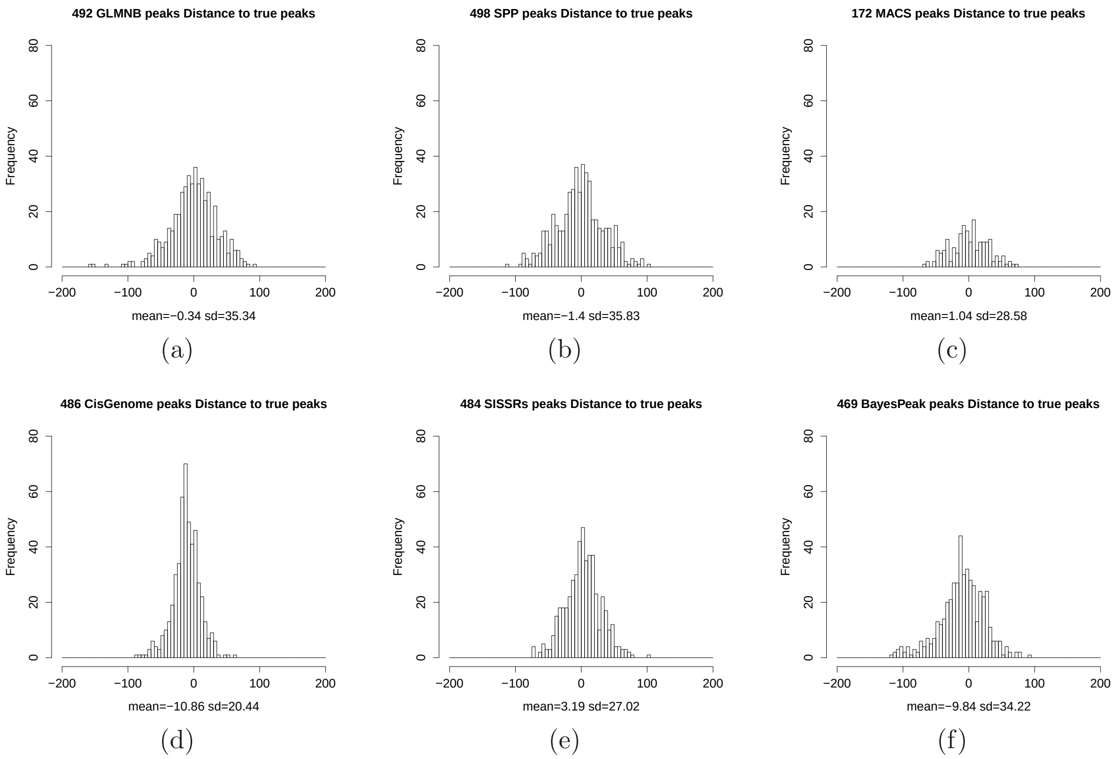
(d) (e) (f)

Figure 3.2. Scatter plots between FDR in log scale and simulated peak strength called by (a) GLMNB, (b) SPP, (c) MACS, (d) CisGenome, (e) SISSRs and (f) BayesPeak. The simulated data contained 500 peaks randomly distributed in a 300Mb region, and each peak was separated from each other by at least 20kb. Background tags are simulated from negative binomial distribution.

After evaluating the power and observed FDR from the six peak callers, I want to evaluate the spacial resolution among them. The spacial resolution for this simulated data set are defined as the distance between predicted peaks and nearest true peak positions. Positive(negative) distance means the predicted peak locates on the right (left) side of true peak position. In Figure 3.3(a)-(f), histograms of such distances are plotted for the six algorithms. The average distance from GLMNB results is -0.34 bp, a value closest to zero among the six algorithms, with a standard deviation of 35.34 bp. It suggests its great central tendency for predicting binding positions for this specific simulated data. The average distance for SPP is -1.4 bp with a standard deviation of 35.83 bp, the greatest standard deviation among the six. MACS and SISSRs achieve similar performance in terms of distance to true peak positions with 1.04 bp on average and a standard deviation of 28.58 bp, and 3.19 bp on average and a standard deviation of 27.02 bp, respectively. However, CisGenome and BayesPeak give average distance of -10.86 bp and -9.84 bp with a standard deviation of 20.44 bp and 34.22 bp, respectively. It indicates that both programs identify peaks about 10 bp upstream to the true peak positions on average. But CisGenome achieves the smallest variation in terms of distances between predicted peak positions to true peak positions. All six algorithms yield an approximately symmetric histogram for the distance.

In summary, GLMNB achieved the best spatial resolution in terms of the average predicted peak distance to true peak position for this specific simulated ChIP sample data, even though the variation is among the two biggest of the six algorithms.

Now we can move to non-binding regions and compare estimated parameters to true simulated values in the study. The estimated parameters of signal profile coefficient β_1, peak shifting parameter θ, baseline tag counts per window β_0 and dispersion parameter α from non-binding regions are plotted in histograms in Figure 3.4. Remember tags from non-binding region come from simulated background only. The sample average for the model parameters is drawn using a red solid line, and the known parameter value when simulating data is drawn in blue dashed line. The estimated β_1 is approximately normally distributed with sample mean -0.01 and a standard deviation of 0.10. Estimated β_1 is expected to center at zero because the profile should not be a statistically significant predictor for background random noise. Therefore, β_1 should be centered at zero. The estimated θ gives a sample mean of 36 bp and a standard deviation 53 bp in non-binding regions, whereas there is no θ or signal profile involved in the non-binding regions. Such non-zero average θ is caused to the initial value setting in my algorithm. Before looking for MLE of model parameters,



(a) (b) (c)

(d) (e) (f)

Figure 3.3. Spatial resolution are plotted as histogram of distance between true peak positions and peaks called by (a) GLMNB, (b) SPP, (c) MACS, (d) CisGenome, (e) SISSRs and (f) BayesPeak with in 200 bp region.

I set a positive initial value of 50 bp rather than 0 bp for θ so that the algorithm has more tendency to search for meaningful solutions. Negative θ value is meaningless for my algorithm. However, since the random noise is not generated from the signal profile, the algorithm is free to locate a MLE solution of θ at any value between 0 bp and profile length. That is why the estimated θ shows a positive sample mean and a large variation.

The baseline parameter β_0 has an sample average -2.0 with a standard deviation of 4, higher than β_0 simulation setting. This is due to the window filtration criteria in my algorithm. All windows with less than 5 forward tags or 5 reverse tags per 500 bp are skipped for modeling, because it is very unlikely that these windows will contain scientifically significant binding positions even if it shows statistical significance, thanks to rare tag counts or unbalanced tag distribution between forward and reverse strands. Similarly, the left skewed α with most estimated values lower than the simulation setting is also caused by the filtration criteria.

One may notice that the Quantile-Quantile plot shown in Figure 3.1(a) is not normally distributed. Figure 3.5(a) and (b) show the QQ plot of Wald test z-score for full model and constant θ from non-binding regions on the same ChIP sample data, respectively. Figure 3.5(c) and (d) show the QQ plot of the likelihood ratio test(LRT) statistics for full model and constant θ model from non-binding regions on the same ChIP sample data, respectively. From Figure 3.5(a), one notices a slight departure of sample quantiles from theoretical quantiles on the positive end. With θ parameter fixed, the z-score for profile coefficient β_1 in GLMNB is approximately normally distributed and the LRT statistic should follow a χ^2 distribution. The sample quantiles are even slightly lower than the theoretical ones on the top right end in Figure 3.5(b). Therefore, the departure of sample quantiles from theoretical quantiles shown in Figure 3.5(a) is due to the involvement of the θ variable in the model, which violates the regulatory conditions of generalized linear model. In fact, predictor x_i in the alternative model is a function of θ, and therefore violates the assumption for asymptotic behavior of maximum likelihood estimator that data points should not depend on unknown parameter θ. If the parameter θ is set as fixed value, then the condition is no longer violated. Therefore, β_1 is asymptotic normal distribution as shown in Figure 3.5(a). Besides Wald test, I also propose the usage of likelihood ratio test. The QQ plots for likelihood ratio test for full model and constant θ model are plotted in Figure 3.5(c) and (d). There is a slight departure for full model (if assuming df=2 in black circle) on the positive end, when compared to asymptotic χ^2 distribution with degree of freedom 1. This is also due to the violation of regulatory conditions of asymptotic normality for maximum likelihood estimator introduced by θ parameter. Suggested by Fan et al [34], however, when there is such a violation, the likelihood ratio test statistic still approximately follows a χ^2 distribution, but no longer the

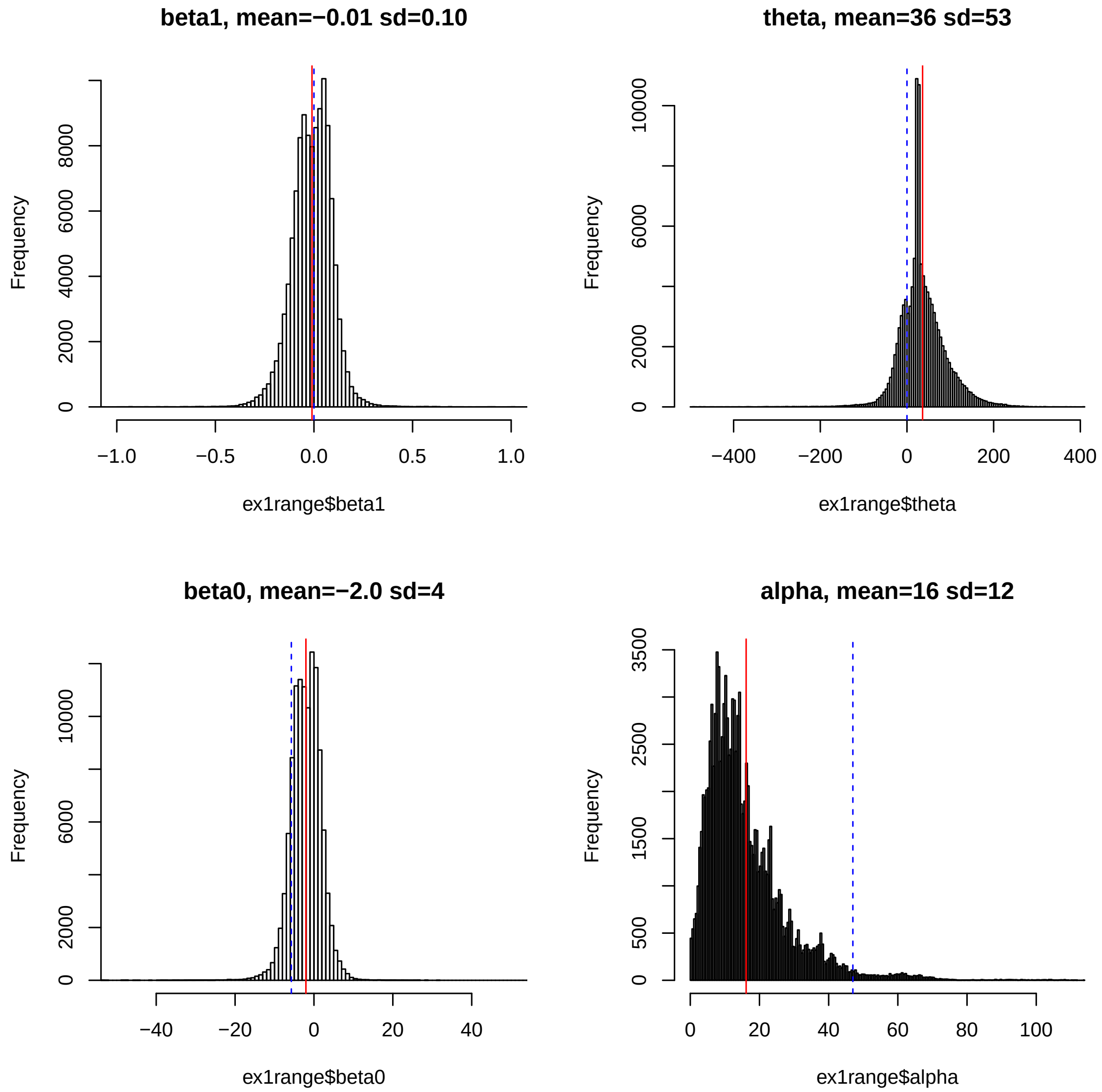
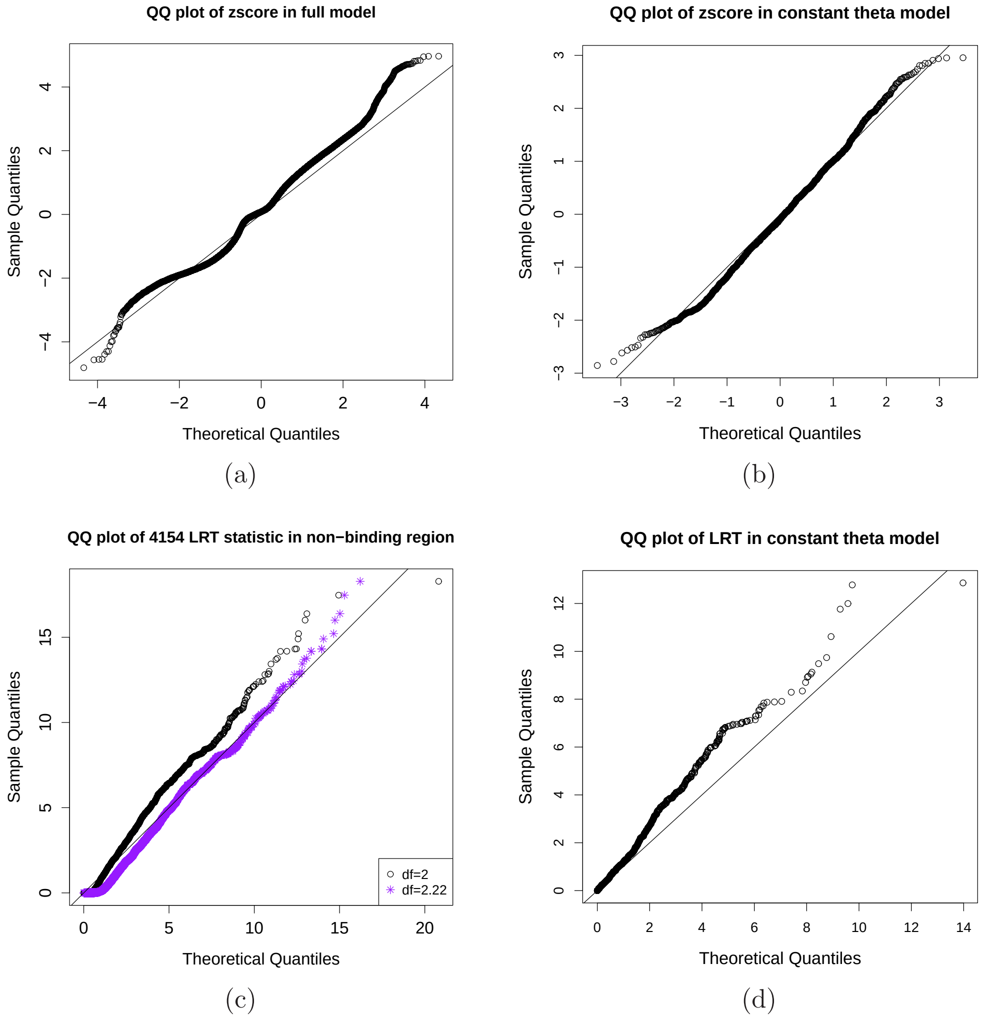


Figure 3.4. Estimated model parameters from non-binding regions on Simulated ChIP sample data, including profile coefficient, β_1, shifting parameter, θ, baseline, β_0 and dispersion parameter, α.

same degrees of freedom as the difference of free parameter numbers between null model and alternative model. Simulations need to be run on rare tag regions to get an estimate of the degrees of freedom. The rare tag regions refer to windows whose forward or reverse tags are less than 8 but more than 5 in practice. The degrees of freedom are estimated as the sample median (= 2.22) of LRT statistic values. Therefore, when we re-plot the QQ plot with the corrected degrees of freedom of 2.22 in purple in Figure 3.5(c), it is asymptotically χ^2 distribution. Therefore, the likelihood ratio test is valid with correctly specified degrees of freedom.

Even though the constant θ model enjoys the beauty of asymptotic normality and validity of likelihood ratio test, fixing θ actually reduces the peak calling power. Figure 3.6 compares GLMNB's performance between full model and simpler model, constant θ model, with respect to its power, FDR and spacial resolution. The constant θ model called 438 true peaks, 54(11%) less than the full model. The peak calling power of constant θ model is 87.6%, 10.8% less than that of full model. Both modeling algorithms yield a conservative FDR, 0.7% and 3.1% under FDR=5% threshold, for constant θ and full models, respectively. Both algorithms give FDR values that are highly correlated with simulated peak strength. However, FDR from full model distinguishes peaks in a more aggressive pattern. For example, the simulated peak with strength above 8 is called with FDR close to 10^{-61} by full model, whereas it is called with FDR around 10^{-32} by constant θ model. Peaks called by constant θ model gives an average of distance to true peak position 14.03 bp with standard deviation of 19.71 bp. It suggests a shifted peak locations even though with a slightly smaller variation. Therefore, fixing θ parameter reduces peak calling power and yields a shifted peak locations, which is not good for our algorithm.

Here, I also want to use simulation study to demonstrate the importance of dispersion parameter α in the negative binomial background modeling in Figure 3.7. Figure 3.7(a) and (c) are the same plots shown in Figure 3.6. Figure 3.7(b) shows the scatter plot between



(a)

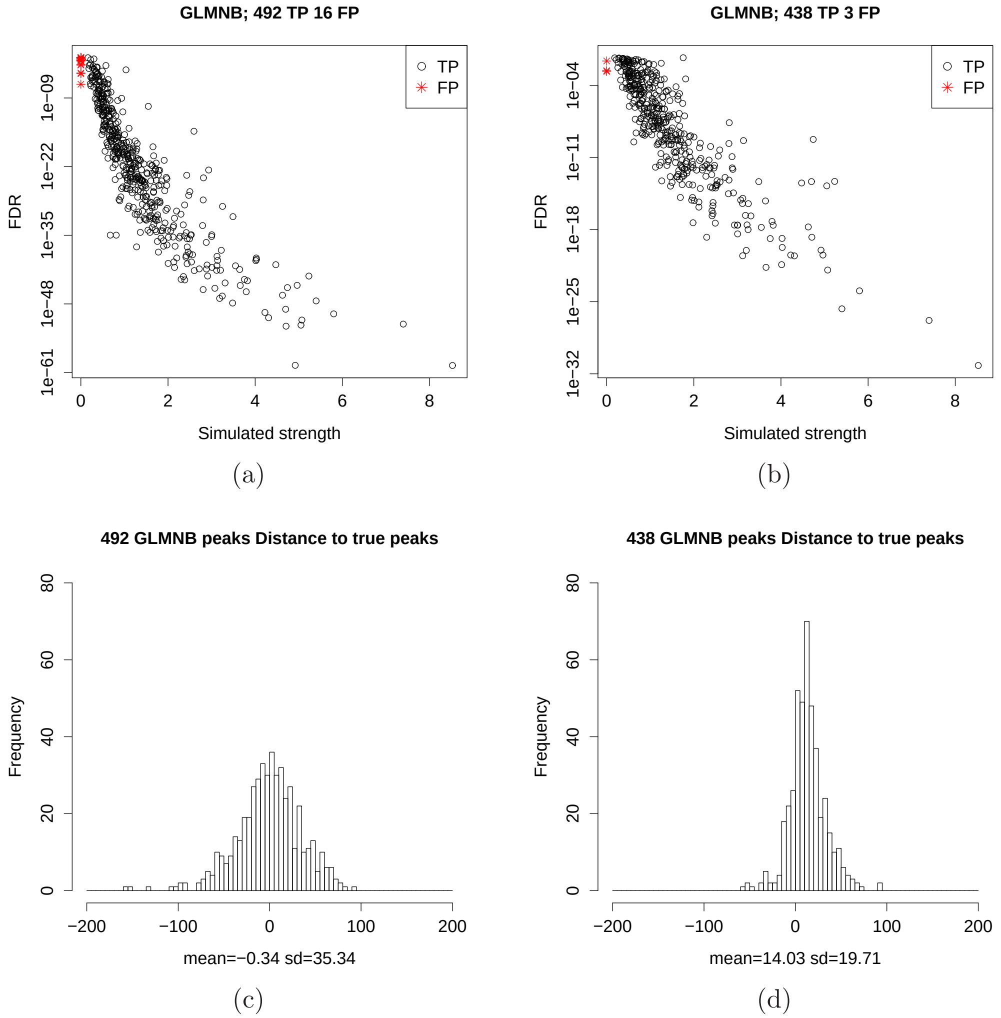
(b)

(c)

(d)

Figure 3.5. Quantile-Quantile(QQ) plot of z-score and LRT from full model and constant θ in non-peak region p-values. The QQ plot of LRT from full model with adjusted degree of freedom is plotted in purple asterisk in (c).

FDR in log_{10} scales and the simulated peak strength from the constant α model. This constant α model achieve 95% power (475/500) and 0.4% observed FDR, only slightly lower power than GLMNB full model. However, the correlation between FDR and simulated peak strength is much weaker than the one in the full model. For example, points with simulated strength between 1 and 4 are flat. FDR values in this area do not strongly correlate with the simulated peak strength. This is due to larger standard error with the fixed α model.



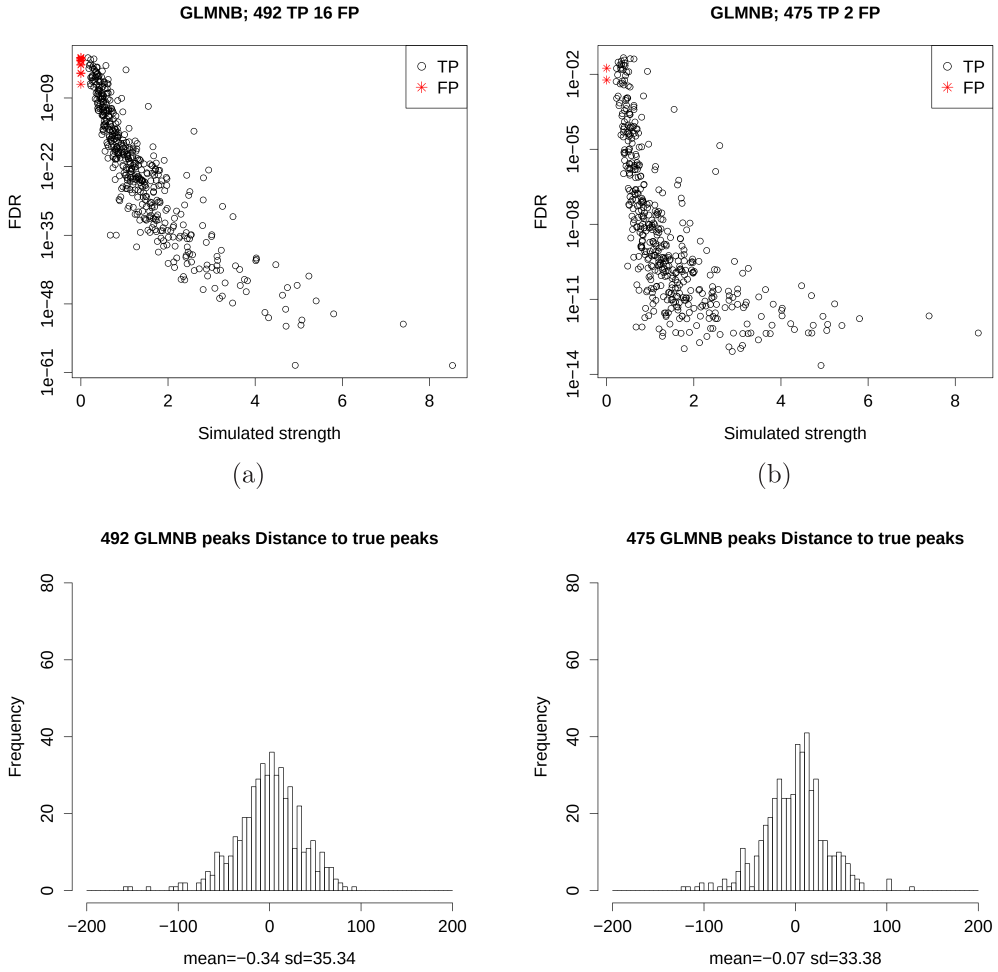
(a) (b)

(c) (d)

Figure 3.6. Scatter plots between FDR and Simulated strength and Predicted peak distances from Simulated peak position from full model (a)(c) and constant θ model (b)(d).

Figure 3.7(d) further plots the predicted peak distances from true peak positions for constant α model. The constant α model achieves an average distance of -0.07 bp and a standard deviation of 33.38 bp, very close to the one in GLMNB full model. In summary, GLMNB full model achieves higher power with variable α and more significant FDR than constant α model.

After checking the necessity of using shifting parameter θ and dispersion parameter α in the alternative model, I want to emphasize the importance of adjusted baseline β_0 using



(a) (b)

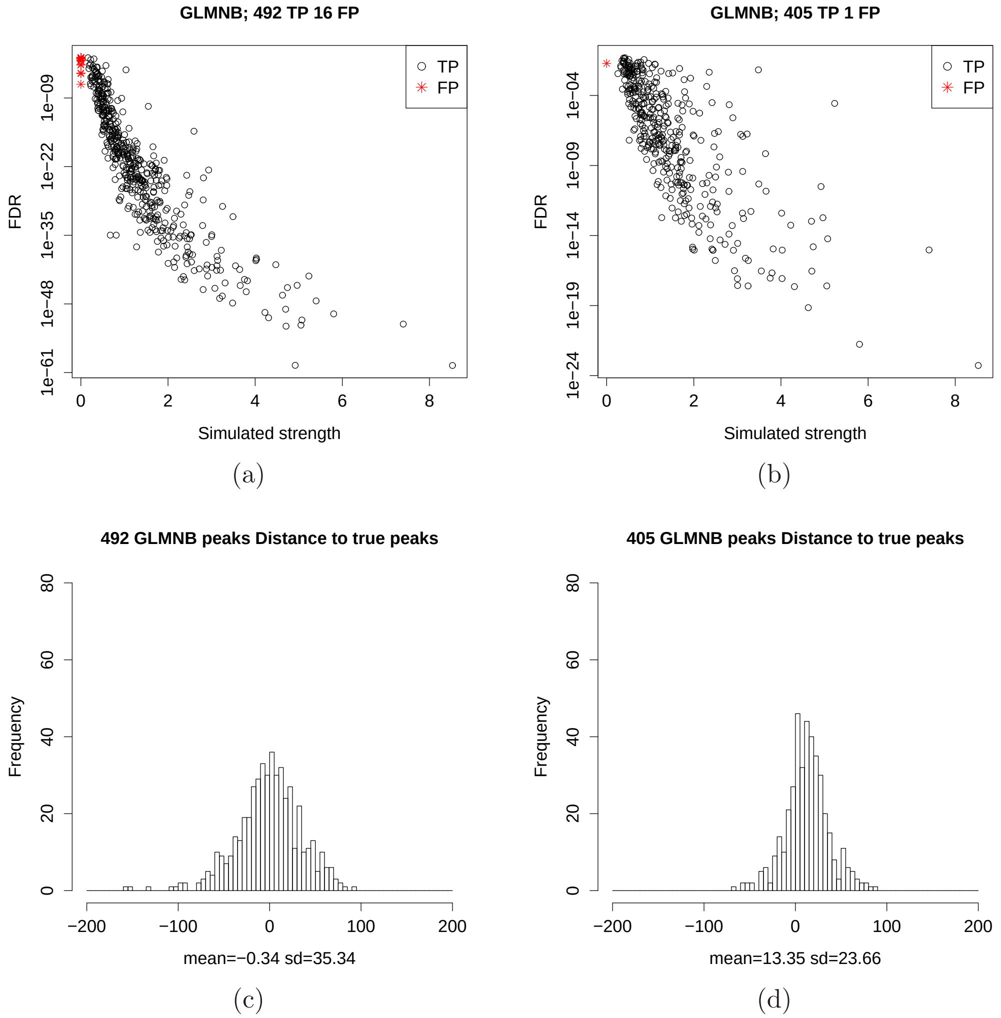
(c) (d)

Figure 3.7. (a)(b)Scatter plot between FDR and (c)(d) Simulated strength and Predicted peak distances from Simulated peak position from full model and constant α model.

current simulated ChIP sample data. Inspired by MACS program, where the Poisson mean estimate uses a higher value when the average tag count in a wider area(1 kb, 5 kb and 10 kb) is higher than the one in the current window (500 bp). This strategy reduces false positives when the neighborhood has stronger non-specific binding affinity than other areas. I adopt this strategy in the GLMNB full model. The initial value for baseline parameter β_0 is the logarithm of maximum values of the overall average tag counts in the entire genome and average tag counts from the current window, 1 kb, 5 kb or 10 kb neighborhood. This is so called adjusted baseline strategy in GLMNB algorithm described in section 2.3.8. If fixed β_0 value is set as the logarithm of the overall average tag counts in the entire genome, it is called fixed global baseline strategy. This strategy does not account for regional tag bias where there are noisy tags observed even in negative control data. However, the adjusted baseline strategy discussed in section 2.3.8 does account for it. Now one can compare the performance between GLMNB model with adjusted baseline strategy in Figure 3.8(a),(c) and the one with fixed global baseline strategy in Figure 3.8(b),(d). Compared with the GLMNB model with adjusted baseline strategy, GLMNB model with fixed global baseline strategy called 405 true peaks and 1 false positive yielding 81% power and 0.2% FDR. However, GLMNB model with fixed global baseline strategy provides less aggressive FDR and ambiguous ranking that does not reflect simulated peak strength. For example, one true peak with peak strength around 5 is called with FDR around 10^{-4}, which is much less significant than many other peaks with smaller peak strength. As shown in Figure 3.8(d), the predicted peak are shifted on average 13.35 bp from true peak positions to downstream with a standard deviation of 23.66 bp. Therefore, there is a non-zero positive shift on peak location for GLMNB with fixed global baseline strategy. In summary, GLMNB with adjusted baseline strategy provides more aggressive FDR values that better distinguish peaks with respect to their true peak strength and a close to zero distance to true peak positions on average.

3.2 Peak calling on Simulated ChIP-Seq data with negative control sample

As described in Section 2.2, 95 of 500 peak positions in ChIP sample are randomly selected as false positives. Tags from these 95 peak positions are combined with previously independently generated background tags and establish a negative control dataset.



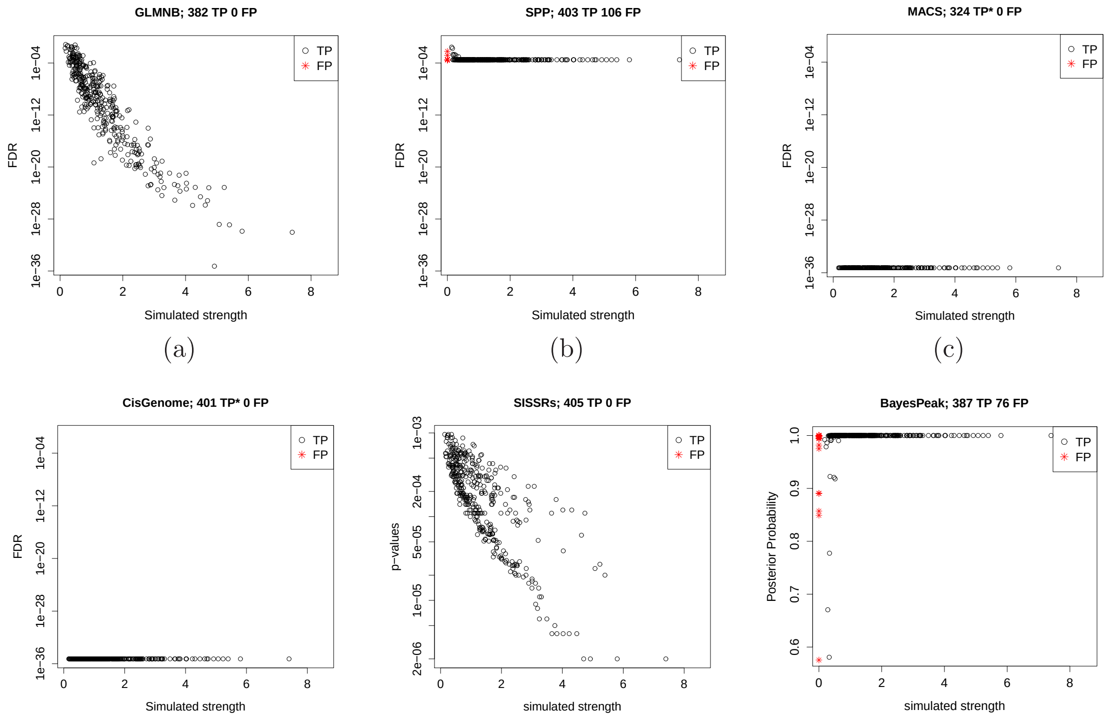
(a) (b)

(c) (d)

Figure 3.8. (a)(b) Scatter plot between FDR and (c)(d) Simulated strength and Predicted peak distances from Simulated peak position from adjusted β_0 full model and constant β_0 full model.

Any predicted peaks close to these 95 peaks are classified false positives. Predicted peaks close to the rest 405 peaks are classified true positives. Figure 3.9 shows the scatter plot between FDR and simulated peak strength for true positives called by (a) GLMNB, (b) SPP, (c) MACS, (d) CisGenome, (e) SISSRs and (f) BayesPeak. GLMNB achieves 94.3%(382/405) power, higher than that of MACS(80%, 324/405), but slightly lower than that of SPP (99.5%, 403/405), CisGenome (99%, 401/405), SISSRs(100%, 405/405) and

BayesPeak(95.6%, 387/405) for this specific simulated dataset. GLMNB achieves 0% FDR, the same level as MACS, CisGenome, SISSRs, but much better than that of SPP(20.8%, 106/509) and BayesPeak(16.4%, 76/463). Therefore, GLMNB offers conservative results with less than expected FDR and slightly lower power. Only CisGenome and SISSRs outperform GLMNB in terms of power and FDR. But GLMNB outputs the best FDR that distinguishes strong peaks from weak ones, which have strongest correlation with the simulated peak strength. CisGenome outputs peaks with FDR=0 and SISSRs output FDR in a narrow range with weaker correlation to simulated peak strength.



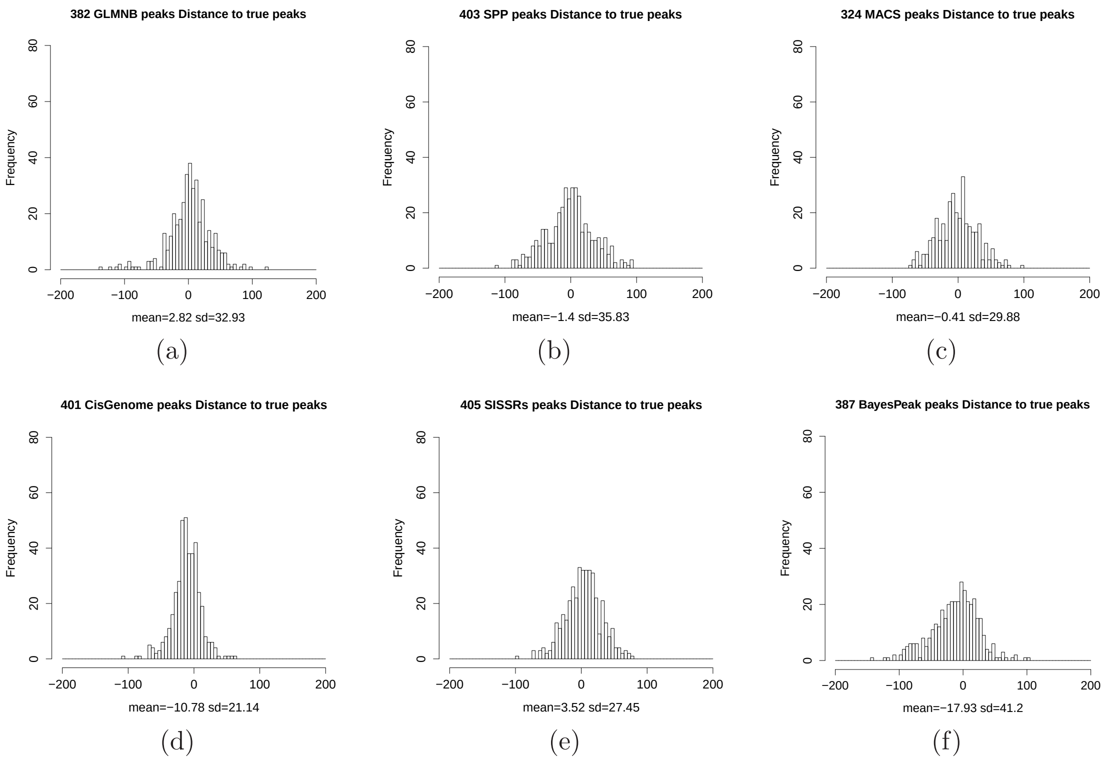
(a) (b) (c)

(d) (e) (f)

Figure 3.9. Scatter plot between FDR in log scale and simulated peak strength called by (a) GLMNB, (b) SPP, (c) MACS, (d) CisGenome, (e) SISSRs and (f) BayesPeak on simulated ChIP data with negative control. The simulated data contained 405 real peaks and 95 pseudo peaks(strong signals in both ChIP sample and negative sample) randomly distributed in a 300Mb region, and each peak was separated from each other by at least 20 kb.
Note: * FDR equals to zero for all peaks called by MACS and CisGenome.

We can take a look at the spatial resolution on true positives predicted by (a) GLMNB, (b) SPP, (c) MACS, (d) CisGenome, (e) SISSRs and (f) BayesPeak with in 200 bp region on simulated ChIP data with negative control in Figure 3.10. GLMNB's true positive peaks has an average distance to actual peak position of 2.8 bp with a moderate standard deviation of 33 bp. SPP, MACS and SISSRs have the similar performance with close to zero average distance and moderate standard deviation. CisGenome and BayesPeak yield predictions with shifted average peak distance.

In summary, GLMNB and SISSRs have similar top performance on this specific simulated ChIP data with negative control in terms of prediction power, actual FDR and spatial resolution. However, GLMNB performs best with respect to rank peaks by simulated peak strength.



(a) (b) (c)

(d) (e) (f)

Figure 3.10. Histogram of distance between true peak positions and peaks called by (a) GLMNB, (b) SPP, (c) MACS, (d) CisGenome, (e) SISSRs and (f) BayesPeak with in 200 bp region on simulated ChIP data with negative control.

Chapter 4

Real Data study

During the simulation study in Chapter 3, I investigate the validity of GLMNB test statistics, evaluate the performance of GLMNB and compare its results with other algorithms on simulated ChIP data with or without negative control data. Since the simulated data are created under a certain artificial assumptions which may not hold in reality (i.e., tags follows a negative binomial distribution with a certain shape), it is important to examine GLMNB's performance compared to other algorithms. In the following chapter, I use GLMNB to call peak in real data from ChIP-Seq experiments. It is not necessary to discuss the performance of GLMNB on all ChIP-Seq data sets mentioned in section 2.1, I mainly use FoxA1 data set as an example. The reason is that this transcription factor has well-known binding conservative motifs, and such a data set has been used in the research article of MACS by Zhang et al [1]. In the following sections, I will illustrate GLMNB's performance on FoxA1 ChIP sample only in section 4.1. Then I will compare the performance of GLMNB with other 4 algorithms, SPP, MACS, CisGenome and SISSRs on the same data in section 4.2. With the help of negative control, one should be able to reduce false positives due to non-specific binding events showing in both ChIP sample and negative control data. In section 4.3, I will compare the performance of GLMNB with other 4 algorithms on FoxA1 ChIP sample data with negative control dataset. In section 4.4, I will brief compare all five algorithms on all other datasets mentioned in 2.1.

4.1 GLMNB peak calling on FoxA1 ChIP sample data only

An example of peak calling by GLMNB on FoxA1 ChIP sample data only is shown in Figure 4.1. The observed forward and reverse tag counts per bin (10bp) are plotted in red/green vertical bars in the figure. There are 13 sliding windows around the binding site, whose -log_{10}(p-value) are illustrated as blue connected dots in the figure. The window centered at chr1:199,518,124 yielded the most significant p-value$= 10^{-11.8}$, which is then called as a binding site. The fitted forward and reverse binding profiles by GLMNB is shown in red and green curves, which are located $\theta = 37$bp away from the window center to each side. The blue horizontal line of length $2 \times \theta = 74$ bp marks the width of the predicted binding interval.

GLMNB were applied on FoxA1 ChIP sample data only with a minimum threshold of 8 tags per 500 bp window for both forward and reverse strands. At 5% FDR, GLMNB detected 4,008 FoxA1 peaks. Figure 4.2(a) shows the ranked FDR in log scale of these peaks. The total number of sliding windows tested by GLMNB was 246,144 after filtration, and thus with p-value$< 10^{-3.09}$ the expected number of false positives is 201 of 4,008 calls, yielding an expected 5% FDR. As shown in our simulation study, the FDR estimated by GLMNB is actually conservative, and thus we expect the actual FDR to be less than 5%. One of GLMNB's feature is to allow the variaty of peak shifting parameter in different windows. Figure 4.2(b) shows the distribution of the estimated peak shifts from all FoxA1 peaks. The estimated peak shifts for FoxA1 have mean 44bp and standard deviation 22bp.

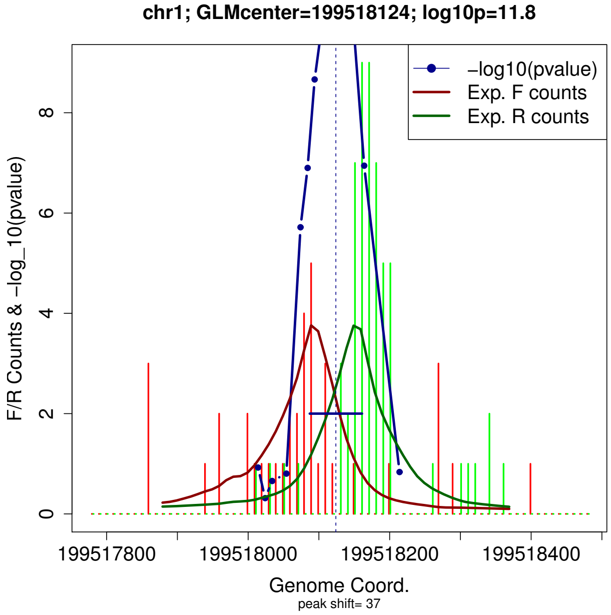


Figure 4.1. A FoxA1 peak detected at chr1:199,518,124 (in blue dashed line) with $-log_{10}$(p-value)=11.8 by GLMNB. All $-log_{10}$(p-value) from adjacent sliding windows are shown in blue connected dots. A blue horizontal line of length $2 \times \theta = 74$ bp represents the distance between the fitted forward peak (red curve) and the reverse peak (green curve). The Y-axis is the tag counts per 10 bp bin and negative log p-values with base 10.

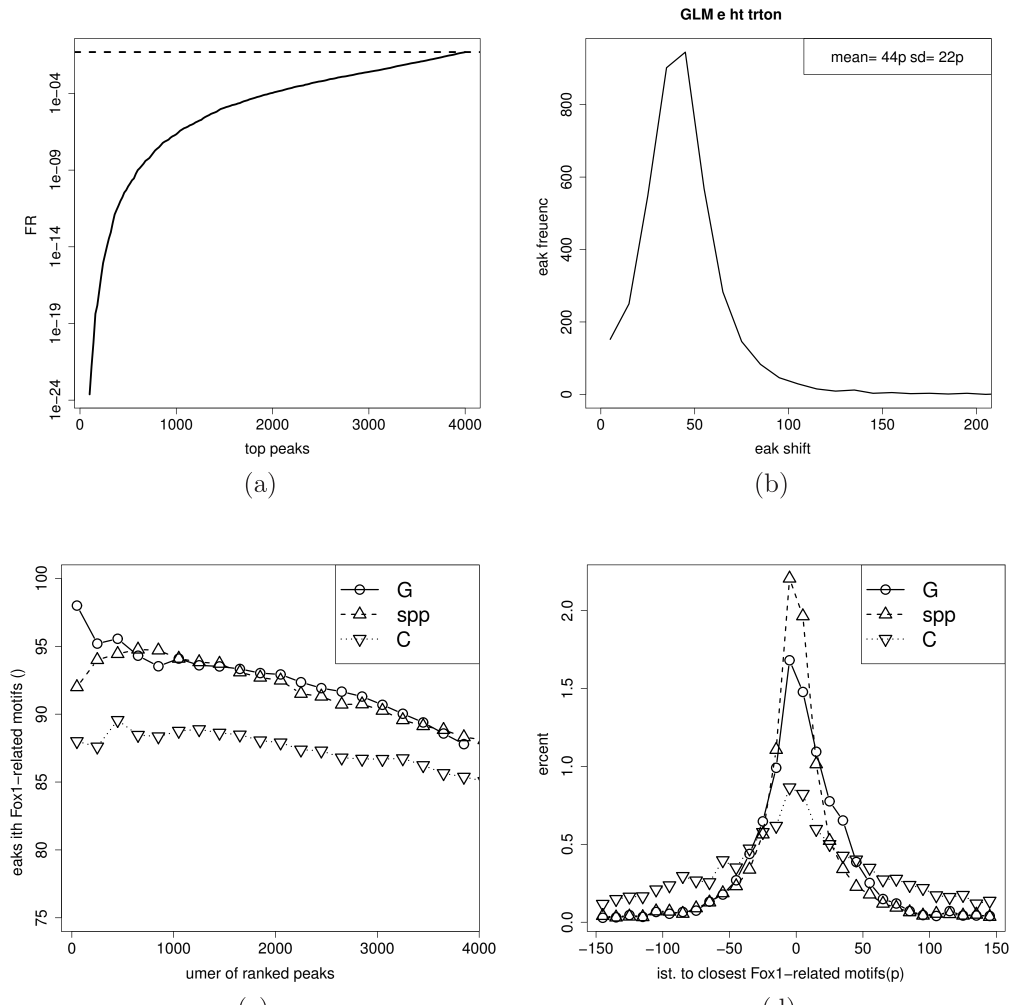
FoxA1 binding sites were reported closely related to FoxA1 motifs, including Forkhead motif(FKHR) ([1, 37]), FoxA1/LNCAP and FoxA1/MCF7 motifs. I plot in Figure4.2(b) the histogram of the distance between each detected FoxA1 peak to its closest FoxA1-related motifs, if there is at least one motif within 150bp of the predicted binding site by GLMNB(circle in solid line), SPP(upper triangle in dashed line) and MACS (inverse triangle in dotted line). We also plot in Figure4.2(c) the percentage of the detected FoxA1 peaks containing a FoxA1-related motif within 150bp distance against ranked top peaks. Among the top 4,008 FoxA1 peaks detected by GLMNB, there were roughly 87.8% to 95% peaks containing at least one

FoxA1-related motif. SPP had similar motif percentage (88.1% to 94.7%). MACS peaks, in contrast, can be matched with 85.4% to 89.6% FoxA1-related motifs within 150bp distance, slightly lower than GLMNB and SPP. This is due to the inaccuracy of MACS predicted binding positions. Figure 4.2(d) further shows the histogram of distance between GLMNB (circle in solid line), SPP(upper triangle in dashed line) and MACS(inverse triangle in dotted line) predicted binding sites and the closest FoxA1-related motifs. The distances between GLMNB peaks and the closest FoxA1-related motifs were mostly within 100 bp, suggesting the high spatial resolution of the predicted binding sites by GLMNB. Both GLMNB and SPP outperformed MACS.

4.2 GLMNB peak calling comparisons with other algorithms on FoxA1 ChIP sample only

One may note that 4,008 peaks were called in section 4.1 using conservatively high tag count threshold, 8 tags for both strands. It is conservative not only because of its tag counts requirement for both strands, but the symmetricity between two strands in terms of tags counts. Biologists interested in these highly conservative peaks for a FoxA1 can use these settings. If such tag count threshold lessens, one should expect more peaks. However, as a peak caller developer, I would like to reduce such tag count threshold to 5 tags per 500 bp window and explore the performance between GLMNB and other peak calling algorithms. Please note that, the peak calling algorithms compared with GLMNB do not include BayesPeak simply because its highly shifted peak resolution and high false discovery rate as shown in Figure3.9(f) and Figure 3.10(f).

Table 4.1 gives number of peaks called by GLMNB and other four algorithms on FoxA1 ChIP data with or without negative control. Among the results from five algorithms on



(a)

(b)

(c)

(d)

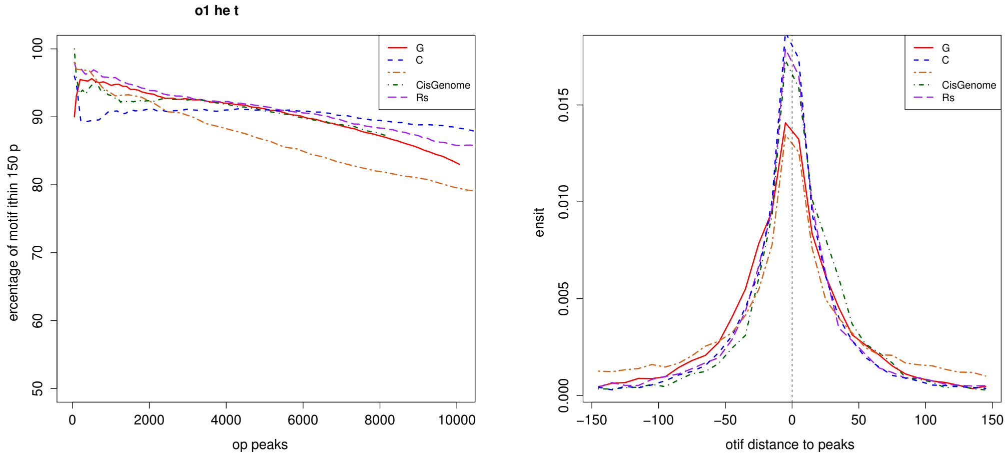
Figure 4.2. GLMNB peak calling results for FoxA1 ChIP-Seq. (a) GLMNB peaks ranked by expected FDR in increasing order. 4,008 GLMNB peaks were called at FDR≤ 5%. (b) Histogram of the estimated peak shifting parameter (θ), with mean 44bp and standard deviation 22bp. Matched motif comparison between GLMNB peaks and SPP, MACS peaks for FoxA1 ChIP-Seq dataset. (c) Percentage of detected peaks carrying at least one FoxA1-related motifs within 150bp to predicted binding sites by GLMNB, SPP and MACS. (d) Histogram of the distance between predicted binding sites and closest FoxA1-related motifs.

FoxA1 ChIP sample only, GLMNB called the least peaks (10,073) at FDR 5%, while SPP, MACS, CisGenome and SISSRs called 33,572, 16,173, 23,406 and 24,189 (at default 0.1% FDR) peaks, respectively. SISSRs is expected to call more peaks than other algorithms with the same 5% FDR level, which is shown in the previous Chapter. Peaks called by GLMNB have a major common parts as peaks called by other algorithms for example SPP(93.8%), MACS(92.4%), CisGenome(94.2%), SISSRs(95.5%). It suggests that GLMNB called the most important peaks. As pointed out by Willbanks and Facciotti in their algorithm evaluation on sensitivity[7], more stringent peaks from some algorithms are almost completely contained in the larger number of calls by others. And calling more peaks gain little in term of sensitivity in their verified binding site comparison study. This is actually the case here. At least 92.4% sharing common peaks between GLMNB and other algorithms suggests GLMNB does not lose too much sensitivity, even though it calls less peaks.

Table 4.1. Peak Number Comparison on FoxA1 with or without negative control(total, percent of GLMNB peaks in common)

FoxA1	GLMNB	SPP	MACS	CisGenome	SISSRs
ChIP only	10,073	33,572(93.8%)	16,173(92.4%)	23,406(94.2%)	24,189(95.5%)
With Input	5,766	33,572(96.9%)	11,778(96.0%)	8,245(92.5%)	11,272(96.7%)

Here I assessed the peak calling accuracy by the percentage of detected peaks matched with at least one motif in 150 bp neighbor region. Figure 4.3(a) plots the percentage of detected peaks matched with at least one FoxA1-related motifs within 150 bp to predicted binding sites among the top 10,073 peaks ranked by FDR values in GLMNB, and other four algorithms on FoxA1 ChIP sample data only. GLMNB's peaks contain higher proportion of peaks matching with a FoxA1 related motif than SPP in all top 10,073 peaks and MACS in top 5,000 peaks, both of which were ranked top two peak callers with highest resolution. SISSRs yields slightly higher (2% on average) percentage of peaks with motifs than GLMNB, partially because it tends to call multiple peaks in tag intensive regions regardless the specific binding or non-specific binding events. Therefore there is more chance to find a motif nearby. Please refer to Figure 4.7(a) for examples. GLMNB is one of the top two peak callers among the five in terms of peak calling accuracy. The spatial resolution is assessed by the distance between predicted binding sites to the nearest motif in Figure 4.3(b). One may notice that GLMNB has a comparable spatial resolution to SPP, the one claimed with best spatial resolution. But GLMNB has wider resolution than MACS, CisGenome and SISSRs on FoxA1 ChIP data without negative control.



(a) (b)

Figure 4.3. ChIP-Seq peak calling comparison with previous methods without negative control sample on FoxA1 ChIP-Seq data. (a) Percentage of detected peaks carrying at least one FoxA1-related motifs within 150 bp of binding sites by GLMNB and previous methods for FoxA1. (b) Histogram of distance between nearest FoxA1 related motif within 150 bp of predicted binding sites.

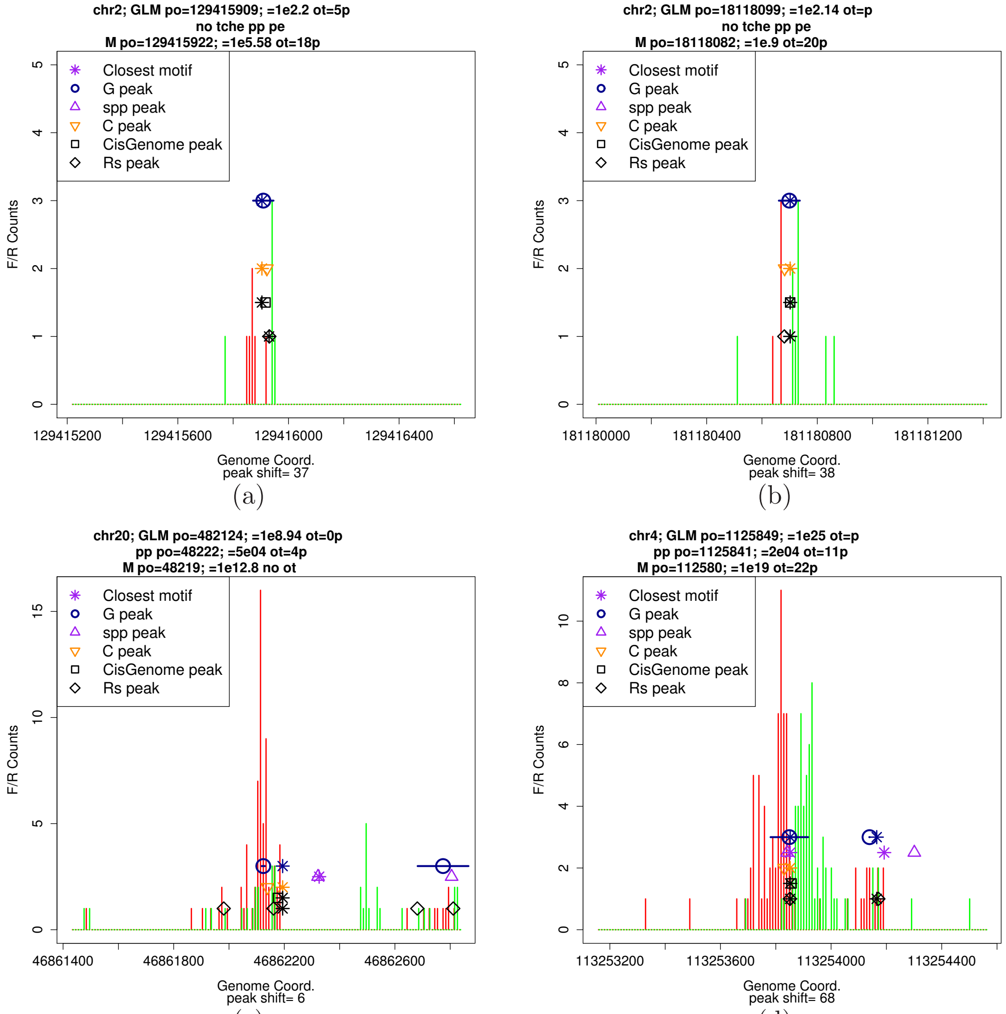
From now on in this section, I want to use some peak examples on FoxA1 ChIP sample data to illustrate the strength and weakness of GLMNB.

Figure 4.4 shows several peak calling examples that illustrate the difference between GLMNB and SPP. Figure 4.4(a) and (b) show two examples of GLMNB peaks not detected by SPP. GLMNB is able to locate peaks at chr2:129,415,909(FDR= $10^{-2.72}$) and chr2:181,180,699(FDR= $10^{-2.14}$) with motifs nearby. MACS, CisGenome and SISSRs did a similar great job. However, SPP did not find them. This is due to low tag counts and no overlapping regions between forward and reverse tags, such that there are no enough tags to provide a significant Pearson's correlation or Chi-square test statistic. It is not unusual in the real data analysis where SPP is not able to locate a binding position when tag counts are low or there is no overlapping region for forward and reverse strands.

Figure 4.4(c) illustrates an interesting case. Around chr20:46,862,124, there is an extremely strong forward strand signal, whereas forward strand signals locate at two relative separated areas, one close to the first peak on the left, and the other one about 400 bp apart on the right. GLMNB, MACS, CisGenome and SISSRs paired the forward signal with the first reverse signal and called peaks at chr20:46,862,124, chr20:46,862,139, chr20:46,862,184 and chr20:46,862,154, matching with a FoxA1 related motif at chr20:46,862,194. However, SPP considered the forward signal should match to the second reverse strand signal, and called a peak at chr20:46,862,322. Given two strategies matching their motifs nearby, it seems no way to tell which strategies are correct. However, the FoxA1 ChIP DNA sample went through a size selection between 150 bp and 400 bp. In this special case, since forward and reverse strand signals are separated by 400 bp, the first strategy is correct.

If one is not convinced, Figure 4.4(d) illustrates a similar example. Regardless the peak identified by all algorithms at around chr4:113,253,849 with a FoxA1 related motif matched, GLMNB and SISSRs identified another peak at chr4:113,254,159 and chr4:113,254,190, with a motif 20 bp to its right and a motif 10 bp to its left, respectively. Both utilized the forward and reverse signals that are close together. However, SPP located a peak at chr4:113,254,300 by pairing the same forward signal as GLMNB and SISSRs but the two reverse tags on the right. SPP's peaks is 110 bp apart from its motif. The strategy used by GLMNB and other algorithms seems more reasonable.

Figure 4.5 shows several peak calling examples that illustrate the difference between GLMNB and MACS. Figure 4.5(a) shows a peak example called by GLMNB, MACS and other algorithms. GLMNB called a peak at chr1:196,770,457 with $FDR = 10^{-8.06}$, with a FoxA1 related motif 8 bp on the right of predicted peak position. MACS called a peak at chr1:196,770,408 with $FDR = 10^{-11.5}$, with a FoxA1 related motif 57 bp on the right. At



(a)

(b)

(c)

(d)

Figure 4.4. Comparison between GLMNB and SPP on FoxA1 ChIP sample only. (a) and (b) show two examples of GLMNB peaks not detected by SPP. (c) An example of SPP peak at the middle of forward and reverse peaks separated by 400 bp, while other algorithms called at another location. Both peaks were matched to FoxA1-related motifs. (d) An example of GLMNB and SPP located two different peak locations that matches to the same FoxA1 motif.

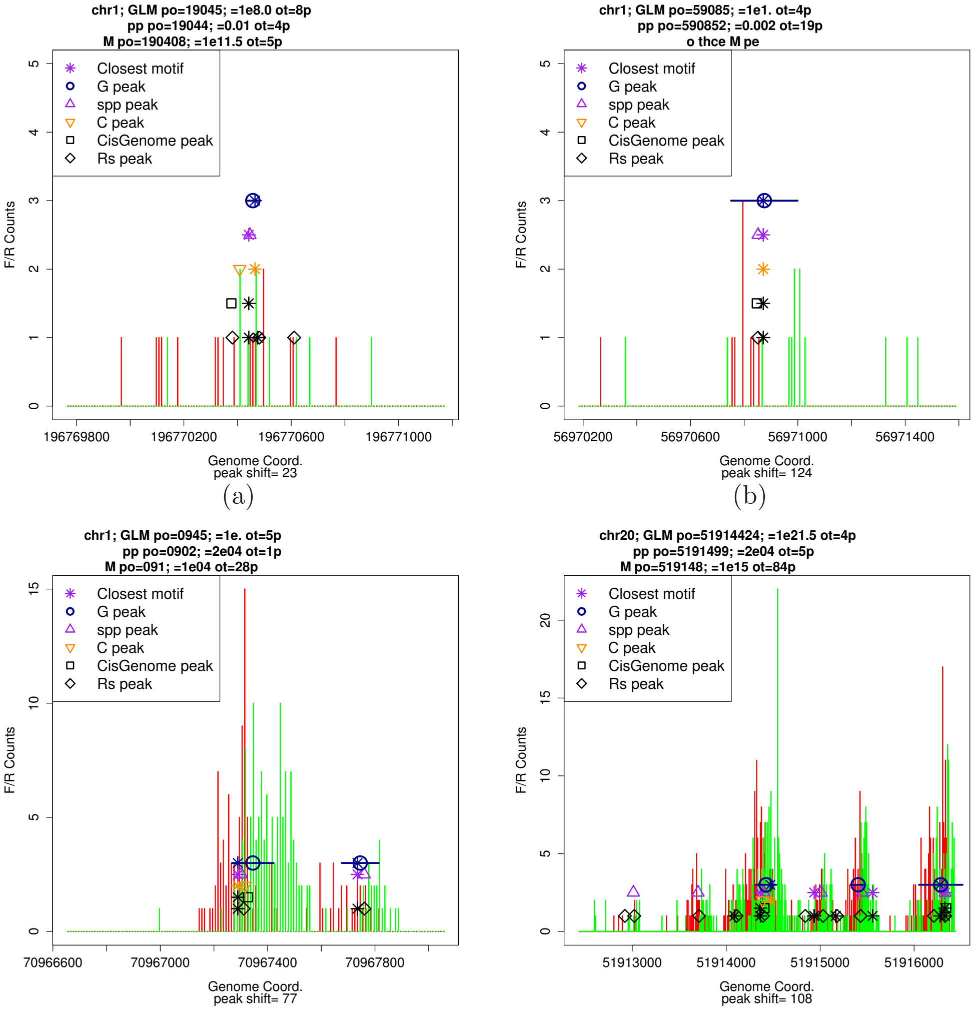
the same time, SPP and CisGenome called a peak at chr1:196,770,446 and chr1:196,770,387, respectively. However, SISSRs called three peaks, two of which have corresponding FoxA1 related motif within 150 bp.

Figure 4.5(b) shows an example of binding sites called by GLMNB and SPP but missed by MACS. GLMNB called a peak at chr17:56,970,895 with FDR= $10^{-1.77}$, matched with a motif 4bp on its left. SPP called a peak at chr17:56,970,852 with FDR=0.2%, matched with the same FoxA1-related motif but 19bp on its right. CisGenome and SISSRs were able to call a peak close to the identical FoxA1-related motif. However, MACS does not call a peak because the tag count in this region is extremely large regardless forward and reverse tags form two peak shapes and there is a FoxA1 related motif between forward and reverse peaks.

In the FoxA1 study, we found that MACS tended to call peaks in larger sizes due to its automatic peak interval expansion procedure. This is why often GLMNB and SPP identify multiple binding sites in a region but only a single MACS peak was found at the strongest peak position. A desirable feature of GLMNB is its capability to call nearby peaks. As shown in Figure 4.5(c), forward and reverse tags form two obvious pairs of peaks separated by about 400 bp, with the stronger pair on the left and a weaker pair on the right. MACS is able to locate the stronger peak at chr17:70,967,317 with a extremely significant FDR(10^{-304}), thanks to MACS modeling mechanism: model combined tag counts from both pairs against background Poisson distribution regardless the amount of peak shapes. There is a FoxA1 motif found at 28 bp on its left. But it missed the peak on the relatively less significant peak at around chr17:70,967,700. GLMNB called two peaks in this region, one centered at 70,967,345 (left blue circle) with $\theta = 77$bp, and the other centered at 70,967,747 (right blue circle) with $\theta = 44$bp. The FDR for the two peaks were $10^{-33.6}$ and $10^{-8.6}$, respectively. GLMNB is capable of capturing multiple peaks within a local region, because the method is a model based approach that fits the data with a specific binding profile. For each pair of forward and reverse strand peaks, GLMNB evaluates its binding significance in each sliding window. Both GLMNB peaks can be matched to two FoxA1-related motifs (blue stars, 56bp and 7bp to the left of the predicted binding sites, respectively). SPP was also able to call two peaks in this region. Two SPP peaks (purple upper triangles) were found, one at 70,967,302 with a motif (left purple star) 13bp to the left, the other at 70,967,760 with a motif (right purple star) 25bp to the left. As an extreme example (Figure 4.5(d)), GLMNB detected 3 peaks in a 5kb region, with FDR ranging from 10^{-5} to $10^{-21.5}$, on chr20:51,913,000-51,918,000. Two of the three GLMNB peaks contained at least one FoxA1-related motifs. SPP identified six peaks in the same region, four of which are within 30bp of GLMNB peaks. Two of three peaks not called by GLMNB do not contain any FoxA1-related motif. In contrast, MACS only called the most significant peak at chr20:51,914,386, with FDR=10^{-135}. CisGenome is able to call two peaks at the two most significant positions at chr20:51,914,391 and chr20:51,917,510 matched to two FoxA1 motifs. However, SISSRs identifies 11 binding peaks, most among the five algorithms. But only six of them can be matched with a FoxA1 motif. In the area containing more than average tags, SISSRs tends to call more peaks than GLMNB and other algorithms.

Figure 4.6(a) illustrates an peak called by GLMNB but missed by CisGenome. There are two pairs of forward and reverse signals nearby each other, with two reverse peaks separated by roughly 200 bp. All five algorithms were able to identify the stronger peak on the right. GLMNB called the peak at chr20:54,743,954, 12 bp to FoxA1 related motif on the left, the one closest to the motif among the five algorithms. However, only GLMNB and SISSRs were able to located the other peak for the left pair of signals. CisGenome was not able to locate it because it was not able to distinguish the two peaks using its default 100 bp sliding window. MACS and SPP were not able to distinguish it due to the reason discussed above.

Figure 4.6(b), (c) and (d) illustrate three situations GLMNB did not call a peak while CisGenome and MACS (and/or SPP, SISSRs) called a peak. There were no FoxA1 motif matched with any peaks called in these three examples. At around chr1:65,186,692 in Figure4.6(b), the tag counts in forward and reverse strand are not comparable(1 forward tags and 11 reverse tags). GLMNB refused to call it because of asymmetric tag counts between two strands. However, SPP, MACS and CisGenome called it with a moderate FDR(2% for SPP and $10^{-10.3}$ for MACS). At around chr1:149,731,382 in Figure4.6(c), the forward



(a) (b)

(c) (d)

Figure 4.5. Comparison example of FoxA1 ChIP sample data between GLMNB and MACS. (a) An example of peak detected by both GLMNB and MACS. (b) An example of GLMNB peak undetected by MACS. (c) An example of multiple GLMNB peaks but called a single one by MACS. Both peaks were matched to FoxA1-related motifs. (d) An extreme example of multiple GLMNB peaks but called a single one by MACS. Three peaks are called by the GLMNB but only the strongest one is called by MACS.

tags locate on the right of reverse tags, which does not match the ChIP-Seq experiment assumption. Therefore, GLMNB refused to call it with negative θ. However, MACS and CisGenome called it because both methods do not use tag direction information. At around chr17:362,269 in Figure4.6(d), tags on both strands show an asymmetric shape. GLMNB failed to call it. However, MACS called it with a FDR= $10^{-3.6}$ and locate a peak at 362,269, close to CisGenome and SISSRs peak locations. In summary, GLMNB is able to locate two peaks located within 200 bp but will not call peaks if tags in both strands do not locate in the correct position or form a nearly symmetric shape.

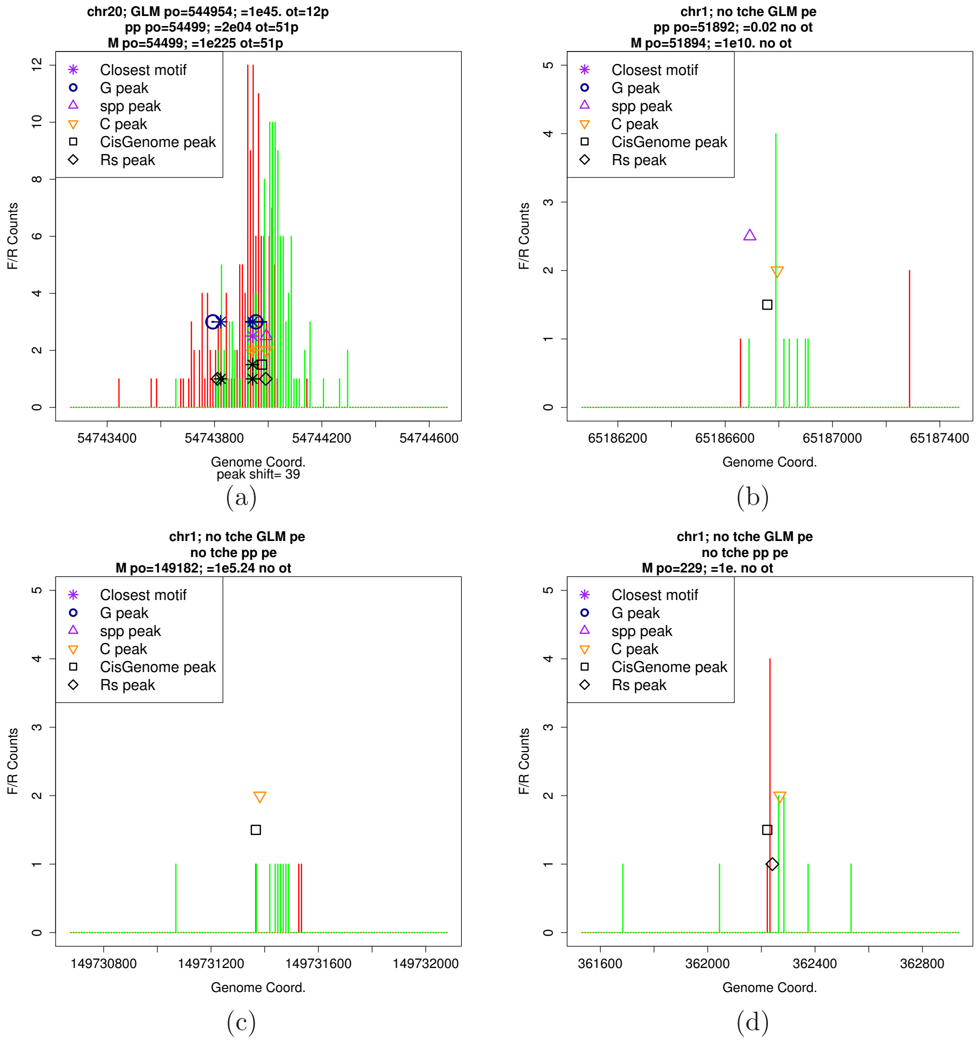
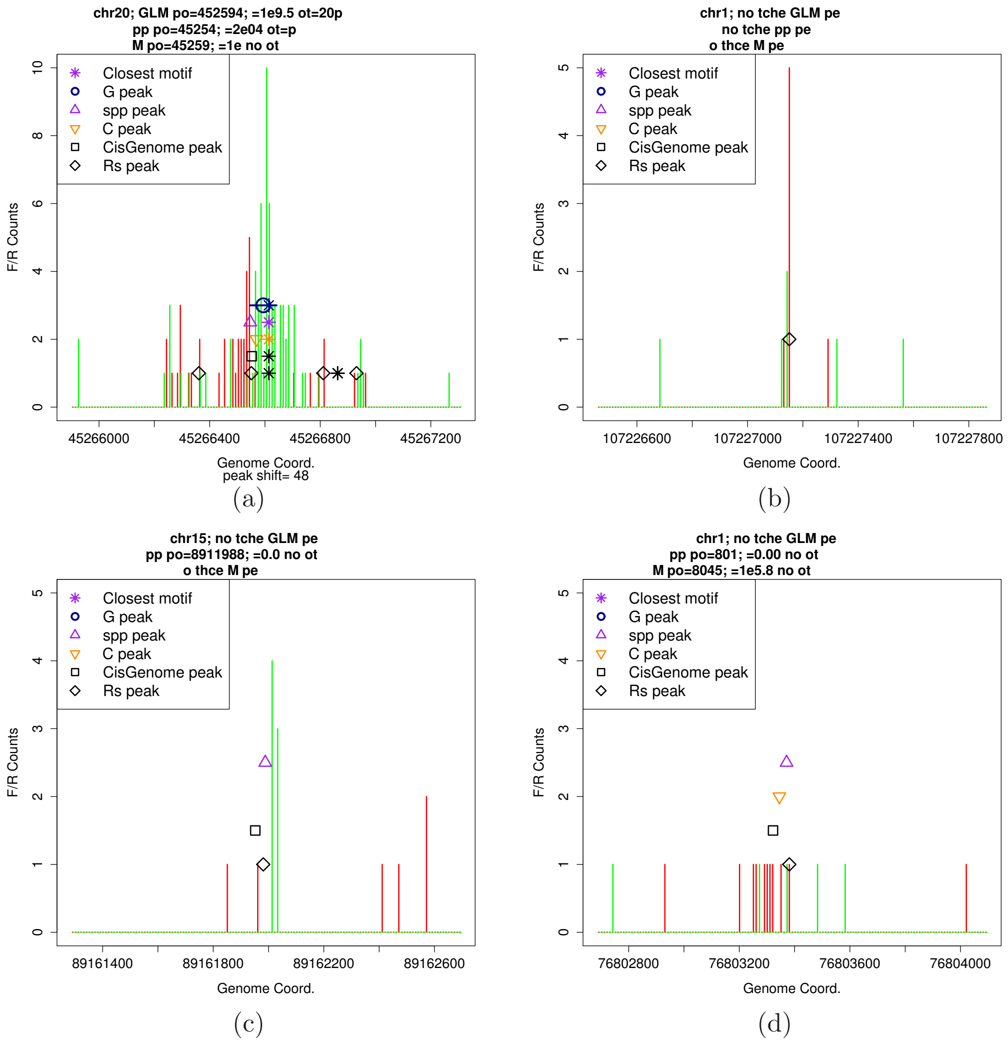


Figure 4.6. Comparison between GLMNB and CisGenome on FoxA1 ChIP sample only. (a) An peak example called by GLMNB but missed by CisGenome. and Three examples called by CisGenome but missed by GLMNB because of (b) asymmetric tag counts, (c) switched forward and reverse strand positions and (d) asymmetric tag shapes.

Figure 4.7(a) illustrates a peak where multiple SISSRs peaks were called whereas other four algorithms only located one. There is a strong signal at chr20:452,665,614. GLMNB called a peak at chr20:452,665,594 with FDR= $10^{-9.56}$, with the smallest distance(20 bp) to a FoxA1 related motif among the five algorithms. However, SISSRs located three extra peaks by utilizing very weak signals nearby, which were considered as background noise by other four algorithms. Only one of these three additional SISSRs peaks can be matched with a FoxA1 motif 50 bp apart.

Figure 4.7(b), (c) and (d) illustrate three situations GLMNB did not call a peak while SISSRS and other algorithms called a peak. There were no motif found in all three examples. For example, at around chr1:107,227,170 in Figure 4.7(b), forward tags are on the right of reverse tags, showing a switched and yet asymmetric shape. Only SISSRs found a peak under such an irregular condition. At around chr15:89,161,988 in Figure 4.7(c), the observed tags displayed an asymmetric shape, 2 forward tags and 7 tags on the right. GLMNB refused to call it because the forward tag count did not exceed the tag count threshold. At around chr16:76,803,371 in Figure 4.7(d), GLMNB refused to call a peak because of its evenly and widely distributed tags for both strands, which might be due to non-specific binding event on the background. However, SISSRs and other three algorithms called a peak without a motif matching within 150 bp.

In summary, GLMNB called relatively less peaks than other algorithms. But it does not reduce its sensitivity. GLMNB refuse to call a peak when forward and reverse tags form asymmetric shapes, or evenly distribute in a wide region. According to the peak examples shown above, GLMNB is able to call multiple peaks even when they are only 200 bp apart.



(a)

(b)

(c)

(d)

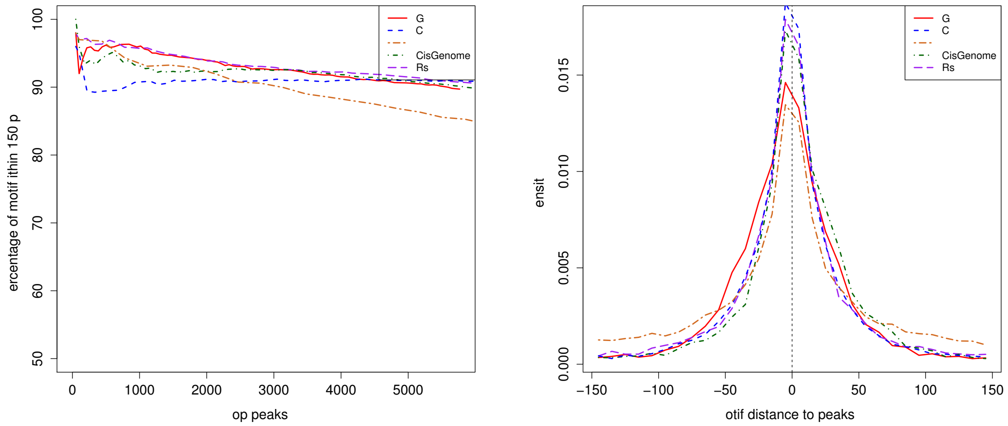
Figure 4.7. Comparison between GLMNB and SISSRs on FoxA1 ChIP sample only. (a) An peak example where SISSRs called more peaks than other four algorithms. and Three examples called by SISSRs but missed by GLMNB because of (b) asymmetric tag tag shapes between both strands, (c) asymmetric tag counts, and (d) evenly distributed tags.

4.3 GLMNB peak calling comparisons with other algorithms on FoxA1 ChIP sample with negative control

By showing the strength of GLMNB on FoxA1 ChIP sample data analysis in section 4.2, we are ready to consider negative control data and reduce false positives. We applied GLMNB on FoxA1 ChIP data with negative control data using the model described in Chapter 2. As listed in Table 4.1, all five algorithms called less peaks after considering negative control data. GLMNB called 5,766 peaks, 57.2% of peaks from ChIP sample data only, whereas CisGenome cut the most peaks and called only 8,245 peaks, 35.2% of peaks before adding negative control. When comparing the common peaks, GLMNB shares 92.5% of its peaks with CisGenome, and even more with SPP(96.9%), MACS(96.0%) and SISSRs(96.7%). This suggests GLMNB' consistent performance and high sensitivity comparing to other methods.

Figure 4.8(a) plots the percentage of detected peaks carrying at least one FoxA1-related motifs within 150 bp to predicted binding sites among the top 5,766 peaks ranked by FDR values in GLMNB, and other four algorithms on FoxA1 ChIP data with negative control. GLMNB's peaks contain higher proportion of peaks matching with a FoxA1 related motif than SPP in all top 5,766 peaks and MACS in top 3,500 peaks, both of which were ranked top two peak callers with highest spatial resolution. SISSRs has slightly higher (0.5-1% on average) percentage of peaks with motifs than GLMNB, particularly because it tends to call multiple peaks in tag intense regions and therefore there is more chance to find a motif nearby. Therefore, GLMNB is one of the top two peak callers among the five in terms of peak calling accuracy. The spatial resolution is assessed using the distance between predicted binding sites to the nearest motif in Figure 4.8(b). GLMNB has a comparable spatial resolution as SPP, the one claimed with best spatial resolution. But GLMNB has wider resolution than MACS, CisGenome and SISSRs on FoxA1 ChIP data with negative control.

Before going to further pairwise comparison between GLMNB and other four algorithms, I want to address the question I proposed in Figure 1.4 in Chapter 1. Remember it was shown there that there is a moderate correlation(0.576) between ChIP sample data and negative control data with respect to tag counts. This greater-than-zero correlation may be due to variable binding affinity and non-specific binding events in different genome locations. However, it is the algorithms' responsibility to distinguish such a non-specific binding event



(a)

(b)

Figure 4.8. ChIP-Seq peak calling comparison with previous methods with the negative control sample on FoxA1 ChIP-Seq data. (a) Percentage of detected peaks carrying at least one FoxA1-related motifs within 150 bp to binding sites by GLMNB and previous methods for FoxA1. (b) Histogram of distance between nearest FoxA1 related motif within 150 bp to predicted binding sites.

from the TF specific binding. Let us see whether GLMNB achieved such separation on FoxA1 ChIP-Seq data with negative control or not. In Figure 4.9, I plotted the sample tag counts scatter plot between ChIP sample and input sample as shown in Figure 1.4. For those windows containing at least one binding site, I plotted a red dot. For those windows without a predicted binding site, I plotted a black dot. GLMNB classified all 10-kb windows along the entire genome into two partitions, binding events with Pearson's correlation between tag counts in ChIP sample and input sample of 0.541, and a non-binding events with correlation of 0.775. It is obvious that most windows in top left regions were classified as binding status, since many more tags were discovered in ChIP sample compared to the negative control sample. A lot of windows plotted in bottom right regions were classified as non-specific binding events, as comparable or even more tags were discovered in the negative control sample than the ChIP sample. And more proportion of windows in the far right end on the bottom of figures were classified as non-specific binding. Even though this is a rough plot based on a wide (10 kb) non-overlapping window, which does not represent tag distribution or shapes, it is plotted independently of GLMNB algorithm. Therefore, it measures the separation between specific binding events and non-specific binding events.

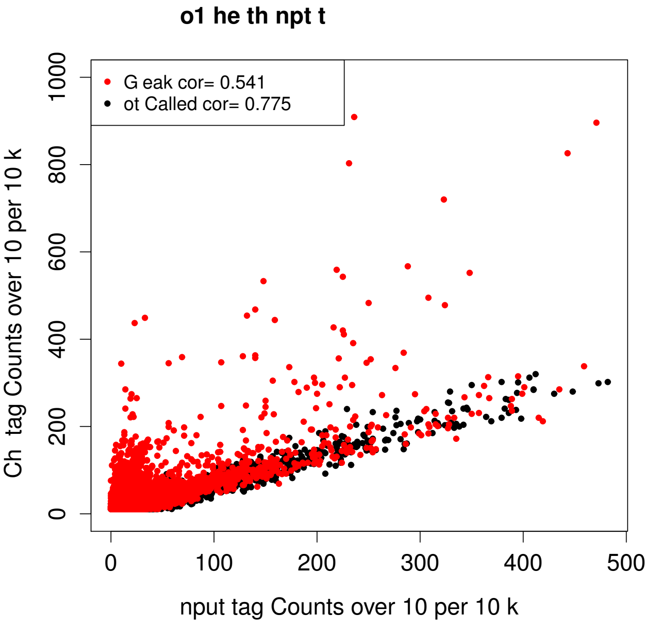


Figure 4.9. Tag counts scatter plot between ChIP sample and input sample after GLMNB peak calling.

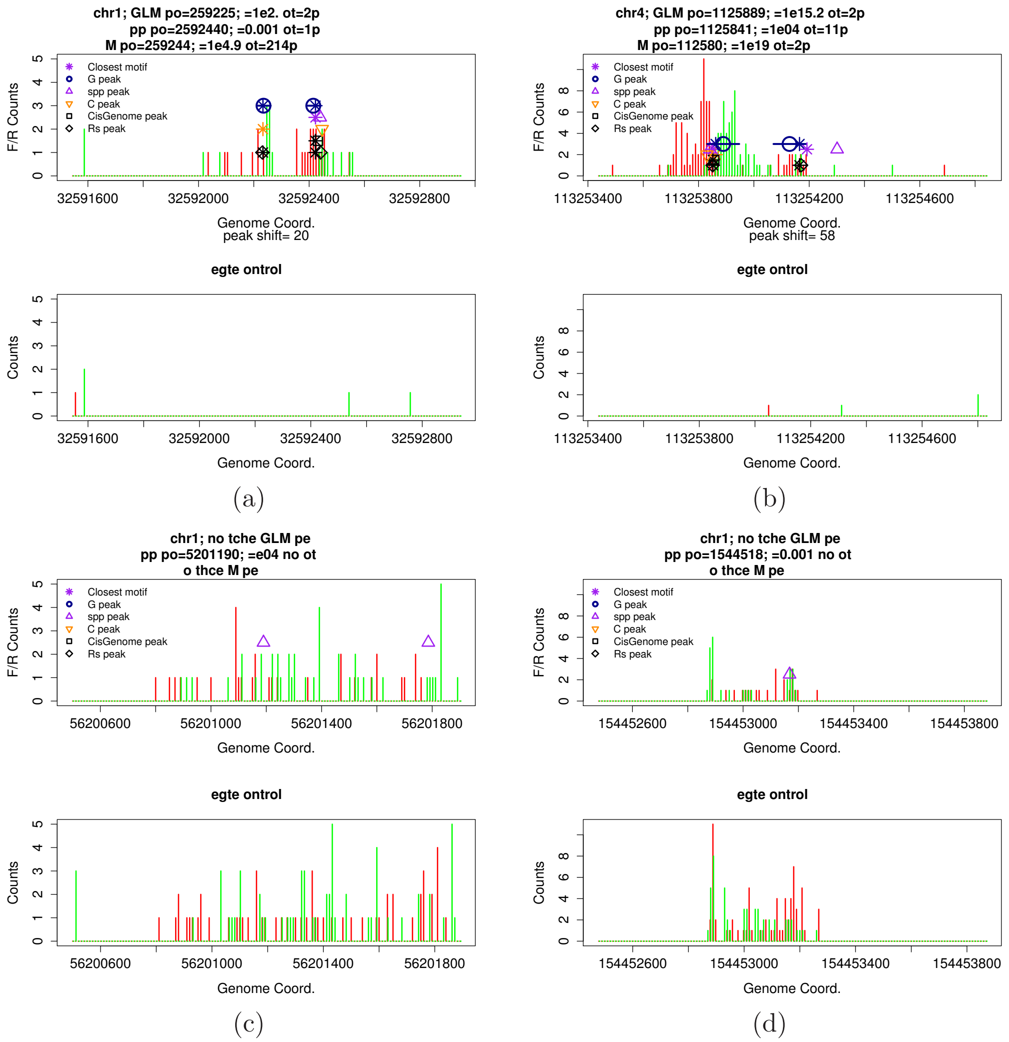
From now on, I will illustrate the different aspects focused by GLMNB and the other algorithms using some representative peak calling examples. I also plotted tags per 10bp bin in negative control data at the same genome location as the ChIP sample on the bottom of each sub-figure. Figure 4.10(a) and (b) give two peak examples called by GLMNB but missed by SPP. At around chr13:32,592,235 and chr13:32,592,417 in Figure 4.10(a), GLMNB called two peaks with moderate FDR $10^{-2.63}$ and $10^{-2.31}$ and two motifs nearby. SPP, MACS, CisGenome, SISSRs were able to call peaks at the second location but not the first location. At around chr4:113,253,889 in Figure 4.10(b), all five algorithms were able to locate a peak with a FoxA1 motif within 150 bp. GLMNB and SISSRs were able to utilize forward and reverse signal at around chr4:113,254,190. However, SPP utilized the same forward signal but a much weaker reverse signal located at around chr4:113,254,290 and called a peak in less plausible position.

Negative control data in the two examples above contained barely any tags. Figure 4.10(c) and (d), however, give two peak examples with strong correlated tags in negative control data. For example, in the region around two SPP peak positions (chr17:56,201,190 with significant FDR 0.01% and chr17:56,201,790), there is comparable or even more tags in the negative control data compared to the ChIP sample data. SPP called two peaks, which should be classified as false positives. There was no FoxA1 related motif close to the two peaks. GLMNB and other 3 algorithms did not call any peaks. At SPP peak position (chr1:154,453,168 with FDR=0.1%), there are even stronger forward and reverse tags in the negative control data compared to the ChIP sample data. It should also be classified as false positive. There were no FoxA1 related motifs close to this SPP peak and no peaks called by other algorithms either.

Figure 4.11(a) and (b) illustrate two peak examples called by GLMNB but missed by MACS. Figure 4.11(c) and (d) show two peak examples called by MACS but missed by GLMNB. Let us first look at Figure 4.11(a). There are two obvious peaks apart by 550 bp. GLMNB was able to call both peaks at chr10:7,315,467 and chr10:7,315,957, with FoxA1 motifs 2 bp and 1 bp on their right. SPP and SISSRs were also able to locate two peaks. However, MACS and CisGenome were only able to call peaks at the first position. MACS failed to call the second peaks due to its automatic binding area extension and tag merging strategy. As a result, MACS called the left one with an extreme FDR(10^{-317}).

Figure 4.11(b) shows another scenario, where negative control contains noisy tags not necessarily correlated with ChIP sample tags. GLMNB, SPP, CisGenome and SISSRS were able to call this peak with relatively low significance. However, MACS completely missed it.

Figure 4.11(c) illustrates the scenario where forward and reverse tags in ChIP sample are asymmetric even though there are few tags in the negative control data. Only MACS called



(a)

(b)

(c)

(d)

Figure 4.10. Comparison between GLMNB and SPP on FoxA1 data with the negative control data.

a peak at chr1:65,706,018 with FDR= $10^{-12.2}$ but there is no FoxA1 related motif nearby. GLMNB refused to call it because the forward tag count does not exceed the minimum tag count threshold.

Figure 4.11(d) illustrate the scenario where there is strong non-specific binding affinity in negative control data at the area of strong ChIP signal. MACS and SPP called a peak at chr17:55,273,416 and chr17:55,273,426 with a motif 25 bp and 15 bp on their right, respectively. Due to the obvious strong similarity on tag counts and shape, GLMNB classified such a region as negative signals, even though there is a motif shown nearby.

In summary, GLMNB keeps its capability of calling nearby multiple peaks with negative control data. It properly adjusts the FDR level to account for the non-specific binding observed in negative control data, rather completely ignoring the signals as MACS does. It also reduces false positives by refusing peaks if forward and reverse tags in ChIP data show asymmetric shape or counts or there is a strong similarity on tag counts and shapes between ChIP sample and negative control sample.

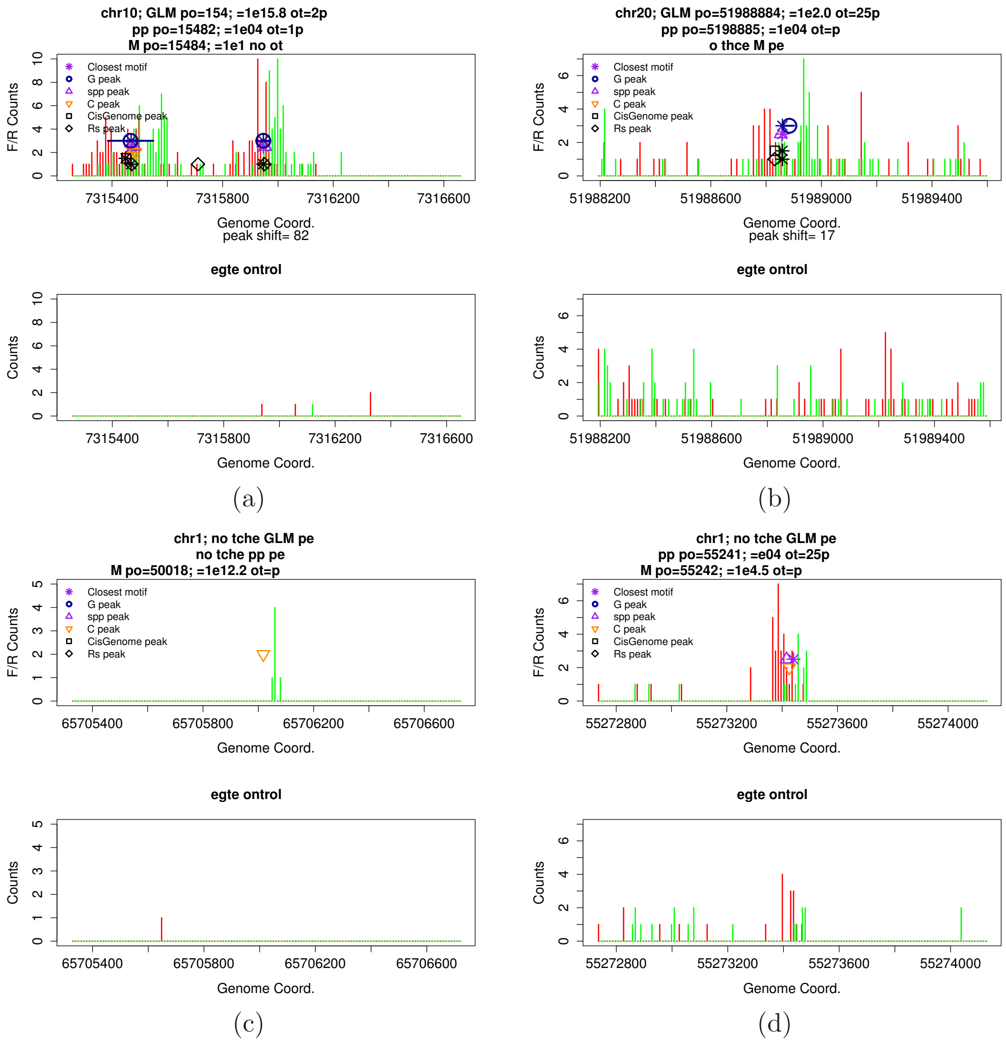
Figures 4.12(a) and (b) illustrate two peak examples called by GLMNB but missed by CisGenome. GLMNB was able to locate a peak at chr20:51,729,874(FDR= $10^{-2.96}$) with a FoxA1 motif 45 bp on its right, after taking the relatively noisy tags in negative control into account in Figure 4.12(a). SPP and MACS is able to locate it too. But CisGenome did not call it a peak.

Figure 4.12(b) shows an similar scenario, where the negative control contains a few tags but not similar in size or shape to those in the ChIP sample. GLMNB was able to call it as MACS and SPP did. But CisGenome and SISSRs failed to call it.

Figure 4.12(c) and (d) show two peak examples called by CisGenome but missed by GLMNB because forward and reverse tags show strong asymmetric shape or size. Forward tags in Figure 4.12(d) even shows a suspected incorrect tag counts caused by amplification bias in ChIP-Seq PCR step. Only CisGenome called it as a binding position.

In summary, GLMNB is able to call a peak after taking into account of the relative noisy background tags appearing in the negative control data. However, GLMNB will not call a peak if forward and reverse tags forms obviously asymmetric size or shape even though there are rare tags in the negative control data. CisGenome fails to make these two contributions.

In the following two cases, 1) noisy but evenly distributed tags in negative control data as shown in Figure 4.13(a) and 2) different size or shape between background tags and ChIP sample tags as shown in Figure 4.13(b), GLMNB was able to call peak after accounting for

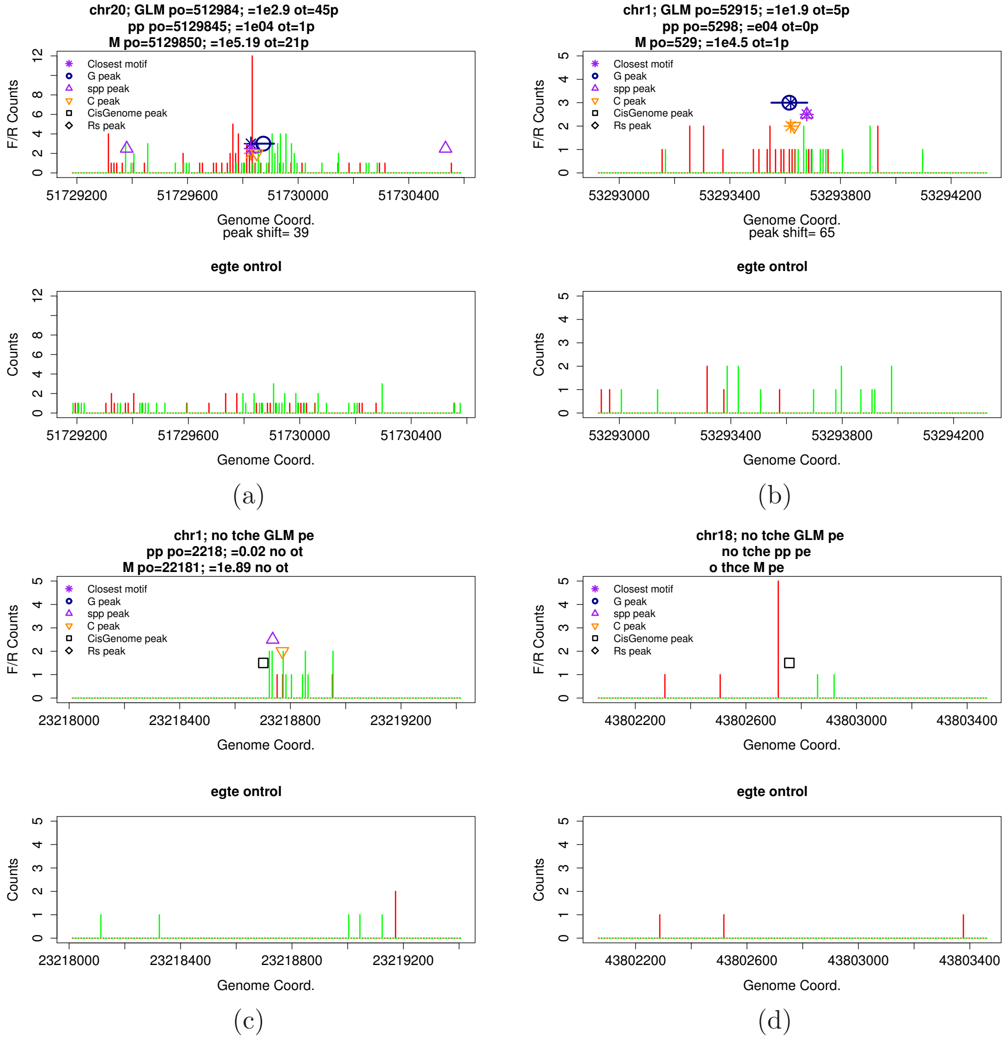


(a) (b)

(c) (d)

Figure 4.11. Comparison between GLMNB and MACS on FoxA1 data with negative control. (a) and (b) illustrate two peak examples called by GLMNB but missed by MACS. (c) and (d) show two peak examples called by MACS but missed by GLMNB.

the background noise. However, SISSRs failed to call a peak in these cases. Again with the asymmetric size or shape of forward tags and reverse tags in ChIP sample as shown in Figures 4.13(c) and (d), GLMNB refuses to call a peak even though there are few non-specific binding events appearing in the negative control data.



(a)

(b)

(c)

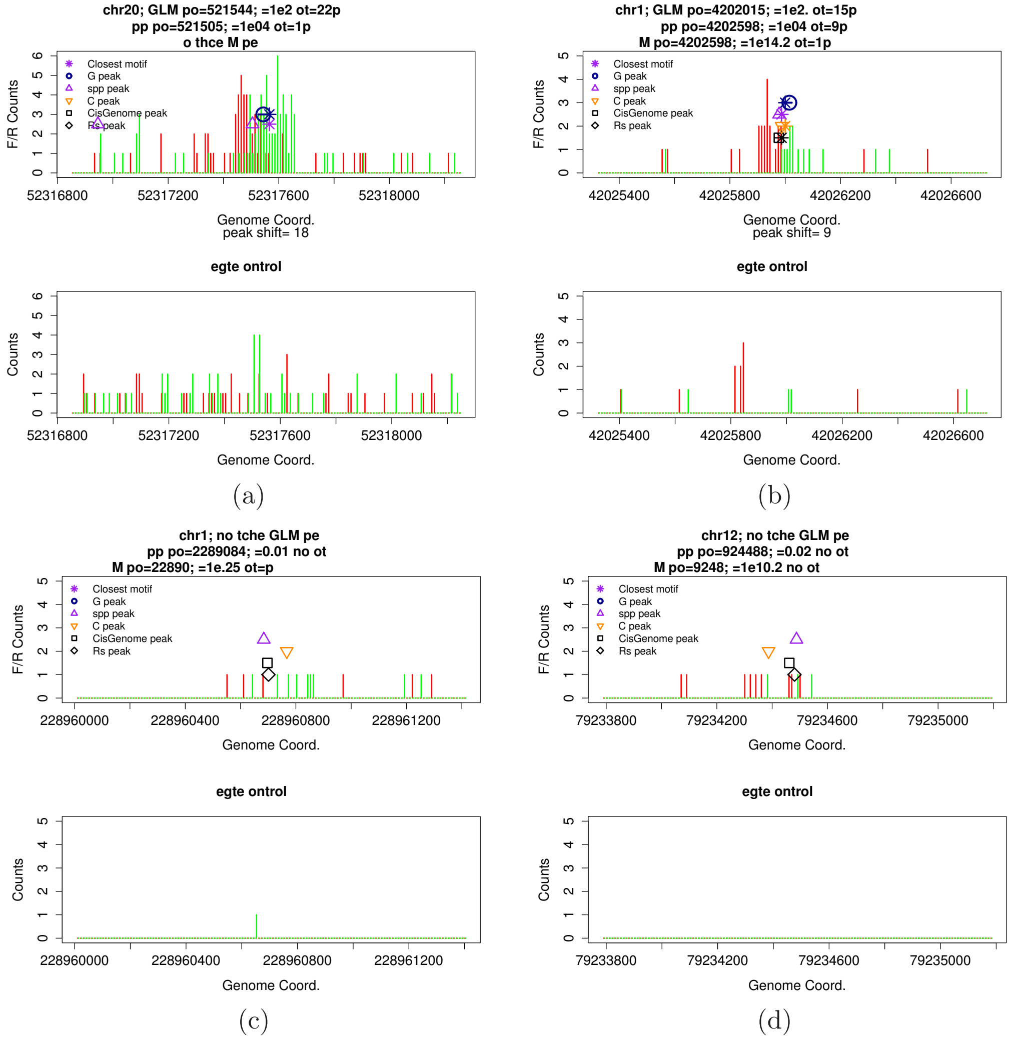
(d)

Figure 4.12. Comparison between GLMNB and CisGenome on FoxA1 data with negative control. (a) and (b) illustrate two peak examples called by GLMNB after accounting for non-specific background noise shown in negative control data, but missed by CisGenome. (c) and (d) show two peak examples called by CisGenome but missed by GLMNB due to asymmetricity in shape or size between forward and reverse tags in ChIP sample.

4.4 Peak calling comparisons on GLMNB and other algorithms on ETV1, RBPJ, EBNA2

.

After going through all detailed pairwise comparison between GLMNB and SPP, MACS,



(a)

(b)

(c)

(d)

Figure 4.13. Comparison between GLMNB and SISSRs on FoxA1 data with negative control. (a) and (b) illustrate two peak examples called by GLMNB after accounting for non-specific background noise shown in negative control data, but missed by SISSRs. (c) and (d) show two peak examples called by SISSRs but missed by GLMNB due to asymmetry in shape or size between forward and reverse tags in ChIP sample.

CisGenome or SISSRS on FoxA1 ChIP data with or without negative controls in sections 4.2 and 4.3, I would like to walk you through results on ETV1, EBNA2 and two RBPJ biological replicates described in section 2.1 and evaluate the performance of GLMNB compared to other peak calling algorithms.

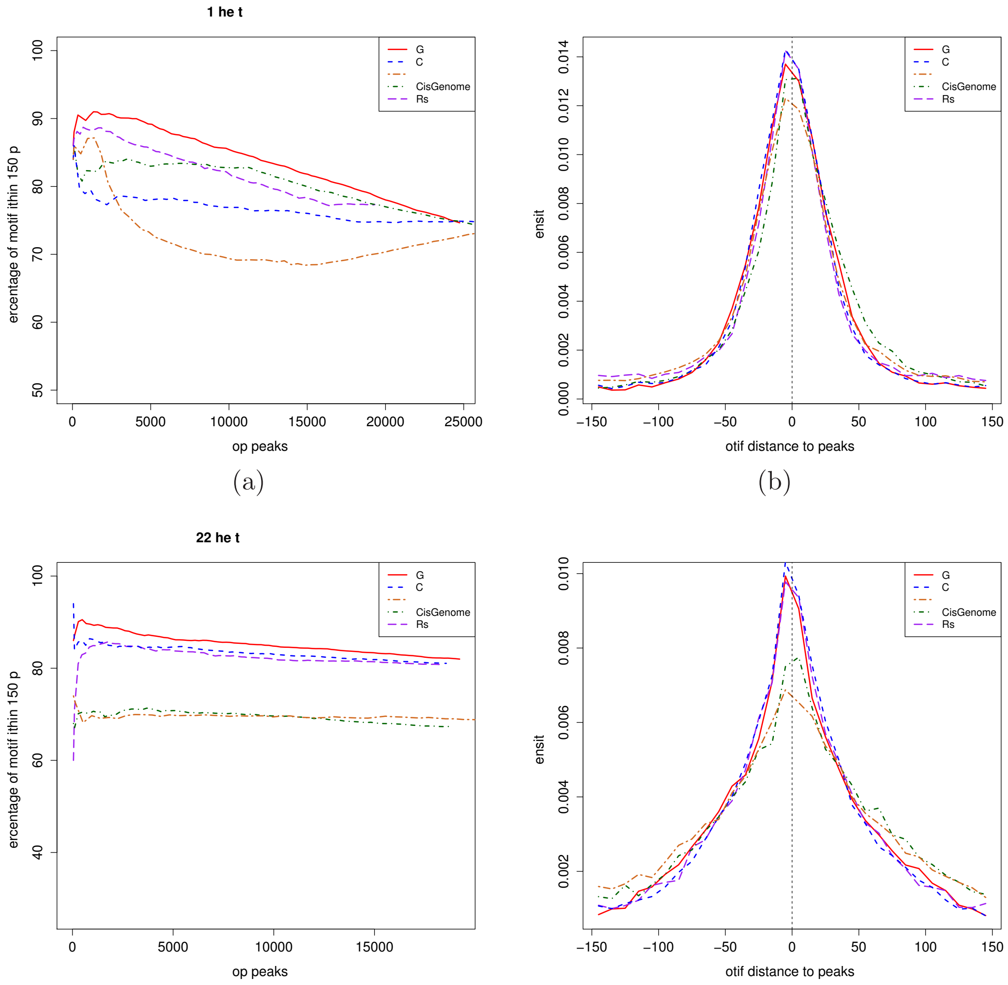
Table 4.2 lists numbers of peaks called by GLMNB, SPP, MACS, CisGenome and SISSRs on ETV1, EBNA2 and two biological replicates of RBPJ ChIP-Seq data only. There seems to be no strong patterns in terms of peak amounts, except that SPP called the most amount peaks among the five algorithms. GLMNB called significant less amount of peaks (24,739) than SPP and MACS for ETV1 ChIP-Seq data, but a similar number to CisGenome and SISSRs. GLMNB's results shared 64.4% to 86.7% peaks in common with the other four algorithms. GLMNB called 19,230 peaks on EBNA2 ChIP-Seq data only, similar amount as that by MACS, CisGenome and SISSRs, but significantly lower than SPP(42,551). GLMNB shared 49.1% to 80.6% peaks with the other four algorithms.

Although RBPJ-1 and RBPJ-2 are two biological replicates on RBPJ protein with a ChIP sample, they differ in the tag amounts in the ChIP sample data (7.6 million tags versus 8.9 million tags). As a result, many more peaks were discovered in RBPJ-2 rather than RBPJ-1 by all programs except for SPP. Peaks called by GLMNB share 45.5% to 78.5% common peaks with the other four algorithms on RBPJ-1 data set. Peaks called by GLMNB share much higher, 61.3% to 79.4%, common peaks with the other four algorithms on RBPJ-2 data set.

Table 4.2. Peak Number Comparison with ChIP sample only(total, percent of GLMNB peaks in common)

ChIP name	GLMNB	SPP	MACS	CisGenome	SISSRs
ETV1	24,739	43,010(82.8%)	34,658(86.7%)	31,648(83.5%)	40,441(82.0%)
EBNA2	19,230	42,551(80.6%)	18,567(69.6%)	18,785(64.2%)	18,318(49.1%)
RBPJ-1	11,612	51,104(78.5%)	11,312(50.1%)	12,638(45.5%)	11,057(46.2%)
RBPJ-2	24,071	46,400(79.4%)	38,539(76.5%)	32,155(71.1%)	12,603(61.3%)*

Although GLMNB called less peaks than SPP and other algorithms, it achieves good sensitivity assessed by percentage of top peaks with a protein binding motif within 150 bp around predicted binding sites. All GLMNB top peaks on ETV1 and EBNA2 ChIP data have the highest proportion of peaks with at least one ETV1 motif found within 150 bp among the five algorithms as shown in Figure 4.14(a) and (c). The spatial resolution is assessed by peaks' distance to the closest motif if any within 150 bp region for GLMNB and other peak calling algorithms. GLMNB achieved about the same spatial resolution as MACS and SISSRs and superior resolution to SPP on ETV1 and EBNA2 ChIP only datasets as shown in Figure 4.14(b) and (d).



(a)

(b)

(c)

(d)

Figure 4.14. ChIP-Seq peak calling comparison with previous methods without negative control sample on ETV1 and ENBA2 ChIP-Seq data. Percentage of detected peaks carrying at least one RBPJ-related motifs within 150 bp to binding sites by GLMNB and previous methods for ETV1(a) and EBNA2(c). Histogram of distance between nearest motif within 150 bp to predicted binding sites for ETV1(b) and EBNA2(d).

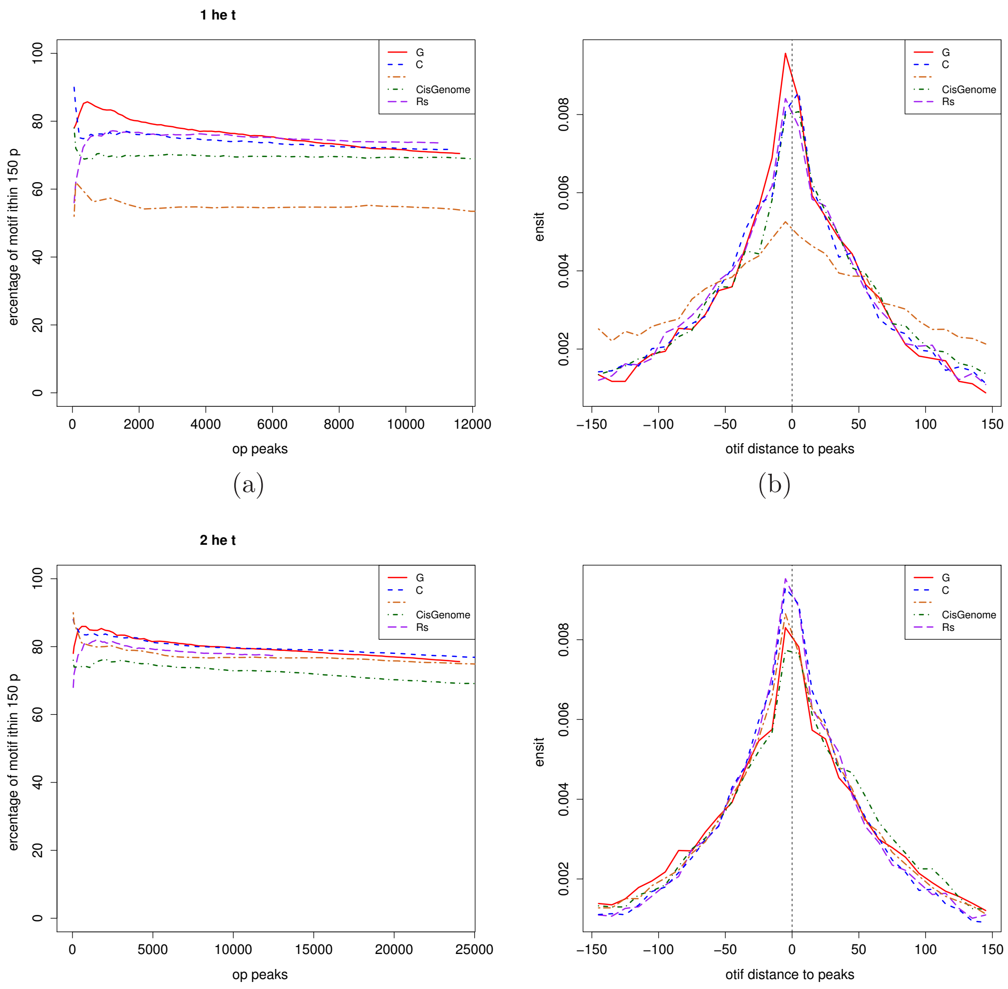
Even though the tag counts vary a lot between RBPJ-1 and RBPJ-2, two biological replicates from the same experiment for RBPJ protein, the performance of GLMNB is consistent in both datasets in terms of peak calling sensitivity and spatial resolution as shown in Figure 4.15. GLMNB achieved highest and consistent sensitivity in both replicates at around 80% of peaks with motif found within 150 bp regions. It also achieved the best spatial resolution in RBPJ-1 and lower but very similar resolution to MACS, SISSRs and SPP. In comparison, SPP does not have an consistent performance in spatial resolution.

After examining the performance of five peak calling algorithms on the four datasets with ChIP sample only, I want to further evaluate their performance after accounting for negative control data. Table 4.3 lists the peak numbers called by all five algorithms on ETV1, EBNA2, RBPJ-1 and RBPJ-2 with negative control data. GLMNB called fewer peaks after accounting for negative controls for all datasets except for RBPJ-2. There were 2,882 more peaks called in RBPJ-2 after accounting for the negative control data. GLMNB shared 60.2% to 91.3% of common peaks with the other four algorithms on ETV1, EBNA2 and RBPJ-1 and RBPJ-2 ChIP-Seq datasets.

Table 4.3. Peak Number Comparison with Negative Control(total, percent of GLMNB peaks in common)

ChIP name	GLMNB	SPP	MACS	CisGenome	SISSRs
ETV1	14,922	36,701(88.6%)	28,432(91.3%)	20,431(87.1%)	19,437(72.3%)
EBNA2	15,840	29,231(79.1%)	17,392(73.0%)	17,545(76.4%)	16,968(60.2%)
RBPJ-1	6,440	32,639(86.0%)	8,742(68.6%)	9,273(69.9%)	7,805(73.7%)
RBPJ-2	26,953	40,011(73.3%)	29,324(75.8%)	27,395(70.4%)	8,396(62.7%)*

I also evaluated the peak calling sensitivity by assessing the percentage of peaks with motif in 150 bp region on ETV1 (Figure 4.16(a)), EBNA2(Figure 4.16 (c)), RBPJ-1 (Figure 4.17(a)) and RBPJ-2 (Figure 4.17(c)) with negative control data. GLMNB kept its superior peak calling sensitivity over other four algorithms on all four datasets after accounting for negative control. The spatial resolution of GLMNB were compared with other four peak calling algorithms in Figure 4.16(b), (d) and Figure 4.17(b), (d). GLMNB also achieved as



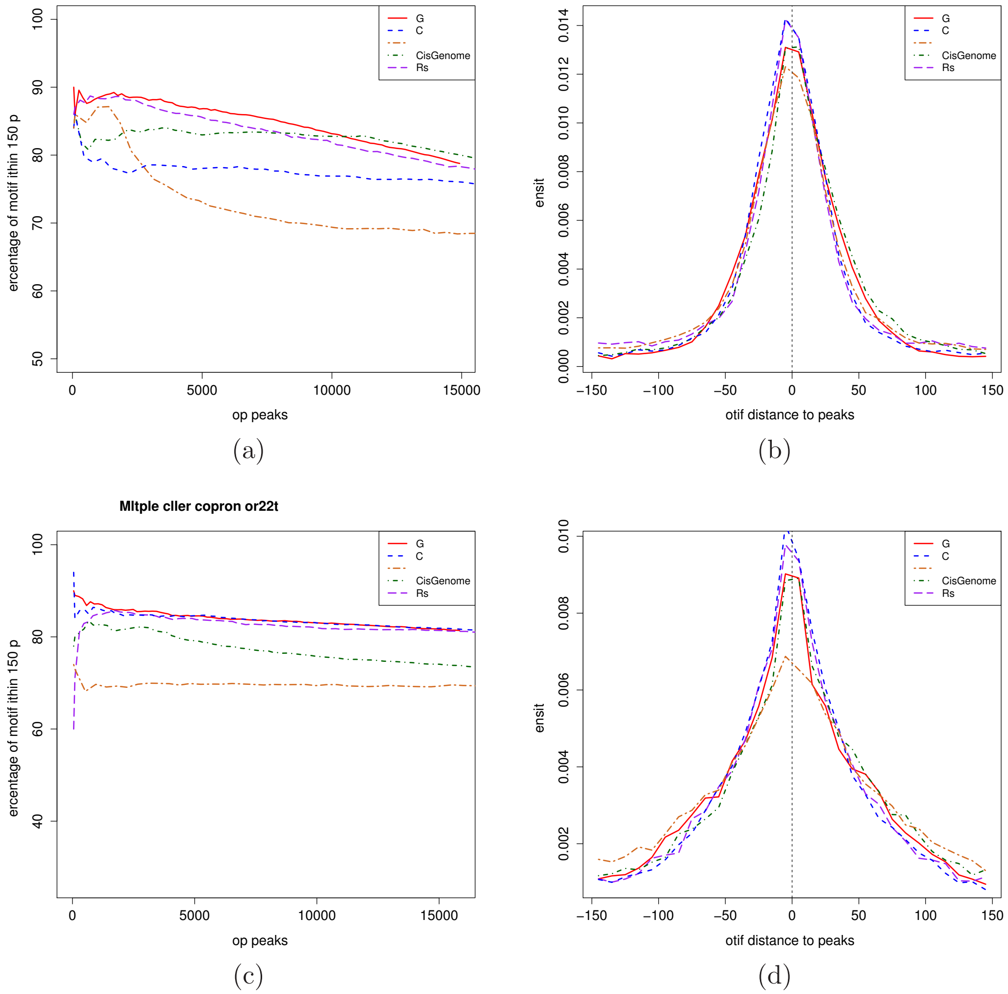
(a) (b)

(c) (d)

Figure 4.15. ChIP-Seq peak calling comparison with previous methods without negative control sample on RBPJ-1 and RBPJ-2. Percentage of detected peaks carrying at least one RBPJ-related motifs within 150 bp to binding sites by GLMNB and previous methods for RBPJ-1 (a) and RBPJ-2(c). Histogram of distance between nearest motif within 150 bp to predicted binding sites for RBPJ-1(b) and RBPJ-2(d).

high resolution as MACS and SISSRs in ETV1 and EBNA2. It achieved the highest spatial resolution in RBPJ-1 dataset but the fourth in RBPJ-2 dataset.

Since the two biological replicates vary a lot at least in total number of tags in the raw data, it is difficult to evaluate the reproducibility of GLMNB. I decided to generate two random replicates from RBPJ-2 dataset by independently randomly sampling 60% of tags



(a)

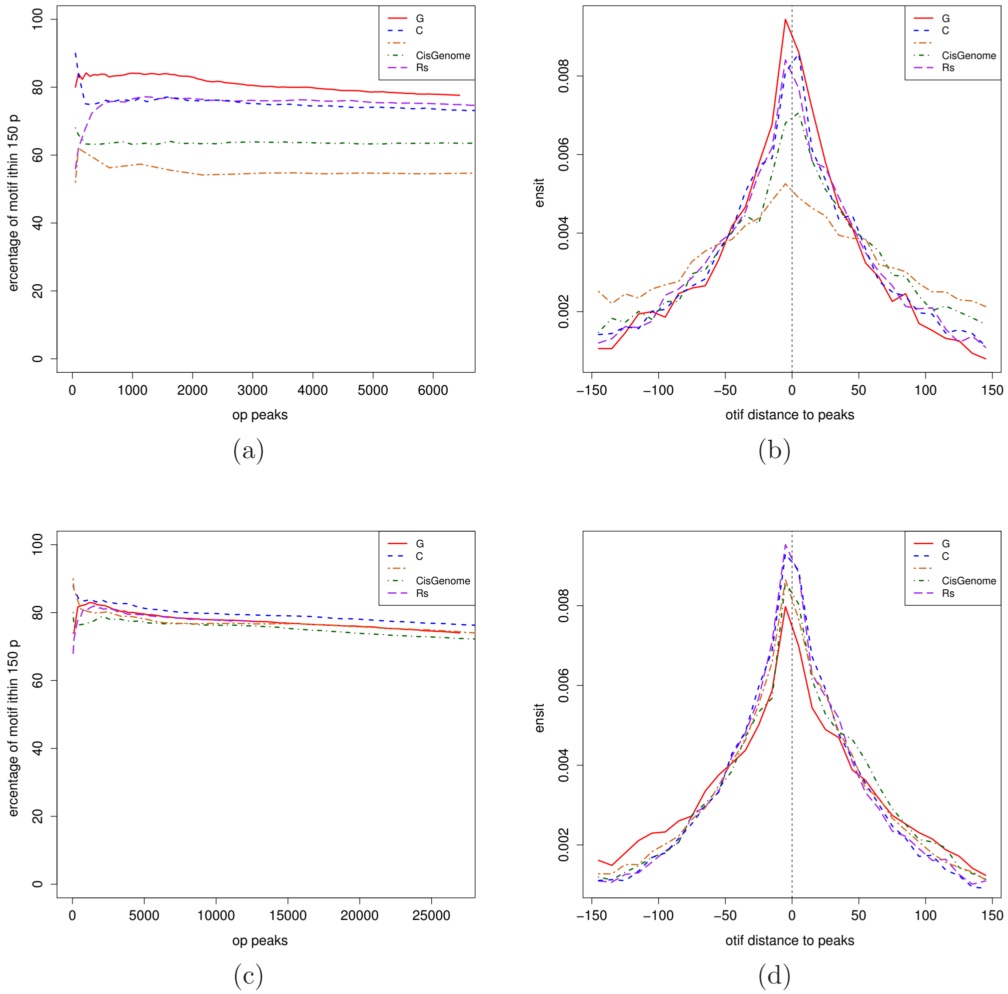
(b)

(c)

(d)

Figure 4.16. ChIP-Seq peak calling comparison with previous methods with control on ETV1 and ENBA2 ChIP-Seq data. Percentage of detected peaks carrying at least one related motif within 150 bp to binding sites by GLMNB and previous methods for ETV1(a) and EBNA2(c). Histogram of distance between nearest motif within 150 bp to predicted binding sites for ETV1(b) and EBNA2(d).

without replacement twice. Such replicates should contain very close number of tags and at least share 20% of original tags in common. The independent sampling is guaranteed by using different random seed in R. GLMNB is used to call peaks on these two replicates using the same negative control data. 18,101 common peaks are discovered from 23,224 peaks in replicate 1 and 21,628 peaks in replicate 2. Their FDRs in logarithm scales have an



(a)

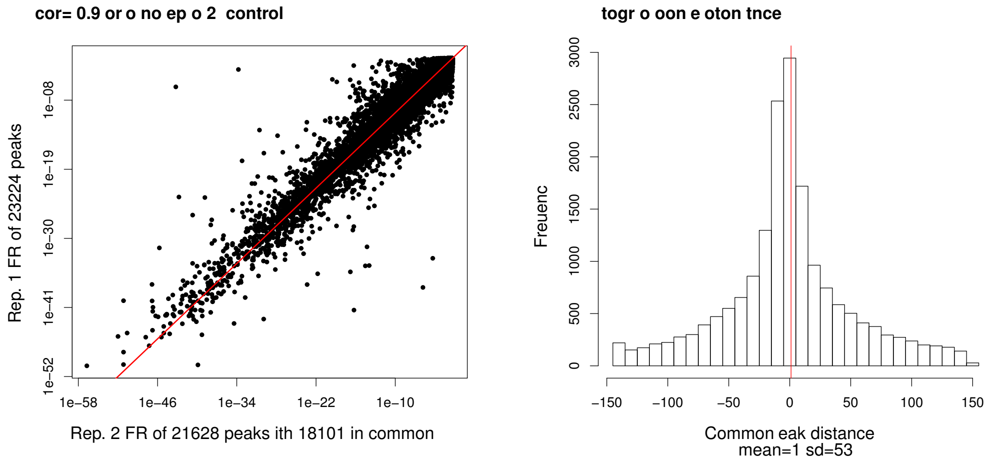
(b)

(c)

(d)

Figure 4.17. ChIP-Seq peak calling comparison with previous methods with control on RBPJ1 and RBPJ2. Percentage of detected peaks carrying at least one RBPJ-related motifs within 150 bp to binding sites by GLMNB and previous methods for RBPJ1 (a) and RBPJ2(c). Histogram of distance between nearest motif within 150 bp to predicted binding sites for RBPJ1(b) and RBPJ2(d).

extremely strong correlation (0.97) in Figure 4.18(a). The common peaks are on average 1 bp apart with standard deviation of 53 bp as shown in Figure 4.18(a). Both results suggest a strong reproducibility of GLMNB with two random replicates sharing common tags in a certain degree.



(a) (b)

Figure 4.18. Reproducibility of GLMNB evaluated by (a) FDR scatter plot and (b) histogram plot on distances between peaks in common on two random replicates generated from RBPJ-2 dataset.

Chapter 5

Discussion and Future work

5.1 Discussion of GLMNB features

One feature of GLMNB is the utilization of the negative binomial distribution. The negative binomial distribution allows the baseline level of tag counts, β_0, and an dispersion parameter α, to vary across different genomic regions ([11]). The flexibility of using these two parameters makes GLMNB a better model for the ChIP-Seq data than a Poisson based model. Using a negative binomial model also allows us to properly account for biological variability from binding affinity variation([3, 5]). ChIP-Seq background tags are frequently unevenly distributed, as they depend on the chromatin structure and the sequence content. By fitting a likelihood function to the data and obtaining a maximum likelihood estimator for β_0 and α within each sliding window, GLMNB can most efficiently and flexibly account for the effects of local genomic features. As shown in Figure 3.7(b), allowing a variable dispersion parameter α helps to increase the prediction power and to enhance the capability of distinguishing strong peaks from weak ones compared to a fixed α model, not to mention Poisson background model over-optimistically sets $\alpha = 0$ and assumes the equality of mean and variance for the entire genome. Furthermore, GLMNB fits the tag data by a binding profile, which is estimated from highly tag-enriched regions. As a result, GLMNB can detect refined protein binding sites that follow a particular tag pattern rather than simply relying on the total tag counts.

Another key feature of GLMNB is its local peak shifting parameter estimation. Most previous peak calling programs estimate a global peak shifting size from highly tag-enriched regions across the genome, and then use such a global estimate to merge forward/reverse strand tags together before peak calling. This strategy ignores the interval variation of DNA sequence protected by the binding protein. For example, a wide interval of DNA sequence may result from sonication in a ChIP-Seq experiment, if the binding protein forms a large complex. Similarly, a narrow interval of DNA sequence may result if a protein only partially binds to a site. If a peak shifting parameter is not accurately estimated, the tags on the forward and reverse strands may not be correctly merged, thus statistical power declines in peak calling and/or reducing the accuracy for pinpointing the binding positions. GLMNB estimates the peak shifting parameter along with other model parameters simultaneously within each window without merging forward and reverse tags together. This strategy not only provides a more accurate estimate for peak shifting parameter based on local tags, but also properly combines peak strength from the two strands of the genome. As shown in Figure 3.6 in the simulation study, modeling the 500 simulated peaks with local estimate of peak shifting parameter gives 10.8% higher power compared to global/constant shifting parameter strategy. And peaks called by constant shifting parameter strategy were on average shifted by 14.03 bp, an obvious reduction on spacial resolution.

As demonstrated in the FoxA1 data, GLMNB was able to detect 220 peaks that were clustered within 98 local regions along the genome. Rather than reporting a single binding site at the strongest peak position among multiple peaks, as did MACS and CisGenome, GLMNB pinpointed every binding location if peaks are separated by at least 200 bp, roughly the DNA filtration size in a ChIP-Seq experiment. Unlike SPP or SISSRs which are more sensitive and usually call more peaks, GLMNB applies a more conservative strategy and calls fewer with higher confidence. For example, in Figure 4.5(d), GLMNB called three peaks at the strongest peaks which can be paired with at least a motif. However, SISSRs identified 11 peaks but only half of them can be matched with a motif within 150 bp range. The output of GLMNB includes the predicted binding location, the local peak shifting parameter that measures the size of protein-protected DNA region and the statistical significance of the peak.

There were several default parameters used by GLMNB: bin size, window size, minimum tag counts in a window, and step size of sliding windows. I evaluated the impact of these parameters on GLMNB's performance using simulated data. I first tested bin size of 5, 10(default) and 20 bp, respectively, which yielded almost the same power and the false positives. In practice, a larger bin size allows a better fit to the model, because there are more bins with non-zero tag counts. Yet at the same time it reduces the mapping resolution, because binding sites are then predicted based on larger interval of data points. I next tested the cutoff value of the minimum tag counts within a window in both strands. With a cutoff of 5, 8 (default value) and 16 tags per 500 bp window in both strands, we again obtained almost the same most significant peaks (i.e. top 500 peaks) in the simulated data. However, we notice a big difference with respect to the number of peaks between using 5 tags per 500 bp and 16 tags per 500 bp. Using a stringent tag cutoff number such as 16 tags per 500 bp window, one arbitrarily filters out those peaks 1) with tags less than cutoff in both strands and 2) with slightly asymmetric tag size but either one strand tag counts may not exceed the cutoff. We do not worry about filtering out true positives in the first case, because these tags counts not significantly larger than background in both strands will not yield a significant peaks anyway. If we set the cutoff too stringent, we may miss true positives in the second case. For example, if a cutoff of 16 tags per 500 bp window is applied, one may miss a peak in a window containing 15 forward tags and 30 reverse tags. If a cutoff of 5 or 8 tags per

500 bp window is applied, GLMNB is able to use the binding profile to call a peak after compensating the significance for its asymmetric tag size. For exploratory analysis purpose, I suggest to use a cutoff of 5 or 8 tags, and leave peak calling task to GLMNB program rather than simply filtering it out. Therefore the default and recommended cutoff for tag counts is 5 tags per 500 bp window. It will take slightly more computing time than a larger cutoff, but the gain is obvious. 4,008 FoxA1 peaks are called using 16 tags per 500 bp window as a cutoff, while 10,073 are called using 5 tags per 500 bp window in the FoxA1 ChIP sample peak calling discussed in Chapter 4.

We further tested the window size of 500 bp (default) and 1,000 bp, respectively. Again we did not observe changes of the performance of GLMNB in the simulated data. In fact, given that our estimated binding profiles have fixed sizes, increasing the window size itself will not largely affect the performance. Finally, the step size of sliding windows is an important parameter in our program. If the step size is too large, GLMNB can easily miss a true binding site. This is true for all methods utilizing sliding windows.

One further extension of GLMNB is to incorporate negative control data into peak calling. If no negative control sample is available, GLMNB constructs a background model based on the ChIP sample data. If a negative control sample is available, GLMNB constructs a background model based on tags from the negative control data. If there are no tags in a certain region in the negative control data, GLMNB can call a peak with high confidence. If there are noisy tags or non-specific binding tags due to regional bias, GLMNB is able to adjust the baseline parameter β_0 & dispersion parameter α and adjust the significance to an appropriate level. If there are strong binding signals shown at the same location in both ChIP and negative control data similar to the scenario in Figure 4.11(d), GLMNB does not call a peak. Even though GLMNB captures the local background variation of tag occurrence via a negative binomial model, the comparison between signal data and control data actually further improves the modeling of background tag variation when the negative control data are available. It thereafter improves the specificity of peak calling and reduces false positives caused by non-specific binding event and background noise, for example GC content in local regions.

GLMNB corrects multiple testing using false discovery rate. It was shown in simulation study in Chapter 2 that GLMNB is able to distinguish peaks with different simulated strength by FDR. Such FDR is calculated based on the Wald test or the likelihood ratio test from negative binomial background model, whose null distribution is shown in Figure 3.5(a) and (c), and is therefore reliable.

GLMNB also offers a function to adjust the sliding window size automatically using the first derivatives of smoothed profiles to help the user to choose an appropriate window size. An appropriate window size can help precisely locate the predicted peak positions. If a window size is set too large, the tag count vector may include too much region with barely any tags. If a window size is set too small, GLMNB may lose important information, for example a particular region that can match with ChIP signal profile. Our approach relies on the first derivatives of smoothed binding profiles from high confident regions on forward and reverse strands. Figure 2.2 shows an example of smoothed first derivatives from FoxA1 ChIP data. GLMNB starts from the center of profile window and extends to both sides and looks for a position where the first derivatives stay around zero for both strands. Please note that the flat area in the first derivatives corresponds to both far ends of smoothed bell-shape profile. This approach will ensure that the proper window size captures adequate information from observed tags and does not mislead the model in a sparse zero-tag area.

5.2 Computing speed

The prototype of GLMNB algorithm is developed in R but takes too long to finish the entire genome due to the weakness of R software. For example, it took roughly a hour to finish

1,000 MLE using maxNR function(a maximum likelihood estimate function in R using the Newton-Raphson method) in R, which covers 30 kb region, a really small region compared to human genome size of 3,098 Mb. Therefore, I encode all functions in a more efficient way in C++. Depending on the sliding window parameter setting, it takes GLMNB about two hours (40 minutes) to call peaks in the entire genome on ChIP sample with negative control (ChIP sample only) using the default setting(minimum tag cutoff 5 tags per 500 bp window). This computing time comes from the FoxA1 ChIP sample(3.9 million tags) with negative control data(5.2 million tags). This is a very fast program among those utilizing directional tag information. It takes 7 hours for BayesPeak, another algorithm utilizing directional tag information , to analyze ChIP-Seq data with negative control sample using 12 cores running in parallel.

5.3 Peak calling using multiple tracks

One benefit of the GLMNB framework based on generalized linear model is its ability to extend peak calling onto multiple track ChIP-Seq data. For example, a biological scientist may be interested in 1) increasing peak calling power using biological replicates with a common negative control data under one biological condition; 2) identifying genome locations where there is a binding event under all c biological conditions other than negative control conditions; 3) identifying genome locations where there are differential binding events under different biological conditions. In this section, I would like to explore hypothesis testing for these three scientific questions and lay out possible future work.

Question 1) is a special case of GLMNB specific in equation (2.13) with k ChIP replicate samples and one negative control sample under one biological condition. To simplify the notation, footnote j is dropped.

$$\begin{aligned}
\log\mu &= \mathbf{X}\beta = \beta_0 + \sum_{j\prime=1}^{k}(\beta_1\vec{x}_{j\prime}) + \beta_2\vec{z}_j \\
\mu &= E(\mathbf{y}) \\
\vec{y} &= \left((y_1^S)^T, \ldots, (y_k^S)^T, (y^C)^T\right)^T \\
\vec{x}_{j\prime} &= ((\vec{0})^T, \ldots, (\vec{0})^T, (x_{j\prime}^S(\theta))^T, \ldots, (\vec{0})^T, \ldots, (\vec{0})^T)^T \\
\vec{z}_j &= ((x^C(\theta))^T, \ldots, (x^C(\theta))^T)^T
\end{aligned}$$

where $x^S_{j\prime(\theta)}, j\prime = 1, \ldots, k$ is the smoothed signal profile for $j\prime$-th replicates, $x^C(\theta)$ is the smoothed profile generated from negative control sample, θ is the common shifting parameter across all samples, and β_1 is the common coefficient for smoothed profile generated from all k ChIP samples. Both $x_{j\prime(\theta)}$ and $x^C(\theta)$ are generated using equations (2.10) and (2.11).

To call peaks that shows strong signals in all k replicates, the following hypothesis is tested.

$$\begin{aligned}
H_0 : \beta_1 &= 0 \qquad (5.1) \\
H_A : \beta_1 &> 0
\end{aligned}$$

GLMNB tests the hypothesis above using the likelihood ratio test. In the null model assuming no binding event, there are three free parameters, coefficient for signal profile β_2, peak shifting parameter θ and dispersion parameter α. In the alternative model assuming the existence of binding event, there is an additional parameter, the common coefficient for signal profiles β_1. GLMNB utilizes the likelihood ratio test with an appropriately estimated degrees of freedom to answer question 1.

Now we can move on to scientific question 2). Assume a ChIP-Seq project involving c biological conditions, each of which contains $k_j (j = 1, \ldots, c)$ ChIP samples and one negative control sample. Then the observed count vector is constructed as follows.

$$\begin{aligned}
\vec{y} &= \left((y^S_{1,1})^T, \ldots, (y^S_{1,k_1})^T, (y^C_1)^T, \ldots, (y^S_{c,1})^T, \ldots, (y^S_{c,k_c})^T, (y^C_c)^T\right)^T \qquad (5.2)\\
y^S_{j,j'} &= \left(y^F_1, \ldots, y^F_n, y^R_1, \ldots, y^R_n\right)^T_{j,j'}, j = 1, \ldots, c, j' = 1, \ldots, k_j \qquad (5.3)\\
y^C_j &= \left(z^F_1, \ldots, z^F_n, z^R_1, \ldots, z^R_n\right)^T_j, j = 1, \ldots, c \qquad (5.4)
\end{aligned}$$

A smoothed profile, $x^S_{j,j'}(\theta)$, are generated for j'-th ChIP sample data under j-th biological condition as follows. The shifted profiles are denoted as $\hat{m}^F_{j,j'}(t+\theta)$ and $\hat{m}^R_{j,j'}(t-\theta)$, where $j = 1, \ldots, c$ and $j' = 1, \ldots, k_j$. And the expected tag counts $x_{j,j'(\theta_j)}$ are generated as described in equations 2.10 and 2.11. The peak shifting parameter θ_j $(j = 1, \ldots, c)$ are the same in all replicates in j-th biological condition. A pair of binding profile are also generated from both forward and reverse strands using the same procedure described in section 2.3.3, denoted as $\hat{m}^F_{C,j}(t+\theta_j)$ and $\hat{m}^R_{C,j}(t-\theta_j)$, where subscript (C, j) indicates that it is from negative control data in j-th condition. And the expected tag counts from negative control data are denoted as $x^C_j(\theta_j)$ by plugging $\hat{m}^F_{C,j}(t+\theta_j)$ and $\hat{m}^R_{C,j}(t-\theta_j)$ into equation 2.12.

Then the generalized linear model is the same as shown in equation (2.13).

$$\begin{aligned}
\log\mu &= \mathbf{X}\beta = \beta_0 + \sum_{j=1}^{c}\left(\sum_{j'=1}^{k_j}(\beta_{1,j}\vec{x}_{j,j'}) + \beta_{2,j}\vec{z}_j\right)\\
\mu &= E(\mathbf{y})\\
\vec{x}_{j,j'} &= ((\vec{0})^T, \ldots, (\vec{0})^T, \ldots, (\vec{0})^T, \ldots, (x^S_{j,j'})^T, \ldots, (\vec{0})^T, \ldots, (\vec{0})^T, \ldots, (\vec{0})^T)^T\\
\vec{z}_j &= ((x^C_1)^T, \ldots, (x^C_1)^T, \ldots, (x^C_j)^T, \ldots, (x^C_j)^T, \ldots, (x^C_j)^T, \ldots, (x^C_c)^T, \ldots, (x^C_c)^T)^T
\end{aligned}$$

where $x^S_{j,j'}(j = 1, \ldots, k)$ is the smoothed signal profile vector from j'-th ChIP sample under j-th biological condition. x^C_j is the smoothed profile vector from negative control sample under j-th biological condition. $\beta_{1,j}$ is the common coefficient for smoothed signal profile

$x^S_{j,j'}$ in all replicates under j-th condition. $\beta_{2,j}$ is the coefficient for smoothed signal profile x^C_j under j-th condition. And β_0 is the baseline parameter.

To find genome locations where there is a binding event under at least one conditions in scientific question 2), the following hypothesis is tested.

$$\begin{aligned} H_0 &: \beta_{1,1} = \beta_{1,2} = \cdots = \beta_{1,c} = 0 \\ H_A &: \text{at least one } \beta_{1,j} > 0 \end{aligned} \tag{5.5}$$

To find genome locations where there are differential binding events under different biological conditions in scientific question 3), the following hypothesis is tested.

$$\begin{aligned} H_0 &: \beta_{1,1} = \beta_{1,2} = \cdots = \beta_{1,c} \\ H_A &: \beta_{1,j} \neq \beta_{1,j^*} \text{where } j \neq j^*, j, j^* = 1, \ldots, c \end{aligned} \tag{5.6}$$

Table 5.1 and 5.2 list numbers of peaks called by GLMNB, MACS, CisGenome and SISSRs at each time point separately on ChIP sample only and with negative control. One may notice that there are a lot more predicted peaks at 3 , 7 and 14 hours compared to 0, 24 and 30 hours, which suggests differential binding of GATA1 between these two categories of time points. It will be an interesting future project to extend the current GLMNB algorithm and explore genome positions with differential TF binding levels under these two categories of time points.

Table 5.1. Peak Number Comparison with ChIP data only on GATA1 multiple time point data(total, percent of GLMNB peaks in common)

Time point	GLMNB	MACS	CisGenome	SISSRs
0hr	6,188	6,907(62.0%)	5,426(58.1%)	6,233(54.9%)
3hr	15,780	26,515(62.1%)	29,251(63.6%)	34,120(66.2%)
7hr	18,951	22,930(88.3%)	24,171(88.9%)	29,488(80.0%)
14hr	12,053	13,088(70.4%)	14,967(84.4%)	20,221(77.8%)
24hr	3,067	5,508(47.6%)	8,592(52.1%)	11,553(47.4%)
30hr	4,996	6,826(66.3%)	9,769(67.7%)	12,868(68.6%)

Table 5.2. Peak Number Comparison with Negative Control on GATA1 multiple time point data(total, percent of GLMNB peaks in common)

Time point	GLMNB	MACS	CisGenome	SISSRs
0hr	6,188	6,907(62.0%)	5,426(66.3%)*	6,233(54.9%)
3hr	10,868	21,177(85.3%)	14,982(74.5%)	6,546(72.7%)*
7hr	13,650	15,985(71.1%)	11,809(73.7%)*	7,148(79.1%)
14hr	14,394	12,535(50.5%)*	8,851(66.7%)	5,039(76.3%)*
24hr	4,131	1,655(60.2%)*	1,398(67.1%)*	2,518(44.1%)*
30hr	4,628	5,619(51.6%)	3,487(55.7%)*	3,852(48.5%)*

* notes programs that calls less peaks than GLMNB.

Appendix A

GLMNB User's manual

A.1 Introduction

This GLMNB software is used to analyze ChIP-Seq Signal data (BED format) and predict transcriptional factor binding sites.

A.2 Software availability

A β version of GLMNB software, glmnb_1.0.tar.gz, is up online at SourceForge.net. The download address is:
https://sourceforge.net/projects/glmnb/files/latest/download

Feel free to download and test it on your ChIP-Seq dataset. The package contains two executive files(for Linux 64 bit and 32 bit systems, respectively), a README file, one ChIP sample dataset, one negative control sample dataset in the package. The flow chart and key functions in GLMNB are listed in Appendix B.

A.3 Performance

The whole procedure takes 1-3 hours, depending mainly on *tagcutoff*, the minimum tags counts for forward and reverse strands, respectively. The larger the *tagcutoff* is, the less time GLMNB will consume. But weak binding sites may be missed. Recommend *tagcutoff*=5.

GLMNB requires at least 2 GB of memory, depending on the number of ChIP-Seq files and their size. It also requires GNU Scientific Library(GSL).

False positives are reduced by including negative control sample. But the computing time increases as well. For the FoxA1 data set provided with the package, it takes about 40 mins to run GLMNB on ChIP sample only. It takes about 2 hours to run GLMNB on ChIP sample with input sample. It is recommended to submit a script to a server if there is a time limit on command line from the server.

A.4 Executive file

GLMNB: an executive file complied by g++ under linux 64 bit system(x86_64) used to call peaks and calculate corresponding false discovery rate(FDR). Please refer to section A.6 for detailed instruction on algorithm parameters.

GLMNB_32: an executive file complied under linux 32 bit system(i686) with identical peak calling function as GLMNB above.

A.5 ChIP-Seq data file in BED format

GLMNB takes ChIP-Seq data, both ChIP sample and negative control sample, in BED format. They can be put in ExpData folder by default or any other directories. There should be six columns separated by Tabs. The first column lists chromosome numbers. The second and third columns list start and end genome coordinates of ChIP tags. The sixth column lists symbol of ”+” or ”-” that represents forward and reverse strands. GLMNB only uses information from the first three columns and the sixth column, while information in the fourth and fifth columns are not used. An example of ChIP-Seq data is listed below:

Table A.1. ChIP-Seq data example

chr1	7324	7359	0	2	+
chr1	522319	522354	0	3	-
chr1	553256	553291	0	3	-
chr1	699985	700020	0	3	+
chr1	745587	745622	0	0	+
chr1	747076	747111	0	0	+
chr1	747461	747496	0	0	+
chr1	748359	748394	0	0	-
chr1	752041	752076	0	0	+
chr1	752047	752082	0	2	-
chr1	774025	774060	0	1	-

A.6 Command

NOTE: Please run GLMNB on linux 64 bit system(x86_64) or GLMNB_32 on linux 32 bit system(i686). The following uses GLMNB executive file as an example. For users in linux 32 bit system, please use GLMNB_32.

```
./GLMNB    ”ExpDatafile” [-ctrlfile Input_tags.bed] [-chipname FoxA1] [-tagcutoff 5]
           [-FDR 0.05] [-binsize 10] [-winsize 500] [-stepsize 10]
           [-printlevel 0] [-pminwincount 50] [-h 10] [-keeptempfile]
           [-buildcommonsymmetricprofile] [-LRT] [-taghalfsize 18] [–help] [-version]
```

For example, command for GLMNB on ChIP sample only (FoxA1 data) is:

```
./GLMNB ExpData/Treatment_tags.bed -chipname FoxA1_ChIPOnly
```

Command for GLMNB on ChIP sample with input sample (FoxA1 data) is:

```
./GLMNB ExpData/Treatment_tags.bed -chipname FoxA1_WithInput -ctrlfile ExpData/Input_tags.bed
```

Arguments are as follows:

ExpDatafile(required input): the ChIP signal file in BED format;

-ctrlfile: the negative control sample file in BED format;

-chipname: the transcription factor name;

-tagcutoff: the minimum tag number for a sliding window to be fitted by generalized linear model, default 5 tags per window for both strands;

-FDR: false discover rate cutoff for output;

-winsize: sliding window size, 500 bp by default;

-binsize: bin size within each sliding window, 10 bp per bin by defaults, therefore 50 bins for a 500-bp window;

-stepsize: step size for a sliding window, 10 bp per window by default. A larger stepsize will increase the computing speed but reduce the spatial resolution;

-printlevel: print level for screen output, 0 for minimum screen output, 5 for detailed debugging output;

-pminwincount: minimum tag counts of high confidence region for profile construction, by default 50 tags per twice of winsize (1,000 bp);

-h: bandwidth for profile smoothing method(kernel regression), do not change it unless for debugging purposes;

-keeptempfile: keep all temporary files for plotting, default not to keep temporary files;

-buildcommonsymmetricprofile: construct mirror profiles for forward and reverse strands(by default, the profile for the two strands are constructed separately without guaranteed mirror shapes);

-LRT: use likelihood ratio test rather than the default z-score test to call peaks;

-taghalfsize: the half length of DNA, for example, for ChIP Seq data with 36bp tag length, one should input 18;

–help: show brief command options;

-version: show version information.

A.7 Output

The final result file is "FoxA1_chr1_GLMNB_toppeaks.txt" at the current folder, containing the following columns separated by comma:

chr: chromosome;

peakpos: predicted peak position;

peakshift: estimated peak shift;

YF: forward strand tag counts within the sliding window;

YR: reverse strand tag counts within the sliding window.

logp: log_{10} of pvalues used to calculate FDR;

FDR: expected false discover rate;

ZF: forward strand tag counts at the same window from input sample (displayed only when input sample is available);

ZR: reverse strand tag counts at the same window from input sample (displayed only when input sample is available).

Appendix B

GLMNB program Flow Chart

B.1 GLMNB program Flow Chart

The flow chart and major functions in GLMNB is illustrated in Figure B.1.

B.2 GLMNB function notations

The comments on major functions used in GLMNB are listed as follows. For further details on GLMNB, please contact the author, Jialin Xu, at jxx120@gmail.com.

CallogGammaDiff: Calculate log Gamma difference as $log(\Gamma(y+1/\alpha)) - log(\Gamma(1/\alpha))$;
CalDigammaDiff: Calculate Digamma (the first derivative of log Gamma function) difference as $\psi(y+1/\alpha) - \psi(1/\alpha)$;
CalTrigammaDiff: Calculate Trigamma (the second derivative of log Gamma function) difference as $Trigamma(y+1/\alpha) - Trigamma(1/\alpha)$;
CalX: calculates a vector of X values for a given bin center position (dm) based on smoothed profile of mF, mR, return a vector x of length 2*winsize/binsize;

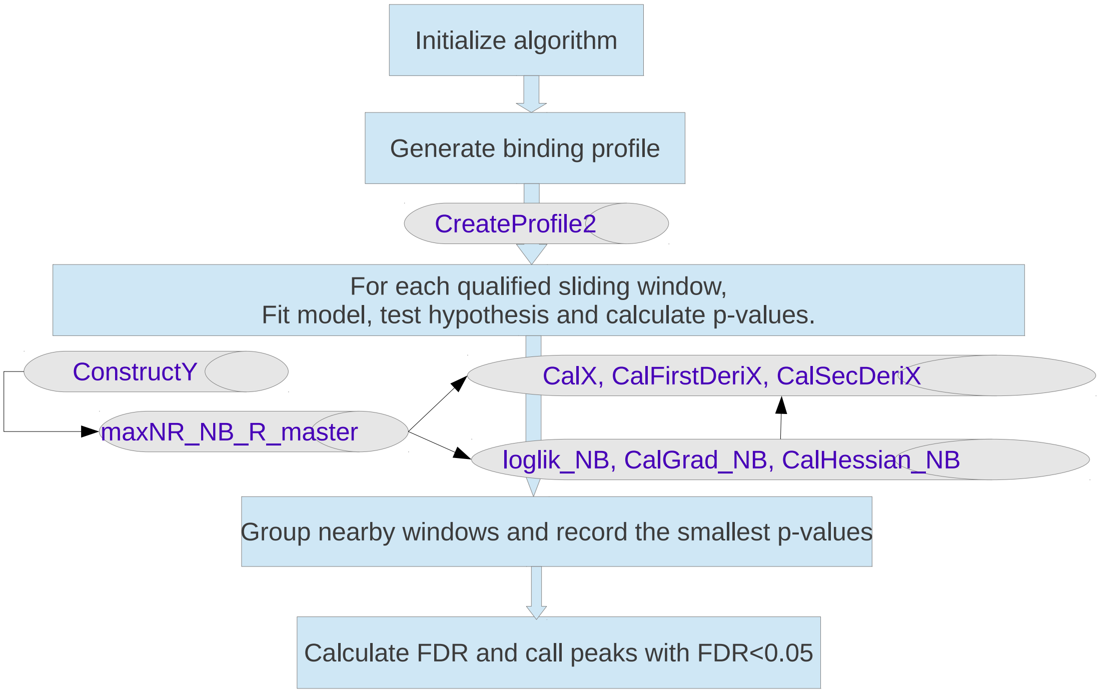

Initialize algorithm
Generate binding profile
CreateProfile2
For each qualified sliding window,
Fit model, test hypothesis and calculate p-values.
ConstructY
maxNR_NB_R_master
CalX, CalFirstDeriX, CalSecDeriX
loglik_NB, CalGrad_NB, CalHessian_NB
Group nearby windows and record the smallest p-values
Calculate FDR and call peaks with FDR<0.05

Figure B.1. Flow Chart of GLMNB program

CalFirstDeriX: Calculate the first derivatives for a given bin center position(dm) based on smoothed average bin height, pmF and pmR, return a vector derix1 of length 2*winsize/binsize;

CalSecondDeriX : Calculate the second derivative derix2 for a given bin center position (dm) based on smoothed version of ppmF, ppmR, return a vector derix2 of length 2*winsize/binsize;

loglik_NB: Calculate log likelihood value for given x, y, z vectors;

CalGrad_NB: Calculate gradient for a given x, y and z vector;

CalHessian_NB: Calculate Hessian matrix;

CalBeta1Stderr_new: Calculate standard error from Hessian matrix;

maxNR_NB_R_sub: function to excute Newton Raphison method and find maximum likelihood estimate when activePar is not full ranked. In other words, this function will be used for constant theta, constant alpha or any models other than full model.

maxNR_NB_R: function to excute Newton Raphison method and find maximum likelihood estimate when activePar is full ranked. This function will be used for full model only.

maxNR_NB_R_master: function to determine which functions above to call, maxNR_NB_R or maxNR_NB_R_sub, depending on whether subdimension = dimension or not.

modeltype2activePar: function to translate model type specification into activePar vector;

ConstructY: a function to construct an observed tag vector, Y, from bincounts vectos from the first(bincountsF and bincountsR), second(bincountsF1 and bincountsR1), third(bincountsF2 and bincountsR2) and forth(bincountsF3 and bincountsR3) ChIP experiment datasets.

CreateProfile2: Create profile from high confidence regions;

Bibliography

[1] Yong Zhang, Tao Liu, Clifford A Meyer, Jerome Eeckhoute, David S Johnson, Bradley E Bernstein, Chad Nusbaum, Richard M Myers, Myles Brown, Wei Li, and X Shirley Liu. Model-based analysis of chip-seq (macs). *Genome Biology*, 9(9):R137.1–R137.9, 2008.

[2] Ben Langmead, Cole Trapnell, Mihai Pop, and Steven L Salzberg. Ultrafast and memory-efficient alignment of short dna sequences to the human genome. *Genome Biology*, 10(3):R25.1–10, 2009.

[3] Christiana Spyrou, Rory Stark, Mike L. Smith, Andy G. Lynch, and Simon Tavare. Bayespeak: Bayesian analysis of chip-seq data. *BMC Bioinformatics*, 10(1):299, 2009.

[4] Jonathan Cairns, Christiana Spyrou, Rory Stark, Mike L. Smith, Andy G. Lynch, and Simon Yavare. Bayespeak - an r package for analysing chip-seq data. *Bioinformatics*, 27(5):713, 2011.

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

[6] Shirley Pepke, Barbara Wold, and Ali Mortazavi. Computation for chip-seq and rna-seq studies. *Nature Methods*, 6(11):S22–S32, 2009.

[7] Elizabeth G. Willbanks and Marc T. Facciotti. Evaluation of algorithm performance in chip-seq peak detection. *PLoS ONE*, 5(7):e11471, 2010.

[8] Peter J. Park. Chip-seq: Advantages and challenges of a maturing technology. *Nature Reviews*, 10:669–680, 2009.

[9] Michael L. Metzker. Sequencing technologies - the next generation. *Nature Reviews*, 11:31–46, 2010.

[10] Peter V Kharchenko, Michael Y Tolstorukov, and Peter J Park. Design and analysis of chip-seq experiments for dna-binding proteins. *Nature Biotechnology*, 26(12):1351–1359, 2008.

[11] Hongkai Ji, Hui Jiang, Wenxiu Ma, David S. Johnson, Richard M. Myers, and Wing H. Wong. An integrated software system for analyzing chip-chip and chip-seq data. *Nature Biotechnology*, 26(11):1293–1300, 2008.

[12] Anton Valouev, David S Johnson, Andreas Sundquist, Catherine Medina, Elizabeth Anton, Serafim Batzoglou, Richard M Myers, and Arend Sidow. Genome-wide analysis of transcription factor binding sites based on chip-seq data. *Nature methods*, 5(9): 829–834, 2008.

[13] Raja Jothi, Suresh Cuddapah, Artem Barski, Kairong Cui, and Keji Zhao. Genome-wide identification of in vivo protein-dna binding sites from chip-seq data. *Nature Biotechnology*, 36(16):5221–5231, 2008.

[14] Zhaohui S. Qin, Juanjun Yu, Jincheng Shen, Christopher A. Maher, Ming Hu, Shanker Kalyana-Sundaram, Jindan Yu, and Arul M. Chinnaiyan. Hpeak: and hmm-based algorithm for defining read-enriched regions in chip-seq data. *BMC Bioinformatics*, 11: 369, 2010.

[15] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Statist. Soc. B*, 57:289–300, 1995.

[16] Gordon Robertson, Martin Hirst, Matthew Bainbridge, Misha Bilenky, Yongjun Zhao, Thomas Zeng, Ghia Euskirchen, Bridget Bemier, Richard Varhol, Allen Delaney, Nina Thiessen, Obi L Griffith, Ann He, Marco Marra, Michael Snyder, and Steven Jones. Genome-wide profiles of stat1 dna association using chromatin immunoprecipitation and massively parallel sequencing. *Nature Methods*, pages 651–657, 2007.

[17] Robert C Gentleman, Vincent J Carey, Douglas M Bates, Ben Bolstad, Marcel Dettling, and et al. Bioconductor: open software development for computational biology and bioinformatics. *Genome Biology*, 5:R80, 2004.

[18] Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *Annals of Mathematical Statistics*, 41(1):164–171, 1970.

[19] Lawrence R. Rabiner. A tutorial on hidden markov-models and selected applications in speech recognition. *Proceedings of the Ieee*, 77:257–286, 1989.

[20] John Ashworth Nelder and Robert Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society, Series A*, 135:370–384, 1972.

[21] Lucien Le Cam and Grace Lo Yang. *Asymptotics in Statistics: Some Basic Concepts*. Springer, 2 edition, 2000.

[22] Erling B. Andersen. *Discrete Statistical Models with Social Science Applications*. North Holland, 1 edition, 1980.

[23] J. W. Hardin. *Advances in Econometrics: Maximum Likelihood of Misspecified Models: Twenty Years Later*. Elsevier, 2003. 45-73 pp.

[24] James W. Hardin and Joseph M. Hilbe. *Generalized Linear Models and Extensions*. STATA, 2 edition, 2007.

[25] Randall L. Eubank. *Non-parametric Regression and Spline Smoothing*. Marcel Dekker, Inc., 2 edition, 1999.

[26] Wolfgang Hardle, Peter Hall, and J. S. Marron. How far are automatically chosen regression smoothing parameters from their optimum? *Journal of the American Statistical Association*, 83(401):86–101, 1988.

[27] John D. Storey and Robert Tibshirani. Statistical significance for genomewide studies. *PNAS*, 100(16):9440–9445, 2003.

[28] Mathieu Lupien, Jerome Eeckhoute, Clifford A. Meyer, Qianben Wang, Yong Zhang, Wei Li, Jason S. Carroll, X. Shirley Liu, and Myles Brown. Foxa1 translates epigenetic signatures into enhancer driven lineage-specific transcription. *Cell*, 132(6):958–970, 2008.

[29] Yannis E. Mavromatakis, Wei Lin, Emmanouil Metzakopian, Anna L.M. Ferri, Carol H. Yan, Hiroshi Sasaki, Jeff Whisett, and Siew-Lan Ang. Foxa1 and foxa2 positively and negatively regulate shh signalling to specify ventral midbrain progenitor identity. *Mechanisms of Development*, 1-2(128):90–103, 2011.

[30] Lexander W. Bruce, Andres J. Lopez-Contreras, Paul Flicek, Thomas A. Down, Pawandeep Dhami, Shane C. Dillon, Christoph M. Koch, Cordelia F. Langford, Ian Dunham, Robert M. Andrews, and David Vetrie. Functional diversity for rest (nrsf) is defined by in vivo binding affinity hierarchies at the dna sequence level. *Genome Research*, 6(19): 994–1005, 2009.

[31] Haruka Abe, Makoto Okazawa, and Shigetada Nakanishi. The etv1/er81 transcription factor orchestrates activity-dependent gene regulation in the terminal maturation program of cerebellar granule cells. *Proceedings of the National Academy of Sciences of the United States of America*, 108(30):12497–12502, 2011.

[32] Bo Zhao, James Zou, Hongfang Wang, Eric Johannsen, Chih wen Peng, John Quackenbush, Jessica C. Mar, Cynthia Casson Morton, Matthew L. Freedman, Stephen C. Blacklow, Jon C. Aster, Bradley E. Bernstein, and Elliott Kieff. Epstein-barr virus exploits intrinsic b-lymphocyte transcription programs to achieve immortal cell growth. *Proc Natl Acad Sci U S A.*, 108(36):14902–14907, 2011.

[33] Daniel Portal, Bo Zhao, Michael Calderwood, Thomas Sommermann, Eric Johannsen, and Elliott Kieff. Ebv nuclear antigen ebnalp dismisses transcription repressors ncor and rbpj from enhancers and ebna2 increases ncor-deficient rbpj dna binding. *Proc Natl Acad Sci U S A.*, 108(19):7808–7813, 2011.

[34] Jianqing Fan, Chunming Zhang, and Jian Zhang. Generalized likelihood ratio statistics and wilks phenomenon. *Annals of Statistics*, 29(1):153–193, 2001.

[35] Sven Heinz, Christopher Benner, Nathaniel Spann, Eric Bertolino, Yin C. Lin, Peter Laslo, Jason X. Cheng, Cornelis Murre, Harinder Singh, and Christopher K. Glass. Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and b cell identities. *Mol Cell*, 38(4):576–589, 2010.

[36] Bradley Efron. *Large-Scale Inference*. Cambridge University Press, 1 edition, 2010.

[37] Hitomi Matsuzaki, Hiroaki Daitoku, Mitsutoki Hatta, Keiji Tanaka, and Akiyoshi Fukamizu. Insulin-induced phosphorylation of fkhr(foxo1) targets to proteasomal degradation. *PNAS*, 100(20):11285–11290, 2003.

Vita

Jialin Xu

Department of Mathematics
The Pennsylvania State University
(814) 777-0467 jxx120@psu.edu

Education

Ph.D. in Statistics, Penn State University	2012(expected)
Bachelor of Science in Mathematics, Beijing Normal University	2004

Professional Experience

Research Assistant *Fall 2009 to Present*
Advisor: Prof. Yu Zhang, Department of Statistics, PSU
Instructor *Summer 2008- Summer 2009*
STAT 100, STAT 200, STAT 240 and STAT 250
Biostatistician in early phase clinical trial group (Intern) *Jan.-Jun. 2011*
Manager: Feng Gao and Timothy Montague, GlaxoSmithKline, King of Prussia, PA
Biostatistician (Intern) *May-August 2010*
Manager: David Landsky, Precision Bioassay, Burlington, VT

Publications

Xu J, Zhang Y, A Generalized Linear Model for peak calling in ChIP-Seq Data, *Journal of Computational Biology*, 19(6), 2012.

Lina Y, Lu L, Kavita P, Eric R, Erica U, **Jialin X** and Byron J, Genetic-based, differential susceptibility to paraquat neurotoxicity in mice. *Neurotoxicology and Teratology* 33 (2011) 415-421.

Xu J, He Y, Qiang B, Yuan J, Peng X, Pan XM (2008) A novel method for high accuracy sumoylation site prediction from protein sequences. *BMC Bioinformatics*. 9: 8.

He Y, **Xu J**, Pan XM (2007) A statistical approach to the prediction of pK(a) values in proteins. *Proteins*. 69 (1): 75-82.

Conference Presentations

Xu J, Zhang Y(2012) A Generalized Linear Model for peak calling in ChIP-Seq Data, ENAR 2012 Spring Meeting, Washington D.C..

Xu J, Carrie W, Liewen J and David L (2010) Fine tuning bioassay design and analysis, United States Pharmacopeia 3rd Bioassay Workshop 2010, Rockville, MD.

Xu J, Zhang Y(2010) A Generalized Linear Model for peak calling in ChIP-Seq Data, ENAR 2010 Spring Meeting, New Orleans, LA.