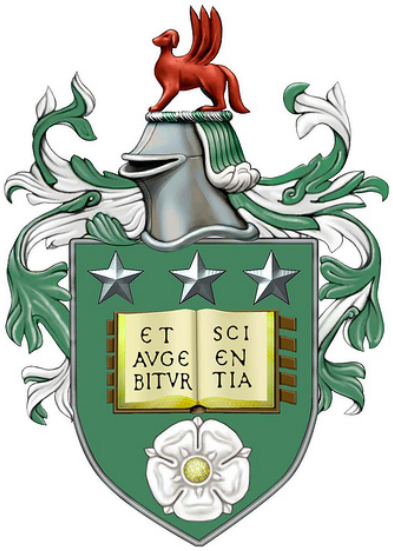
# Statistical analysis of genomic binding sites using high-throughput ChIP-seq data




Ibrahim Ali H Nafisah

Department of Statistics

University of Leeds

Submitted in accordance with the requirments for the degree of

*Doctor of Philosophy*

November 2015

# Abstract

This thesis focuses on the statistical analysis of Chromatin immunoprecipitation sequencing (ChIP-Seq) data produced by Next Generation Sequencing (NGS). ChIP-Seq is a method to investigate interactions between protein and DNA. Specifically, the method aims to identify the binding sites of a particular protein of interest, such as a transcription factor, in the genome. In the context of cancer research, this information is important to check whether, for example, a particular transcription factor can be considered as a therapeutic target.

The sequence data produced by ChIP-Seq experiment are in the form of mapped short sequences, which are called reads. The reads are counted at each single genomic position, and the read counts are the data to be analysed. There are many problems related to the analysis of ChIP-Seq data, and in this research we focus on three of them.

First, in the analysis of ChIP-Seq data, the genome is not analysed in its entirety; instead the intensity of read counts is estimated locally. Estimating the intensity of read counts usually involves dividing the genome into small regions (windows). If the window size is small, the noise level (low read counts) would dominate and many empty windows would be observed. If the window size is large, the windows would have many small read counts, which would smooth out some important features. The need exists for an approach that enables researchers to choose an appropriate window size. To address this problem, an approach was developed to optimise the window size. The approach optimises the window size based on histogram construction. Note, the developed methodology is published in [46].

Second, different studies of ChIP-Seq can target different transcription factors and then give different conclusions, which is expected. However, they are all ChIP-Seq datasets and many of them are performed on the same genome, for example the human genome. So is there a pattern for the distribution of the counts? If the answer is yes, is the pattern

common in all ChIP-Seq data? Answering this question can help in better understanding the biology behind this experiment. We try to answer this question by investigating RUNX1/ETO ChIP-Seq data. We try to develop a statistical model that is able to describe the data. We employ some observed features in ChIP-Seq data to improve the performance of the model. Although we obtained a model that is able to describe the RUNX1/ETO data, the model does not provide a good statistical fit to the data.

Third, it is biologically important to know what changes (if any) occur at the binding sites under some biological conditions, for example in knock-out experiments. Changes in the binding sites can be either in the location of the sites or in the characteristics of the sites (for example, the density of the read counts), or sometimes both. Current approaches for differential binding sites analysis suffer from major drawbacks. First, unclear underlying models as a result of dependencies between methods used, for example peak finding and testing methods. Second, lack of accurate control of type-I error. Hence there is a need for approach(es) to address these drawbacks. To address this problem, we developed three statistical tests that are able to detect significantly differential regions between two ChIP-Seq datasets. The tests are evaluated and compared to some current methodologies by using simulated and real ChIP-Seq datasets. The proposed tests exhibit more power as well as accuracy compared to current methodologies.

# Declaration

The candidate confirms that the work submitted is his/her own, except work that has formed part of jointly authored publications that has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

Chapter 2, Section 2.2 was published under Gusnanto A, Taylor CC, Nafisah I, Wood HM, Rabbitts P, Berri S (2014) Estimating optimal window size for analysis of low-coverage next-generation sequence data, Bioinformatics, 30, 1823-1829.

AG, HMW and SB discussed the initial idea and research problem. AG, CCT and IN developed the statistical method. IN analysed the LS041 data while AG analysed the LS010 data. IN developed the R package NGSoptwin to perform the calculation. AG performed the simulation analysis with both real and computer-generated data, and drafted the first manuscript. All authors gave comments that improved the manuscript and contributed to the discussion.

# Acknowledgements

Anything is possible with passion. Doing a PhD would not be an easy mission for me without the support of many people around me. Support in the form of ideas, discussion and even questioning was very helpful for me during my PhD journey. I would like to thank my supervisors Arief Gusnanto, Charles Taylor and David Westhead for all the support, guidance and patience throughout my study. They were available with useful advice, encouragement and very valuable discussion. Arief and Charles provided support in the mathematical and statistical aspects, while David supported the biological aspects. I would also like to thank Vijaya Shanmugiah for his support in biological and programming aspects.

Finally, I am more than grateful to my mother Sarah and my wife Fatimah for their support and help in everything in my life. I am also very grateful to my children Ali, Sarah and Norah, who make me smile whenever I look at them. I would like to thank my friends for their support and useful discussion. Throughout my PhD, I was financially supported by King Saud University.

# Contents

# List of figures

# List of tables

# Chapter 1

# Introduction

In this chapter, we introduce the basic biological background that is relevant to our research. We highlight some biological issues that motivate the work in this thesis. The objectives of this research are also stated. We give a brief summary of our achievements. Finally, the structure of the remaining parts of this thesis is outlined.

## 1.1 Background

### Genome

The genome consists of deoxyribonucleic acid (DNA), which carries the genetic information of an organism [47]. DNA is a chain (sequence) consisting of four nucleotides, which are adenine (A), thymine (T), guanine (G) and cytosine (C) bases, which are attached to a sugar phosphate. This chain of DNA constructs a DNA strand and the DNA strands build what is called a chromosome [42]. The human genome has 24 chromosomes (including the sex chromosomes X and Y) whereas other organisms have different numbers, for example the mouse genome has 21 chromosomes. Each chromosome normally has two copies of DNA and each copy has two strands of DNA

sequence. In normal DNA, the bases in the two strands are paired where A is paired with T and C with G [47]. For instance, if TACATCGG is a DNA sequence in one strand then the DNA sequence in the same location in the other strand is ATGTAGCC. The human genome is about three billion base pairs long [42] (other organisms vary).

## Transcription factors

Many biological processes are related to DNA, one of which is the transcription of many parts of the DNA into RNA and then into proteins. Proteins (or transcription factors, TF) are type of genes that perform a specific function , for example maintaining structures, generating movements or other genes' activities [47, 28]. A TF binds to DNA via a quite complicated biological process and the DNA defines where the TF begins and ends [47]. The beginning and ending of TFs are known as TF binding sites.

## Next-generation Sequencing

Next-generation sequencing (NGS) technologies have transformed genetic research into a new era. NGS helps to investigate and understand the human genetics (and other organisms' genetics) from various aspects. For instance, understanding the relationships between some TFs and different types of diseases, as many studies have shown that there are associations between TFs and certain diseases [66, 45, 19, 49]. The recent development in NGS technologies has accelerated the research process in genetics. There are different NGS technologies. For instance, Illumina (or Solexa), SOLiD and 454 systems. All technologies produce the same data but they differ in some details. For example, The length of the fragment, run time, quality and cost. The principle behind NGS is to isolate and chop DNA (or RNA) into short fragments to construct a genomic library. Then, these fragments are sequenced and mapped to a reference genome according to their sequences, which are then called *reads* (a genetic map can be described

as a representation of the gene position on the chromosome [51]). Figure 1.1 shows the work flow of NGS technologies.

In the mapping step, the reads are signed with either "+" or "–", referring to which strand they come from. This is illustrated as follows [30]: Let TTCATCG be a DNA sequence that needs to be mapped to a reference genome. Mapping software will look for the same DNA sequence in the reference genome. If the software finds the sequence, then the sequence will be mapped with a "+" sign. On the other hand, if the software does not find matching sequence in the reference genome, it will consider the complementary sequence (which is the paired in the other strand) for TTCATCG, which is AAGTAGC, and search for a matching sequence in the reference genome. If a match is found for the complementary sequence, then it will be mapped with a "–" sign. That is, there are two strands in the genome, and the mapping is done by using one reference, which can be considered as one strand. A naive way can be to consider two references, so each strand has a reference. However, this would be time consuming. Hence, it is more efficient to check the complementary sequence for those that do not match directly. By completing the mapping of the sequenced DNA (Figure 1.1), we end up with quantitative data, which is called *read counts*. The reads are counted at each position.

[53] argued that although NGS has shorter reads compared to the old generation (which is Sanger sequencing), it has high coverage (sample size), which improves its accuracy. This argument was raised as a result of mapping accuracy issues. Since NGS technologies appeared, the process of mapping short reads with enough accuracy has been a big concern [65, 16]. However, NGS technologies have been developed and have become able to produce longer reads. Moreover, many mapping programmes provide a filtering parameter that filters reads that can be mapped to many positions in the genome, and hence high-quality data are produced [62]. However, the filtering parameter needs to be chosen by the user, and to obtain data with an acceptable quality it is recommended to refer to the data producer [62].

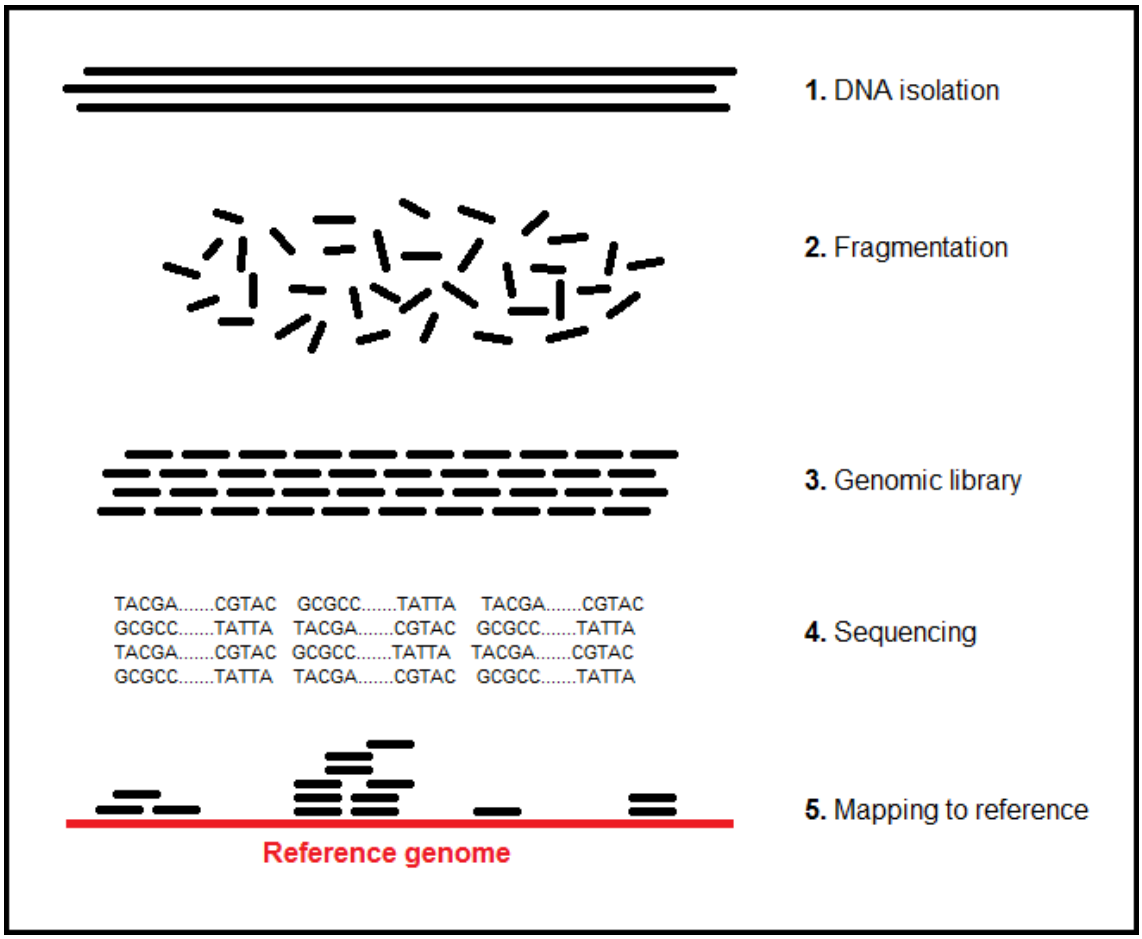



Figure 1.1: NGS work flow, which consists of five main steps. First, isolating target DNA sites. Second, to fragment the isolated DNA, and third, to construct a genomic library using the fragments. Fourth, perform sequencing of the fragments. The sequencing can be done by using the different available NGS methods. Finally, the sequenced fragments are mapped to a reference genome and the mapped sequences are called *reads*. The mapping can be done by using different methods, which can also involve a filtering step. By completing step five, we end up with a quantitative dataset that is *read count* per position across the genome. Note that for "+" stranded reads the starting position is counted, whereas for "–" stranded reads the ending position is counted (because the "–" stranded reads are mapped to their complementary sequences in the reference genome).

### ChIP-Seq experiment

The chromatin immunoprecipitation sequencing (ChIP-Seq) experiment is one of the methodologies that are based on NGS. ChIP-Seq experiments are used to sequence fragments of target DNA sites bound by a particular TF in living cells [29, 38, 26]. ChIP-Seq is a direct way to identify TF binding sites in the whole genome [18, 57, 64, 39]. ChIP-Seq experiments are used to study the association of targeted DNA and TFs across the genome [29]. ChIP-Seq experiments can be done in various protocols for different targets of association studies. For instance, a *knock-out* experiment in which two genetic samples are produced where in one of them a target TF is kept silent (knocked-out or inactive). This type of experiment aims to study the effect of a specific TF on the genome regulation.

## 1.2 Motivation

In genomic studies, for example binding site analysis where the whole genome is considered, the genome is not analysed in its entirety. For instance, in binding site analysis, researchers are interested in specific regions across the genome. Moreover, in ChIP-Seq experiments, most of the genome regions have low read counts because the experiment targets and chips specific regions. In such genomic studies, the intensities of the read counts across the genome need be estimated. Usually, estimating the intensity involves dividing the genome into small regions (windows) in which the regions of interest are not missed in the analysis. Choosing the window size is critical [24]. If the window size is small, the noise level (low read counts) would dominate and many empty windows would be observed. If the window size is large, the windows would have many small read counts, which would smooth out important features. In the context of copy number variation (CNV) analysis, [70, 48, 72] have proposed approaches that optimise the window size based on some underlying assumptions. These approaches are limited

and rely on certain assumptions. On the other hand, in many studies researchers chose the window size arbitrarily. There is no approach that enables researches to optimise the window size for NGS data in general and under any context. The need exists for an approach that is able to optimise the window size.

As mentioned, NGS technologies accelerate the research process in genetics. This leads to a large number of genetic datasets for different purposes, and many of these datasets are publicly available. More specific for ChIP-Seq experiments, Encyclopaedia of DNA Elements (ENCODE) [12] is a project that provides ChIP-Seq data for public researchers. ChIP-Seq data are counts that are used to investigate the association between DNA and TFs across the genome. Different studies target different TFs and give different conclusions, which is expected. However, they are all ChIP-Seq datasets and many of them are performed on the same genome, for example the human genome. Hence, they are count data and targeted regions of the genome are chopped. A question can be asked, is there a model for the distribution of the counts in given ChIP-Seq data? Furthermore , if there was a pattern for the counts in a given ChIP-Seq data , would it be a common pattern for all ChIP-Seq data? Hence, investigating the pattern of ChIP-seq data is needed.

Analysis of TF binding sites by using ChIP-Seq experiments is one of the contexts of NGS. It is biologically important to know the changes in the binding sites under some biological conditions. Specifically, in knock-out experiments there are two paired ChIP-Seq datasets in which a targeted TF is knocked out in one of them. Under these two biological conditions, the goal is to determine the effect of that TF on other genes or TFs by analysing their binding sites across the genome. Changes in the binding sites can be in either the location of the sites in the genome, the characteristics of the sites (for example the density of the read counts) or both. Some approaches have been proposed to analyse the binding sites. However, the current approaches suffer from some major drawbacks. First, the underlying model and assumptions are not very clear. Second, there is a lack of accurate control of type-I error. Hence, there is a need for an approach that is clear in

terms of underlying assumptions as well as exhibiting good control for the false positive rate.

## 1.3 Objective

There are many problems related to NGS data. In this thesis, we will focus on the three motivating problems in Section 1.2, which can be summarised as follows.

- Developing an approach that enables researchers to optimise window size in NGS datasets when optimising the variability of the genome is needed.
- Investigating the pattern of ChIP-Seq counts and whether there is a common pattern in ChIP-Seq datasets.
- Developing an approach that is able to detect significant differential binding sites accurately based on a clear underlying model and assumptions.

In this research, we will provide statistical methodologies that will address these problems as follows:

### 1.3.1 Optimal window size

From a statistical viewpoint, this problem can be formalised as optimising the bin size in The histogram of count data, so bin and window have the same concept in histogram construction. We address this problem by developing a novel approach based on histogram construction to optimise the window size of NGS datasets. The approach uses the raw count data to optimise the window size. That means that the approach enables researches to find an optimal window size before starting to analyse the data. This will save time and lead to meaningful analysis. The approach will be introduced and discussed in Chapter 2.

### 1.3.2 Pattern of ChIP-Seq counts

This problem is addressed by investigating the distribution of the counts in RUNX1/ETO ChIP-Seq data. In statistical terms, we try to find a statistical model that is able to describe the data. There are some clear features in ChIP-Seq data. For instance, non-stationarity and lengthy gaps of zero counts. We tried several statistical models and we employed some of the data features in the models. Although we obtained a model that is able to describe the RUNX1/ETO data, the model does not provide a good statistical fit to the data. We give some recommendations based on our findings, which might help in further investigation. This will be presented and discussed in Chapter 3.

### 1.3.3 Detecting differentially binding sites

This problem can be viewed from a statistical perspective as a task of detecting statistically significant differences between two paired samples where the samples here are the windows. We address this problem by developing three statistical tests that are able to detect significantly different windows. The detected windows can refer to differences in binding sites of TFs or to differences in genome regulation. The proposed tests detect only significant differences between two paired samples wherever they exist across the genome. We evaluate the tests by using simulated and real ChIP-Seq datasets. The tests show good control of the false positive rate. The tests will be presented in Chapter 5. Furthermore, we conduct a comparison study between our tests and some current methodologies for differential binding site analysis. We find that our proposed tests exhibit more power and accuracy compared to current methodologies.

## 1.4 Data

### 1.4.1 RUNX1/ETO

RUNX1/ETO data is ChIP-Seq data obtained from a knock-out experiment for a target TF named RUNX1/ETO. RUNX1/ETO is associated with *acute myeloid leukaemia*. DNA samples were taken from a patient diagnosed with acute myeloid leukaemia. The sequencing of the data was done by NGS technology Illumina. The reads length is 36 bp. A filtering process was considered in the mapping step as well as removal of duplicate reads. More biological details are given in [37].

The data are two samples of read counts, where in one of them RUNX1 is knocked out; call it *test*, and the other one *control*. The samples are structured in three columns *chromosome*, *position* and *read counts*. Both samples contain 24 chromosomes. The sample sizes are different; test contains 26,122,739 reads whereas control has 24,399,638 reads. We provide an exploratory data analysis of RUNX1/ETO in Section 2.3.

### 1.4.2 ENCODE and simulation data

In this research, we use many ChIP-Seq datasets that are publicly available from the ENCODE project [12]. We also simulate many datasets, including most ChIP-Seq dataset characteristics, as well as satisfying some required assumptions of the models under consideration. Both types of dataset are used for the purpose of evaluation. We use them to evaluate the performance of our proposed test and some current methodologies in terms of the false positive rate and power.

## 1.5 Outline of thesis

The rest of this thesis is constructed as follows: In Chapter 2, we develop new methods to optimise the window size for ChIP-Seq data; this is one of the main steps. We also statistically explore and describe the RUNX1/ETO data. In Chapter 3, we try to model the RUNX1/ETO data based on its features. In Chapter 4, we consider and evaluate some of the current methodologies for differential binding site analysis. In Chapter 5, we introduce and evaluate our proposed methods with our underlying assumption and model. In Chapter 6, a comparison study is constructed between our methods and some of the previously proposed methods. In Chapter 7, we summarise our findings and show the final conclusion. In the same chapter, we also provide some suggestions for further work.

# Chapter 2

# Optimal window size and exploratory data analysis of RUNX1/ETO

## 2.1 Introduction

In this chapter, we introduce a procedure for choosing an optimal window size for ChIP-Seq data, which is a critical step in the analysis of Chip-Seq data (or other genetics data). We also give a descriptive statistic for the RUNX1/ETO ChIP-Seq dataset.

### Preparing data for analysis

The genome is not analysed in its entirety as most of the genome regions are not of interest. In addition, most of the genome regions have low read counts, and hence variability of the read counts would not be observed properly . A natural question to ask is whether the window size is important, and if so how should it be selected?

## 2.2 Optimal window size

The problem of estimating an optimal window size is not new. Many studies have considered the problem in the context of NGS data and they concluded that an optimal window size is critical in inferring any pattern from data [24].

An optimal window size for the read counts of genetic data can be obtained by a method based on histogram construction [46]. That is, a histogram is usually constructed using windows of a common size, say $h$. [50] shows how to obtain the optimal window size in a histogram by considering the histogram function. The histogram function, $f(x)$, can be defined as follows: Let $x_1, \ldots, x_n$ be the observed position of reads in the genome, assumed to be a random sample from a density $f(x)$, where $n$ is the sample size. The density $f(x)$ represents the underlying true density of reads. Here we consider the issue of windowing the reads into equally spaced windows. Let $I_i(x)$ be the $i$-th genomic window, and $t_i(x)$ be the left-hand point of $I_i(x)$, where $i = 1, 2, \ldots, m$. We denote $h = t_{i+1} - t_i$ to be the window size.

Let $v_i(x)$ be the number of reads falling in the genomic window $I_i(x)$. $v_i(x)$ has Binomial distribution with parameters $n$ and $p_i(x)$, Bin$(n, p_i(x))$, where $p_i(x)$ is the probability of reads in the $i$-th window, $I_i(x)$. [50] assumed that $f(x)$ is a continuous and regular probability density function, and can be estimated by

$$\hat{f}(x) = \frac{v(x)}{nh}. \tag{2.1}$$

An optimal window size can be defined by that which minimises the integrated mean square error [50],

$$IMSE = \int E\left\{\hat{f}(x) - f(x)\right\}^2 dx.$$

Through some approximations based on the derivative $f'(x)$ [50] obtained the following approximation:

$$IMSE = \frac{1}{nh} + \frac{h^2}{12}\int_{-\infty}^{\infty} f'(x)^2 dx + O\left(\frac{1}{n} + h^3\right). \tag{2.2}$$

By minimising the first two terms in Equation (2.2), we obtain an approximate optimal window size

$$h^* = \left\{\frac{6}{n\int f'(x)^2 dx}\right\}^{1/3}.$$

The above procedure by [50] is not applicable to ChIP-Seq data. That is because, first, the true underlying function $f(x)$ is not available in practice. Second, the function $f(x)$ would not be continuous as the $x$ is a discrete variable (positions). Hence the derivative $f'(x)$ cannot be evaluated. To deal with this issue, we propose to estimate the optimal window size empirically using cross-validation (CV) and, second, Akaike's information criterion (AIC). Both CV and AIC are used to evaluate model selection [10].

To proceed, instead of assuming $f(x)$ to be a regular density function, [63] defined $f(x)$ as a step function

$$f(x) = c_i, \quad x \in I_i(x), \tag{2.3}$$

over a given window $I_i(x)$ where $c_i$ is a standardised reads in that window. The likelihood function is given by

$$L(c) = \prod_{j=1}^{n} \hat{f}(X_j) = \prod_{i=1}^{m} c_i^{\nu_i(x)}. \tag{2.4}$$

Under $c \geq 0$ and $\sum_i c_i = 1/h$, we the maximum likelihood estimate as histogram

$$c_i = \frac{\nu_i(x)}{nh}, \tag{2.5}$$

where it is just the standardised reads in the $i$-th window.

### 2.2.1 Cross-validation

Cross-validation is a general statistical method that can be used for various purposes, such as re-sampling [56]. Although there are different types of cross-validation, they generally share a common target, which is improving the accuracy of the suggested model(s) [44]. Leave-one-out cross-validation is one type and we consider it in estimating the optimal window size based on the log of the likelihood in Equation (2.4). The log of the likelihood in Equation (2.4) can be written as

$$\begin{aligned} \log L(c) &= \log\left\{\prod_{i=1}^{m} c_i^{\nu_i(x)}\right\} \\ &= \sum_{i=1}^{m}\sum_{s=1}^{\nu_i(x)} \log c_i \\ &= \sum_{i=1}^{m}\sum_{s=1}^{\nu_i(x)} \log\left(\frac{\nu_i(x)}{nh}\right) \end{aligned} \tag{2.6}$$

Where, as previously defined, $\nu_i(x)$ is the number of reads in window $i$.

For a given window size, a leave-one-out cross-validated (CV) log-likelihood can be calculated as

$$\log L^{CV}(c) = \sum_{i=1}^{m}\sum_{s=1}^{\nu_i(x)} \log\left(\frac{\nu_i^{(-s)}(x)}{(n-1)h}\right) \tag{2.7}$$

where $(-s)$ denotes that the $s$-th read in window $i$ is not considered in the calculation. We aim to identify the optimal window size that maximizes the CV log-likelihood in Equation (2.7).

### 2.2.2 Akaike's information criterion

Akaike's information criterion (AIC) was introduced by [3]. AIC is based on the idea of comparing suggested models by comparing the likelihood value of the models and penalising them on their number of the parameters [2, 27]. For more than one suggested

model for a specific problem, AIC can be evaluated as:

$$\text{AIC} = -\log(\text{likelihood value of a model}) + (\text{number of parameters in a model}). \tag{2.8}$$

From the suggested models, the model with the minimum AIC value can be described as the best model [3].

For a given window size, the AIC of the log-likelihood, Equation (2.6), can be calculated as follows.

$$\text{AIC} = m - \sum_{i=1}^{m} \log\left(\frac{v_i(x)}{nh}\right) \tag{2.9}$$

where the optimal window size is estimated as the one that minimises the AIC above.

Thus, obtaining the optimal window size for a given ChIP-Seq data or any counts data can be done in a simple process as follows: First, we consider some suggested window sizes. Second, for each of the suggested window sizes we apply either CV or AIC, or both of Them, based on the histogram function, as shown in Equations (2.7) and (2.9). Finally, the window size that is associated with maximum CV log-likelihood value or minimum AIC value can be considered as the optimal window size.

## 2.3 Exploratory data analysis of RUNX1/ETO

Here, we consider in more detail the project data that was introduced above. We start with a descriptive analysis. Then, we move deep into the data and try to understand its characteristics. Finally, we use some features of the data and try to find a statistical model that can work with the data.

### 2.3.1 Descriptive statistics

The data are two ChIP-Seq samples from a knock out experiment. We named the samples test and control, where in test RUNX1 is knocked out. The length of the reads is 36 bp. The sample sizes are different; the test sample has 26,122,739 reads whereas the control has 24,399,638 reads. Both samples contain reads covering all 24 chromosomes. Table 2.1 shows the actual length in base pairs, minimum observed base pairs, and sample mean of observed reads of each chromosome. From the table, it can be seen that the sample means of the reads in the chromosomes are quite low and do not vary. Although the actual sizes of the chromosomes are different, the observed sample means do not show big differences. Moreover, it can be noticed that all chromosomes in the test sample have slightly larger sample means compared to those in the the control sample.

In Table 2.2, we present the first sight of read counts of chromosome 1. From the table it can be noticed that there are many zero counts in both samples. In other words, many base pairs have not been observed in the genome, and this is true in all chromosomes (this can be spotted in Table 2.1 from the minimum observed positions of the reads column). By checking the proportion of zero counts, it is found that more than 99% of the counts are zeros in both samples. It is expected to observe a large proportion of zeros in a ChIP-Seq experiment. In ChIP-Seq experiments, the whole genome is not sampled, but only selectively targeted regions. These selected regions are quite few compared to the whole genome, hence most of the genome's positions are expected to have zero counts as they are not sampled.

Table 2.3 shows descriptive statistics for the counts of the whole genome. From the table, it can be seen that the sample means are affected by the large proportion of zero counts. The sample means are 0.0093 for test and 0.0086 for control; note the unit is number of reads per position. Although the test sample has larger sample mean, it has a slightly larger proportion of zero counts compared to the control sample. As these zero counts

| | Actual | Test sample | | Control sample | |
|---|---|---|---|---|---|
| Chr | length | Min | Mean | Min | Mean |
| 1 | 249,250,621 | 44,564 | 0.0084 | 41,785 | 0.0078 |
| 2 | 243,199,373 | 790 | 0.0088 | 200 | 0.0082 |
| 3 | 198,022,430 | 36,943 | 0.0109 | 35,782 | 0.0101 |
| 4 | 191,154,276 | 45 | 0.0100 | 230 | 0.0095 |
| 5 | 180,915,260 | 64,925 | 0.0087 | 64,921 | 0.0082 |
| 6 | 171,115,067 | 92,794 | 0.0097 | 67,540 | 0.0090 |
| 7 | 159,138,663 | 132,549 | 0.0091 | 54,400 | 0.0085 |
| 8 | 146,364,022 | 11,249 | 0.0102 | 11,239 | 0.0096 |
| 9 | 141,213,431 | 199 | 0.0062 | 274 | 0.0058 |
| 10 | 135,534,747 | 85,234 | 0.0129 | 53,567 | 0.0122 |
| 11 | 135,006,516 | 113,266 | 0.0092 | 128,596 | 0.0085 |
| 12 | 133,851,895 | 17,239 | 0.0092 | 17,335 | 0.0085 |
| 13 | 115,169,878 | 1 7,918,673 | 0.0068 | 17,920,244 | 0.0065 |
| 14 | 107,349,549 | 18,070,203 | 0.0074 | 18,070,186 | 0.0069 |
| 15 | 102,531,392 | 18,260,967 | 0.0061 | 18,260,047 | 0.0056 |
| 16 | 90,354,753 | 4,375 | 0.0061 | 1,042 | 0.0057 |
| 17 | 81,195,210 | 112 | 0.0090 | 355 | 0.0085 |
| 18 | 78,077,248 | 210 | 0.0108 | 691 | 0.0102 |
| 19 | 59,128,983 | 42,116 | 0.0071 | 41,950 | 0.0068 |
| 20 | 63,025,520 | 8,268 | 0.0083 | 8,049 | 0.0078 |
| 21 | 48,129,895 | 9,720,061 | 0.0060 | 9,720,033 | 0.0058 |
| 22 | 51,304,566 | 14,434,683 | 0.0048 | 14,432,757 | 0.0047 |
| X | 155,270,560 | 2,709,713 | 0.0043 | 2,709,535 | 0.0039 |
| Y | 59,373,566 | 2,716,169 | 0.0001 | 2,746,623 | 0.0001 |

Table 2.1: Summary of observed reads in each chromosome from each of the samples. The columns in order represent the chromosome number, the actual chromosome lengths in base pair units and the minimum observed positions of the reads and sample means in reads per base pair unit for the test and control samples. To calculate the sample mean, we divide the total number of reads of the chromosomes by the actual sizes.

| Position | Test sample Reads count | Control sample Reads count |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 0 |
| 2 | 0 | 0 |
| ⋮ | ⋮ | ⋮ |
| 41784 | 0 | 0 |
| 41785 | 0 | 1 |
| 41786 | 0 | 0 |
| ⋮ | ⋮ | ⋮ |
| 44563 | 0 | 0 |
| 44564 | 1 | 0 |
| 44565 | 0 | 0 |
| ⋮ | ⋮ | ⋮ |
| 81318 | 0 | 0 |
| 81319 | 1 | 1 |
| 81320 | 0 | 0 |
| ⋮ | ⋮ | ⋮ |

Table 2.2: Chromosome 1 Control and Test samples, sight of the read counts. It can be seen that most of the positions show no changes between the two samples. In addition, most of the positions in both samples either have no or low read counts.

show more than 99% of the positions, the first three quartiles will be zero and that is shown in the table. Moreover, in the table it can be noticed that the maximum read counts are quite large. That means that there are some *enriched* regions.

| Sample | Min | $Q^{1st}$ | Median | Mean | $Q^{3rd}$ | Max |
|---|---|---|---|---|---|---|
| Test | 0 | 0 | 0 | 0.0093 | 0 | 236 |
| Control | 0 | 0 | 0 | 0.0087 | 0 | 197 |

Table 2.3: Descriptive statistics for the number of reads at each position in the whole genome.

In Table 2.4, we show percentages of the observed *non-zero* counts, which represent less than 1% of the whole genome. As mentioned, the read counts are generally low in both samples, and the table confirms that. In the test sample it can be seen about 82% of the counts are 1 and 2, whereas the same read counts represent 98% of the non-zero counts in the control. The table shows that test has generally larger percentages for read counts higher than 1.

| | Percentage | |
|---|---|---|
| read counts | test | control |
| 1 | 57.48% | 88.46% |
| 2 | 24.15% | 10.37% |
| 3 | 10.91% | 1.06% |
| 4 | 4.59% | 0.1% |
| 5 | 1.8% | 0.01% |
| $\geq 6$ | 1% | $< 0.01\%$ |

Table 2.4: Percentages of *non-zero counts*, which represent less than 1% of the whole genome in test and control samples.

### 2.3.2 Optimal window size

In Figure 2.1, we show histograms of read counts of chromosome 1 in the test and control samples. By looking at the plots, the first thing that can be spotted is the empty region in the middle of each histogram. This region of chromosome is called the *centromere*. It exists in all chromosomes. The centromere region is the part where the two strands of a chromosome meet. The centromere region is hard to sequence, and hence it has not been fully understood. Thus, no reads are mapped to this region, and then it appears empty. Note, the location of the centromere differs from one chromosome to other and is not necessarily exactly in the middle.

Furthermore, from Figure 2.1 the large number of read counts around the centromere in both samples can be observed. Biologically, it is known that regions near to centromere are regions of multiple repetitive reads. That means that it is likely that many reads can be mapped to these regions. However, the filtering process reduces the likelihood of mapping error. To produce the histograms in Figure 2.1, we used a window size of 10 bp. This window size is chosen for visualising purposes; an optimal window size is discussed in Section 2.3.2. In addition, it can be seen that the test sample has larger read counts compared to the control, and this confirms the percentages in Table 2.4.

As mentioned, ChIP-Seq data are not studied and analysed directly. We need to divide the data into windows, then analyse them. To obtain an optimal window size, we employed the histogram-based method. which was described in Section 2.2. By finding the optimal window size in a histogram, we actually find the optimal window size for the data. That is, by optimising the window size we obtain the optimal visualisation for the variability in the data. Hence, a window size that shows the best variability of the data is optimal.

To optimise the window size, each of the samples is individually considered. In addition, within a sample, each of the chromosomes is considered and optimised separately. Moreover, we mentioned that the centromere region has zero counts only, so we decided

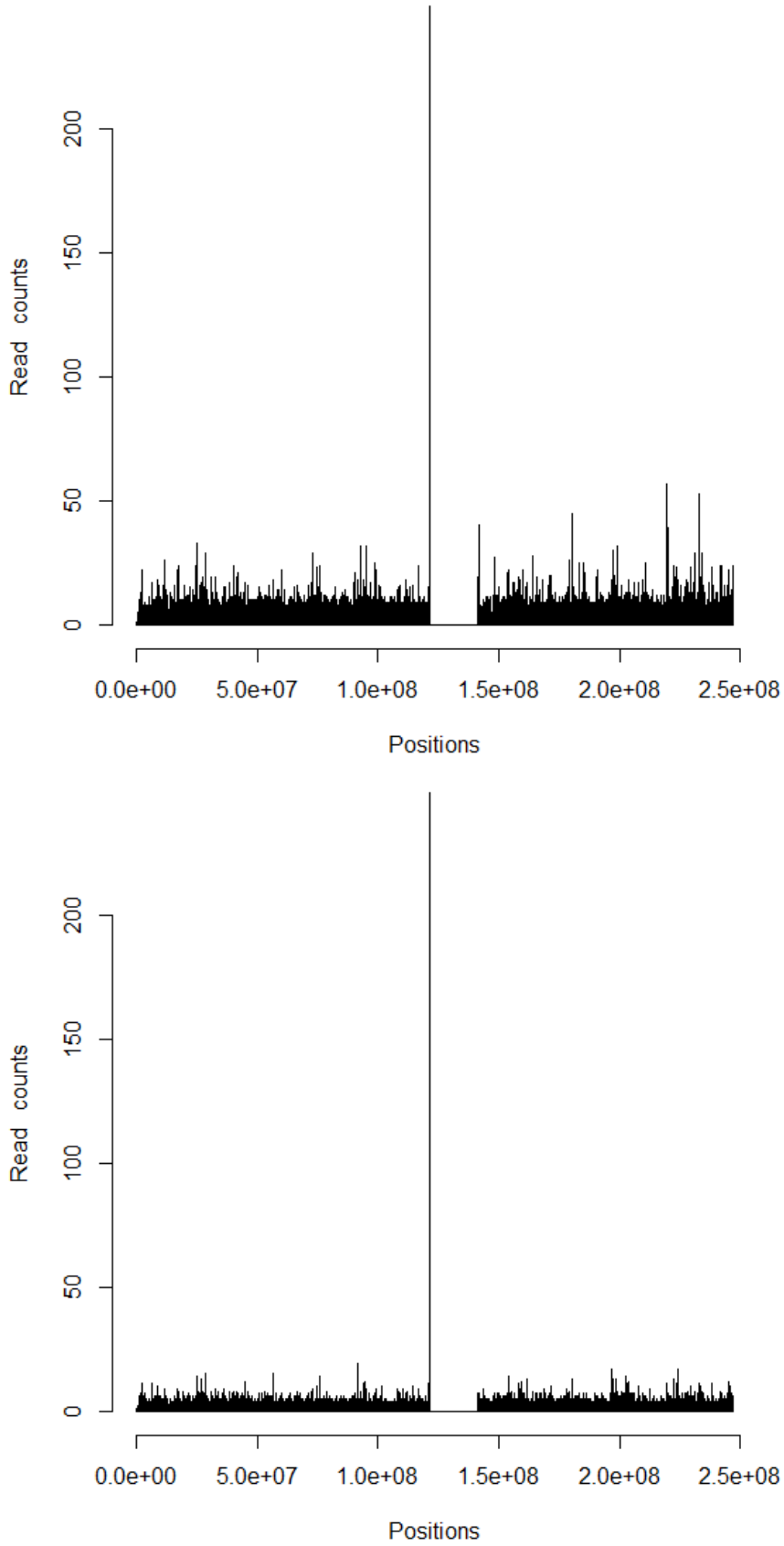



Figure 2.1: Chromosome 1 from the test (top) and control (bottom) samples.

to exclude this region from all chromosomes. Hence, each chromosome is divided into two parts, the first and second parts, which refer to the parts of the chromosome before and after centromere, respectively. For each part, we have an optimal window size. The aim is to find a common optimal window size for all chromosomes in both samples.

From the given different window sizes, we used cross-validation and AIC, as shown in Equations (2.7) and (2.8), with the histogram function in Equation (2.1) to choose an optimal size. We suggested sizes from 1 bp to 1 Kbp per window. Hence, for each of the sizes, the log-likelihood functions were calculated and then validated. The optimal window size is the one that shows maximum cross-validated log-likelihood (CV) value or minimum AIC value. For the first part of chromosome 1 (before the centromere), we show the results in Figure 2.2 by using LO-CV and AIC for the test and the control samples. Note that we considered the left-hand end of the reads in the windowing process. One could question what would happen if we considered the right-end or the middle, would this have an effect? We can say that it would not a have significant impact on the optimisation process because considering different ends can be considered as shifting all the windows by several base pairs, and this shifting would not have a significant effect, especially when the window size is much larger than the reads' length.

In Figure 2.2 it can be seen that panels (a) and (b) show the results of CV for the test and the control samples, respectively. In the same figure, panels (c) and (d) show the results by using AIC for the test and control samples, respectively. From (a) and (b) it can be seen that LO-CV shows different solutions. That is, for the test sample, the maximum cross-validated log-likelihood value falls in the interval between $200$ and $300$ bp sizes, while in the control sample it falls between $300$ and $400$ bp sizes. On the other hand, both samples show the same interval, which includes the minimum AIC values, and this interval is from $200$ to $250$ bp. By zooming in on the AIC interval from both samples it can be seen that the AIC values are fluctuating within quite a small range as the window size changes. Hence, it can be said that any window sizes in the interval from $200$ bp

to 250 bp can be considered optimal in both samples for the first part of chromosome 1. Furthermore, we optimised the window size for the second part of chromosome 1 as well as for all other chromosomes and found that the optimal window sizes fall into a common interval from 200 bp to 250 bp per window.

From a biological point of view, genomic variations can be detected within a genomic window of size 200 bp or less. Hence, from the optimal interval of window sizes, a window size of 200 bp is chosen to be the optimal window size for the two samples in all chromosomes. In Figure 2.3 we present the first 1Mbp of chromosome 1 in the test and control samples by using the optimal window size.

From Figure 2.3 it can be clearly seen that most of the windows have zero counts. In addition, in regions which are dense in reads, the variabilities between the read counts in the windows can be clearly seen by using the optimal window size. Moreover, in regions that are dense in reads, we can see that the test sample has a higher level of read counts compared to the control sample.

As mentioned, the optimal window size reflects the optimal window size. That is, the optimal window size represents a single point for each window, and these points represent the read counts in the optimal representation. For the optimal window size, we have windows of width 200 bp. In other words, each window is of width 200 bp. These windows are optimal to investigate the differences between the two samples as both samples have the same optimal size. In Figure 2.4, we show a window from chromosome 1 by using the optimal window size. In the figure we show the counts of both samples, test and control. As mentioned, by using such a window we want to investigate whether or not there is a significant difference between the test's and control's read counts.

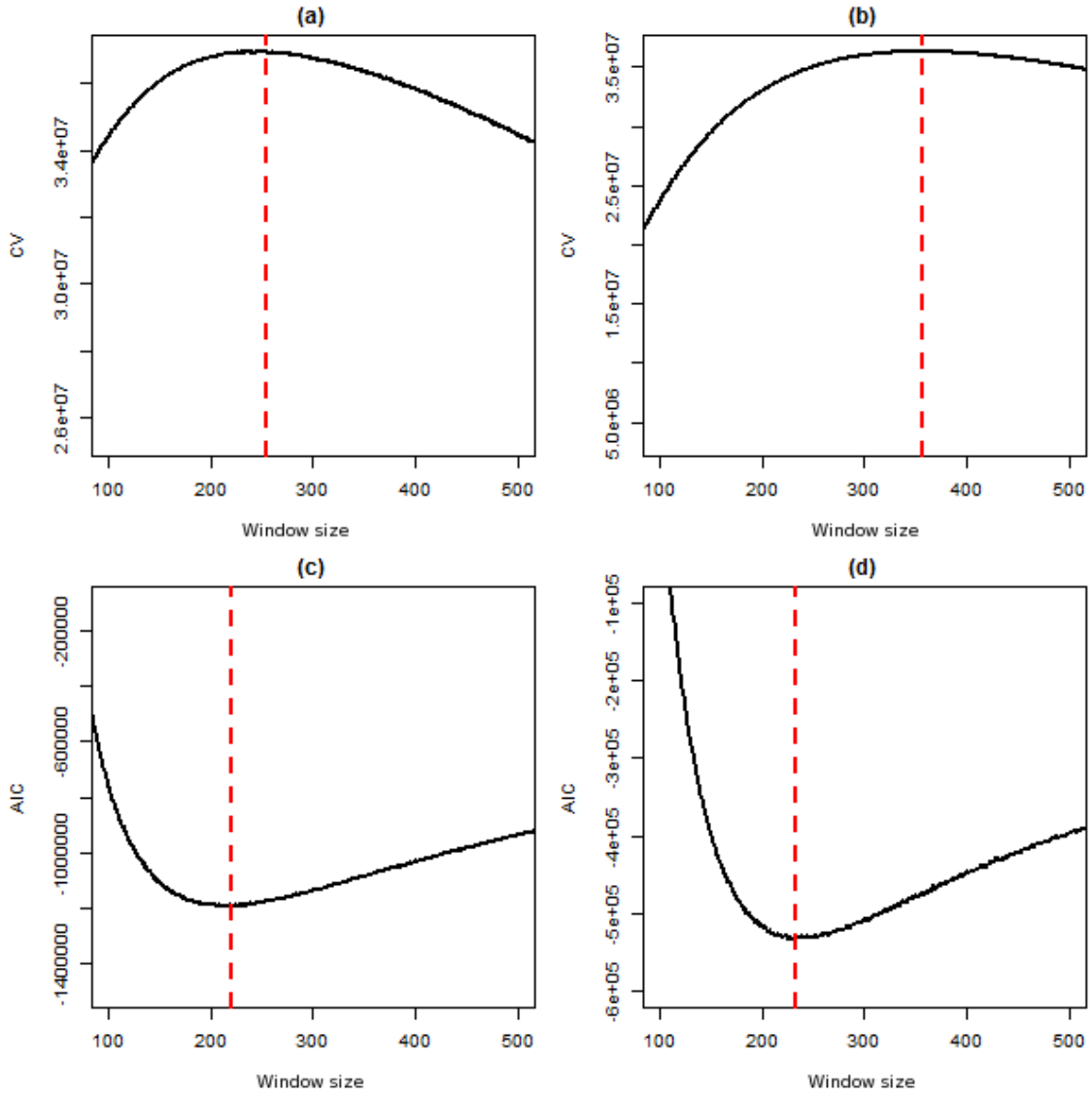



Figure 2.2: Cross-validated log-likelihood and AIC values by using the histogram function with different window sizes. Panels (a) and (b) show cross-validated log-likelihood CV (vertical axis) against window sizes (horizontal axis) of the first part of chromosome 1 from the test and the control samples, respectively. Vertical red lines indicate the window sizes associated with maximum CV values. Panels (c) and (d) show AIC (vertical axis) against window sizes (horizontal axis) of the first part of chromosome 1 from the test and the control samples, respectively. Vertical red lines indicate the window sizes associated with minimum AIC values.

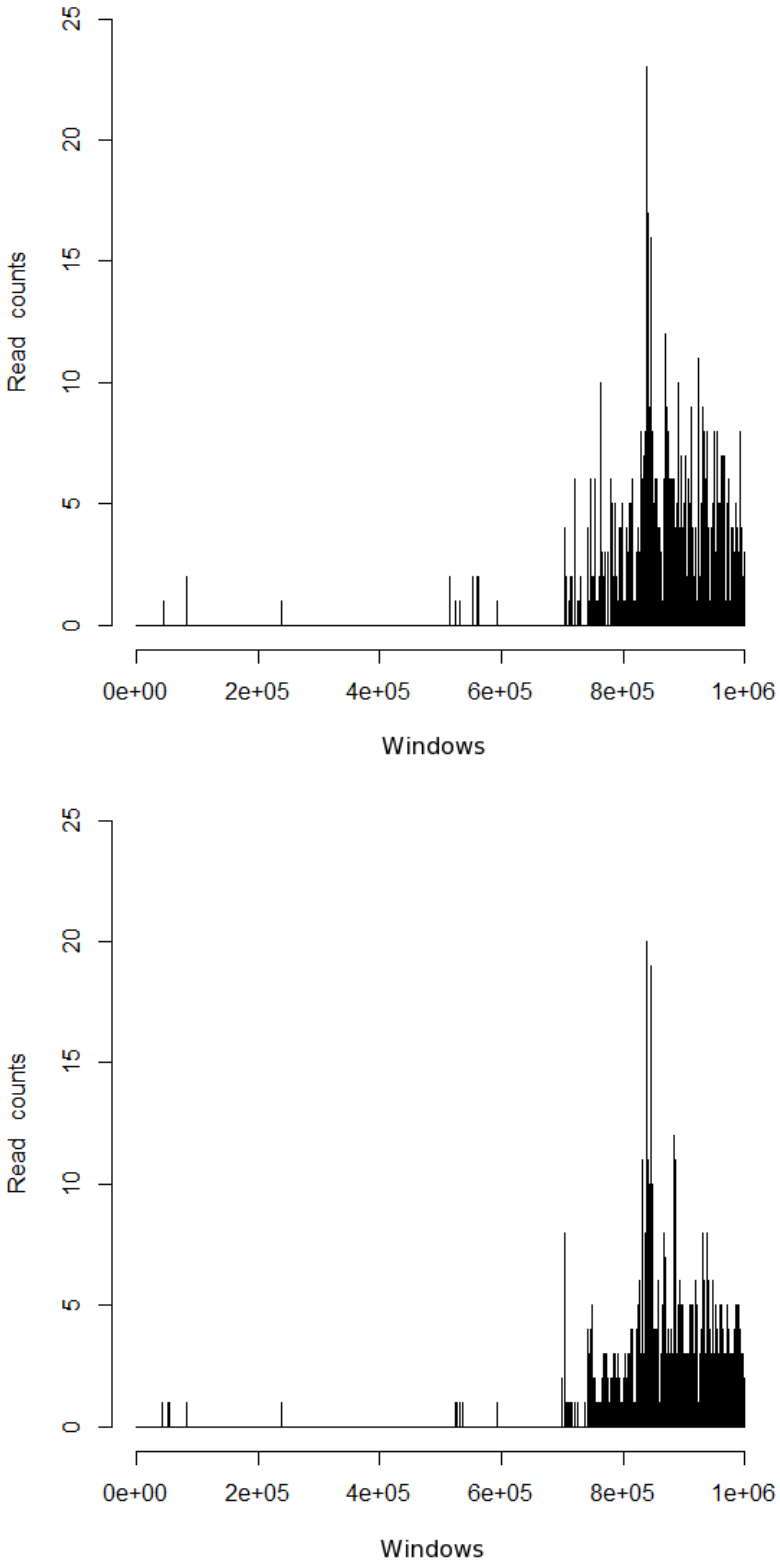



Figure 2.3: Chromosome 1 from the test (top) and the control (bottom) samples, the first 1 Mbp by using the optimal window size, which is 200 bp per window. Note, it can be seen that there are many consecutive empty windows; these are the regions where no reads are observed, hence they show zero read counts.

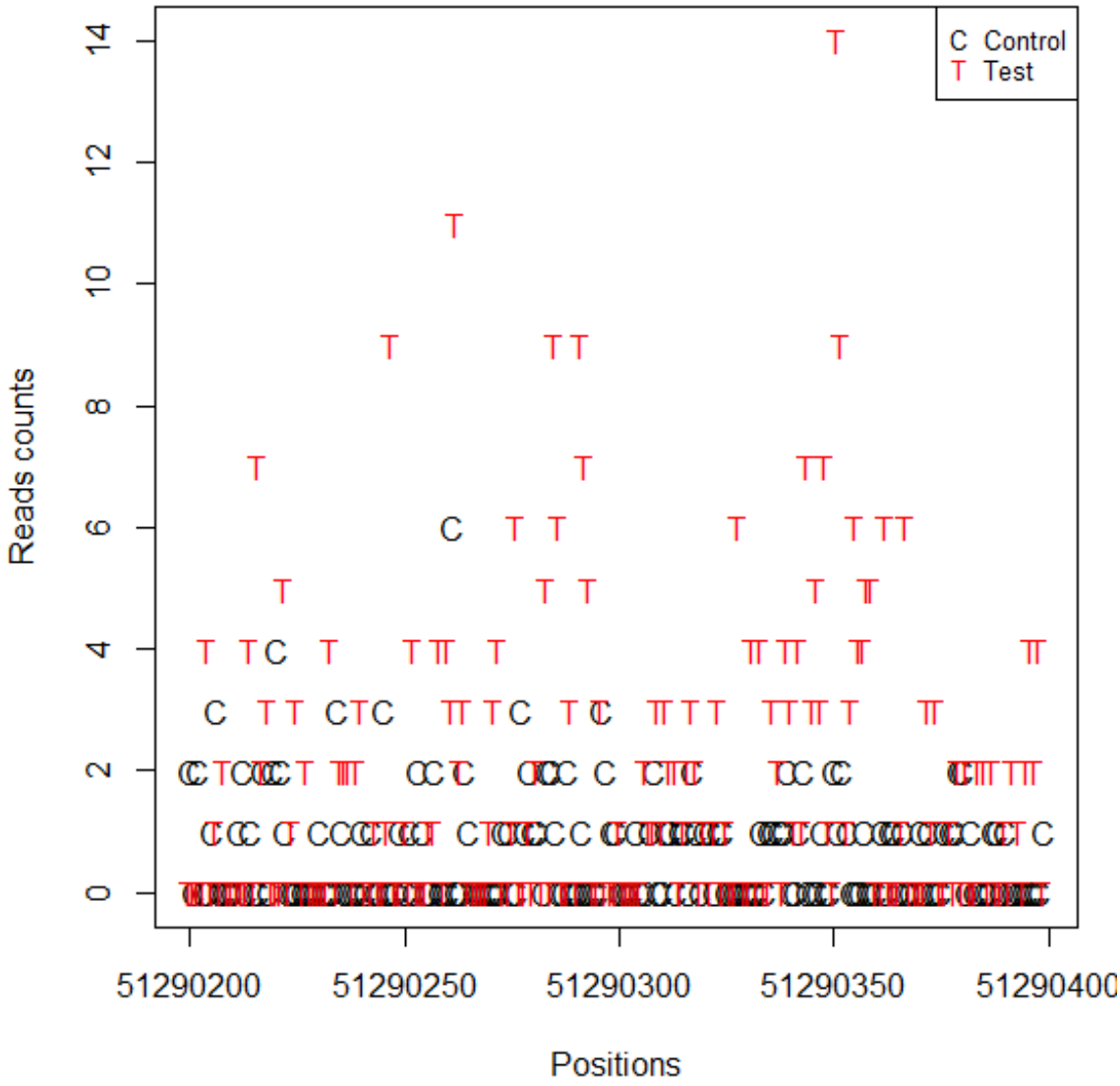



Figure 2.4: Read counts of test and control in a single window from chromosome 1 by using the optimal window size, which is 200 bp.

### 2.3.3 Characteristics of the data

#### Correlation

In the data, it can be seen that there is a weak correlation between the read counts. In Figure 2.5, we show the auto correlation function (acf ) for the first part of chromosome 1 from the test sample (the control sample shows similar behaviour). From the figure it can be seen that the test sample shows a weak correlation between the read counts compared. It is generally known that genetic data are serially correlated, but the amount of correlation varies. We notice that the read counts show weak correlation if we consider the whole test sample (the whole genome). On the other hand, by considering small regions of the genome, the read counts show little correlation.

Although the read counts show weak correlation globally, they show no correlation locally within small regions. That is, the read counts within windows of sizes $200$ bp show no correlation. For instance, the read counts in Figure 2.5 appear uncorrelated in both the test and the control samples. One can argue that the read counts show no correlation within small regions because there are not enough read counts, *i.e,* the sample size is small so the correlation between the counts is not significant. We can say that if the interest is only in the small regions, then considering the whole genome with its all features is not necessary.

#### Zero counts

It was mentioned that most of the read counts in the data are zeros at more than 99%. We mean by zero counts the unobserved base pairs so we observe zero counts in these base pairs. Hence, the zero counts are worth further investigation. As there are few non-zero counts, there are many consecutive zero counts. In Table 2.5, we show summary statistics of the observed lengths of consecutive zero counts. In addition, in Figure 2.6, we show

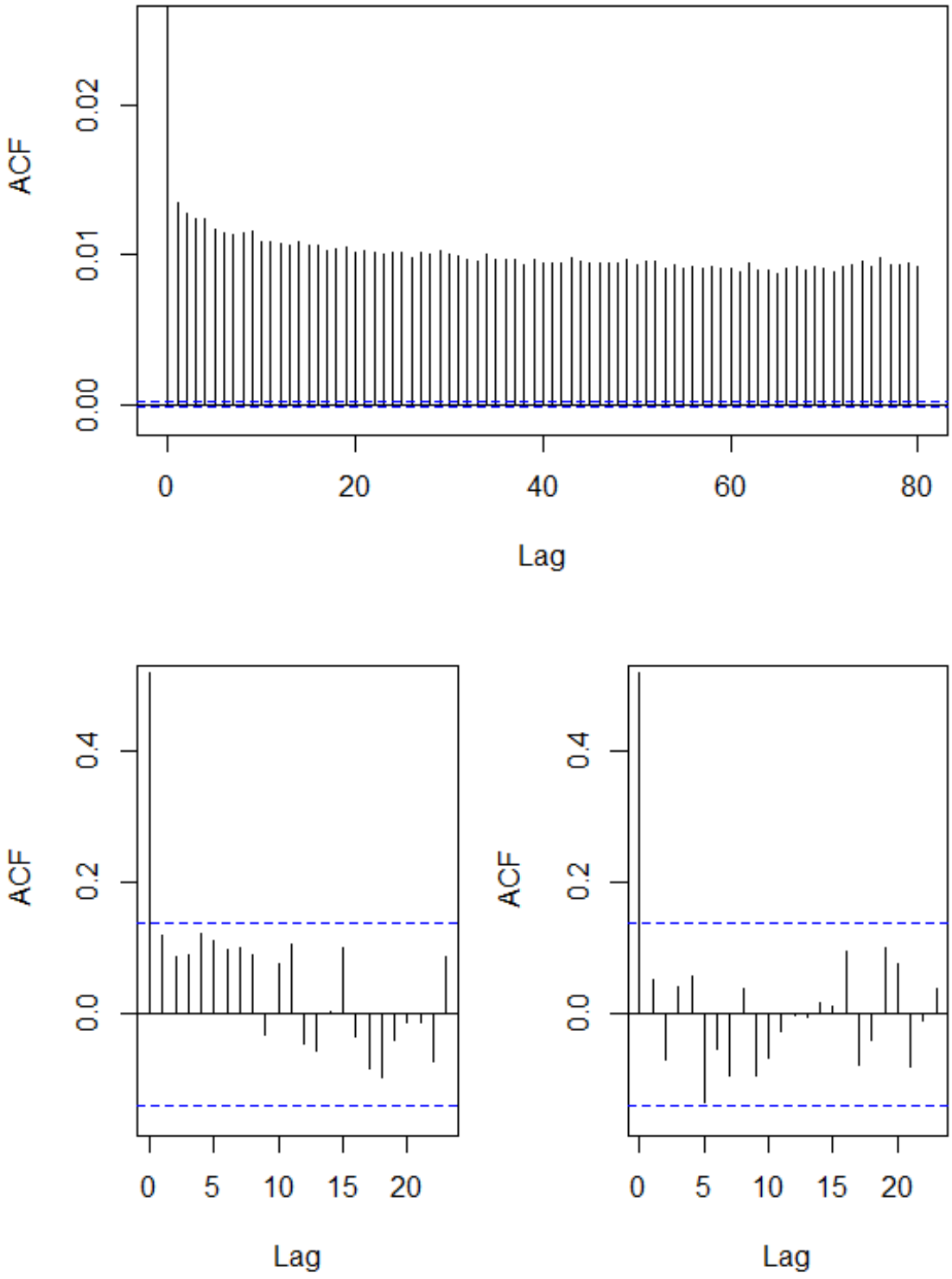



Figure 2.5: The top panel shows the acf between the read counts of the first part of chromosome 1 from the test sample. Note, no windowing is used for the reads (window size 1 bp). The lower panel shows the acf between read counts of a window from chromosome 1, which is shown in Figure 2.4, for the test (left) and the control (right) samples. Note, the vertical axis in all plots ranges from 0 to 1, and here we show the lower part of it where the correlation exists.

the observed distribution of lengths of consecutive zero counts. Note that the table and the figure are produced by using the first part of chromosome 1, and in total there are $616,638$ lengths of consecutive zeros.

| Min. | $1^{st}$ Qu. | Median | Mean | $3^{st}$ Qu. | Max. |
|---|---|---|---|---|---|
| 0 | 17 | 49 | 97.56 | 118 | 184900 |

Table 2.5: Summary statistics of observed lengths of consecutive zero counts in the first part of chromosome 1.

From Table 2.5, it can be seen that there are quite long gaps of zeros between non-zero counts. In addition, it can be noticed that the difference between the median and the mean is about 50, and this is a large difference. On the other hand, the difference between the median and the third quartile is not as large as that between the mean and the median. In general, the lengths vary and 75% of them are of lengths less than 120 zeros. In Figure 2.6, we can see that there are some large lengths, but they are not many From the figure in (c) and (d), it can be seen that the lengths have a clear pattern, more clearly in (d) where more than 75% of the lengths are represented.

In Figure 2.6 (d), we can see three clear features. First, an exponential decay from $0$ until around $20$ zeros. Second, a bump from $30$ to around $70$ zeros, and third, a linear component after length $70$ zeros. These components suggest that the lengths of the zeros might not follow a single distribution. As a result, the read counts might not follow a single distribution either. If the read counts followed a Poisson distribution, for instance, then the lengths of consecutive zeros would follow a geometric distribution, hence the decay in Figure 2.6 (d) would be linear throughout. In Section 3.3, we will see how these features can be useful in finding a statistical model for the read counts.

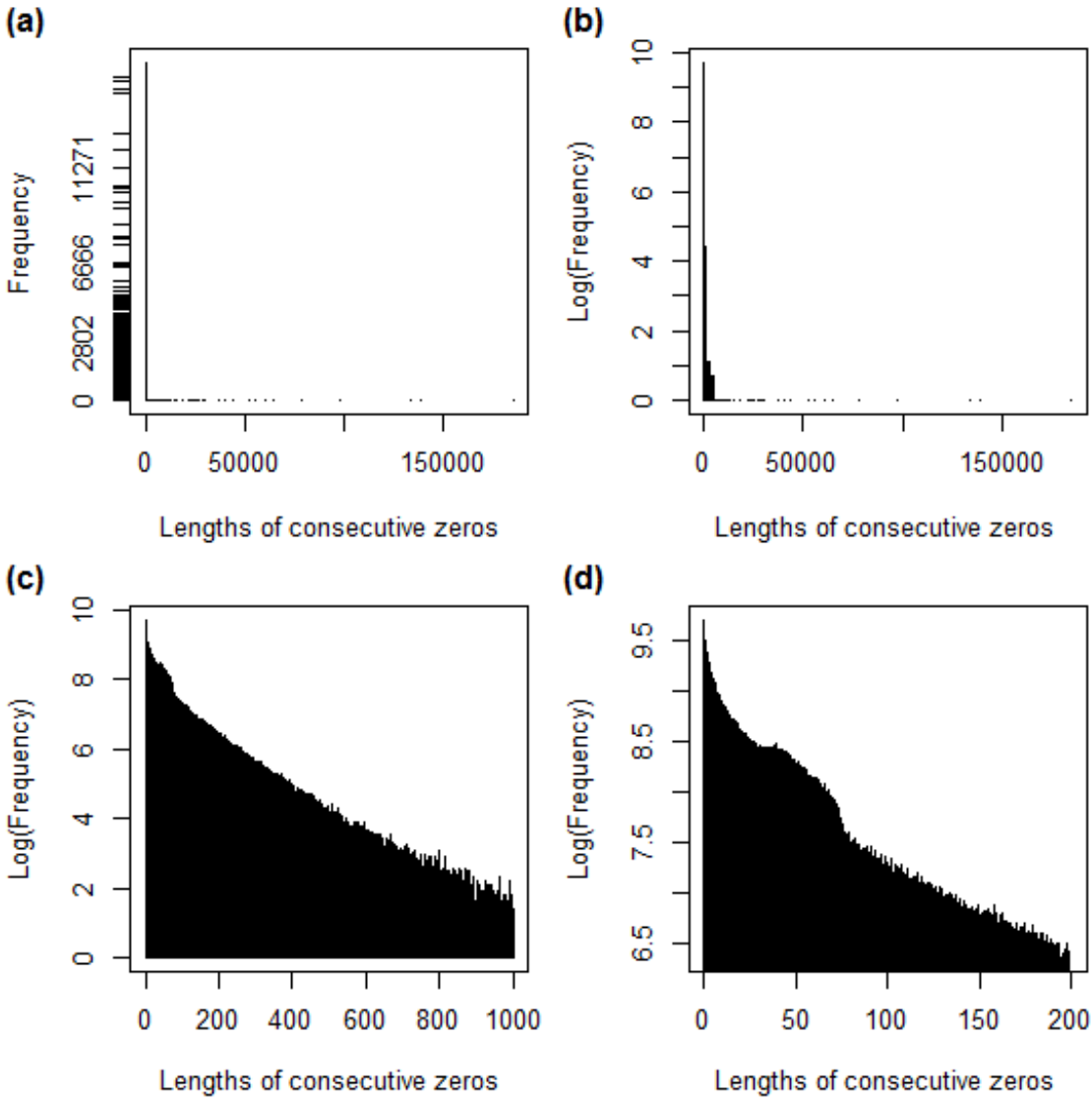



Figure 2.6: Distribution of lengths of consecutive zero counts in the first part of chromosome 1. (a) shows raw distribution for consecutive zeros lengths. (b) represents the frequencies in log scale. (c) shows lengths from 0 to 1000 consecutive zeros with frequencies in log scale, and (d) shows lengths from 0 to 200 consecutive zeros. Note, we mean by zero counts the unobserved base pairs so we observe zero counts in these base pairs.

# Chapter 3

# Modelling the distribution of ChIP-Seq data

## 3.1 Introduction

In this section, we look for a statistical model that can be fitted to the read counts. For the read counts, we seek a statistical model that can be used to achieve the objective of the study, which is to detect regions in the genome that are significantly different.

A model that can be used to describe the read counts is Poisson Distribution. That is, the Poisson model describes the number of events at a particular location. The number of events is a non-negative and discrete variable. Hence, the read counts can be considered as the number of events, and they are associated with positions, which can be considered as the location. We noticed in RUNX1/ETO, for example, that most of the regions across the genome have low read counts (or no reads at all). Although we observed a few regions with quite large read counts, these regions represent a very small fraction of the rest of the genome (see Table 2.4). This means that the read counts are quite similar across the genome and the average read count per base pair is low as well. Hence, it can be said that

there is no difference between the mean and the variance of the data. Thus, the Poisson distribution can be considered to model the read counts.

Although the filtering process is considered in the read-mapping step, some mis-mapped reads might exist. Assuming a random model for the read counts like Poisson can handle . That is, the mis-mapped reads would be considered as a part of the randomness of the model under consideration. Furthermore, the mis-mapped reads and non-zero reads make up a very small fraction compared to the zeros in ChIP-Seq data (see Table 2.4).

We observed in the previous chapter that many zeros are observed in RUNX1/ETO data and in any ChIP-Seq data in general. In many of the ChIP-Seq data studies most of these zeros are thrown away. However, this common feature is an important part of the nature of the ChIP-Seq data. Hence considering this common and huge part of the data might lead to find a common model that can generally describe ChIP-Seq data. Thus, the aim is to find a model that can handle this common and natural feature of ChIP-Seq data.

It was seen that the read counts are weakly correlated. Hence, it can be said that the read counts are not independent, which violates a simple Poisson model. Correlation can exist between variables that are drawn from mixture models [15]. On the other hand, this correlation can be an issue when a single model is assumed for the whole genome. However, there is a direction that can be followed to satisfy conditional independence. That direction is Hidden Markov Models (HMM), which are introduced in the following section, Section 3.2.

## 3.2 Hidden Markov Models

It is known that in CHIP-Seq experiments, a few regions of the genome are selectively targeted, so the zero counts may not be random events. It can be assumed that some of the zero counts are structural whilst others are stochastic. Given these assumptions, HMM

can be used to model the read counts. We assume a model with two types of hidden states. First, a state that produces structural zero counts. Second, states that produce Poisson random variables. To clarify the idea, let $C_t$ be a hidden state at position $t$ and let $x_t$ be an observed read count at position $t$ (see Figure 3.1).

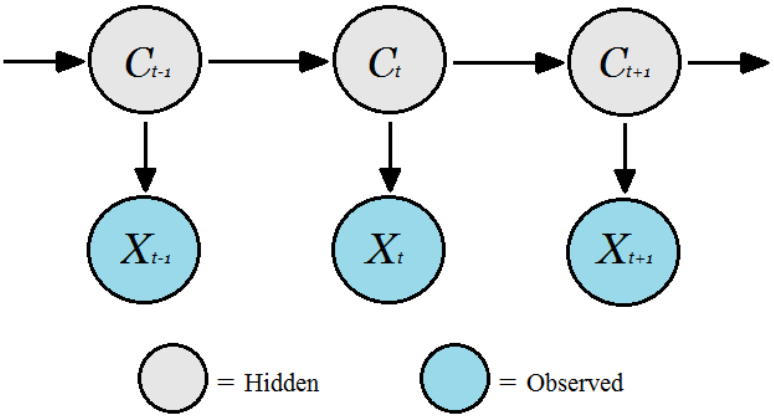



Figure 3.1: A simple structure of Hidden Markov Models.

HMM provide a property of conditional independence. That is, given the hidden state $C_t$, $X_t$ is independent from any other read counts. The property deals with the dependence issue between the read counts. Markov's property is held here, which is a hidden state $C$ at position $t$ that is independent from all states at all previous positions except the hidden state at position $t-1$. This can be shown as follows:

$$P(C_t|C_{1:(t-1)}) = P(C_t|C_{t-1}). \tag{3.1}$$

Let us start with a simple HMM to model the read counts. We assume two hidden states $C = 1$ and $C = 2$, where $C = 1$ is for structural zero counts, and $C = 2$ is for a Poisson model with rate parameter $\lambda$, Poi($\lambda$). $C = 1$ produces only counts, $x = 0$. $C = 2$ produces all possible counts from Poi($\lambda$), $x = 0, 1, 2, \ldots$. In term of probabilities, the model can be formalized as follows:

$$P(X_t = x|C_t = 1) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise,} \end{cases} \tag{3.2}$$

$$P(X_t = x|C_t = 2) = e^{-\lambda}\frac{\lambda^x}{x!}, \quad x = 0, 1, 2, \ldots \text{ and } \lambda > 0. \tag{3.3}$$

Equations (3.2) and (3.3) are called emission probabilities. The marginal probability function of $X$ can be written as follows:

$$\begin{aligned} P(X_t = x) &= P(C_t = 1) \times P(X_t = x|C_t = 1) + P(C_t = 2) \times P(X_t = x|C_t = 2) \\ &= \begin{cases} P(C_t = 1) + P(C_t = 2) \times e^{-\lambda} & \text{if } x = 0 \\ P(C_t = 2) \times e^{-\lambda}\frac{\lambda^x}{x!} & \text{if } x = 1, 2, \ldots. \end{cases} \end{aligned} \tag{3.4}$$

The model requires several parameters, which are:

- probabilities of the hidden states at position 1, $u(1) = c(P(C_1 = 1), P(C_1 = 2))$, where $P(C_1 = 1) + P(C_1 = 2) = 1$;
- transition probabilities between the hidden states, $\gamma_{ij} = P(C_t = j|C_{t-1} = i)$ where $i = 1, 2$ and $j = 1, 2$; note that the transition probabilities are constrained, where $\gamma_{11} + \gamma_{12} = 1$ and $\gamma_{21} + \gamma_{22} = 1$;
- rate parameter $\lambda$, see Figure 3.2.

By considering the HMM for the read counts, we can obtain the posterior probability distribution of $C_{t+1}$ given the observed read counts $x_1, x_2, \ldots, x_t$. That can be formalised as follows.

$$P(C_{t+1} = i|x_{1:T}) = \frac{P(X_{1:T} = x_{1:T}, C_t = i)}{P(X_{1:T} = x_{1:T})}, \quad i = 1, 2; \tag{3.5}$$

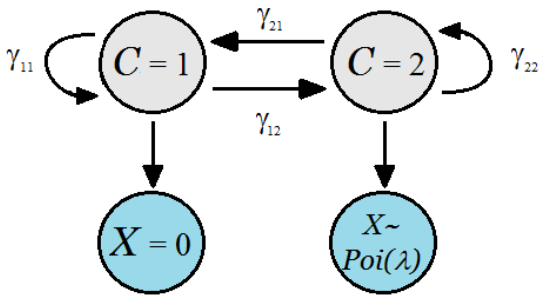



Figure 3.2: The suggested HMM to model the read counts.

where

$$P(X_{1:T} = x_{1:T}, C_t = i) = P(C_t = i)P(X_{1:T} = x_{1:T}|C_t = i), \tag{3.6}$$

$$P(X_{1:T} = x_{1:T}) = \sum_{i=1}^{2} P(X_{1:T} = x_{1:T}, C_t = i),$$

and $T$ is the last position. By obtaining the posterior distribution, we obtain a full HMM for the read counts.

### 3.2.1 Parameter estimation

There are four parameters that need to be estimated and we seek maximum likelihood estimates (mle ). In fact, there are seven parameters in the model, but we need to estimate four owing to the constrains on them. The parameters are the probability of the hidden state at position $t = 1$ either P$(C_1 = 1)$ or P$(C_1 = 2)$, transition probabilities between the hidden states either $\gamma_{11}$ or $\gamma_{12}$ and either $\gamma_{21}$ or $\gamma_{22}$, and the rate parameter $\lambda$. The joint probability function, Equation (3.6), is used to obtain the estimates by maximising its likelihood function. If the full data are provided $x_{1:T}$ and $C_{1:T}$ where $1 : T$ are all possible positions, then the likelihood function $L$ can be written as follows:

$$L(.|x_{1:T}, C_{1:T}) = P(C_{t=1}) \prod_{t=2}^{T} P(C_t|C_{t-1}) \prod_{t=1}^{T} P(x_t|C_t).$$

The $\log$ likelihood function, $\ell$, can be shown as,

$$\ell(.|x_{1:T}, C_{1:T}) = \log\left(P(C_{t=1})\right) + \sum_{t=2}^{T} \log\left(P(C_t|C_{t-1})\right) + \sum_{t=1}^{T} \log\left(P(x_t|C_t)\right). \quad (3.7)$$

Providing the full data, the above log-likelihood can be written in slightly different formalisation as follows. Let us define two indicator functions $V_i(t)$ and $W_{ij}(t)$ for $i = 1, 2$ and $j = 1, 2$ as;

$$\begin{aligned} V_i(t) &= \begin{cases} 1 & \text{if } C_t = i \\ 0 & \text{else;} \end{cases} \\ W_{ij}(t) &= \begin{cases} 1 & \text{if } C_{t-1} = i \text{ and } C_t = j \\ 0 & \text{else.} \end{cases} \end{aligned} \quad (3.8)$$

Then, the log-likelihood function, Equation (3.7), can be rewritten as follows:

$$\begin{aligned} \ell(.|x_{1:T}, C_{1:T}) &= \sum_{i=1}^{2} \log u_i(1) V_i(1) \\ &+ \sum_{i=1}^{2} \sum_{j=1}^{2} \log \gamma_{ij} \sum_{t=2}^{T} W_{ij}(t) \\ &+ \sum_{i=1}^{2} \sum_{t=1}^{T} \log P_i(x_t) V_i(t); \end{aligned} \quad (3.9)$$

where $u_i(1) = P(C_{t=1} = i)$, $\gamma_{ij} = P(C_t = j|C_{t-1} = i)$, and
$P_i(x_t) = P(X_t = x_t|C_t = i)$.

As mentioned, Equation (3.9) would be used if the full data were provided. However, in realty it is hard to provide full data, $i.e$, full population. What is provided normally is a sample that is, under ideal conditions, a representation of the population. Hence

in such cases with HMM, the *Baum-Welch* algorithm is used to obtain the mle of the needed parameters. BaumWelch is simply an EM algorithm (Expectation Maximisation) for HMM. That is, the expected $\ell$ is computed given the observed data $x_{1:T}$ and initial starting parameters $\theta_0$, where $\theta$ is a vector parameter to be estimated. This expectation can be shown as:

$$\begin{aligned} E[\ell(x_{1:T}, C_{1:T}|\theta)|x_{1:T}, \theta_0] &= E\left[\sum_{i=1}^{2} \log u_i(1) V_i(1) + \sum_{i=1}^{2}\sum_{j=1}^{2} \log \gamma_{ij} \sum_{t=2}^{T} W_{ij}(t) \right. \\ &\quad \left. + \sum_{i=1}^{2}\sum_{t=1}^{T} \log P_i(x_t) V_i(t) \,\middle|\, x_{1:T}, \theta_0 \right]. \end{aligned} \tag{3.10}$$

In Equation (3.10), it can be seen that the only random parts are the indicator functions. Hence, the expectation, Equation (3.10), can be rewritten as follows:

$$\begin{aligned} E\left[\ell(x_{1:T}, C_{1:T}|\theta)|x_{1:T}, \theta_0\right] &= \sum_{i=1}^{2} \log u_i(1) E\left[V_i(1)|x_{1:T}, \theta_0\right] \\ &+ \sum_{i=1}^{2}\sum_{j=1}^{2} \log \gamma_{ij} \sum_{t=2}^{T} E\left[W_{ij}(t)|x_{1:T}, \theta_0\right] \\ &+ \sum_{i=1}^{2}\sum_{t=1}^{T} \log P_i(x_t) E\left[V_i(t)|x_{1:T}, \theta_0\right]. \end{aligned} \tag{3.11}$$

Let $\tilde{V}_{i0} = E\left[V_i(t)|x_{1:T}, \theta_0\right]$ and $\tilde{W}_{ij0} = E\left[W_{ij}(t)|x_{1:T}, \theta_0\right]$. As $V_i$ and $W_{ij}$ are indicators, then we have:

$$\tilde{V}_{i0}(t) = P(V_i(t) = 1|x_{1:T}, \theta_0) = P(C_t = i|x_{1:T}, \theta_0), \tag{3.12}$$

and

$$\tilde{W}_{ij0}(t) = P(W_{ij}(t) = 1|x_{1:T}, \theta_0) = P(C_{t-1} = i, C_t = j|x_{1:T}, \theta_0). \tag{3.13}$$

In HMM, there are two components called forward and backward probabilities, $\alpha$ and $\beta$, respectively. These two components are defined as follows.

$$\begin{aligned} \alpha_i(t) &= P(X_{1:t} = x_{1:t}, C_t = i), \\ \beta_i(t) &= P(X_{(t+1):T} = x_{(t+1):T} | C_t = i). \end{aligned} \tag{3.14}$$

By using the forward and backward probabilities, Equation (3.14), Equations (5.2) and (3.13) can be rearranged as;

$$\tilde{V}_{i0}(t) = \frac{\alpha_{i0}(t)\beta_{i0}(t)}{\alpha_0(T)\mathbf{1}'}; \tag{3.15}$$

and

$$\tilde{W}_{ij0}(t) = \frac{\alpha_{i0}\gamma_{ij0}P_{j0}(x_t)\beta_{j0}(t)}{\alpha_0(T)\mathbf{1}'}; \tag{3.16}$$

where $\mathbf{1}$ is the identity matrix. In addition, the posterior probability, Equation (3.5), can be rewritten in terms of forward and backward probabilities as;

$$P(C_t = i|x_{1:T}) = \frac{\alpha_i(t)\beta_i(t)}{P(X_{1:T} = x_{1:T})}, \quad i = 1, 2. \tag{3.17}$$

Obtaining Equations (3.15) and (3.16) means achieving the expectation step in the Baum-Welch algorithm. Hence, still the maximisation step . The expected $\ell$ is maximised with respect to the needed parameters, which are $u_i(1), \gamma_{ij}$ and $\lambda$. These maximisations can be derived as follows:

$$\hat{u}_i(1) = \frac{\tilde{V}_{i0}(1)}{\sum_{i=1}^{2} \tilde{V}_{i0}(1)} = \tilde{V}_{i0}(1), \quad \text{where } \sum_{i=1}^{2} \tilde{V}_{i0}(1) = 1; \tag{3.18}$$

$$\hat{\gamma}_{ij} = \frac{\sum_{t=2}^{T} \tilde{W}_{ij0}(t)}{\sum_{j=1}^{2}\sum_{t=2}^{T} \tilde{W}_{ij0}(t)}; \tag{3.19}$$

$$\hat{\lambda} = \frac{\sum_{t=1}^{T} x_t \tilde{V}_{20}(t)}{\sum_{t=1}^{T} \tilde{V}_{20}(t)}. \tag{3.20}$$

By achieving the maximisation step, updated estimates for the parameters are obtained. These estimates are considered as improved estimates compared to the initial estimates. Hence, these improved parameters are used as initials , $\theta_0$, and the two steps of the Baum-Welch algorithm are applied again. The two steps are repeated recursively until convergence, and the final estimates are considered to be the mle.

We applied the Baum-Welch algorithm to the read counts for each sample separately. We started with the initial parameters obtained from the data. That is, $\lambda_0$ is the average read counts excluding the zero counts, $P(C_{10} = 1)$ is the proportion of zero counts excluding the expected zero counts under Poi$(\lambda_0)$, and a transition matrix $\Gamma_0(2 \times 2)$ contains the probability of moving from zero to non-zero counts and from non-zero to zero counts and their cumulates . We applied this method for different chromosomes. For instance, in chromosome 22 from the test sample, we started with the initial parameters as:

$$\begin{aligned} \lambda_o &= 1.61, \\ u_o(1) &= (0.97, 0.03), \\ \Gamma_o &= \begin{pmatrix} \gamma_{11} = 0.97 & \gamma_{12} = 0.03 \\ \gamma_{21} = 0.94 & \gamma_{22} = 0.06 \end{pmatrix}. \end{aligned}$$

### 3.2.2 Results

By applying the proposed HMM with the Baum-Welch algorithm for the read counts, we faced a problem in the parameter estimation process. That is, in Equation (3.14), it can be seen that if the sample size $T$ is large, then $\alpha_i(t) \longrightarrow 0$ as $t \longrightarrow \infty$, and $\beta_i(t) \longrightarrow 0$ as $t \longrightarrow 0$. In this case, the expectation step, Equations (3.15) and (3.16), cannot be

achieved. As a result, the parameters cannot be estimated. Unfortunately, this is the case in the read counts data. For example, in chromosome 1 from the test sample, we can obtain estimates using a maximum number of around 1000 positions, and by increasing the number of positions, the estimates cannot be obtained any more (this is discussed further in Section 3.4).

Although the parameters cannot be estimated, the posterior probabilities, Equation (3.17), can be estimated. That is, we can estimate the posterior using the possible maximum number of positions, for example, 1000 positions in chromosome 1 from the test sample. Then, to calculate the posterior for the next possible number of positions, and next, update the old posterior using the last one. We carry on updating the posterior until we complete all positions.

## 3.3 Mixture Model

It was shown that the read counts are globally weakly correlated. We tried to model the counts by using HMM to take advantage of the conditional independence property, as well as to control the huge number of zero counts that we think are structured. However, we could not estimate the required parameters in the model. On the other hand, we also tried to model the read counts as a Poisson model with a single rate parameter and two rate parameters, but the models do not fit the data. Hence, we had to look for a different way to model the counts.

### 3.3.1 Mixture of two components

We believe that the zero counts are effectively influencing the distribution of the read counts. By revisiting Figure 2.6, it can be seen the distribution of the lengths of consecutive zero counts (or the gaps between non-zero counts) appears as a mixture of

distributions. It is known that if a random variable follows a Poisson distribution with rate parameter $\lambda$, then the the lengths of consecutive zeros in that variable follow Geometric distribution. Therefore, if the read counts followed Poisson with a single rate parameter, the distribution of the consecutive zero lengths in Figure 2.6 (c,d) would appear as a single component with a linear decay as the lengths get larger. The linear decay starts in the figure after lengths of around 80, but for lengths lower than that the behaviour is different. Based on that. we can say the read counts do not follow a single Poisson distribution.

We seek a mixture model that takes into account the zero read counts. We still employ the Poisson distribution to model the read counts, but not a single Poisson. We started with a model of two Poisson components with two rate parameters $\lambda_1$ and $\lambda_2$. The two components can describe two types of regions in the data. First, regions that contain few or no reads. For these regions, we expect the rate to be very low. Second, regions that are rich in reads. These regions can exist where TF biding sites are located, or where some other genetic activations exist. Hence, for the second type of region, we expect for the rate to be higher than the rate of the first type. How can the zero counts be involved in the model? We let the zero counts determine the weights of the components in the model. However, we do not mean all zeros . That is, the model is built in a way that keeps a memory of the number of consecutive zero counts since the last non-zero count has been observed. This memory is used to calculate the weights. The model can be formulated as follows:

Let $x_i$ be the read count at position $i$ and let $r_i$ be the distance since the last non-zero count has been observed until position $i$. Then, the probability function can be written as:

$$\Pr(X = x|R_i = r) = \delta_1(r)e^{-\lambda_1}\frac{\lambda_1^x}{x!} + \delta_2(r)e^{-\lambda_2}\frac{\lambda_2^x}{x!}, \tag{3.21}$$

where $x = 0, 1, 2$, $r = 0, 1, 2$, and $\delta_1(r)$ and $\delta_2(r)$ are the weights, which are functions of $r$, and under the constraint $\sum_{j=1}^{2} \delta_j(r) = 1$.

In the model, denoted by Equation (3.21), we treat the read counts as a type of series

with respect to consecutive zero counts, where the series keeps a short memory of the consecutive zero counts until a non-zero count appears, then this memory is reset immediately after that. As an illustration, at position $i = 1$, the length of previous consecutive zeros is zero, $r_1 = 0$. Moreover, if the observed read counts at positions $i = 1, 2, 3$ are zeros, then at position $i = 4$ the lengths of zeros is $r_4 = 3$. Furthermore, if the observed read counts at positions $i = 1, 2, 3$ are zeros and at position $i = 4$ the observed is a non-zero read count, then at position $i = 5$ the length of consecutive zeros is $r_5 = 0$.

The temporary memory determines the weights of each component in the model. The weight of each component is defined as a function of $r$, which is the short memory of the consecutive zero counts. The problem now is how to define the functions $\delta_1$ and $\delta_2$. First, one of the weights' functions needs to be defined, as the second weight can be calculated by using the constraint $\sum_{j=1}^{2} \delta_j(r) = 1$. Second, we use the information shown in Figure 2.6 to define the function of the weight as follows:

Let the first component of the model, Equation (3.21), represent read counts of low rate, Poi$(\lambda_1)$. Hence, the second component represents the higher rate of the read counts, Poi$(\lambda_2)$. The weight of the first component, $\delta_1(r)$, can be calculated by using the constraint. Therefore, $\delta_2$ only needs to be defined. From Figure 2.6, we can see a bell shape centred around zeros of length 50. Hence we assume $\delta_2$ to be a Normal probability density function with mean $\mu$ and standard deviation $\sigma$, $\delta_2 \sim N(\mu, \sigma)$. That means that the weight of the second component, Poi$(\lambda_2)$, increases for lengths of zeros of around $\mu$ and the maximum weight occurs when the length of consecutive zeros is equal to $\mu$, $R = \mu$. Hence, for a given length of consecutive zeros $r$, the weight can be calculated:

$$\delta_2(R = r) = \gamma \times \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-1}{2\sigma^2}(r-\mu)^2} \tag{3.22}$$

where $\gamma$ is a scaling parameter. For the mixture model Equation (3.21), the expectation,

the second moment and the variance can be shown as:

$$\begin{aligned} E\left(X|R\right) &= \delta_1(r)\lambda_1 + \delta_2(r)\lambda_2, \\ E\left(X^2|R\right) &= \delta_1(r)\left(\lambda_1^2+\lambda_1\right) + \delta_2(r)\left(\lambda_2^2+\lambda_2\right), \\ V\left(X|R\right) &= E\left(X^2|R\right) - E\left(X|R\right)^2 \\ &= \delta_1(r)\left(\lambda_1^2+\lambda_1\right) + \delta_2(r)\left(\lambda_2^2+\lambda_2\right) - \left(\delta_1(r)\lambda_1 + \delta_2(r)\lambda_2\right)^2. \end{aligned} \tag{3.23}$$

**Parameter estimation**

We built the mixture model in Equation (3.21) based on the consecutive zero lengths $r$, which is shown in Figure 2.6. There are four parameters in the model that need to be estimated, which are $\lambda_1$, $\lambda_2$, $\mu$ and $\sigma$. By obtaining $\hat{\mu}$ and $\hat{\sigma}$, we can calculate the second weight for a given $r$ ($\delta_2(r)$), and then compute the first weight by using the constraint $\delta_1(r) = 1 - \delta_2(r)$. We also want to employ the consecutive zero lengths to estimate the required parameters in the model. Therefore, let us derive the probability zero(s) from the model, Equation (3.21). The probability of a zero count at position $i$ and for any given $r$ is:

$$\text{Pr}(X = 0|R = r) = \delta_1(r)e^{-\lambda_1} + \delta_2(r)e^{-\lambda_2}. \tag{3.24}$$

From Equation (3.24), the probability of observing a non-zero count at any position $i$ and for any given length of consecutive zeros $r$ is:

$$\begin{aligned} \text{Pr}(X > 0|R = r) &= 1 - \text{Pr}(X = 0|R = r) \\ &= 1 - \left\{\delta_1(r)e^{-\lambda_1} + \delta_2(r)e^{-\lambda_2}\right\}. \end{aligned} \tag{3.25}$$

Now, let us consider the probability of observing a gap of two zeros. The probability can be derived as follows.

$$\begin{aligned} \text{Pr}(X_i = 0, X_{i+1} = 0, X_{i+2} > 0|R_i = 0) &= \text{Pr}(X_i = 0|R_i = 0) \times \text{Pr}(X_{i+1} = 0|R_{i+1} = 1) \\ &\quad \times \text{Pr}(X_{i+2} > 0|R_{i+2} = 2) \\ &= \left(\delta_1(0)e^{-\lambda_1} + \delta_2(0)e^{-\lambda_2}\right) \times \left(\delta_1(1)e^{-\lambda_1} + \delta_2(1)e^{-\lambda_2}\right) \\ &\quad \left(1 - \left\{\delta_1(2)e^{-\lambda_1} + \delta_2(2)e^{-\lambda_2}\right\}\right) \\ &= \left(1 - \left\{\delta_1(2)e^{-\lambda_1} + \delta_2(2)e^{-\lambda_2}\right\}\right) \prod_{j=0}^{2-1} \left(\delta_1(j)e^{-\lambda_1} + \delta_2(j)e^{-\lambda_2}\right). \end{aligned} \tag{3.26}$$

Similarly, the probability of observing a gap of three consecutive zeros is:

$$\begin{aligned}\Pr(X_i=0,X_{i+1}=0,X_{i+2}=0,X_{i+3}>0|R_i=0) &= \Pr(X_i=0|R_i=0)\times\Pr(X_{i+1}=0|R_{i+1}=1)\\ &\quad\times\Pr(X_{i+2}=0|R_{i+2}=2)\times\Pr(X_{i+3}>0|R_{i+3}=3)\\ &= \left(1-\left\{\delta_1(3)e^{-\lambda_1}+\delta_2(3)e^{-\lambda_2}\right\}\right)\prod_{j=0}^{3-1}\left(\delta_1(j)e^{-\lambda_1}+\delta_2(j)e^{-\lambda_2}\right).\end{aligned} \tag{3.27}$$

Hence, for any gap of length $k$, where $k>1$, with consecutive zeros starting at position $i$, the probability function can be shown as:

$$\Pr(X_i=0,\ldots,X_{i+k-1}=0,X_{k+1}>0|R=0)=\left(1-\left\{\delta_1(k)e^{-\lambda_1}+\delta_2(k)e^{-\lambda_2}\right\}\right)\prod_{j=0}^{k-1}\left(\delta_1(j)e^{-\lambda_1}+\delta_2(j)e^{-\lambda_2}\right). \tag{3.28}$$

There are two special cases, which are when $k$ equals 0 or 1. For $k=0$ Equation (3.25) is used, and for $k=1$ we use Equation (3.24), which is as follows:

$$\Pr(X_i>0|R=0)=1-\left\{\delta_1(0)e^{-\lambda_1}+\delta_2(0)e^{-\lambda_2}\right\}. \tag{3.29}$$

$$\Pr(X_i=0|R=0)=\delta_1(0)e^{-\lambda_1}+\delta_2(0)e^{-\lambda_2}. \tag{3.30}$$

The probability function of lengths of consecutive zeros was derived and obtained in Equation (3.28) with its two special cases in Equations (3.29,3.30). A question can be asked here: is it possible to estimate the parameters by using only the lengths of consecutive zeros, and not the full data in a Poisson model? The answer is yes, it is possible, and that can be shown by simulation.

We conducted a simulation study by using a single Poisson model, Poi($\lambda$), to check the ability to estimate the model's parameter by using the length of the consecutive zeros in the observations. Let $X$ be a random variable that follows a Poisson distribution with rate $\lambda$ and probability mass function:

$$\Pr(X=x)=e^{-\lambda}\frac{\lambda^x}{x!},$$

where $\lambda > 0$ and $x = 0, 1, 2, \ldots$. Hence, we have the following probabilities:

$$\begin{aligned} \Pr(X = 0) &= e^{-\lambda}, \\ \Pr(X > 0) &= 1 - e^{-\lambda}, \end{aligned}$$

and by using independence property we have:

$$\Pr(X_1 = 0, X_2 = 0, X_3 = 0, \ldots, X_k = 0) = \{\Pr(X = 0)\}^k = e^{-\lambda k}.$$

Let $k$ be the length of consecutive zero observations followed by non-zero observation Then, the probability mass function of the lengths of consecutive zero variable, $K$, can be written as:

$$\begin{aligned} \Pr(K = k) &= \{\Pr(X = 0)\}^k \Pr(X > 0) \\ &= e^{-\lambda k}\left(1 - e^{-\lambda}\right), \quad k = 0, 1, 2, \ldots. \end{aligned} \tag{3.31}$$

By using data $k$, which is a subset of the full data $x$, we want to obtain the mle of $\lambda$. As only the subset $K$ is used, its probability function, Equation (3.31), is used in the estimation process. The likelihood function $L(\lambda|k)$ can be shown as follows.

$$\begin{aligned} L(\lambda|k) &= \prod_{i=1}^{n} \Pr(K = k_i) \\ &= \prod_{i=1}^{n} e^{-\lambda k_i}\left(1 - e^{-\lambda}\right) \\ &= \left(1 - e^{-\lambda}\right)^n \prod_{i=1}^{n} e^{-\lambda k_i}, \end{aligned}$$

where $n$ is the total number of observations $k$. For simplicity, let us consider the log likelihood function $\ell(\lambda|k) = \log(L(\lambda|k))$, and then we have,

$$\ell(\lambda|k) = n \log\left(1 - e^{-\lambda}\right) - \lambda \sum_{i=1}^{n} k_i. \tag{3.32}$$

To obtain the mle of $\lambda$, we differentiate Equation (3.32) with respect to $\lambda$, and set the derivative to zero as follows:

$$\begin{aligned} \frac{\partial \ell(\lambda|k)}{\partial \lambda} &= \frac{ne^{-\lambda}}{1 - e^{-\lambda}} - \sum_{i=1}^{n} k_i \\ 0 &= \frac{ne^{-\hat{\lambda}}}{1 - e^{-\hat{\lambda}}} - \sum_{i=1}^{n} k_i \\ \hat{\lambda} &= -\log\left(\frac{\bar{k}}{1 + \bar{k}}\right). \end{aligned} \tag{3.33}$$

Hence, by using data $k$ we can obtain the mle of $\lambda$ directly from Equation (3.33). Moreover, the mle of $\lambda$ can also be obtained by using numerical optimisation methods by maximising the log likelihood in Equation (3.32).

Returning to the real data, which is the read counts, we want to use the observed consecutive zeros to estimate the mles of the parameters of the model in Equation (3.21), and by using Equations (3.28, 3.29 and 3.30). Hence, the likelihood function when $k > 1$, $L(.|k > 1)$, is,

$$L(.|k>1) = \prod_{i=1}^{n_{k>1}} \left\{ \left(1 - \left\{\delta_1(k_i)e^{-\lambda_1} + \delta_2(k_i)e^{-\lambda_2}\right\}\right) \prod_{j=0}^{k_i-1} \left(\delta_1(j)e^{-\lambda_1} + \delta_2(j)e^{-\lambda_2}\right) \right\}, \tag{3.34}$$

where $n_{k>1}$ is the number of observations $k$ where $k$ is larger than 1.

Again, for simplicity we consider the log likelihood, $\ell(.|k>1)$, as the following:

$$\begin{aligned}
\ell(.|k>1) &= \log\left[\prod_{i=1}^{n_{k>1}} \left\{ \left(1 - \sum_{s=1}^{2} \delta_s(k_i)e^{-\lambda_s}\right) \prod_{j=0}^{k_i-1} \left(\sum_{s=1}^{2} \delta_s(j)e^{-\lambda_s}\right) \right\}\right] \\
&= \sum_{i=1}^{n_{k>1}} \log\left\{ \left(1 - \sum_{s=1}^{2} \delta_s(k_i)e^{-\lambda_s}\right) \prod_{j=0}^{k_i-1} \left(\sum_{s=1}^{2} \delta_s(j)e^{-\lambda_s}\right) \right\} \\
&= \sum_{i=1}^{n_{k>1}} \left[\log\left(1 - \sum_{s=1}^{2} \delta_s(k_i)e^{-\lambda_s}\right) + \log\left\{ \prod_{j=0}^{k_i-1} \left(\sum_{s=1}^{2} \delta_s(j)e^{-\lambda_s}\right) \right\}\right] \\
&= \sum_{i=1}^{n_{k>1}} \log\left(1 - \sum_{s=1}^{2} \delta_s(k_i)e^{-\lambda_s}\right) + \sum_{i=1}^{n_{k>1}} \sum_{j=0}^{k_i-1} \log\left(\sum_{s=1}^{2} \delta_s(j)e^{-\lambda_s}\right).
\end{aligned} \tag{3.35}$$

Similarly, when $k = 0$ and $k = 1$, Equations (3.29) and (3.30), the log likelihood can be respectively shown as:

$$\ell(.|k=0) = n_0 \times \log\left(1 - \sum_{s=1}^{2} \delta_s(0)e^{-\lambda_s}\right), \tag{3.36}$$

and

$$\ell(.|k=1) = n_1 \times \log\left(\sum_{s=1}^{2} \delta_s(0)e^{-\lambda_s}\right), \tag{3.37}$$

where $n_0$ and $n_1$ are the numbers of observation $k$ where $k$ equals 0 and 1, respectively. Hence, the full log likelihood, $\ell(.|k)$, is the summation of the three previous log

likelihoods. That is:

$$\ell\left(.|k\right) = \ell\left(.|k=0\right) + \ell\left(.|k=1\right) + \ell\left(.|k>1\right). \tag{3.38}$$

The mles of the parameters in the mixture model (3.21) can be obtained by obtaining the values of the parameters that maximise the log likelihood in Equation (3.38) for the observed length of consecutive zero counts. The parameters that need to be estimated are $\lambda_1$, $\lambda_2$, $\gamma$, $\mu$ and $\sigma$, where the later three parameters are used in the calculation of the weight $\delta_2$. To estimate these parameters, we use numerical iterative methods, for example Nelder-Mead, to optimise the parameters. In order to use such methods, we need to remove any constraints on the parameters. In the parameters, we have $\lambda_1$, $\lambda_2$, $\gamma$ and $\sigma$ constrained to be larger than zero, whereas $\mu$ ranges from $-\infty$ to $\infty$. Hence, to remove the constraints, re-parameterisations (or transformations) are considered as follows:

$$\begin{aligned}
\eta_1 &= \log(\lambda_1), \\
\eta_2 &= \log(\lambda_2), \\
\pi &= \log(\gamma), \\
\nu &= \log(\sigma).
\end{aligned}$$

By using the above transformations, we need to estimate new unconstrained parameters, which are $\eta_1$, $\eta_2$, $\pi$ and $\nu$. Then, recovering the original parameters from the new estimates as:

$$\begin{aligned}
\hat{\lambda}_1 &= e^{\hat{\eta}_1}, \\
\hat{\lambda}_2 &= e^{\hat{\eta}_2}, \\
\hat{\gamma} &= e^{\hat{\pi}_1}, \\
\hat{\sigma} &= e^{\hat{\nu}}.
\end{aligned}$$

Hence, final equations to be used in the optimisation process are:

$$\begin{aligned}
\ell\left(.|k>1\right) &= \sum_{i=1}^{n_{k>1}} \log\left(1-\sum_{s=1}^{2}\delta_s(k_i)e^{-e^{\eta_s}}\right) + \sum_{i=1}^{n_{k>1}}\sum_{j=0}^{k_i-1}\log\left(\sum_{s=1}^{2}\delta_s(j)e^{-e^{\eta_s}}\right), \\
\ell\left(.|k=0\right) &= n_0 \times \log\left(1-\sum_{s=1}^{2}\delta_s(0)e^{-e^{\eta_s}}\right), \\
\ell\left(.|k=1\right) &= n_1 \times \log\left(\sum_{s=1}^{2}\delta_s(0)e^{-e^{\eta_s}}\right).
\end{aligned} \tag{3.39}$$

where:

$$\begin{aligned}
\delta_1(a) &= 1-\delta_2(a), \\
\delta_2(a) &= e^{\pi_1} \times \tfrac{1}{\sqrt{2\pi e^{2\nu}}} e^{\frac{-1}{2e^{2\nu}}(a-\mu)^2},
\end{aligned}$$

and $\gamma$ is a scaling. The full likelihood is calculated as in Equation (3.38).

**Results**

After processing the optimisation in Equations (3.39) and (3.38), we found that the values of the estimated parameters depend on the initial values. That is, for different starting values for the parameters, we obtained different estimated values for the parameters. However, for some of the parameters, we observed frequent estimated values. That is, $\lambda_1$ was frequently estimated as $0.01$, and $\mu$ was frequently estimated to be in the range from $46$ to $52$. It is expected to have estimates for $\lambda_1$ and $\mu$ around these frequently observed solutions. On the other hand, the estimated values for parameters $\lambda_2$, $\gamma$ and $\sigma$ vary for different initial values.

We tried to generate count data from the mixture model in Equation (3.21) by using some of the optimised solutions, and to see whether or not the generated data simulate the characteristics of the real data. We tried many of the solutions and found the solution where $\hat{\lambda}_1 = 0.01$, $\hat{\lambda}_2 = 0.513$, $\hat{\gamma} = 0.95$, $\hat{\mu} = 51$, and $\hat{\sigma} = 15$ is the closet to the real data between the others in terms of distribution of lengths of consecutive zeros, which is shown in Figure 3.3. The figure is compared to Figure 2.6, (d). Figure 3.3 shows that the mixture model in Equation (3.21) can simulate the bell shape around length 50 and

the linear part after it. However, the first part, which is the sharp exponential decay, does not appear in the figure. This suggests that the mixture model of two components is not enough to simulate the data, and hence an additional component may be needed.

### 3.3.2 Mixture of three components

From the simulation results, we found that a mixture model of two Poisson components is not able to simulate the characteristics of the real data. It was noticed that the suggested model is not able to simulate the full observed distribution of lengths of consecutive zeros, where the sharp exponential decay at the beginning of the distribution is missing. Hence, we decided to add one component more and have a mixture model of three Poisson components with three different rate parameters. We expect from the additional component to simulate the missing part by using two components only, which is the exponential decaying at the beginning of the distribution of lengths of consecutive zeros, in Figure 2.6, (d).

In the two-component mixture model, we expected to have two rate parameters representing low and high read counts in the data. Now, we expect to have the same two components, low and high rates, and another component with a rate in between the two. IN addition, to obtain the exponential decay, we set the weight for the additional component as Exponential probability density function (pdf), with parameter $\theta$, where it is a function of a given length of consecutive zeros. The new mixture model can be shown as follows:

$$\begin{aligned} \Pr(X_i = x | R_i = r) &= \delta_1(r) e^{-\lambda_1} \frac{\lambda_1^x}{x!} + \delta_2(r) e^{-\lambda_2} \frac{\lambda_2^x}{x!} + \delta_3(r) e^{-\lambda_3} \frac{\lambda_3^x}{x!} \\ &= \sum_{s=1}^{3} \delta_s(r) e^{-\lambda_s} \frac{\lambda_s^x}{x!}, \end{aligned} \tag{3.40}$$

where $\delta_s(r)$ is the weight of component $s$, which are functions of $r$ under the constraint $\sum_{s=1}^{3} \delta_s(r) = 1$. For the model (3.40), the expectation, the second moment and the variance

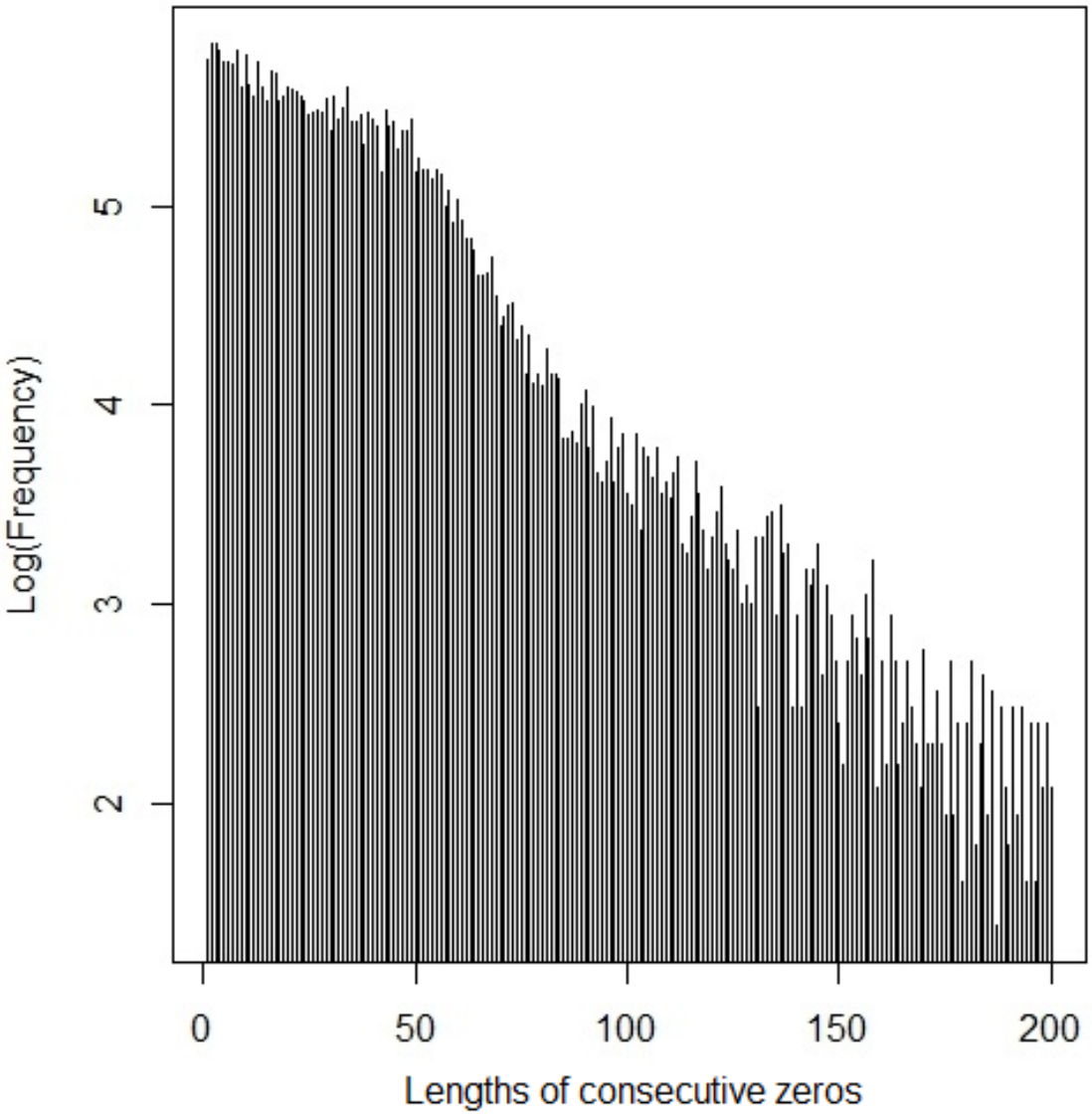



Figure 3.3: Distribution of lengths of consecutive zeros from simulated counts data by using one of the optimal solutions for Equation (3.38), where the mles of the parameters are $\hat{\lambda}_1 = 0.017$, $\hat{\lambda}_2 = 0.513$, $\hat{\gamma} = 0.95$, $\hat{\mu} = 51$ and $\hat{\sigma} = 15$. The vertical axis is the frequency in $\log$ scale, and the horizontal axis is the lengths of consecutive zeros.

can be shown as:

$$\begin{aligned} E\left(X|R\right) &= \delta_1(r)\lambda_1 + \delta_2(r)\lambda_2 + \delta_3(r)\lambda_3, \\ E\left(X^2|R\right) &= \delta_1(r)\left(\lambda_1^2+\lambda_1\right) + \delta_2(r)\left(\lambda_2^2+\lambda_2\right) + \delta_3(r)\left(\lambda_3^2+\lambda_3\right), \\ V\left(X|R\right) &= E\left(X^2|R\right) - E\left(X|R\right)^2. \end{aligned} \tag{3.41}$$

Given the constraint on the weights, we need to estimate two only and calculate the third weight from them. Again, a subset of the data is used, which is the observed lengths of consecutive zeros. In the twocomponent mixture model, Equation (3.21), we set the weight of the second component, Poi$(\lambda_2)$, as pdf of Normal distribution, $N(\mu, \sigma)$, and we expected this component to represent the high rate read counts. The same setting and expectation are still being held for the same component. For the third component, Poi$(\lambda_3)$, we set the weight to be pdf of Exponential, Exp$(\theta)$, and expect this component to represent read counts of low rate but not as low as the first component, Poi$(\lambda_1)$. Hence, we expect to have rates $\lambda_1 < \lambda_3 < \lambda_2$. As the second and the third weights are pdfs and have parameters that need to be estimated, they were chosen to be estimated, and then by using the constraint, the first weight, $\hat{\delta}_1(r)$, is calculated.

As mentioned, the weights $\delta_2$ and $\delta_3$ are pdf of Normal and Exponential distributions. As the weights are constrained to be summed up to 1, then the pdfs need to be normalised. That is, the weights need to be scaled as follows:

$$\begin{aligned} \delta_2(R=r) &= \gamma_1 \times \tfrac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-1}{2\sigma^2}(r-\mu)^2}, \\ \delta_3(R=r) &= \gamma_2 \times \tfrac{1}{\theta} e^{-\frac{r}{\theta}}, \end{aligned} \tag{3.42}$$

where $\gamma_1$ and $\gamma_2$ are scaling parameters and $\gamma_1, \gamma_2 > 0$.

### Parameter estimation

In the mixture model in Equation (3.40) there are eight parameters to be estimated. These are $\lambda_1$, $\lambda_2$, $\lambda_3$, $\mu$, $\sigma$, $\theta$, $\gamma_1$ and $\gamma_2$. We seek mle for the parameters by using a subset of the data, which is the lengths of consecutive zeros. Similar to the two-component mixture

model, Equations (3.28, 3.29, 3.30), we need to define the probability of observing $k$ consecutive zeros starting at position $i$. By using model (3.40), the probability function with its two special cases, which are when $k = 0$ and $k = 1$, can be shown as:

$$\begin{aligned}
\Pr(X_i = 0, \ldots, X_{i+k-1} = 0, X_{k+1} > 0 | R = 0) &= \left(1 - \sum_{s=1}^{3} \delta_s(k) e^{-\lambda_s}\right) \prod_{j=0}^{k-1} \left(\sum_{s=1}^{3} \delta_s(j) e^{-\lambda_s}\right), \quad \text{where } k > 1; \\
\Pr(X_i > 0 | R = 0) &= 1 - \sum_{s=1}^{3} \delta_s(0) e^{-\lambda_s}, \quad \text{where } k = 0; \\
\Pr(X_i = 0 | R = 0) &= \sum_{s=1}^{3} \delta_s(0) e^{-\lambda_s}, \quad \text{where } k = 1.
\end{aligned} \tag{3.43}$$

To obtain the mle, a likelihood function needs to be provided. For simplicity, we directly consider the log likelihood function. Hence, for a given $k$ data, the log likelihood function $\ell(.|k)$ of Equation (3.43) can be shown as the following (similar to Equations (3.35, 3.36, 3.37, 3.38)):

$$\begin{aligned}
\ell\left(.|k > 1\right) &= \sum_{i=1}^{n_{k>1}} \log\left(1 - \sum_{s=1}^{3} \delta_s(k_i) e^{-\lambda_s}\right) + \sum_{i=1}^{n_{k>1}} \sum_{j=0}^{k_i - 1} \log\left(\sum_{s=1}^{3} \delta_s(j) e^{-\lambda_s}\right); \\
\ell\left(.|k = 0\right) &= n_0 \times \log\left(1 - \sum_{s=1}^{3} \delta_s(0) e^{-\lambda_s}\right); \\
\ell\left(.|k = 1\right) &= n_1 \times \log\left(\sum_{s=1}^{3} \delta_s(0) e^{-\lambda_s}\right); \\
\ell\left(.|k\right) &= \ell\left(.|k = 0\right) + \ell\left(.|k = 1\right) + \ell\left(.|k > 1\right).
\end{aligned} \tag{3.44}$$

The mles of the parameters are those maximising the value of $\ell(.|k)$ in Equation (3.44). We follow the same procedure used in the two-component mixture model to obtain the estimates. That is, we use numerical optimisation methods to optimise the estimates after removing any existing constraints on the parameters. In model (3.40), there is only one unconstrained parameter, which is $\mu$. Hence, the constraints need to be removed from the other seven parameters. Similar to the two-component mixture model, we remove the

constraints by doing re-parameterisations as follows:

$$\begin{aligned}
\eta_1 &= \log(\lambda_1) \iff \lambda_1 = e^{\eta_1};\\
\eta_2 &= \log(\lambda_2) \iff \lambda_2 = e^{\eta_2};\\
\eta_3 &= \log(\lambda_3) \iff \lambda_3 = e^{\eta_3};\\
\nu &= \log(\sigma) \iff \sigma = e^{\nu};\\
\tau &= \log(\theta) \iff \theta = e^{\tau};\\
\pi_1 &= \log(\gamma_1) \iff \gamma_1 = e^{\pi_1};\\
\pi_2 &= \log(\gamma_2) \iff \gamma_2 = e^{\pi_2}.
\end{aligned}$$

Hence, the unconstrained parameters are used in the optimisation process, and then the original parameters can be recovered as shown above.

**Results**

After applying the optimisation process and observing the resulting estimates for the parameters, we noticed the same problem found in the two-component mixture model. That is, it was noticed that the values of the resulting estimates depend on the initial values being given. It was observed that some the parameters were frequently having the same solutions, even if the starting values were different. It was observed in most of the resulting solutions that $\hat{\lambda}_1 \simeq 0.01$, $0.02 \leq \hat{\lambda}_3 \leq 0.3$, $45 \leq \hat{\mu} \leq 52$ and $1/6 \leq \hat{\theta} \leq 1/12$. On the other hand, the estimated values of $\lambda_2$, $\sigma$, $\gamma_1$, and $\gamma_2$ vary depending on the starting values.

What was expected was observed for some of the parameters. That is, in all optimised solutions we observed $\hat{\lambda}_1 < \hat{\lambda}_3 < \hat{\lambda}_2$. In addition, $\hat{\mu}$ and $\hat{\theta}$ were estimated in the range we expected. Hence, to understand the reasons for not obtaining consistent solutions for the parameters, we created some likelihood profiles. The purpose of such profiles is to observe the behaviour of the likelihood function for some given values of the parameters in the model. We made many profiles for different possible pair combinations of the

parameters. The process of creating a profile is simple. For a given sequence of values for a certain parameter, other parameters in the model are optimised at each given value and the likelihood (or log likelihood) value is recorded, and then the recorded values are plotted against the given sequence. The result is called a likelihood profile for the given parameter.

Figure 3.4 shows a likelihood profile for $\lambda_2$ and $\sigma$. From the figure, it can be seen that the log likelihood function has an irregular spiky surface. This behaviour explains the unstable estimations for the model's parameters. That is, for such a surface there are many local maximums, and then any optimisation method can stick with any of these maximums and report the estimations as optimal. We created other likelihood profiles for other parameters and observed the same behaviour pattern.

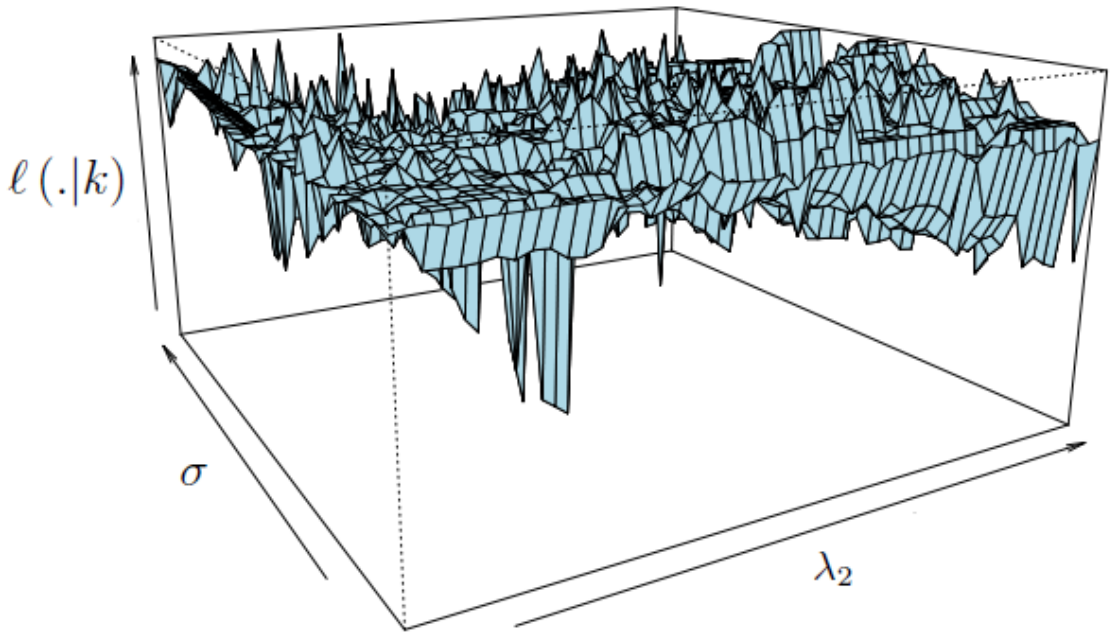



Figure 3.4: A likelihood profile for parameters $\lambda_2$ and $\sigma$ by using log likelihood function, $\ell(.|k)$, in equation (3.44). The values of $\lambda_2$ range from 1 to 10, and of $\sigma$ from 1 to 100. The maximum $\ell(.|k)$ value occurs at $\lambda_2 = 1.73$ and $\sigma = 84.49$.

We picked some of the optimal solutions and used them to generate counts data to see whether or not any of the solutions have the same characteristics as the real data. Figure 3.5 shows a close distribution of lengths of consecutive zeros by using one of the optimal solutions for Equation (3.44). The mle used in this simulation are $\hat{\lambda}_1 = 0.01$, $\hat{\lambda}_2 = 3.1$, $\hat{\lambda}_3 = 0.02$, $\hat{\mu} = 48$, $\hat{\sigma} = 16$, $\hat{\theta} = 1/11$, $\hat{\gamma}_1 = 0.9$ and $\hat{\gamma}_2 = 0.99$. This figure is compared to the distribution of observed lengths of consecutive zeros in the real data (Figure 2.6, (d)). From the simulation's result, it can be seen that the distribution of consecutive zero lengths is quite similar to that of the real data, but not close enough. However, by comparing this result with the simulation's result from the two-component mixture model, Figure 3.3, we can say that the three-component mixture model is much closer to the real data.

**Additional parameter $\alpha$**

We noticed in Figure 3.5 that the exponential decaying at the beginning is not as sharp as the observed one in the real data, Figure 2.6, (d). One way to control the decay is to set a power on the random variable that has Exponential distribution. That is, an additional parameter $\alpha$ is added to the weight of the third component to be as follows:

$$\delta_3(R = r) = \gamma_2 \times \frac{1}{\theta} e^{-\frac{r^\alpha}{\theta}}, \quad \text{where} \quad \alpha > 0. \tag{3.45}$$

We expect that a larger $\alpha$ leads to faster decaying.

**Redundancy issue**

The additional parameter $\alpha$ needs to be estimated, which means the total number of parameters increases by one to be nine. Simply, we remove the constraint from the parameter by the re-parameterisation and optimise the parameters for log likelihood function, $\ell(.|k)$, to obtain the mle. Hence, the optimisation process was done and the result was different after adding $\alpha$. We noticed that different sets of estimated parameters

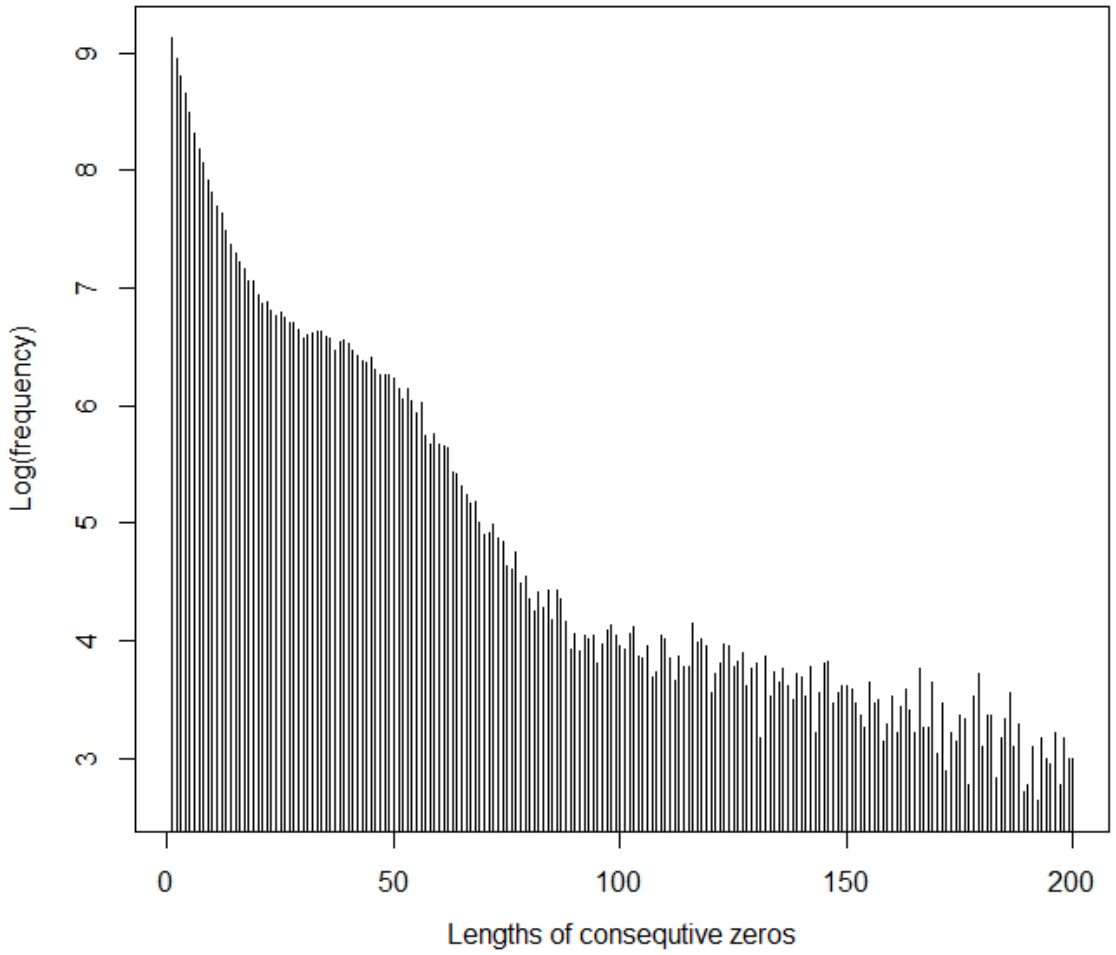



Figure 3.5: Distribution of lengths of consecutive zeros from simulated counts data by using one of the optimal solutions for Equation (3.44), where the mles of the parameters are $\hat{\lambda}_1 = 0.01$, $\hat{\lambda}_2 = 3.1$, $\hat{\lambda}_3 = 0.02$, $\hat{\mu} = 48$, $\hat{\sigma} = 16$, $\hat{\theta} = 1/11$, $\hat{\gamma}_1 = 0.9$ and $\hat{\gamma}_2 = 0.99$. The vertical axis is the frequency in $\log$ scale, and the horizontal axis is the lengths of consecutive zeros.

give the same value for the likelihood function. This finding suggests a redundancy in the model. That is, the parameters in the model are not independent, and hence these dependent parameters make the model redundant. We created a likelihood profile for $\lambda_2$ and $\lambda_3$ and show the result in Figure 3.6.

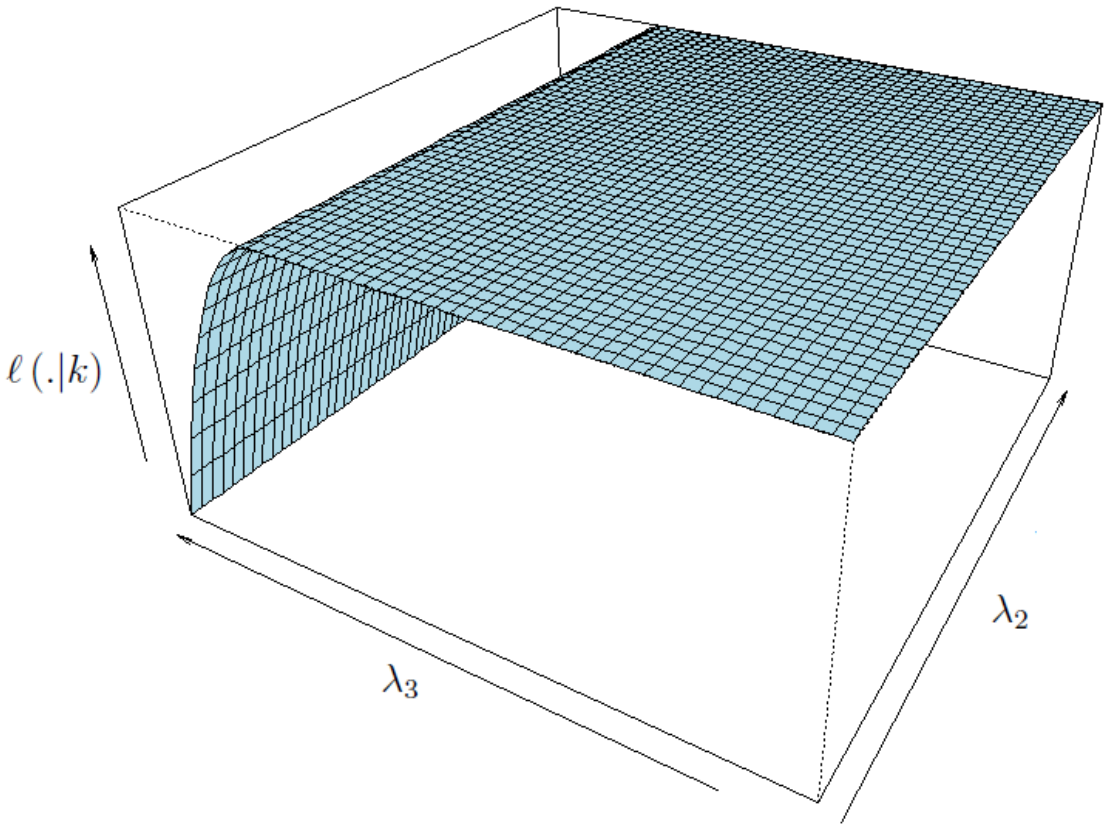



Figure 3.6: A likelihood profile for parameters $\lambda_2$ and $\lambda_3$ by using log likelihood function, $\ell(.|k)$, in equation (3.44) and after adding the parameter $\alpha$ as shown in Equation (3.45).

The likelihood profile of $\lambda_2$ and $\lambda_3$ in Figure 3.6 shows a smooth straight surface for the likelihood function. Hence, we want to find the relationship between the parameters that leads to this behaviour. We found the following. Let $\bar{x}$ be the observed sample mean of

the read counts, then for Equation (3.40) the following is true.

$$\begin{aligned} \bar{x} &= \delta_1(r)\lambda_1 + \delta_2(r)\lambda_2 + \delta_3(r)\lambda_3 \\ \delta_1(r)\lambda_1 &= \bar{x} - \{\delta_2(r)\lambda_2 + \delta_3(r)\lambda_3\} \\ \lambda_1 &= \tfrac{\bar{x} - \{\delta_2(r)\lambda_2 + \delta_3(r)\lambda_3\}}{\delta_1(r)}. \end{aligned} \tag{3.46}$$

The above means that given two of the rate parameters with other weight parameters, the other rate parameter can be calculated from the given ones. Such relation can lead to the behaviour observed in Figure 3.6. Hence, if we decided to use the above equation by calculating $\lambda_1$ from the others, then we would consider an additional constraint, which is:

$$\bar{x} - \{\delta_2(r)\lambda_2 + \delta_3(r)\lambda_3\} > 0. \tag{3.47}$$

Given the above constraint, we need to redefine the ranges of $\lambda_2$ and $\lambda_3$ as:

$$\begin{aligned} 0 < \lambda_2 < \tfrac{\bar{x} - \delta_3(r)\lambda_3}{\delta_2(r)}, \\ 0 < \lambda_3 < \tfrac{\bar{x} - \delta_2(r)\lambda_2}{\delta_3(r)}. \end{aligned} \tag{3.48}$$

The parameters that need to be estimated are $\lambda_2$, $\lambda_3$, $\gamma_2$, $\gamma_3$, $\mu$, $\sigma$, $\theta$ and $\alpha$, then parameter $\lambda_1$ and weight $\delta_1(r)$ are calculated by using the constraints. Hence, after doing the optimisation process and considering the relation between the parameters, the redundancy issue was solved. However, like previous problems, the estimates of the parameters depend on the initial starting values. We created many likelihood profiles for many different combinations of the parameters and found that the likelihood function has irregular behaviour with many local maximums, and this makes the estimation process difficult. Although we could not obtain the mle, there were values of the estimates that appeared frequently. From these frequent solutions we could get an idea of the values of the estimates as follows:

$$
\begin{aligned}
&\hat{\lambda}_1 \simeq 0.01, \\
&\hat{\lambda}_2 \simeq 3.07, \\
&\hat{\lambda}_3 \simeq 1.17, \\
&\hat{\gamma}_2 >> \hat{\gamma}_1, \text{ and } \hat{\gamma}_1 \in (1/3, 1/2), \text{ and } \hat{\gamma}_2 \in (200, 800), \\
&\hat{\mu} \in (44.5, 49), \text{ and } \hat{\sigma} \in (8, 12)' \\
&\hat{\theta} \in (1/50, 1/40), \\
&\hat{\alpha} \in (2, 3.5).
\end{aligned}
\tag{3.49}
$$

Some of the optimised solutions were chosen and used to simulate counts data. Figure 3.7 shows the closest result to the real data in terms of the observed lengths of consecutive zeros. In the figure, it can be seen that the distribution of consecutive zeros is quite similar to that of the real data in Figure 2.6, (d). Three components can be clearly seen, as in the real data figure .

## 3.4 Discussion

In RUNX1/ETO data we noticed two such features, which are the high proportion of zero counts and serial correlation between the read counts across the genome. Based on these two features we tried to functionally model the data. Having a model can serve the objective of the study, which is detecting differential regions between the two samples, by inference. Our aim is to model the data without removing or changing the naturality of it. Hence, the desired model is that which is able to describe the data with its natural features.

We tried to model the read counts by using HMM with two hidden states. The states are for structural zeros and random read counts from the Poisson distribution. This model did not work with the data because of too many lengthy consecutive zeros in the data, which stopped the estimation process for the parameters. Hence, we failed to obtain the estimates

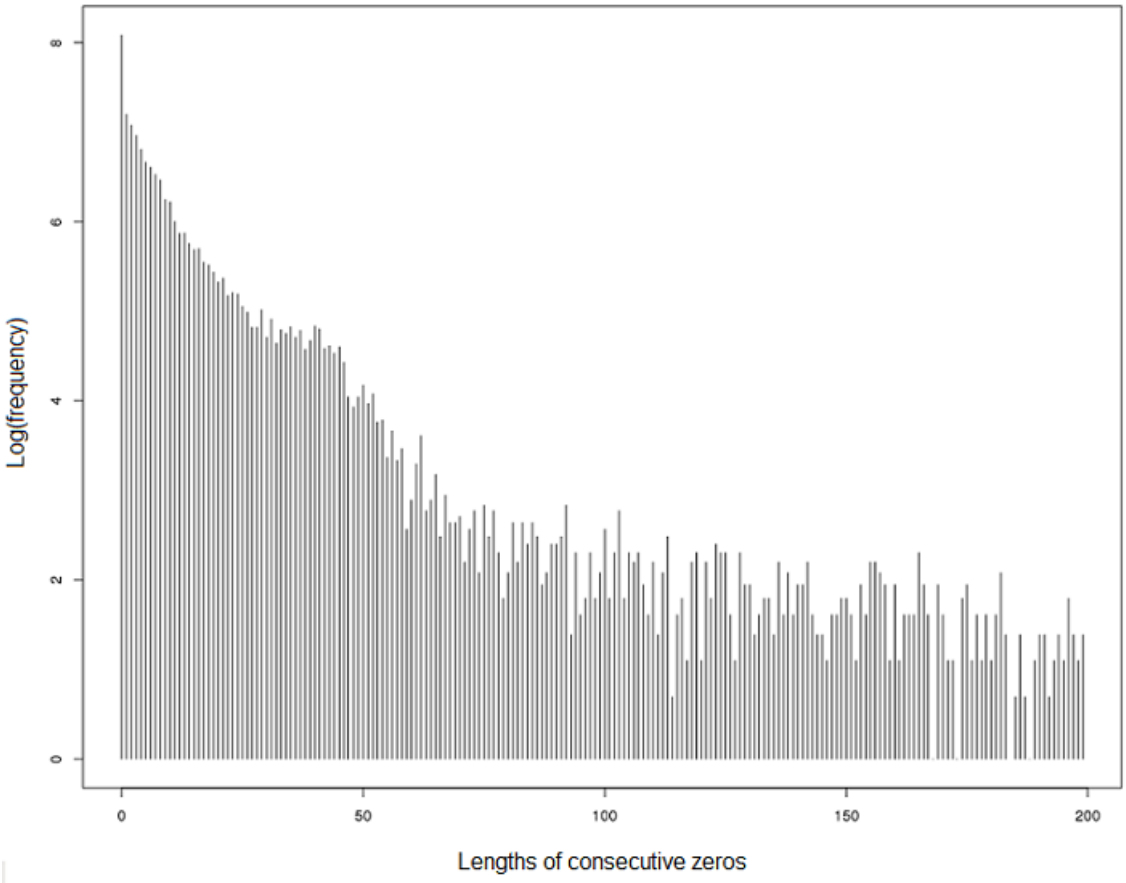



Figure 3.7: Distribution of lengths of consecutive zeros from simulated counts data by using one of the optimal solutions for Equation (3.44), where the mles of the parameters are $\hat{\lambda}_1 = 0.01$, $\hat{\lambda}_2 = 3.1$, $\hat{\lambda}_3 = 1.2$, $\hat{\mu} = 44.6$, $\hat{\sigma} = 8.1$, $\hat{\theta} = 1/42$, $\hat{\gamma}_1 = 0.5$, $\hat{\gamma}_2 = 200$ and $\hat{\alpha} = 2$. The vertical axis is the frequency in $\log$ scale, and the horizontal axis is the lengths of consecutive zeros.

of the parameters of the model. Although HMM can handle the correlation, we thought that it is not suitable to model the data because of the consecutive zero read counts. Note that in the estimation process we transformed the probabilities in log transformation but we failed in estimating the parameters as well.

There are previously proposed methods that are used to analyse ChIP-Seq data for differential binding sites detection, and some of these methods use HMM with Baum-Welch. For instance, ODIN [75], which is distinct from peak- finding methods. The main difference between this method and the HMM we used is the filtering step (which is the case in most of the analysis methods, whether or not they are peak-finding methods). That is, ODIN does a filtering step where about 90% of the genome regions are filtered out, and only dense regions are kept, which are more likely to be peaks. Hence HMM is applied only on the peaks, and other regions are completely ignored. By doing this filtering, low read count regions are removed including zero count regions, so the problems of consecutive and high proportions of zero read counts do not occur. Thus, we believe that HMM with the Baum-Welch algorithm cannot be used for RUNX1/ETO data. It would work if we filtered the read counts. We would generalise this conclusion for all ChIP-Seq data if we applied without filtering HMM with Baum-Welch on some other ChIP-Seq datasets.

In relevance to HMM, Hidden Semi-Markov Model (HSMM) might work with RUNX1/ETO data. The main difference between HMM and HSMM is that in HSMM the hidden state can consider a sequence of observations, whereas the assumption in HMM is a single observation per state [74]. Hence, HSMM might be a better way to deal with the consecutive zero counts.

We also tried another model that handles the characteristics of the data, and proposed a model of mixture Poisson distributions. We attempted to fit this model after checking the fit of the single Poisson model and found that it does not fit the data (by using the Chi-squared goodness of fit test). Correlation can exist between data drawn from a mixture

of random variables and consecutive zeros can be employed in the model. The data were treated as a type of time series, where a short temporary memory of lengths of consecutive zeros is kept. This memory determines the weight of the mixture components. We started with a mixture of two Poisson components and moved to three mixture components as the two-component mixture did not show a reasonable simulation of the real data in the simulation's results. Moreover, the three-component mixture model showed better simulation of the data compared to the two-component model in terms of distribution of lengths of consecutive zeros. However, none of the mixture models showed a close behaviour to the real data in terms of correlation.

In the three-component mixture model and after solving the redundancy problem, we obtained better simulation results, which have similar the characteristics to the real data in terms of consecutive zeros and correlation. On the other hand, similar to the two-component mixture model, we could not reliably obtain the mles of the parameters of the model. It was found that for different initial starting values, the optimisation process returned different solutions, and we found that was caused by the surface of the likelihood function.

By observing RUNX1/ETO data it can be said that a mixture of three Poisson components would be a good model to describe it. That is, a Poisson component with a very low rate for the zero and very low read counts; another component with low rate but larger than the previous component and this would be for the low read counts but larger than zero; and the third component with a slightly large rate for the larger read counts. A model in this structure would handle the changes in the variability of the read counts. Furthermore, we did some comparisons between single-, two- and three-component mixture models to see which one fits the data better by using the Chi-squared test. We found that none of the models fit the data. However, by comparing the test statistics of the models, we can say that the three-component mixture model is better because it has the smallest test statistic compared to the others. Note that the compared models are those shown in Figures 3.5

and 3.7, which are the closet simulation results to the real data.

We checked some ENCODE datasets to see whether the characteristics of RUNX1/ETO data are observed under similar biological conditions. Hence, we chose ChIP-Seq datasets with knock-out experiment. We found the following. In terms of correlation, we found a similar pattern to what we observed in RUNX1/ETO data. The read counts are serially correlated across the genome and the correlation is not significant between the read counts within the window (similar to Figure 2.5). We found that the proportion of zero counts is huge compared to the nonzero counts, also similar to the findings for RUNX1/ETO. However, the distribution of the consecutive zero counts varies. Although we found that some datasets have similar distributions of consecutive zero counts, we cannot say that there is common distribution for the consecutive zero counts in ChIP-Seq datasets. So what is shown in Figure 2.6 is only observed in RUNX1/ETO data. It is logical to observe different distributions for consecutive zero counts in ChIP-Seq experiment because the targeted regions to be chopped are different from experiment to other. Hence, the gaps or non-sequenced regions are different and these regions represent large part of consecutive zero counts.

Finding a statistical model for the whole data is not an easy task. The model consisting of mixture of three Poisson variables seems to simulate the characteristics of the data, although it does not fit it (by using $\chi^2$ goodness of fit test). We found that this model is able to mimic the features of the RUNX1/ETO data. By dividing the data into windows of $200$ bp, the characteristics of these windows are slightly different and easy to model. The correlation between the read counts is not significant within windows. Moreover, the single Poisson component model fits more than two thirds of the non-empty windows.

To perform statistical inference on the data, windows can be used individually. The Poisson model can be assumed for the read counts within windows. Statistical inference can be applied on the two samples by inferring the differences between the same location windows in the two samples. There are many available inference methods that can be

employed (Chapters 4 and 5).

# Chapter 4

# Current methods for identifying differential binding sites

## 4.1 Introduction

In this chapter, we introduce and discuss some previously developed methods for transcription factor (TF ) binding site studies. We consider in detail two methods. First, a combination of MACS, which is a peak-calling method, and MAnorm, which is a peak-calling-based method. Second, diffReps method, which is independent from peaks-calling methods. These methods are applied to simulated data as well as the project data RUNX1/ETO. Finally, we discuss and conclude the findings of these methods.

It is biologically known for TFs to have binding sites located before and after the actual TF locations. These binding sites are enriched with DNA reads. These dense regions of the genome are called peaks. Biologically, changes in the densities of the locations of the binding sites are considered to be signs for changes in the TF. Hence, the binding sites are analysed to detect changes occurring in the TF under different biological conditions.

Two types of methods have been suggested and developed to analyse TF binding sites.

First, peak- based methods. For instance, MAnorm [43], DiffBind [60], DESeq [5] and EdgeR [35], where the last two were proposed initially for gene expression analysis. Second, non-peak-based methods. For instance, diffReps [14] and ODIN [75].

What are peak-calling methods? Peak-calling methods simply search across the genome for enrichment regions (peaks) and report them. There are many proposed methods for calling peaks. For instance, MACS [31], HPeak [73], BayesPeak [34], ZINBA [61], Qeseq [25], OccuPeak [7] and JAMM [20]. Each of these methods has its own methodology to declare a region in the genome as a peak. In binding site analysis, peak-based methods are more commonly used.

From the above previous developed methods, we chose three to be considered in detail and applied to the RUNX1/ETO data. That is, from the peak-calling methods we chose MACS. MACS seems to be the most popular peak calling method as it has around 2,000 citations according to *google scholar* (2015). From peak-based methods, MAnorm was chosen as it was initially proposed for the analysis of ChIP-Seq data. Also unlike many other methods, MAnorm does not need biological replicates to perform the analysis. Finally, from non-peak-based methods we chose diffReps.

The rest of the chapter is constructed as follows. In Section 4.2, we introduce the chosen methods from the current methods for differential binding sites analysis. In Section 4.3, the methods are evaluated by using simulated and real data. In Section 4.4, we apply the methods to RUNX1/ETO data. Finally, in Section 4.5 we discuss and conclude our findings from these methods.

## 4.2 Current methods

### 4.2.1 MACS peak identification

Model-based analysis of ChIP-Seq (MACS) was proposed in 2008 for the purpose of detecting ChIP-Seq peaks. Its inputs from ChIP-Seq data are two samples, *test* and *control*. The control can be either a real control sample or an *input* sample. An input sample contains the biological background (noise) for the ChIP-Seq experiment that is used to produce test and control samples. Moreover, the input sample is DNA sequences, and it can be treated in the same way as test and control samples are treated. In MACS, both test and control samples are compared to the input, separately, to detect the peaks, and then the resulting peaks can be analysed.

MACS is based on three components. These components are user's given information, biological and statistical components. In the user component, window size and fold-enrichment need to be given by the user. Fold-enrichment here is a threshold for the ratio of densities (read counts) of test over input within windows. That is, for a given window size, MACS divides the genome into windows of the given size. MACS slides a window of double the given size across the genome searching for windows that satisfy the given fold-enrichment. Let $x$ and $y$ be the number of reads of test (or control) and input in a window, respectively; the ratio $x/y$ is compared to the given fold-enrichment. By default, the window size is 300 bp and this the minimum window size the MACS software can consider. By default also, the fold-enrichment is the range of ratios from 10 to 30. This means that any ratios less than 10 and more than 30 are not called. Specifying a maximum fold-enrichment is to prevent calling peaks (or windows) with outliers.

In the biological component, DNA fragments in ChIP-Seq experiments are equally likely to be sequenced from both positive and negative ends [31]. That is, for each peak with a positive strand, it is more likely to have a negative stranded peak located close to it,

and vice versa, where both peaks have similar density. MACS benefits from this and uses it as an assumption. MACS introduces a parameter $d$ based on that assumption. $d$ is defined as the distance between the summits of positive and negative stranded peaks that are adjacent. Estimating the distance $d$ is done as follows. After detecting all peaks that satisfy the given fold-enrichment, MACS randomly samples $1,000$ peaks. Then, it separates the positive from negative stranded peaks, then considers them in pairs where each pair has positive and negative peaks. In each pair, the distance between the positions of the summits of the peaks is calculated, say $d_i$ for pair $i$ where $i = 1, 2, \ldots, 500$. The distance $d$ is then estimated as the average of all calculated $d_i$ as:

$$d = \frac{1}{500} \sum_{i=1}^{500} d_i.$$

After obtaining the estimate of $d$, MACS combines each pair of positive and negative peaks, which are located next to each other, in a single peak. That is, MACS shifts the reads of each peak of the pair toward each other by $d/2$ bp.

In the statistical component, MACS employs Poisson distribution to declare the significant peaks. It assumes that read counts of genome have Poisson distribution with rate parameter $\lambda_{\text{BG}}$. The parameter $\lambda_{\text{BG}}$ is estimated as the sample mean of the read counts in the whole genome. MACS uses Poisson as follows. After estimating $d$ and combining the peaks, MACS slides a window of size $2d$ across the genome searching for significant peaks with p-values below a given threshold, which is by default $10^{-5}$. A p-value of a peak is calculated as the probability of observing a read count that is larger than the rate of the peak (sample mean of the peak). Mathematically, this can be defined as follows. Let $x$ be the observed read counts, and hence $X \sim \text{Poi}(\lambda_{\text{GB}})$ where $\lambda_{\text{GB}}$ is estimated by the sample mean of the read counts per position by using whole sample, $\bar{x}$. Let $\lambda_{\text{peak}}$ be the rate of a detected peak, which is estimated as the sample mean of the read counts within the peak. The p-value, $P_{\text{GB}}$, of that peak is calculated as

$$P_{\text{GB}} = \Pr(X > \lambda_{\text{peak}}), \quad X \sim \text{Poi}(\lambda_{\text{GB}}).$$

After obtaining these peaks, MACS does a final step to report the final peaks from the candidates. In addition to the global testing by using $\text{Poi}(\lambda_{\text{GB}})$, MACS does local checking for each peak by using Poisson with rate $\lambda_{\text{local}}$, $\text{Poi}(\lambda_{\text{local}})$, where $\lambda_{\text{local}}$ is estimated as follows.

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

where $\lambda_{\text{1k}}$, $\lambda_{\text{5k}}$, and $\lambda_{\text{10k}}$ are rates estimated from 1 kp, 5 kp and 10 kp windows centred at the location of the peak. By using $\lambda_{\text{local}}$, another p-value is calculated for each candidate peak, say $P_{local}$. The calculation of this p-value is done in the way of calculating $P_{\text{GB}}$ above, but the random variable $X$ follows Poisson with rate parameter $\lambda_{\text{local}}$. That is, $P_{\text{local}}$ is computed as

$$P_{\text{local}} = \Pr(X > \lambda_{\text{peak}}), \quad X \sim \text{Poi}(\lambda_{\text{local}}).$$

Finally, peaks with p-values $P_{\text{local}}$ under the threshold by using $\lambda_{\text{local}}$ are reported by MACS. Note that the identification of the peaks is done for the test and the control samples separately with the input sample.

### 4.2.2 MAnorm

MAnorm is a peak-based method. It uses output of a peak calling method, for example MACS. It is based on two assumptions. First, most of the called peaks are common between the two samples. Second, most of the common peaks have the same densities. These assumptions are practically known, but there is no evidence to say they are true in any ChIP-Seq experiment. These assumptions are employed in the normalisation step of MAnorm, which is the first step in the method where it is constructed into two steps. The second step is the significance checking.

In the normalisation step, MAnorm normalises the given peaks by using only the regions of the common peaks. It also employs MA-plot's idea to normalise the peaks' regions. MA-plot was proposed to visualise and normalise gene expression data in cDNA microarray experiments [11]. MA-plot is a two-dimensional plot, with $M$ components in vertical axis and $A$ components in horizontal axis. These two components, $M$ and $A$, can be defined as follows. Let $x_j$ and $y_j$ be the number of reads of the $j$-th pair of common peaks under two different biological conditions, where $j = 1, \ldots, k$ and $k$ is the number of pairs of common peaks. Then, $M_j$ is $\log_2$ ratio of read densities and $A_j$ is the average of the intensity of the densities of the two peaks in $\log_2$ transformation. Mathematically, $M_j$ and $A_j$ can be represented as

$$\begin{aligned} M_j &= \log_2\left(\frac{x_j}{y_j}\right), \\ A_j &= \frac{1}{2} \times \log_2\left(x_j \times y_j\right). \end{aligned} \tag{4.1}$$

After defining $M$ and $A$, which are vectors of length $k$, a robust linear regression model is fitted to model the dependence between $M$ and $A$ values as follows.

$$M = a + b \times A,$$

where the parameters are estimated by using iterative re-weighted least square with Bisquare weighting function [36]. By obtaining the estimates of $a$ and $b$ by using the common peaks, read counts of regions of all called peaks are normalised as follows:

$$\begin{aligned} \log_2\left(x^*\right) &= \log_2\left(x\right) \quad \Rightarrow \quad x^* = x \\ \log_2\left(y^*\right) &= \frac{2\times a}{2-b} + (2+b) \times \frac{\log_2(y)}{2-b} \quad \Rightarrow \quad y^* = 2^{\frac{2\times a}{2-b} + (2+b)\times \frac{\log_2(y)}{2-b}}, \end{aligned} \tag{4.2}$$

where $x^*$ and $y^*$ are the normalised numbers of reads.

The second step in MAnorm is the significance checking. MAnorm does not assume any statistical model to declare the significant differential peaks, instead it calculates p-value numerically for each of the peaks' regions by using a conditional probability as follows [6].

$$\Pr\left(y^* | x^*\right) = \frac{(x^* + y^*)!}{x^*! y^*! 2^{x^*+y^*+1}}. \tag{4.3}$$

The above is driven by using Bayes' theorem. That is, let $x^*$ and $y^*$ be the observed and normalised number of reads within a pair of peaks from test and control samples, where $X^*$ and $Y^*$ follow the same distribution, which is Poisson distribution with rate parameter $\lambda$ where $\lambda > 0$, i.e, $X^* \sim \text{Poi}(\lambda)$ and $Y^* \sim \text{Poi}(\lambda)$. So the probability of observing $y^*$ given the rate $\lambda$ is

$$P(Y^* = y^*|\lambda) = e^{-\lambda}\frac{\lambda^{y^*}}{y^*!}.$$

Audic and Claverie, [6], defined a conditional probability of observing number of reads $y^*$ given number of observed reads $x^*$ in the other peak where $x^*$ is treated as the rate parameter $\lambda$, i.e, $P(Y^* = y^*|\lambda = x^*)$. They approximate the probability of observing $y^*$ number of reads by using $x^*$ as the maximum likelihood estimate for $\lambda$. By taking into account the fact that $\lambda$ is unknown, an integration over all possible values of $\lambda$ needs to be considered as follows.

$$P(Y^* = y^*|\lambda = x^*) = \int_0^\infty P(\lambda|x^*)P(y^*|\lambda)d\lambda.$$

Through some approximations and by using Bayes' theorem, [6] arrived to Equation (4.3). Note that as the samples are normalised, $P(Y^* = y^*|\lambda = x^*) = P(X^* = x^*|\lambda = y^*)$ is hold.

### 4.2.3 diffReps

diffReps is a statistical method that is independent from peak-calling methods. It applies a direct testing between two given count samples. It can also be used when there are biological replicates. diffReps provides two methods of testing difference between two conditions based on the presence and the absence of the biological replicates. If the experiment has biological replicates, diffReps applies an *exact negative binomial test*. If there are no replicates in the experiment, two tests are provided, *Chi-squared test* and *G test*. The project data RUNX1/ETO does not have replicates, hence an exact negative

binomial test is not of interest here. Chi-squared test and G test give similar results, but G test is preferable [58]. The advantage of G test is that it is less restricted on the expected value compared to Chi-squared. Thus, we decided to consider in detail diffReps with G test only.

Given two ChIP-Seq samples, window size and shift size, diffReps simply slides a window with the given size by the given shift size across the genome on both samples to detect differential regions. In detail, the working process of diffReps consists of three steps: First, filtering the data. Second, normalising the data. Third, differential testing.

In the data-filtering step, diffReps' authors claim that most genomic regions do not show significant enrichment. Hence, diffReps suggests a threshold procedure for the read counts where any window with read counts less than that threshold is filtered out. The threshold is constructed of three components, two statistical measures calculated from the data and a user-given statistic. From the data, diffReps calculates right trimmed mean, $\mu$, and median absolute deviation, $k$. To calculate these two statistics, diffReps samples $100,000$ non-overlapping windows from the given samples, and then calculates the statistics. The user's given statistic is a tunable integer, $t$, which in diffReps' default setting is either $2$ or $20$ for broad and sharp peaks, respectively. Hence, providing the above three statistics, $\mu$, $k$, and $t$, and by using sliding window procedure, diffReps filters any window that contains read counts less than the threshold,

$$\mu + t \times k.$$

In the normalisation step, diffReps proposes a twostep process to normalise the read counts for those windows that pass the threshold. First, calculate a numeric normalising factor for each of the ChIP-Seq samples. Second, consider the median of each factor to normalise its sample. That is, let $x_j$ and $y_j$ be read counts in window $j$ from two ChIP-Seq

data $x$ and $y$. A normalising factor for sample $x$, $f_{xj}$, is calculated as,

$$\begin{aligned} f_{xj} &= \frac{x_j}{\text{geometric-mean}(x_j, y_j)} \\ &= \frac{x_j}{\sqrt{x_j y_j}} \\ &= \sqrt{\frac{x_j}{y_j}}. \end{aligned}$$

After calculating $f_{xj}$ for all windows $j$, the median of $f_{x.}$ is used to normalise sample $x$. Similarly, the same process is done for sample $y$ and median($f_{y.}$) is used to normalise the sample.

The third step in diffReps is the testing. G test is applied to the normalised windows by using the sliding window procedure. In G test, the null assumption is the difference between the windows' density in both samples. That is, given total number of read counts in two windows from the two samples, it is expected to have equal an number of read counts in each window. Mathematically, let $n_j$ be the total read counts in window $j$ from both samples $x$ and $y$. Then, under the null of G test, the following is true.

$$E_j = E_{xj} = E_{yj} = \frac{1}{2} \times n_j,$$

where $E_{xj}$ and $E_{yj}$ are the expected number of read counts in window $j$ from $x$ and $y$ samples, respectively. For a window $j$, G test has a test statistic $G_j$ as,

$$G_j = 2 \times \left\{ O_{xj} \ln\left(\frac{O_{xj}}{E_j}\right) + O_{yj} \ln\left(\frac{O_{yj}}{E_j}\right) \right\},$$

where $O_{xj}$ and $O_{yj}$ are the observed read counts in window $j$ from $x$ and $y$ samples, respectively. Under the null, $G_j$ follows $\chi^2$ with a single degree of freedom. Hence, rejecting the null hypothesis means that read counts in window $j$ show significant difference, and diffReps reports it.

## 4.3 Simulation study

In this section, we evaluate the performance of MACS and MAnorm jointly, and diffReps. The evaluation is made around two points, controlling the false-positive rate and the power

(sensitivity) of the methods. To do that, we used simulated data and real ChIP-Seq data from the Encyclopedia of DNA Elements project (ENCODE).

### 4.3.1 Simulated data

In the simulation study, we want to simulate data that behave like real ChIP-Seq data. In ChIP-Seq data, it is known most of the genome's regions have low levels of read counts and a few regions have quite high levels of read counts. For instance, in RUNX1/ETO data most of the windows have a rate of read counts of 1 read per window, approximately, and a few windows have rates larger than $10$ reads per window. It was also mentioned that the read counts can be modelled with a Poisson model. Moreover, the optimal window for RUNX1/ETO is $200$ bp and we seek simulated data with the same structure, hence the same window size is used in the simulation.

To determine whether or not the methods control the false-positive rate at a given nominal level, $e.g.$, $5\%$, we constructed a simulation study as follows:

1. Simulate two samples each of $10,000$ windows (samples) of size $200$ bp from Poi($\lambda$). $\lambda$ ranges from 0.01 to 0.5 with step 0.02. The simulation is made for each value in $\lambda$. By simulating in this setting, we simulate from the null hypothesis, which assumes no difference between the two samples.

2. Perform the testing methods, and calculate the proportion of significant windows out of $10,000$ for each testing method, where any window with p-value $< 0.05$ is considered significant.

3. Repeat the above three steps $50$ times, and consider the average of them as the final results of false-positive rate for each of the used methods.

To evaluate the power of the methods, we constructed a simulation study as follows:

1. Simulate two samples each of $10,000$ windows of size $200$ bp from Poi$(\lambda_1 = 0.01)$. $\lambda_1$ here represents the low rate read counts, which is most of the genome's regions.

2. Simulate $2000$ windows out of the $10,000$ in both samples at the same locations with Poi$(\lambda_2 = 0.2)$

3. Simulate another $1000$ windows at the same location in both samples by using Poisson with fixed rate $\lambda_2 = 0.02$ and rate $\lambda_3$ ranges from $0.22$ to $1$ by step $0.02$. To keep the difference existing after the normalising process of the methods, in the first sample we simulate the first $500$ windows by using $\lambda_2$ and the second $500$ windows by using $\lambda_3$, and in the second sample the first $500$ from $\lambda_3$ and the second $500$ from $\lambda_2$. Note that the choice of $\lambda_2 = 0.2$ will be discussed in the comparison chapter, Chapter 6.

4. Perform the testing methods, and calculate the proportion of detected significant windows out of $1000$ windows for each of the testing methods.

5. Repeat the above steps $50$ times and consider the average of them as the final results of the power for each of the used methods.

**MACS and MAnorm**

As MAnorm is a peak- based method, we used MACS and evaluated the performance of MACS and MAnorm jointly. MAnorm can be evaluated by itself, if we give it the exact regions that we simulate as peaks. Although MAnorm is a peak-based method, we did give MAnorm exact windows to evaluate its individual performance. That is, for false-positive determination we used MACS and MAnorm jointly, as well as using MAnorm only by giving it the exact window's starting and ending points, so MAnorm would do the significance test for each of the windows. Similarly for the power evaluation, we used the methods jointly in one run, and MAnorm only by giving it the 1000 windows with

different rates as the peak's regions. The results are shown in Figure 4.1 Note, MACS was used in its default setting.

From Figure 4.1 it can be seen that MACS and MAnorm jointly do not show good control of the false-positive rate, especially if $\lambda$ is very low. By investigating the results we found that MACS cannot detect the peaks when $\lambda$ is quite low, and then it does not report any peak to MAnorm. From the figure, we can see the false-positive rate of MACS and MAnorm starts to recover around $\lambda = 0.1$. On the other hand, by considering MAnorm only it can be seen that the behaviour of the false-positive rate is opposite to the behaviour of MACS and MAnorm jointly when $\lambda$ is low. Moreover, we can see that MACS and MAnorm jointly and MAnorm only show close behaviour when $\lambda > 0.2$.

In the power plot, Figure 4.1, it can be seen that MACS and MAnorm jointly do start detecting significant results from differences of around 0.1, whereas MAnorm only starts much earlier with differences of around 0.02. In fact, from the results of the false-positive rate, it is expected from MAnorm only to show higher power compared to MACS and MAnorm jointly when the rate is low. But in the power simulation we set all the rates to be larger than or equal to 0.2, and at that level the methods can be considered comparable.

**diffReps**

diffReps is also evaluated in terms of controlling the false-positive rate and power. In the diffReps experiment, we used a window size of 200 bp, and the rest of the options as in the default setting. Note that in diffReps the shift process has to be made, and the shift size has to be a fraction of the window size; by default the shift size is 100 bp. Hence, we applied diffReps with its G test and the result is shown in Figure 4.2.

In Figure 4.2 the false-positive rate plot, it can be seen that diffReps shows A quite high false-positive rate for all used values of rate $\lambda$, and the highest associated with low $\lambda$'s. It is assumed that diffReps controls the false positive at 5% level, but the actual level which

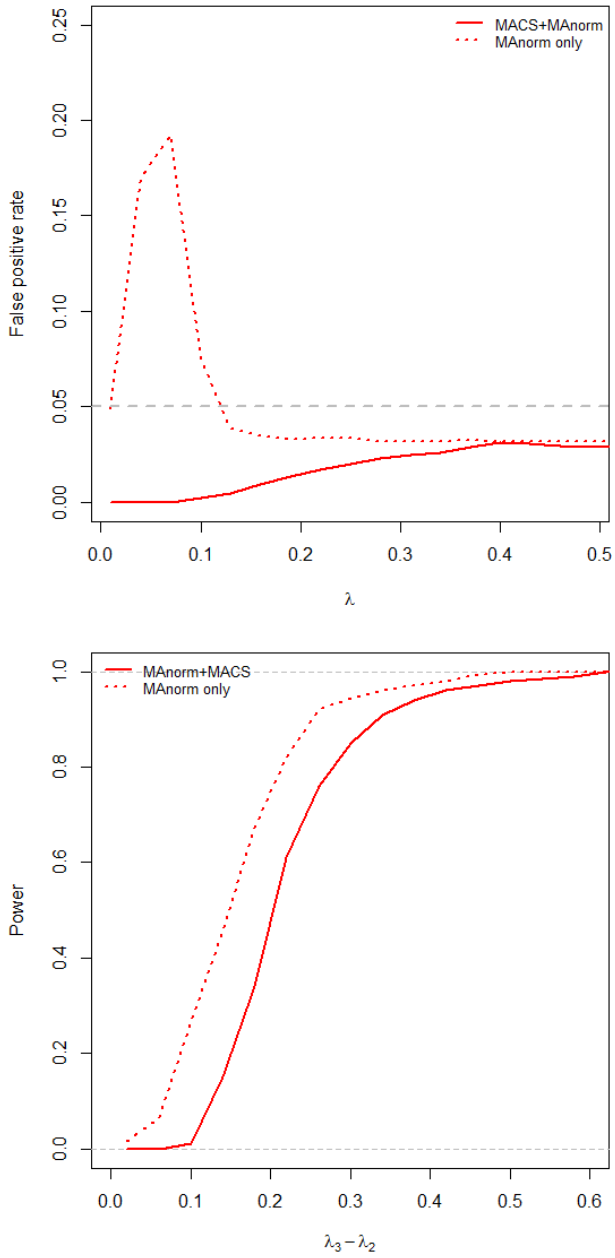



Figure 4.1: Top panel shows the determination of the false-positive rate by using MACS and MAnorm jointly, and MAnorm only based on simulated data. In that panel, the horizontal axis represents the $\lambda$ used in the simulation, and the vertical axis represents the average of 50 false-positive rates for each of the $\lambda$'s. The lower panel shows the power of MACS and MAnorm jointly, and MAnorm only based on simulated data. In that panel, the horizontal axis represents the difference in rates between the two simulated data, where we enrich the windows with different rates, and the vertical axis represents the average power of 50 simulations for each difference point $\lambda_3 - \lambda_2$.

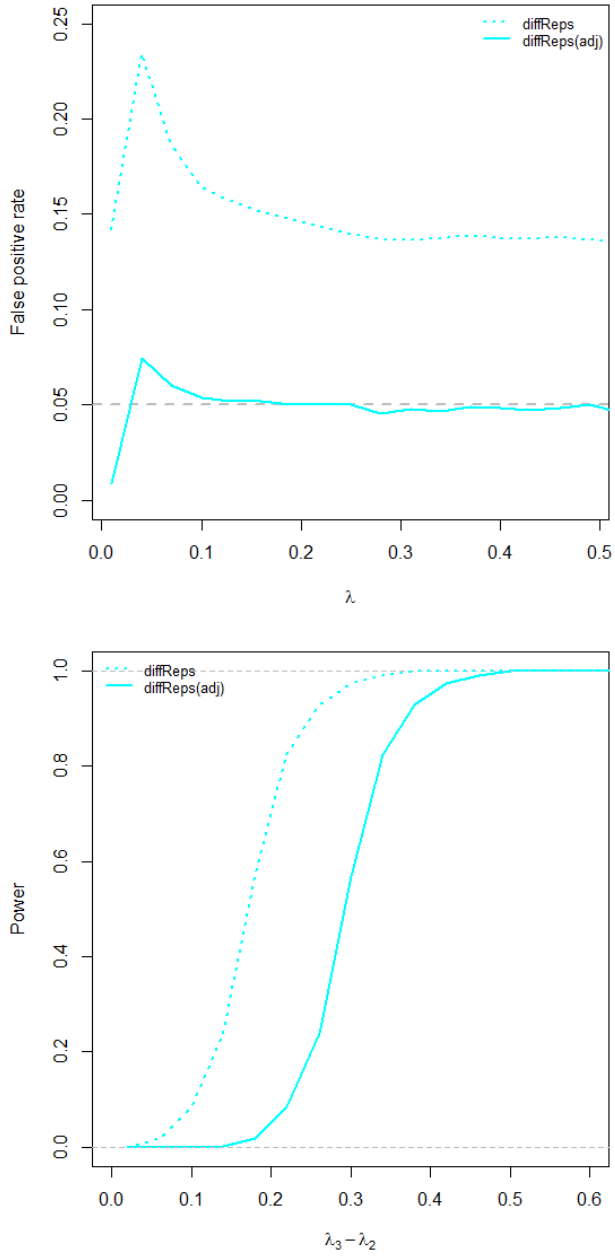



Figure 4.2: Top panel shows the determination of the false-positive rate of diffReps based on the simulated data. In that panel, the horizontal axis represents the $\lambda$ used in the simulation and the vertical axis represents the average of 50 false-positive rates for each of the $\lambda$s. The lower panel shows the power of diffReps based on simulated data. In that panel, the horizontal axis represents the difference in rates between the two simulated datasets, where we enrich the windows with different rates, and the vertical axis represents the average power of 50 simulations for each difference point $\lambda_3 - \lambda_2$.

can be seen from the plot is 0.15 for $\lambda \geq 0.2$. As we deal with simulated data and we know the underlying assumption, hence we can adjust the p-values of diffReps so it can meet the assumed level of false-positive rate. By running diffReps on the simulated data with $\lambda = 0.2$ and at many levels of significance lower than 0.05, we found that by assuming the significance level at 0.012, it is actually controlled at 0.05. Hence, p-values of diffReps from the simulated data can be adjusted as follows. Let $p$ and $p^*$ be the observed and adjusted p-values, respectively. The adjusted p-values can be calculated as:

$$p^* = a \times p^b, \tag{4.4}$$

where $a$ and $b$ are parameters to be calculated by solving the following two equations.

$$\begin{aligned} 0.15 &= a \times (0.05)^b, \\ 0.05 &= a \times (0.012)^b. \end{aligned}$$

After few steps of algebra, we found $a = 1.51$ and $b = 0.78$. Hence, these two parameters are used to adjust the resulting p-values from diffReps in the simulation study by using Equation (4.4). By performing this adjustment, the false-positive rate is controlled at the assumed level, as shown in Figure 4.2.

From the power plot, Figure 4.2, we can see that diffReps starts to detect the differences quite early by using its raw p-values. However, we saw that the raw setting returns a high false-positive rate. Hence, an adjustment procedure is needed here as well. As we fixed $\lambda_2$ to be 0.2 in the simulation, we can used the same adjusting values $a$ and $b$ to adjust the p-values in the power simulation. After the adjustment, it can be seen that the power curve shifted to the right. Hence, diffReps starts to detect significant results around a difference of 0.2, as shown in Figure 4.2.

Note that the adjustment parameters $a$ and $b$ would be different from one dataset to another. In the simulation, we used simulated data with the same rate $\lambda$ to adjust the p-values. In the case where we have real data, the adjustment would be made if biological replicates were provided (see Section 4.3.2). That is, in real data, we expect replicates to have the same rate, and hence the adjustment can be made.

### 4.3.2 ENCODE data

We used simulated data to evaluate the performance of the testing methods. Although we set the simulation in a way that is close to the observed structure of ChIP-Seq data, real ChIP-Seq datasets still have their special biological structures. Hence, the testing methods need to be applied and evaluated on real ChIP-Seq data. From the ENCODE project, we looked for ChIP-Seq data from knock-out experiments. We downloaded the following three samples:

1. Cell line Gm12878 with transcription factor ATF2;
2. Cell line Gm12878 with transcription factor BCLAF1;
3. Cell line H1-hESC with transcription factor ATF2.

Note that for each of these three samples, there are two biological replicates. By checking the average read counts in the samples, we found it to be around 0.02 bp per genomic position. From the simulated data in the previous section, we know that none of the testing methods behave well at low rate level. Hence, in order to increase the rate, we combined the first replicates of the three samples together and the second three replicates together. After that, we also divided the combined samples into chromosomes, and each chromosome was divided into two halves before and after the centromere region.

To perform the false-positive evaluation, we used the replicates. Ideally, biological replicates of the same experiment would not show significant differences. Hence, the difference testing is performed between the replicates of the same experiment, and the detected significant differences can be considered as false positives. We used the first half of chromosomes 1 to 10 to do the false-positive evaluation, therefore 10 values of false-positive rate for each testing method.

The power evaluation in the simulated data was easy as we can generate differences between samples and we know where they are located. However, in real data it is not

a straightforward process. We want to keep the samples with their natural variabilities. Hence, we adapted a procedure that guarantees no change to the distribution of read counts as follows. First, to consider only one sample with window size 200 bp. Second, the second sample to compare with is the first sample, but by shifting one window at a time. That is, if $m$ is the total number of windows, then windows $i = 1, 2, \ldots, m$ in the first sample are matched to windows $i = 2, 3, \ldots, m, 1$ in the second sample. After each shift, the testing methods are applied. In addition, after each shift, we record the observed difference in estimated rates and hence we can plot the power as a function of the observed difference in rates. That is, the power of a difference $k$ is the number of significant windows with difference $k$ over the total number of observed difference $k$ across the windows. To do this, we used the first half of chromosome 10.

**MACS and MAnorm**

MACS and MAnorm jointly and MAnorm only are applied to the ENCODE data. MACS is used in its default settings in both false-positive and power evaluations. For MAnorm only, regions of windows are given as peaks in both types of evaluations. The result is shown in Figure 4.3.

The results in Figure 4.3 show that MACS and MAnorm jointly and MAnorm only both exhibit high false-positive rates. To control the false-positive rate at the nominal level, an adjustment is needed. That is, we adjust the p-values that result from MAnorm by using Equation (4.4). After adjusting the p-values, we can see both testing methods controlling the false-positive rate at $5\%$, as shown in the same plot.

The power results, Figure 4.3, show that MACS and MAnorm are jointly start to detect a significant difference of around 0.4 difference in rates. In addition, their power shows a slow increase until 0.8 difference, then it starts to increase sharply. On the other hand, MAnorm only appears quite powerful when all windows are reported to it. It can be

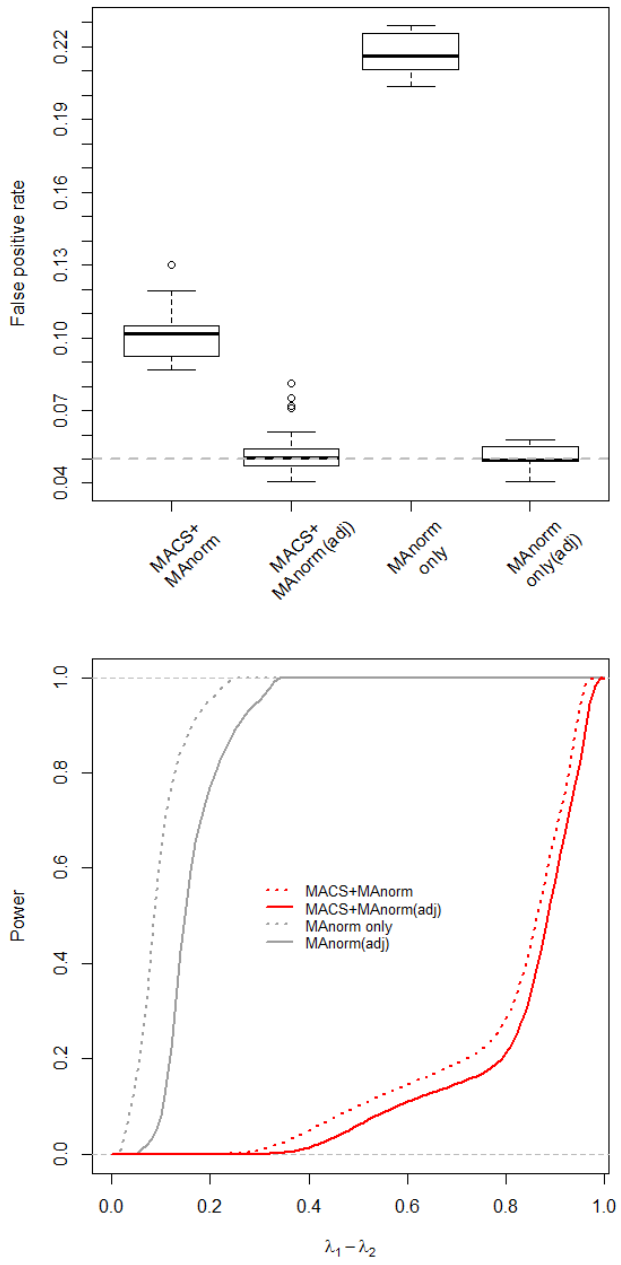



Figure 4.3: The top panel shows the determination of the false-positive rate by using MACS and MAnorm jointly, and MAnorm only with their adjustments based on real ENCODE data. In that panel, the horizontal axis represents the used methods and the vertical axis is false-positive rates for each of the methods. Each point in the plot represents the false-positive rate by using a subsample that is a half of a chromosome. The lower panel shows the power of MACS and MAnorm jointly, and MAnorm only and their adjustments based on real ENCODE data. In that panel, the horizontal axis represents the observed difference in rates between the two samples, and the vertical axis represents the average power by using ENCODE data. More details are provided in the text.

seen after adjustment that MAnorm only starts to detect a significant result around a 0.05 difference in rates, and dramatically reaches its full power around a 0.35 difference in rates, which is even before MACS and MAnorm jointly start their power curve.

**diffReps**

We also evaluated the controlling of the false-positive rate of diffReps by using the real data and the results are shown in Figure 4.4. From the figure it can be seen that diffReps shows quite good control of the false-positive rate compared to the nominal level, which is 5%.

Unfortunately, we are not able to evaluate the power of diffReps because the settings of the evaluation do not work with the working process of diffReps. That is, we cannot compare or record the observed difference in estimated rates with difference in rates of the significant regions after applying the method. This is because of the shifting process made by diffReps.

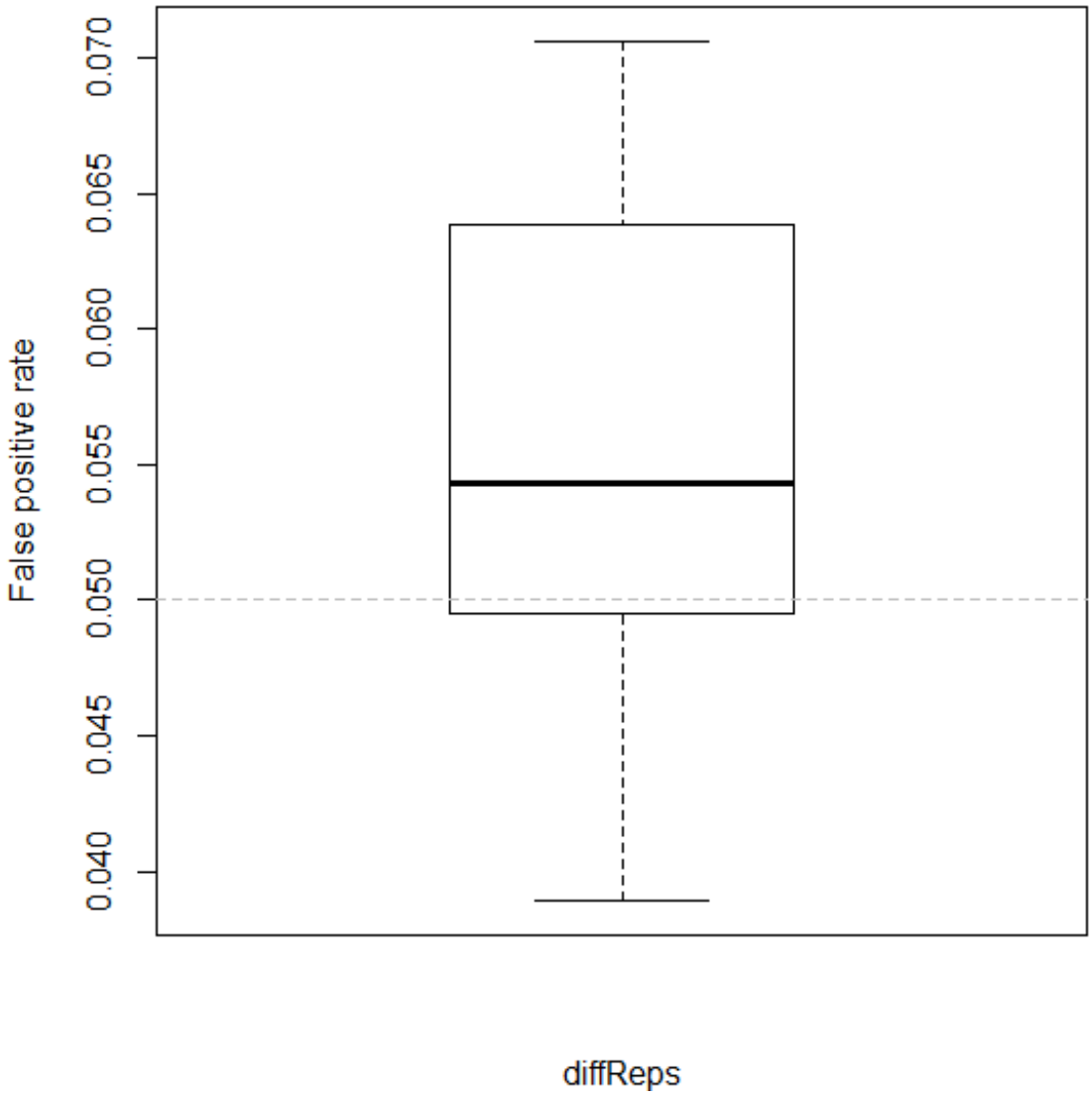



Figure 4.4: The determination of the false-positive rate by using diffReps based on real ENCODE data. In the plot, the horizontal axis represents the used methods and the vertical axis shows the false-positive rates. Each point in the plot represents the false-positive rate by using a subsample that is a half of a chromosome.

## 4.4 Project data RUNX1/ETO

### 4.4.1 MACS and MAnorm

We applied MACS to the project data. In RUNX1/ETO data, we have three data files for three samples: test, control and input. We used the test and control data separately, and used the input data as control for both samples. It was shown that the optimal window size for RUNX1/ETO data is 200 bp. However, as mentioned, MACS sets a lower bound for the given window size with a minimum of 300 bp (which is the default). Given that minimum, the optimal window size of the data cannot be used in MACS. Thus, we used MACS in its default settings, and the results follow.

By comparing the control to the input, MACS called 3790 peaks. In addition, by comparing the test to the input, MACS called 8243 peaks. The intersection between the test and control peaks is 3671 peaks. That means there are 4572 peaks that are uniquely called in the test, whereas only 119 peaks are uniquely called in the control. Figure 4.5 shows a Venn diagram of the peak numbers and the intersection.

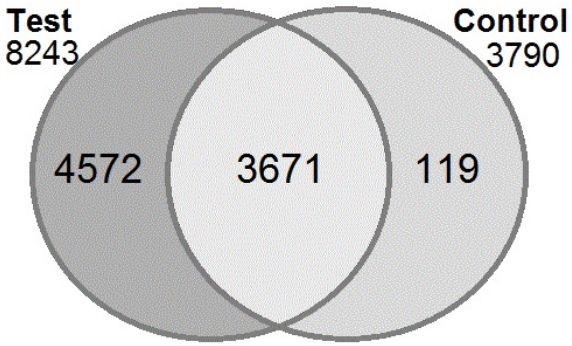



Figure 4.5: Peaks of test and control samples by using MACS in the default settings. The test and control were compared to the input sample separately, then we verified the common and unique peaks.

The widths of the called peaks in test range from 287 bp to 19140 bp with average width

and read counts of 940 bp and 115 bp, respectively. In the control sample, the peaks range from 60 bp to 7787 bp in terms of the width with a mean of 438 bp and average read counts of 95 bp per peak. Note that MACS estimates the parameter $d$ as 43 and 148 in the control and test samples, respectively.

The called peaks are given to MAnorm for detecting significant differences. In MAnorm, the 3671 common peaks are used to normalise the peak's regions. MAnorm declares that there are 509 regions showing significant difference.

## 4.5 Discussion

In this chapter, we present some of the current methods for differential binding site analysis. The common path to be followed in this analysis is based on peak-calling methods, as prior step to the analysis. Some proposed analysis methods do perform the peak-calling step. For instance, ChIPDiff [32] and ODIN [75]. Other proposed analysis methods, which represent the majority, depend on any peak-calling methods. This dependency makes, first, evaluating the performance of the analysis methods not possible. Second, many of the proposed peak-calling methods suffer from subjectivity and ambiguity of underlying assumptions. Therefore, different peak callers can give different results for the same data. To remove some of this confusion, PFMS [13] was proposed. This method combines several peak-calling methods and reports the consensus peaks only. Moreover, many of the used analysis methods suffer from poor control of the false-positive rate [59].

We used the most popular peak calling method of MACS. MACS has shown good performance compared to many other peak-calling methods [68]. We used MAnorm, which is an analysis method, jointly with MACS to analyse the called peaks. In MAnorm, it is claimed that the p-value in Equation (4.3) is numerically calculated and there is no assumption made for the variables in it. However, we can re-write it as

$\Pr(y|x) = 1/2 \times \Pr(Y = y)$ where $Y \sim \text{Bin}(n = x + y, p = 1/2)$. Hence, it is a scaled point probability, while it should be a tail probability or equivalent. Moreover, it was mentioned by MAnorm's authors that it does not behave well when the observed rate of read counts is low in most of the peaks, and we observed this in our simulated data. In addition, it requires a substantial number of common peaks to work well as it does the normalisation based on the common peaks.

diffReps offers two tests when there are no biological replicates, Chi-square and G test, where it recommends the latter. It was mentioned by its authors that these two tests can exhibit high false-positive rates. That is true and was observed in the simulation study. Because of this high rate of false positives, diffReps might not be a reliable analysis method.

We used an adjustment formula to adjust the p-values of the methods that do not control the false positive at the targeted level. However, we expect that when the rate parameter is large, then the values of the parameters $a$ and $b$ will be close to those in the formula. That is, when the rate is large we expect any method to behave well in terms of controlling the false positives.

In RUNX1/ETO data, MACS and MAnorm jointly detect 509 regions showing significant differences. Although diffReps shows unreliable performance, we used it here as well and it reports 80,163 regions with significant differences. 476 regions from MACS and MAnorm are intersected with the diffReps' results. That huge number by diffReps does not mean that it is powerful, but it is a result of the high rate of false positive error. Note that this result is obtained after performing FDR correction [8] based on the full human genome. That is, the human genome is $3 \times 10^9$ bp, and by using a window size of 200 bp, the number of windows is $(3 \times 10^9)/200$, and this is the number to be considered in the correction process.

Currently available methods for differential binding sites are not efficient. They suffer from subjectivity, poor controlling for the false positive rate and unclarity of the

underlying assumptions. The need exists for objective methods with a clear underlying model that enables control of the false-positive rate.

# Chapter 5

# Parametric statistical methods for differential binding site studies

## 5.1 Introduction

We showed and discussed in Chapter 4 some of the current proposed methods for differential binding site studies. It was shown that the methods suffer from poor controlling for false- positive error and unclear underlying models in some of them. In this chapter, we address these issues by first defining a clear assumptions and underlying model for the ChIP-Seq data, which is the read counts. We also employ a hypothesis test with three parametric tests to infer significant differential binding sites. Moreover, we evaluate the behaviour of the proposed tests in terms of false-positive rate and power. The proposed tests show good control of the false-positive error, and a good power as well.

In the rest of this chapter, we introduce parametric statistical tests that can be used in differential binding site studies. We first introduce the underlying model, which is Poisson Difference (PD), Section 5.2, including parameter estimation and the challenges. We then propose three statistical tests to be used based on the underlying model in Section 5.3.

These tests are *Exact*, *Wald* and *Likelihood Ratio Tests*. We evaluate the proposed tests by using simulated and real ENCODE data, Section 5.4. Fourth, we apply the tests on the project data RUNX1/ETO, Section 5.5. Finally, we discuss and conclude the findings, 5.6.

## 5.2 Poisson Difference

It was mentioned, in Chapter 3, that the read counts of ChIP-Seq data can be modelled as a Poisson random variable. Clearly, not the whole genome (full sample) is modelled as Poisson with a single rate parameter, but read counts within windows can be modelled as Poisson where each of the windows is modelled separately. Modelling each window separately allows the rate of the read counts to vary between windows across the genome. Note that all windows across the genome are not overlapping and have the same width (size).

Many of the previously developed methods for detecting differential binding sites assume a Negative Binomial (NB) model for the read counts, for example diffReps [14] and DESeq [5]. NB is able to deal with the changes in the variability of the read counts. However, it was shown that such methods have high false-positive rates [59]. In RUNX1/ETO we observed that most of the windows have very low read counts (or none at all). This observation might indicate the equality of the mean and the variance within the windows. As a result, the Poisson model with a single rate parameter can be an acceptable model for the read counts in this case. We fitted the Poisson model in many of them and found that the model fitted. However, in a few windows where a large number of read counts exist (enriched), we observed large changes in the variability of the read counts. This observation might violate the equality of the mean and the variance within windows, and hence the Poisson model would not be preferable in this case. In such windows we tried to fit Poisson with a single rate parameter, but it did not fit. Given the above about

NB and the observation in RUNX1/ETO data, we decided to start by assuming the Poisson model with a single rate parameter for the read counts within windows. We want to see how the Poisson model does compared to other methods under different assumptions, for example NB.

All previous proposed methods for differential binding site studies consider read counts in different samples as independent. However, that is not true in such studies. That is, a read count is simply the number of times a specific position is observed, and that position must be compared with the same position in other samples, but not with any other position. For instance, the read count at position 100 in sample one is compared to the read count at position 100 in sample two. This means that the read counts in the two samples are paired. Hence, assuming independence between the two samples ignores the location relationship between them.

As explained above, the read counts in two samples are paired, hence studying differential binding sites can be achieved by studying the difference in read counts between the two samples. Again, the analysis is done for the difference within windows, but not the whole genome. The read counts in a window are modelled as Poisson with a single rate parameter. The difference in read counts in a window between two samples is then the difference between two Poisson variables with different rate parameters. The difference between two Poisson variables can be described as the Poisson Difference (PD) random variable [55].

Let $x_i$ and $y_i$ be vectors of the observed read counts in the $i$-th window from the test and control samples, respectively, where $x_i = (x_1, \ldots, x_n)$ and $y_i = (y_1, \ldots, y_n)$, and $n$ is the window size (which is fixed for all windows). Let $z_i$ be the difference in read counts in the $i$-th window where $z_i = x_i - y_i$, i.e, $z_i = z_1, \ldots, z_n$. If $X_i \sim \text{Poi}(\lambda_{xi})$ and $Y_i \sim \text{Poi}(\lambda_{yi})$, then $Z_i = X_i - Y_i$ would follow the PD distribution with two rate parameters $\lambda_{xi}$ and $\lambda_{yi}$, and is denoted as $Z_i \sim PD(\lambda_{xi}, \lambda_{yi})$.

A PD random variable $Z_i$, where $Z_i \sim PD(\lambda_{xi}, \lambda_{yi})$, is a discrete random variable

(integers) that ranges from $-\infty$ to $\infty$. Furthermore , it has probability mass function as follows:

$$P(Z_i = z) = e^{-(\lambda_{xi}+\lambda_{yi})}\left(\frac{\lambda_{xi}}{\lambda_{yi}}\right)^{\frac{z}{2}} I_z\left(2\sqrt{\lambda_{xi}\lambda_{yi}}\right), \quad z \in \{\ldots,-2,-1,0,1,2\ldots\}, \quad \lambda_{xi},\lambda_{yi} > 0, \tag{5.1}$$

where $z$ is an integer, and

$$I_z(a) = \left(\frac{a}{2}\right)^z \sum_{k=0}^{\infty} \frac{\left(\frac{a^2}{4}\right)^k}{k!(z+k)!},$$

is a modified Bessel function of order $z$ and of the first type, where $a > 0$. Assuming independence, the random variable $Z_i$ has expectation and variance as:

$$\begin{aligned} E(Z_i) &= \lambda_{xi} - \lambda_{yi}, \\ V(Z_i) &= \lambda_{xi} + \lambda_{yi}. \end{aligned} \tag{5.2}$$

### 5.2.1 Parameter estimation

To work with any parametric model, it is essential to know how to obtain the estimates of the model's parameters, so it can be applied in real problems where the true parameters' values are unknown. As shown above in Equation (5.1), PD has two rate parameters, $\lambda_{xi}$ and $\lambda_{yi}$. Two types of estimations can be obtained for these parameters, which are maximum likelihood and moment estimates. We consider the maximum likelihood estimates (mle ), as the moment estimates do not exist under certain conditions (see the discussion, Section 5.6).

The mle of $\lambda_{xi}$ and $\lambda_{yi}$ are obtained by using the likelihood function $L(\lambda_{xi}, \lambda_{yi}; z_i)$ with the observed difference in read counts $z_i$. Note that each window $i$ has its PD model with its own parameters. Hence each window $i$ will have its mle of $\lambda_{xi}$ and $\lambda_{yi}$ based on its observed differences $z_i$. $L(\lambda_{xi}, \lambda_{yi}; z_i)$ can be obtained by using Equation (5.1) as follows:

$$L\left(\lambda_{xi}, \lambda_{yi} | z_i\right) = \prod_{j=1}^{n} e^{-(\lambda_{xi}+\lambda_{yi})} \left(\frac{\lambda_{xi}}{\lambda_{yi}}\right)^{\frac{z_{ij}}{2}} I_{z_{ij}}\left(2\sqrt{\lambda_{xi}\lambda_{yi}}\right), \tag{5.3}$$

where $n$ is the window size, and it is fixed for all windows.

For simplicity, in all upcoming equations in this section, Section 5.2.1, we will remove the window index $i$ from the parameters $\lambda_{xi}$ and $\lambda_{yi}$ and the difference data $z_i$. However, it is important to keep in mind that the estimation process is made individually for each of the windows across the genome. Hence, the likelihood function in Equation (5.3) can be rewritten as:

$$L\left(\lambda_x, \lambda_y; z\right) = \prod_{j=1}^{n} e^{-(\lambda_x+\lambda_y)} \left(\frac{\lambda_x}{\lambda_y}\right)^{\frac{z_j}{2}} I_{z_j}\left(2\sqrt{\lambda_x\lambda_y}\right). \tag{5.4}$$

To obtain the mle of $\lambda_x$ and $\lambda_y$ from Equation (5.4), the equation needs, first, to be differentiated with respect to (w.r.t) each of the parameters separately. Second, to set the derivatives to zero and solve them to obtain the mle $\hat{\lambda}_x$ and $\hat{\lambda}_y$. To make the differentiation simpler, the log-likelihood function, $\ell\left(\lambda_x, \lambda_y; z\right)$, is considered instead of the likelihood function, Equation (5.4). The log-likelihood function can be shown as follows:

$$\begin{aligned} \ell\left(\lambda_x, \lambda_y; z\right) &= \log\left\{\prod_{j=1}^{n} e^{-(\lambda_x+\lambda_y)} \left(\tfrac{\lambda_x}{\lambda_y}\right)^{\frac{z_j}{2}} I_{z_j}\left(2\sqrt{\lambda_x\lambda_y}\right)\right\} \\ &= -n(\lambda_x+\lambda_y) + \tfrac{\log(\lambda_x)-\log(\lambda_y)}{2} \sum_{j=1}^{n} z_j + \sum_{j=1}^{n} \log\left(I_{z_j}\left(2\sqrt{\lambda_x\lambda_y}\right)\right). \end{aligned} \tag{5.5}$$

Next, Equation (5.5) is differentiated w.r.t $\lambda_x$ and $\lambda_y$, and the derivatives are set to zero. Note that the equation contains a Bessel function, and its derivative can be shown as follows [1]:

$$\frac{\partial I_z\left(a\right)}{\partial a} = \frac{z}{a} I_z\left(a\right) + I_{z+1}\left(a\right). \tag{5.6}$$

Now, Equation (5.5) is differentiated w.r.t $\lambda_x$ as follows:

$$\begin{aligned}
\frac{\partial \ell(\lambda_x,\lambda_y;z)}{\partial \lambda_x} &= -n + \frac{1}{2\lambda_x}\sum_{j=1}^{n} z_j + \sqrt{\frac{\lambda_y}{\lambda_x}}\sum_{j=1}^{n} \frac{\frac{z_j}{2\sqrt{\lambda_x\lambda_y}} I_{z_j}\left(2\sqrt{\lambda_x\lambda_y}\right) + I_{z_j+1}\left(2\sqrt{\lambda_x\lambda_y}\right)}{I_{z_j}\left(2\sqrt{\lambda_x\lambda_y}\right)} \\
&= -n + \frac{1}{2\lambda_x}\sum_{j=1}^{n} z_j + \sqrt{\frac{\lambda_y}{\lambda_x}}\sum_{j=1}^{n} \frac{z_j}{2\sqrt{\lambda_x\lambda_y}} + \frac{I_{z_j+1}\left(2\sqrt{\lambda_x\lambda_y}\right)}{I_{z_j}\left(2\sqrt{\lambda_x\lambda_y}\right)} \\
&= -n + \frac{1}{2\lambda_x}\sum_{j=1}^{n} z_j + \frac{1}{2\lambda_x}\sum_{j=1}^{n} z_j + \sqrt{\frac{\lambda_y}{\lambda_x}}\sum_{j=1}^{n} \frac{I_{z_j+1}\left(2\sqrt{\lambda_x\lambda_y}\right)}{I_{z_j}\left(2\sqrt{\lambda_x\lambda_y}\right)} \\
&= -n + \frac{1}{\lambda_x}\sum_{j=1}^{n} z_j + \sqrt{\frac{\lambda_y}{\lambda_x}}\sum_{j=1}^{n} \frac{I_{z_j+1}\left(2\sqrt{\lambda_x\lambda_y}\right)}{I_{z_j}\left(2\sqrt{\lambda_x\lambda_y}\right)}.
\end{aligned} \tag{5.7}$$

Next is to set the derivative, Equation (5.7), to zero as follows:

$$\begin{aligned}
0 &= -n + \frac{1}{\hat{\lambda}_x}\sum_{j=1}^{n} z_j + \sqrt{\frac{\hat{\lambda}_y}{\hat{\lambda}_x}}\sum_{j=1}^{n} \frac{I_{z_j+1}\left(2\sqrt{\hat{\lambda}_x\hat{\lambda}_y}\right)}{I_{z_j}\left(2\sqrt{\hat{\lambda}_x\hat{\lambda}_y}\right)} \\
0 &= -n\hat{\lambda}_x + \sum_{j=1}^{n} z_j + \sqrt{\hat{\lambda}_x\hat{\lambda}_y}\sum_{j=1}^{n} \frac{I_{z_j+1}\left(2\sqrt{\hat{\lambda}_x\hat{\lambda}_y}\right)}{I_{z_j}\left(2\sqrt{\hat{\lambda}_x\hat{\lambda}_y}\right)}
\end{aligned} \tag{5.8}$$

Similarly, by differentiating $\ell(\lambda_x, \lambda_y; z)$, Equation (5.5), w.r.t $\lambda_y$ and setting the derivative to zero we obtain the following:

$$0 = -n\hat{\lambda}_y + \sqrt{\hat{\lambda}_x\hat{\lambda}_y}\sum_{j=1}^{n} \frac{I_{z_j+1}\left(2\sqrt{\hat{\lambda}_x\hat{\lambda}_y}\right)}{I_{z_j}\left(2\sqrt{\hat{\lambda}_x\hat{\lambda}_y}\right)}. \tag{5.9}$$

From Equations (5.8) and (5.9), it can be seen that solving for $\hat{\lambda}_x$ and $\hat{\lambda}_y$ is not explicit, because of the Bessel term. Note that there is a relationship between $\hat{\lambda}_x$ and $\hat{\lambda}_y$. That is, by subtracting Equation (5.9) from (5.8) we obtain the following:

$$\begin{aligned} 0 &= -n\hat{\lambda}_x + \sum_{j=1}^{n} z_j + n\hat{\lambda}_y \\ \bar{z} &= \hat{\lambda}_x - \hat{\lambda}_y, \end{aligned} \tag{5.10}$$

where $\bar{z}$ is the sample mean of the difference in read counts per position in a window. This relationship, Equation (5.10), can be employed in the estimation process of the parameters. That is, one of the parameters, either $\lambda_x$ or $\lambda_y$, needs to be estimated by using the likelihood function, Equation (5.5), and then Equation (5.10) can be used to calculate the other parameter for a given $\bar{z}$. From Equation (5.10) we have the following:

$$\begin{aligned} \hat{\lambda}_x &= \bar{z} + \hat{\lambda}_y \\ \hat{\lambda}_y &= \hat{\lambda}_x - \bar{z} \end{aligned} \tag{5.11}$$

Although the relationship reduces the number of parameters in the estimation process, solving the derivatives w.r.t the estimates is still analytically not easy . For instance, by substituting $\hat{\lambda}_x = \bar{z} + \hat{\lambda}_y$ into Equation (5.9), we obtain the following:

$$0 = -n\hat{\lambda}_y + \sqrt{\left(\bar{z} + \hat{\lambda}_y\right)\hat{\lambda}_y} \sum_{j=1}^{n} \frac{I_{z_j+1}\left(2\sqrt{\left(\bar{z} + \hat{\lambda}_y\right)\hat{\lambda}_y}\right)}{I_{z_j}\left(2\sqrt{\left(\bar{z} + \hat{\lambda}_y\right)\hat{\lambda}_y}\right)}. \tag{5.12}$$

If $\hat{\lambda}_y$ can be separated in one side in Equation (5.12), then it can be analytically estimated. By writing full Bessel terms in Equation (5.12), we can obtain the following:

$$\begin{aligned} 0 &= -n\hat{\lambda}_y + \sqrt{\frac{\bar{z}+\hat{\lambda}_y}{\hat{\lambda}_y}} \sum_{j=1}^{n} \frac{\left(\sqrt{\left(\bar{z}+\hat{\lambda}_y\right)\hat{\lambda}_y}\right)^{z_j+1} \sum_{k=0}^{\infty} \frac{\left((\bar{z}+\hat{\lambda}_y)\hat{\lambda}_y\right)^k}{k!\left(z_j+1+k\right)!}}{\left(\sqrt{\left(\bar{z}+\hat{\lambda}_y\right)\hat{\lambda}_y}\right)^{z_j} \sum_{k=0}^{\infty} \frac{\left((\bar{z}+\hat{\lambda}_y)\hat{\lambda}_y\right)^k}{k!(z_+k)!}} \\ 0 &= -n\hat{\lambda}_y + \sqrt{\frac{\bar{z}+\hat{\lambda}_y}{\hat{\lambda}_y}} \sum_{j=1}^{n} \left(\sqrt{\left(\bar{z} + \hat{\lambda}_y\right)\hat{\lambda}_y}\right) \frac{\sum_{k=0}^{\infty} \frac{\left((\bar{z}+\hat{\lambda}_y)\hat{\lambda}_y\right)^k}{k!\left(z_j+1+k\right)!}}{\sum_{k=0}^{\infty} \frac{\left((\bar{z}+\hat{\lambda}_y)\hat{\lambda}_y\right)^k}{k!\left(z_j+k\right)!}} \\ 0 &= -n\hat{\lambda}_y + \left(\bar{z} + \hat{\lambda}_y\right) \sum_{j=1}^{n} \frac{\sum_{k=0}^{\infty} \frac{\left((\bar{z}+\hat{\lambda}_y)\hat{\lambda}_y\right)^k}{k!\left(z_j+1+k\right)!}}{\sum_{k=0}^{\infty} \frac{\left((\bar{z}+\hat{\lambda}_y)\hat{\lambda}_y\right)^k}{k!\left(z_j+k\right)!}}. \end{aligned} \tag{5.13}$$

From Equation (5.13) it can be seen that solving for $\hat{\lambda}_y$ analytically is not possible, as $\hat{\lambda}_y$ cannot be separated from $z$ because of the non-linearity of the equation, and it is the same situation for $\hat{\lambda}_x$ in Equation (5.8). In such cases, numerical optimisation methods can be used to obtain the required estimates. There are many developed numerical optimisation methods [40, 52, 17, 33, 9]. For a given function and observations, these optimisation methods search for values of parameters that minimise the value of the function. In our situation, we seek the estimates of the $\lambda_x$ and $\lambda_y$ that maximise the log-likelihood function in Equation (5.5). Hence, in order to use the optimisation methods we instead consider the negative log-likelihood function, which can be shown as follows:

$$-\ell\left(\lambda_x, \lambda_y; z\right) = n(\lambda_x + \lambda_y) - \frac{\log(\lambda_x) - \log(\lambda_y)}{2} \sum_{j=1}^{n} z_j - \sum_{j=1}^{n} \log\left(I_{z_j}\left(2\sqrt{\lambda_x \lambda_y}\right)\right). \tag{5.14}$$

By using the observed difference $z$ in a window, $\lambda_x$ and $\lambda_y$ can be optimised numerically for that window by using the negative $log$ likelihood function, Equation (5.14), where $\hat{\lambda}_x$ and $\hat{\lambda}_y$ minimise the value of $-\ell\left(\lambda_x, \lambda_y; z\right)$. The values of $\hat{\lambda}_x$ and $\hat{\lambda}_y$ that minimise $-\ell\left(\lambda_x, \lambda_y; z\right)$, are in fact the same as those that maximise $\ell\left(\lambda_x, \lambda_y; z\right)$, and hence they are the mle of $\lambda_x$ and $\lambda_y$. Moreover, the optimisation can be done for one of the parameters as they are related, Equation (5.11). Hence, one of the following equations can be used in the optimisation process to obtain the mle:

$$-\ell\left(\lambda_x | z\right) = n(2\lambda_x - \bar{z}) - \frac{\log(\lambda_x) - \log\left(\lambda_x - \bar{z}\right)}{2} \sum_{j=1}^{n} z_j - \sum_{j=1}^{n} \log\left(I_{z_j}\left(2\sqrt{\lambda_x\left(\lambda_x - \bar{z}\right)}\right)\right). \tag{5.15}$$

$$-\ell\left(\lambda_y | z\right) = n(\bar{z} + 2\lambda_y) - \frac{\log\left(\bar{z} + \lambda_y\right) - \log(\lambda_y)}{2} \sum_{j=1}^{n} z_j - \sum_{j=1}^{n} \log\left(I_{z_j}\left(2\sqrt{\left(\bar{z} + \lambda_y\right)\lambda_y}\right)\right). \tag{5.16}$$

## 5.3 Hypothesis testing

Hypothesis testing is a statistical tool that is used for inference purposes. In differential binding site studies, the interest is to infer the differences between two genomic samples under different biological conditions. In a knock-out experiment, for instance, hypothesis testing can be employed to infer differential binding sites between test samples, where a transcription factor is knocked-out, and the control sample.

To use hypothesis testing, underlying assumptions or a model need to be considered. We described the characteristics of the ChIP-Seq data, which are used for differential binding site studies. In addition, we assumed a statistical model for them. Hence, in the following we clearly state and formalise the assumptions and the underlying model that are made on the ChIP-Seq data.

### Assumptions

For clarity purposes we will use a window index $i$. Let $x_{ij}$ and $y_{ij}$ be the read counts in the $j$-th position within the $i$-th window for the test and control samples, respectively, where $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$, and $n$ is the size of the window and it is fixed for all windows, and $m$ is the total number of windows, which is equal in both samples. Let $z_{ij}$ be the difference in read counts between the test and control samples in the $j$-th position and the $i$-th window, i.e., $z_{ij} = x_{ij} - y_{ij}$.

- For the test sample, we assume $X_{i1}, X_{i2}, \ldots, X_{in}$ to be independent and identically distributed ($iid$) as Poisson distribution with unknown rate parameter $\lambda_{xi}$ where $i = 1, 2, \ldots, m$, i.e, $X_{ij} \overset{iid}{\sim}$ Pois$(\lambda_{xi})$. Similarly, for the control sample we assume $Y_{i1}, Y_{i2}, \ldots, Y_{in}$ to be independent and identically distributed ($iid$) as Poisson distribution with unknown rate parameter $\lambda_{yi}$ where $i = 1, 2, \ldots, m$, i.e $Y_{ij} \overset{iid}{\sim}$ Pois$(\lambda_{yi})$.

- Given the previous assumptions on $X_{ij}$ and $Y_{ij}$, the random variable $Z_{i1}, Z_{i2}, \ldots, Z_{in}$, which is the difference in read counts within the $i$-th window, is independent and identically distributed as Poisson Difference (PD) with unknown rate parameters $\lambda_{xi}$ and $\lambda_{yi}$ for $i = 1, 2, \ldots, m$, i.e., $Z_{ij} \overset{iid}{\sim} \text{PD}(\lambda_{xi}, \lambda_{yi})$ where the values of $Z_{ij}$ are integers and $Z_{ij} \in \mathbb{Z}$.

The above assumptions are the basis of any hypothesis testing we are about to introduce. More specifically, PD distribution is the underlying model in our hypothesis testing. Given this, the differential binding sites (or differential genomic regions) can be identified by testing whether the mean of $Z_{ij}$, which is $E(Z_{ij}) = \lambda_{xi} - \lambda_{yi}$ (see Equation (5.2)), is equal to a pre-specified value $c$. Specifically, the interest is in testing the following hypotheses:

$$\begin{aligned} H_{0i}: &\quad \lambda_{xi} - \lambda_{yi} = c, \\ H_{1i}: &\quad \lambda_{xi} - \lambda_{yi} \neq c. \end{aligned} \tag{5.17}$$

The value of $c$ depends on the total read counts of the test and control. That is, let $N_x$ and $N_y$ be the total read counts in the test and control samples, respectively, i.e., $N_x = \sum_{i,j} x_{ij}$ and $N_y = \sum_{i,j} y_{ij}$. If $N_x = N_y$, then $c$ is equal to zero, otherwise $c$ is equal to the observed mean of the genome-wide difference $\bar{z}$, i.e., $\bar{z} = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} z_{ij}$. In simple words, the value $c$ replaces the normalisation process, which is often required when the sample sizes are different.

One can argue that the null hypothesis can be rewritten as $H_{0i} : \lambda_{xi} = c\lambda_{yi}$, where $c$ is a normalising constant. Therefore, the parameters can be estimated from each sample separately, then the test can be performed. We can say that this is true analytically but not in practice. By simulation we observed that under the PD model the sample size needs to be huge (larger than $1 \times 10^6$) to obtain estimates of parameters that match the estimates of the parameters of each Poisson sample alone. Hence, we need to use the difference data to obtain the right estimates, and normalising the read counts by constant $c$, the differences

will not be integers any more (at least some of them). Thus, we stick to the hypotheses in (5.17).

As shown above in the hypotheses of interest (5.17), the testing is performed in each window $i$. For a given measure of significance, for example $5\%$, a window is considered significantly differential between the two samples if the null hypothesis $H_{0i}$ is rejected, where $i = 1, 2, \ldots, m$. In addition, given the hypotheses (5.17) and the underlying model, PD, we employ three parametric tests to identify differential windows between the two samples. These tests are Exact Test (ET), Wald Test (WT) and Likelihood Ratio Test (LRT). In ET, the underlying model PD is used to calculate the test statistic, whereas WT and LRT are approximate tests and hence approximated distributions are used to calculate the test statistics (more details in Sections 5.3.1, 5.3.2 and 5.3.3). Note that the tests formalise the hypotheses of interest (5.17) differently. That is, WT and LRT have the same formalisation as in (5.17), while in ET the hypotheses are re-formalised to be as follows (more details in Section 5.3.1):

$$\begin{aligned} H_{0i}: &\quad n\lambda_{xi} - n\lambda_{yi} = nc, \\ H_{1i}: &\quad n\lambda_{xi} - n\lambda_{yi} \neq nc, \end{aligned}$$

where $n$ is the window size, which is the number of bins (or positions) within a window.

### 5.3.1 Exact Test (ET)

We mentioned that the PD model is the basis of our testing methods. This underlying model can be used itself to identify differential windows. In the $i$-th window, a test statistic, say $E_i$, is calculated and a p-value can be computed by using the exact distribution under the null hypothesis, which is PD , hence it is called an exact test.

In ET, we still assume PD as an underlying model, but with different parameters. That is, let $X_i^* = \sum_{j=1}^{n} X_{ij}$ and $Y_i^* = \sum_{j=1}^{n} Y_{ij}$ be sum of read counts within $i$-th window in

test and control samples, respectively. It can be shown that $X_i^*$ and $Y_i^*$ follow Poisson distribution with parameters $n\lambda_{xi}$ and $n\lambda_{yi}$, respectively, i.e. $X_i^* \sim \text{Pois}(n\lambda_{xi})$ and $Y_i^* \sim \text{Pois}(n\lambda_{yi})$. Now, let $Z_i^* = X_i^* - Y_i^*$ be the difference between the sums of read counts in window $i$, $i = 1, 2, \ldots, m$. Then it can be shown that $Z_i^*$ follows PD distribution with rate parameters $n\lambda_{xi}$ and $n\lambda_{iy}$, i.e., $Z_i^* \sim \text{PD}(n\lambda_{xi}, n\lambda_{yi})$. Given this, the expected value of $Z_i^*$ can be shown as $E(Z_i^*) = n\lambda_{xi} - n\lambda_{iy}$. In ET, we use $Z_i^*$ where $Z_i^* = \sum_{j=1}^{n} Z_{ij}$. Hence, the hypotheses of interest (5.17) will be re-formalised as follows:

$$\begin{aligned} H_{0i}: &\quad n\lambda_{xi} - n\lambda_{yi} = nc, \\ H_{1i}: &\quad n\lambda_{xi} - n\lambda_{yi} \neq nc, \end{aligned} \tag{5.18}$$

where $n$ is the number of bins (or positions) within a window and $c$ is a pre-specified value that is equal to $\bar{z}$. Let $z_i^*$ be the observed value of $Z_i^*$. Then, the test statistic, $E_i$, can be defined as:

$$E_i = z_i^* = \sum_{j=1}^{n} z_{ij} \quad i = 1, 2, \ldots, m. \tag{5.19}$$

Under the null hypothesis $H_{0i}$, $E_i \sim \text{PD}(n\lambda_{xi}^0, n\lambda_{yi}^0)$ exactly, that where $\lambda_{xi}^0$ and $\lambda_{xi}^0$ are parameters which satisfy the null constraint, i.e., $\lambda_{xi}^0 - \lambda_{xi}^0 = \bar{z}$.

It was mentioned that PD distribution is not symmetric when its expectation is not equal to zero. Hence, to compute the p-value, $p_i$, we compute the complement probability of the non-rejection region. That is, we sum up probabilities of all possible $Z_i^*$, say $t$, such that the probabilities are larger than or equal to the probability of the observed test statistic $E_i$, then subtract this summation from one. By doing this, we compute the probability of the rejection region under the null model. This can be formalised as follows:

$$p_i = 1 - \sum_{t \in \Omega_i} P_{H_{0i}}(Z_i^* = t) \quad \text{where} \quad \Omega_i = \left\{ t : P_{H_{0i}}(Z_i^* = t) \geq P_{H_{0i}}(Z_i^* = E_i) \right\}. \tag{5.20}$$

If $p_i$ is lower than some threshold, the null hypothesis is rejected and window $i$ is declared to be significantly different between the two samples, otherwise it is not. Note, a multiplicity correction needs to be used to adjust p-values $p_i$ before drawing a conclusion about the significance.

### 5.3.2 Wald Test (WT)

For the given hypotheses (5.17) and the underlying model, WT can be employed to identify differential binding sites in ChIP-Seq data. WT was first introduced by the Hungarian statistician Abraham Wald [67]. In WT, if $\theta$ is a parameter of interest and $\hat{\theta}$ is the mle of it, and we want to compare that estimate to a specific value $\theta_0$, then a test statistic $\frac{\hat{\theta}-\theta_0}{se(\hat{\theta})}$ would follow a standard normal distribution asymptotically, where $se(\hat{\theta})$ is standard error.

We are interested in the mean $E(Z_i)$, where $E(Z_i) = \lambda_{xi} - \lambda_{yi}$, and in each window $i$ we want to test the hypothesis in (5.17), $H_{0i} : \lambda_{xi} - \lambda_{yi} = c$. Hence in this case, $\theta = \hat{\lambda}_{xi} - \hat{\lambda}_{yi}$, $\theta_0 = c$, and $se(\hat{\theta}) = \sqrt{(\hat{\lambda}_{xi} + \hat{\lambda}_{yi})/n}$ which is an approximation for the standard error, where $\hat{\lambda}_{xi}$ and $\hat{\lambda}_{yi}$ are the mle of $\lambda_{xi}$ and $\lambda_{yi}$ under the constraint $\lambda_{xi} - \lambda_{yi} = \bar{z}_i$, and $\bar{z}_i$ is the sample mean of the difference in read counts in $i$-th window. Then, a test statistic, denoted as $W_i$, can be written as follows:

$$W_i = \frac{\left(\hat{\lambda}_{xi} - \hat{\lambda}_{yi}\right) - c}{\sqrt{(\hat{\lambda}_{xi} + \hat{\lambda}_{yi})/n}}, \quad i = 1, 2, \ldots, m.$$

Under the null $H_{0i}$, $W_i$ is approximated as standard normal, $W_i \sim \mathrm{N}(0, 1)$. However, we know tha $Z_{ij}$ is a discrete random variable, which follows PD distribution. Hence, as we approximate discrete variable as continuous variable, a continuity correction needs to be considered [71]. The continuity correction simply is adding $\pm 0.5$ to the discrete

value of interest. But here we are interested in the mean, hence the continuity correction is $\pm 0.5/n$. In this case, the test statistic $W_i$ can be rewritten as,

$$W_i = \frac{\left(\hat{\lambda}_{xi} - \hat{\lambda}_{yi}\right) \pm \frac{0.5}{n} - c}{\sqrt{(\hat{\lambda}_{xi} + \hat{\lambda}_{yi})/n}}, \quad i = 1, 2, \ldots, m, \tag{5.21}$$

where $c$ is equal to $\bar{z}$, which is the observed mean of the genome-wide difference, and under the null $H_{0i}$, $W_i \sim \mathrm{N}(0, 1)$ approximately. Given this, p-value, $p_i$, can be computed directly from the standard normal distribution at each window $i$ by, first, calculating the probability of observing values larger than or equal to the absolute value of the test statistic $W_i$. Second, multiplying this probability by two to get the probabilities of the rejection regions in both tails of the distribution as it is a symmetric distribution. In mathematical notations that is:

$$p_i = 2 \times P\left(W \geq |W_i|\right) \quad \text{where} \quad W \sim \mathrm{N}(0, 1). \tag{5.22}$$

For a given threshold $\alpha$, $H_{0i}$ would be rejected and window $i$ would be declared significantly differential, if $p_i < \alpha$ after applying the multiplicity correction in $p_i$.

### 5.3.3 Likelihood Ratio Test (LRT)

Similar to WT, for given hypotheses (5.17) and the underlying model, LRT can be employed to detect differential binding sites for ChIP-Seq data. LRT was first introduced by Samuel Wilks, [69], when he found the asymptotic distribution for the observation testing function, which is the test statistic, by Neyman and Pearson [41]. In LRT, two likelihood functions under two different models are compared. These models are considered as the null and the alternative models. That is, we can identify differential windows by comparing the PD likelihood with parameters $\hat{\lambda}_{xi}$ and $\hat{\lambda}_{yi}$, say $L(\lambda_{xi} = \hat{\lambda}_{xi},$

$\lambda_{yi} = \hat{\lambda}_{yi}$), and the PD likelihood with parameters under the null model $\lambda^0_{xi}$ and $\lambda^0_{yi}$, say $L_0(\lambda_{xi} = \lambda^0_{xi}, \lambda_{yi} = \lambda^0_{yi})$. The hypotheses of interest are shown in (5.17), which are:

$$\begin{aligned} H_{0i}: &\quad \lambda_{xi} - \lambda_{yi} = c, \\ H_{1i}: &\quad \lambda_{xi} - \lambda_{yi} \neq c. \end{aligned}$$

where $c = \bar{z}$, which is the sample mean of the genome-wide difference data. The test statistic of LRT, say $R_i$, can be computed in each window $i$ as follows:

$$R_i = -2 \times \log \left( \frac{L_0(\lambda_{xi} = \lambda^0_{xi}, \lambda_{yi} = \lambda^0_{yi})}{L(\lambda_{xi} = \hat{\lambda}_{xi}, \lambda_{yi} = \hat{\lambda}_{yi})} \right), \quad i = 1, 2, \ldots, m. \tag{5.23}$$

Under the null hypothesis $H_{0i}$, $R_i$ approximately follows a Chi-squared distribution, $\chi^2$. To calculate the likelihood under the null PD model, $L_0$, we estimate the parameters by using constraint $\hat{\lambda}^0_{xi} - \hat{\lambda}^0_{yi} = \bar{z}$. On the other hand, the constraint $\hat{\lambda}_{xi} - \hat{\lambda}_{yi} = \bar{z}_i$ is used to estimate the parameters of the observed PD model, where $\bar{z}_i$ is the sample mean of the difference data in window $i$. Note that given these constraints, we need only to estimate one of the parameters (see Equation (5.11)). In this case and under the null hypothesis, the test statistic $R_i$ follows $\chi^2$ distribution with one degree of freedom, approximately, owing to the used constraint in the estimation process.

Given the test statistic $R_i$, the p-value $p_i$ can be calculated from $\chi^2_1$ as follows:

$$p_i = P\left(R \geq R_i\right) \quad \text{where} \quad R \sim \chi^2_1. \tag{5.24}$$

For a given threshold $\alpha$, the null hypothesis would be rejected and window $i$ would be declared significantly differential, if $p_i < \alpha$ after applying a multiplicity correction in $p_i$.

## 5.4 Simulation study

We perform simulation studies to determine several factors. First, to determine the ability of obtaining mle of $\lambda_{xi}$ and $\lambda_{yi}$ under the PD model by using the difference data. Second, to determine whether the test statistics behave well with the distributions under the null hypothesis in ET, WT and LRT. Third, to determine whether ET, WT and LRT control the false-positive rate at a specified level of significance $\alpha$. Finally, to determine the power of ET, WT and LRT in detecting truly differential regions.

To do so, we consider two simulation models. First, a model based on simulated data, Section 5.4.1. In this simulation model, we determine the four previous factors. Second, a model based on real data, which is ENCODE project data. In this simulation model, we only determine the false-positive rate and the power factors; more details can be found in Section 5.4.2.

### 5.4.1 Simulated data

#### Determination of parameter estimation

To determine the ability of obtaining the mle of $\lambda_{xi}$ and $\lambda_{yi}$, we simulate two Poisson samples, $x_i = x_{i1}, x_{i2}, \ldots, x_{in}$ and $y_i = y_{i1}, y_{i2}, \ldots, y_{in}$ with known rates $\lambda_{xi}$ and $\lambda_{yi}$, where $i$ indicates the window number (it can be considered as sample number) and $n$ is the sample size. Then, we take the difference $z_i = x_i - y_i$, i.e., $z_i = z_{i1}, z_{i2}, \ldots, z_{in}$. By using observations $z_i$, we want to obtain the mle $\hat{\lambda}_{xi}$ and $\hat{\lambda}_{yi}$. Note that to obtain the mle of PD, we use either Equation (5.15) or (5.16).

By using the above setting, we run many simulations by using different values of rates, $\lambda_{xi}$ and $\lambda_{yi}$, and different sample sizes $n$. In these simulations, we notice that the mle of the parameters are quite different from the true values, and from the direct mle from

each sample, which are $\hat{\lambda}_{xi} = \bar{x}_i$ and $\hat{\lambda}_{yi} = \bar{y}_i$. For instance, in Figure 5.1 we show a profile of the likelihood of a single simulation by using the difference data $z_i$ where the true rate parameters are $\lambda_{xi} = 0.3$ and $\lambda_{yi} = 0.2$, with sample size $n = 20$. In the figure, it can be seen that the behaviour of the likelihood function is regular for the given values of $\lambda_{xi}$ and $\lambda_{yi}$ with maximum values at $\hat{\lambda}_{xi} = 0.22$ and $\hat{\lambda}_{yi} = 0.12$. However, the mle from each of the simulated samples are $\hat{\lambda}_{xi} = 0.31$ and $\hat{\lambda}_{yi} = 0.21$, which are close to the true values. On the other hand, the difference between $\hat{\lambda}_{xi}$ and $\hat{\lambda}_{yi}$ is the same in both estimators , either by using $z_i$ or $x_i$ and $y_i$, owing to the constraint (5.11). To address this issue of different estimations, we conduct a bias simulation study for the estimators of PD distribution (more details follow).

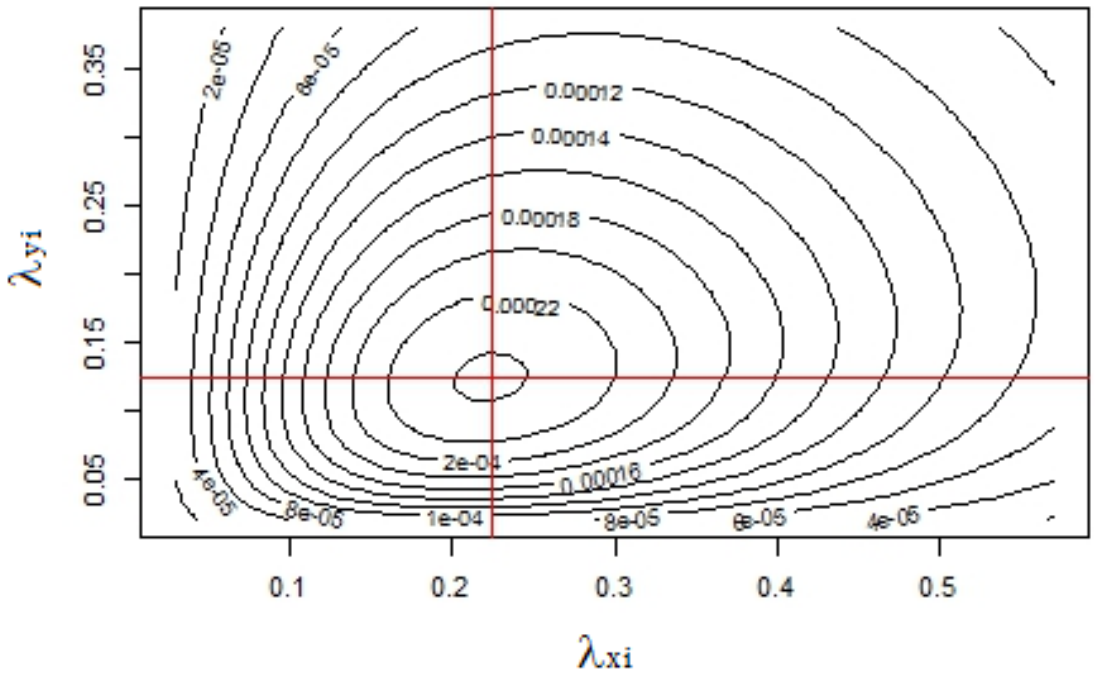



Figure 5.1: A profile likelihood for $\lambda_{xi}$ and $\lambda_{yi}$ by using the difference data $z_i$. Horizontal and vertical axes represent values of parameters $\lambda_{xi}$ and $\lambda_{yi}$, respectively. The sample size is $n = 20$. The intersection between the two red lines indicates the maximum likelihood value.

In the bias study, for simplicity we consider the case when the two parameters of PD are equal [21], say $\lambda_i$, i.e., $\lambda_i = \lambda_{xi} = \lambda_{yi}$. In this case, the probability mass function in Equation (5.1) can be re-written as follows:

$$P(Z_i = z) = e^{-2\lambda_i} I_z(2\lambda_i), \quad z \in \{\ldots, -2, -1, 0, 1, 2\ldots\}, \quad \lambda_i > 0. \tag{5.25}$$

In the above probability function, it can be seen that only one parameter needs to be estimated, which is $\lambda_i$. Hence, to obtain the mle of $\lambda_i$, we use the likelihood function of (5.25), $L(\lambda_i|z_i)$, which can be shown as:

$$L(\lambda_i|z_i) = \prod_{j=1}^{n} e^{-2\lambda_i} I_{z_{ij}}(2\lambda_i),$$

and the log-likelihood, $\ell(\lambda_i|z_i)$, as:

$$\ell(\lambda_i|z_i) = -2n\lambda_i + \sum_{j=1}^{n} \log\left(I_{z_{ij}}(2\lambda_i)\right).$$

Again, the mle of $\lambda_i$ cannot be derived explicitly, as the derivation includes a Bessel term. Hence, we use numerical optimisation methods to obtain the mle of $\lambda_i$. The negative log-likelihood function to be used in the optimisation process is: $-\ell(\lambda_i|z_i)$.

The simulation is done as follows. We simulate samples $x_i$ and $y_i$ from Poisson distribution with rate parameter $\lambda_i$, Pois$(\lambda_i)$, and sample size $n$. Then, we consider the difference $z_i = x_i - y_i$, hence $Z_i$ would follow PD$(\lambda_i, \lambda_i)$. Finally, we estimate $\lambda_i$ by using $-\ell(\lambda_i|z_i)$. For the rate parameter, we use $\lambda_i$ ranges from 0.01 to 1. Furthermore, for each rate parameter $\lambda_i$ we simulate $10,000$ samples for each $x_i$ and $y_i$, i.e. $i = 1, \ldots, m = 10000$. Moreover, for each of the rate $\lambda_i$ we simulate by using sample sizes $n = 5, 25, 45, \ldots, 485$. Finally, we compute the bias by subtracting the true rate $\lambda$ from the average of the estimates. Mathematically, this can be formalised as follows. Let $\hat{\lambda}_i$ be the mle of $\lambda$ by using $i$-th sample where $i = 1, 2, \ldots, m$. Then, the bias of the estimate can be shown as:

$$\text{Bias}\left(\hat{\lambda}\right) = \frac{1}{m}\sum_{i=1}^{m}\hat{\lambda}_i - \lambda. \tag{5.26}$$

Moreover, for each sample size $n$ we compute the bias. For instance, for $\lambda_i = 0.01$ we compute the bias for sample size $n = 5$, $n = 25$, $n = 45$ and so forth up to $n = 485$. We do the same strategy for other proposed values of $\lambda_i$, which ranges from 0.01 to 1. Hence for each value $\lambda_i$, we have a number of estimates that is equal to the number of the sample sizes. In other words, we want to see the bias as a function of the sample size $n$. In Figure 5.2, we show some of the findings.

In Figure 5.2, it can be seen that the relative bias goes to zero as the sample size and the rate parameter increase. Furthermore, there is a sharp decay at the beginning of the figure. Specifically, from $n = 5$ to $n = 25$ the bias decreases sharply, and after $n = 25$ it continues decreasing but slower compared to the first part. The explanation of this behaviour is simply because sample size $n = 5$ is very small, and hence it is less likely to be an accurate. On the other hand, sample sizes $n = 25$ and more can be considered large, hence the bias from size to size decreases smoothly as the size increases. Thus, it appears that the maximum likelihood estimator of PD is asymptotically unbiased.

**Determination of test statistic distribution**

To determine whether the test statistics behave well with the distributions under the null hypothesis in ET WT and LRT, we can compare the observed distribution of the test statistics to the null distributions. That is, in ET the test statistics are compared to PD, in WT the test statistics are compared to the standard normal, and in LRT the test statistics are compared to Chi-squared with one degree of freedom. More simply, the same comparisons can be achieved by analysing the p-values of the test statistics. That is, we can say the test statistics behave well with the null distribution, if the cumulative distribution of their p-values is uniform and exhibits a continuous increasing function of the observed p-values.

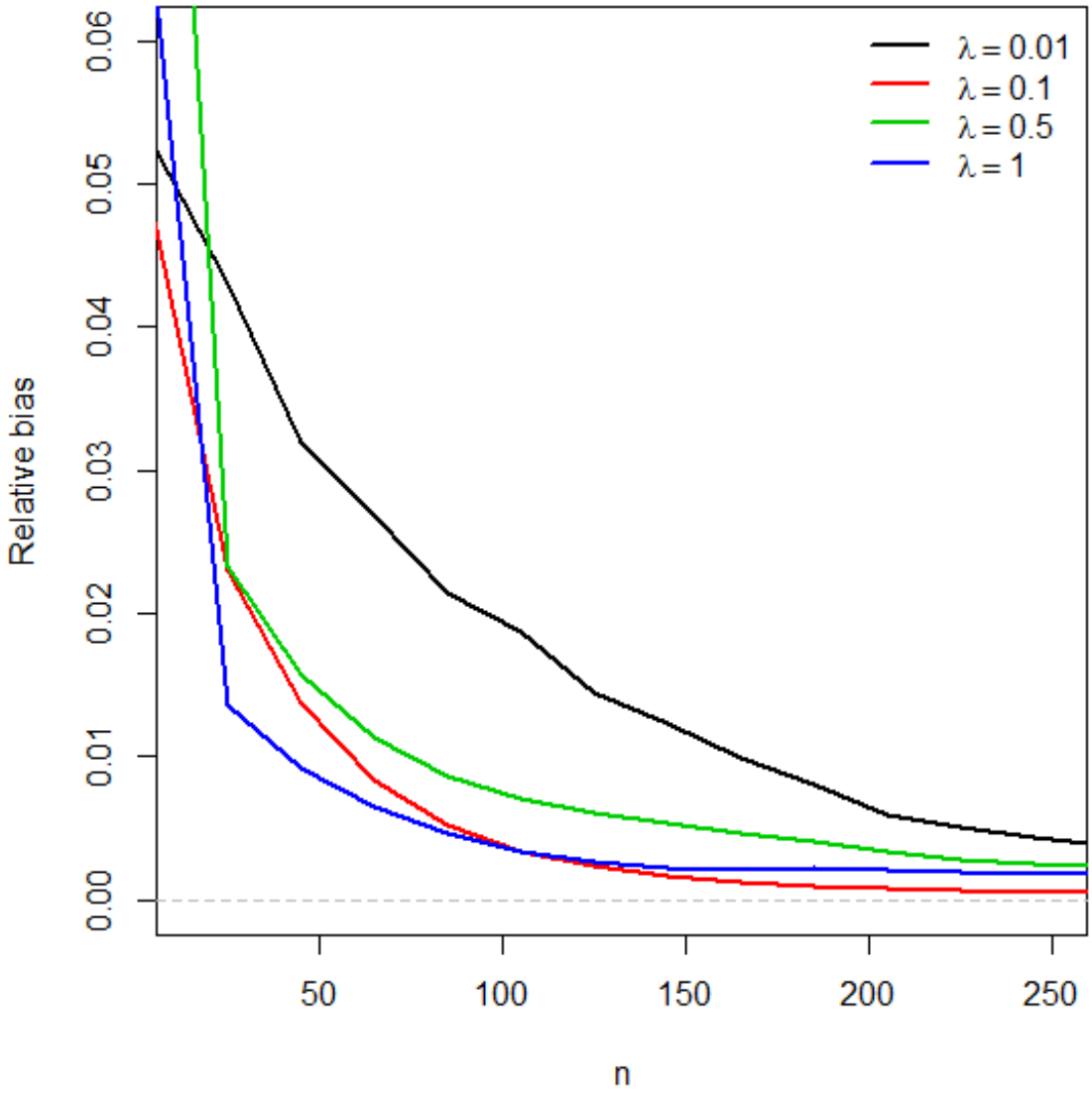



Figure 5.2: Bias of maximum likelihood estimator of PD as a function of sample size. Horizontal and vertical axes are the sample size and the relative bias, which is Bias$(\hat{\lambda})/\lambda$, respectively. Each point in the figure represents an average of $10,000$ bias values by using a sample size $n$. Four lines for $\lambda = 0.01, 0.1, 0.5, 1$ are shown in the figure.

To do the above comparisons, we conduct a simulation study as follows. First, we simulate two Poisson samples $x_i$ and $y_i$ with the same rate $\lambda = \lambda_{xi} = \lambda_{yi}$, which corresponds to the null hypothesis in (5.17), and sample size $n$. Second, we consider the difference $z_i = x_i - y_i$. Finally, we perform the tests to obtain the test statistics $E_i$, $W_i$, and $R_i$. In the simulation, we consider values of the rate $\lambda$ ranging from $0.01$ up to $1$, and sample size $n = 200$. In addition, for each $\lambda$ we simulate $10,000$ samples from each $x_i$ and $y_i$, hence for each rate $\lambda$ we end up with $10,000$ values of $E_i$, $W_i$, and $R_i$, i.e. $i = 1, 2, \ldots, 10000$. These test statistics are compared to PD$(n\lambda, n\lambda)$, N$(0, 1)$ and $\chi^2_1$ in ET, WT and LRT, respectively. Moreover, for these test statistics we can compute p-values as shown in Equations (5.20, 5.22 and 5.24). In addition, we repeat all previous simulations $100$ times and consider an average result for each of the rates $\lambda$. The result is shown in Figure 5.3.

In Figure 5.3, we show the empirical CDF of p-values of test statistics $E$, $W$, and $R$, by using rate $\lambda = 0.01$, $0.1$, $0.5$ and $1$. In the figure, it can be seen that none of the tests behave well when the rate is low. More specifically, by using $\lambda = 0.01$, the distributions of the p-values appears discrete under ET, WT and LRT. From the behaviour of the CDF it can be said that ET and WT would show a rejection rate for the null hypothesis lower than expected at nominal significance level $5\%$, while LRT would show larger (more details in the following section). However, it is noticed that for rate $\lambda \geq 0.1$, the CDF improved in all tests. It can be seen that WT and LRT show quite similar behaviour as the rate increases. ET improves as well by increasing the rate $\lambda$, but not as much as WT and LRT.

**Determination of false-positive rate and power**

To determine the controlling of the false-positive rate and the power in ET, WT and LRT, we use the same simulated data to evaluate the false-positive rate in diffReps and MACS with MAnorm methods, Section 4.3.1. We perform the tests on the simulated data and show the results in Figure 5.4.

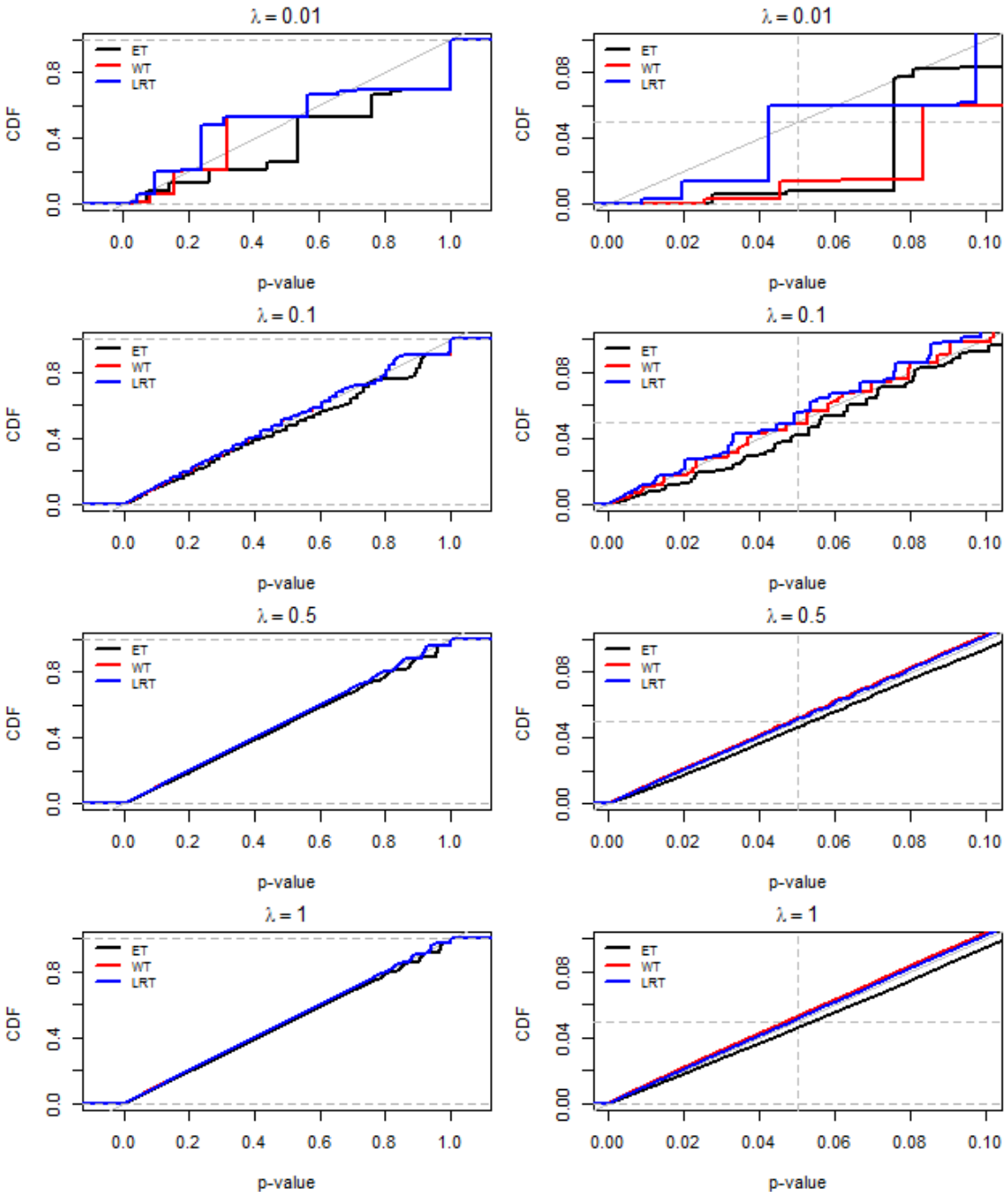



Figure 5.3: Empirical cumulative distribution function (CDF) of p-values of test statistics of ET, WT and LRT. The left hand column of the figure represents the full CDF, whereas the right hand column represents zoom plots of the left plot around $5\%$. Each of the coloured lines represents an average of $100$ lines, each of which contains $10,000$ p-values. In the simulation we simulate two Poisson samples with the same rate $\lambda$.

In the top panel of Figure 5.4, we can see the tests behave as we expected in the previous section. That is, for low rate values $\lambda$ none of the tests show good control for the false-positive rate at the significance level of $5\%$. It can be seen that ET and WT seem to be conservative and show a low false-positive rate associated with low values of $\lambda$. On the other hand, LRT shows a quite high rate of false-positive error associated with a low rate of $\lambda$, lower than $0.1$. However, it can be noticed that for values of $\lambda$ around $0.2$ and more, all tests show quite good control of the false- positive rate.

In the bottom panel of Figure 5.4, we represent the evaluation of the tests' power. It can be seen that LRT appears as the most powerful test, and next to it WT then ET. However, it can be said that the differences between the tests in term of power are not large, and the tests show similar power in general. It also can be seen that all tests start to detect the differences around $0.05$, and reach maximum power with difference in rates around $0.3$.

### 5.4.2 ENCODE data

In this section, we evaluate the proposed testing methods, ET, WT and LRT by using ChIP-Seq data from the ENCODE project [12]. Initially, we aim to determine four factors, parameter estimation, test statistic distribution, false-positive rate and power. However, by using real data we are only able to evaluate two factors, which are false-positive rate and power. That is, in real data the true values of the parameters $\lambda_{xi}$ and $\lambda_{yi}$ are unknown. Hence, in parameter estimation determination we can obtain the mle $\hat{\lambda}_{xi}$ and $\hat{\lambda}_{yi}$, but we cannot compare them to the actual $\lambda_{xi}$ and $\lambda_{yi}$ because they are unknown. About test statistics, we can have a single comparison for the test statistics distribution in each dataset with the null distribution. However, we cannot combine more than one dataset in a single comparison as the true rate parameters are unknown. Hence, it acceptable for the checking in specific data, but not enough to draw a conclusion.

To determine the false-positive rate and power of ET, WT and LRT, we use the same

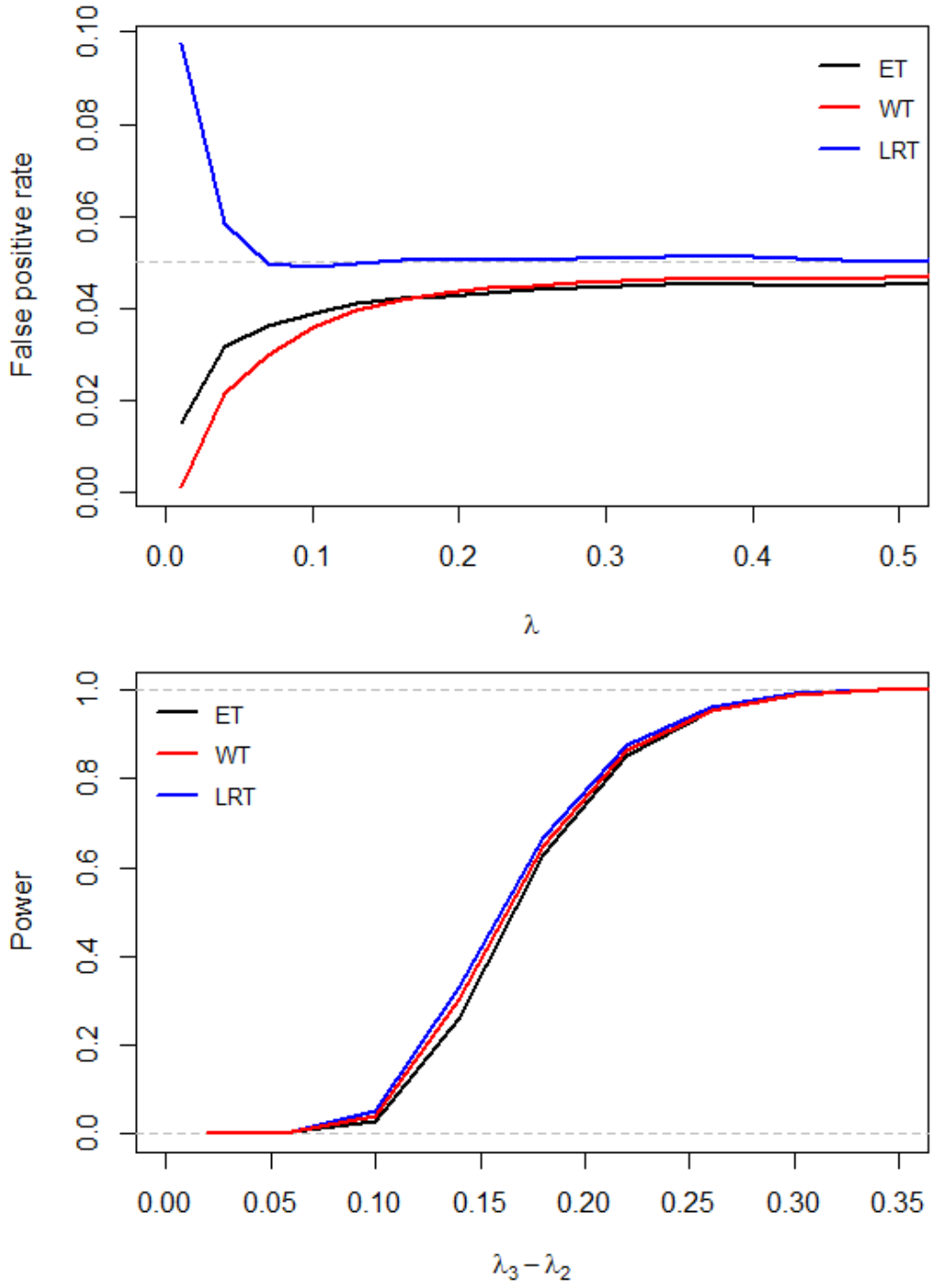



Figure 5.4: The top panel shows the evaluation of the false-positive rate by using ET, WT and LRT based on simulated data. In that panel, the horizontal axis represents the used $\lambda$ in the simulation and the vertical axis represents the average of 50 false-positive rates for each of the $\lambda$s. The lower panel shows the power of ET, WT and LRT based on simulated data. In that panel, the horizontal axis represents the difference in rates between the two simulated datasets, where we enrich the windows with different rates, and the y axis represents the average power of 50 simulations for each difference $\lambda_3 - \lambda_2$. More details about the evaluation structure are provided in Section 4.3.1

ENCODE data in Section 4.3.2, which are used in MACS and MAnorm, and diffReps methods evaluation. Hence, we performed the tests ET, WT and LRT, and show the results in Figure 5.5. In the top panel of the figure, we represent the false positive evaluation. We can see that ET and LRT show good control of the false-positive rate at the $5\%$ level of significance, whereas WT shows over controlling of it. The figure indicates that ET is the most accurate test, and in second place is LRT. Furthermore, it indicates that WT is quite a conservative test.

The lower panel of Figure 5.5 represents the power determination. It can be seen that the tests show slightly different power, although they all start detecting significant results with very small differences in rates. That is, the tests start to declare significant results and continue with a sharp increase at differences in observed rates lower than $0.04$, in order LRT, ET and then WT. However, around differences of $0.1$ ET and WT continue their sharp increasing and become more powerful compared to LRT. ET and WT reach the maximum power at a difference of around $0.4$, whereas LRT reaches it at a difference of around $0.6$.

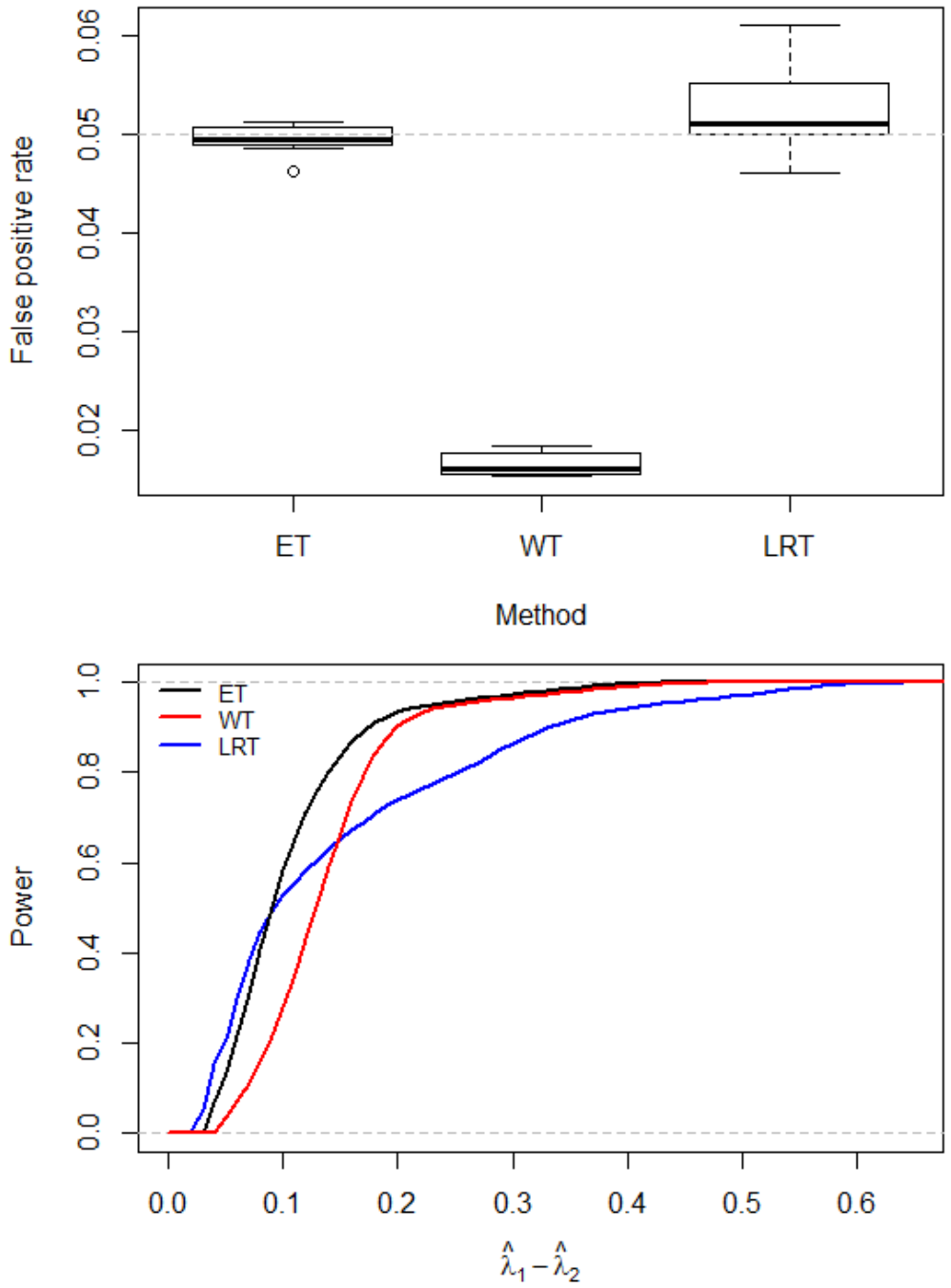



Figure 5.5: The top panel shows the determination of the false-positive rate by using ET, WT and LRT based on real ENCODE data. In that panel, the horizontal axis represents the used methods and the vertical axis shows the false-positive rates for each of the methods. Each point in the plot represents the false-positive rate by using a subsample that is a half of a chromosome. The lower panel shows the power of ET, WT and LRT based on real ENCODE data. In that panel, the horizontal axis represents the observed difference in rates between the two samples and the vertical axis represents the average power by using ENCODE data. More details about the evaluation structure are provided in Section 4.3.2.

## 5.5 Project data RUNX1/ETO

We perform the tests ET, WT and LRT for RUNX1/ETO data. We use window size 200 bp, which is optimal, and bin size 1 bp within windows. The result is represented in a Venn diagram in Figure 5.6. Note that all the following findings were obtained after applying multiplicity correction FDR [8] based on the human genome-wide (hence total number of tests is $(3 \times 10^9)/200$, where 200 is the window size).

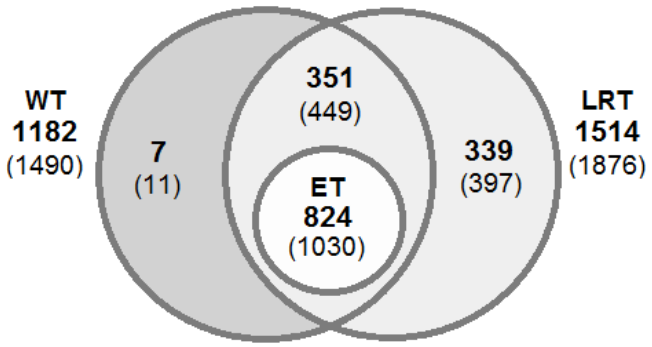



Figure 5.6: The number of significant regions (either as single windows or a group of adjacent windows) in RUNX1/ETO data detected by using ET, WT and LRT. As some windows are adjacent to each other, the numbers within brackets represents the corresponding number of significant windows of size 200 bp.

In the figure, we can see that ET declares 824 regions corresponding to 1030 significant windows of size 200 bp. In addition, 1182 regions corresponding to 1490 significant windows are detected by WT. Moreover, LRT detects 1514 regions corresponding to 1876 significant windows. In total, there are 1521 regions that are declared by either ET, WT or LRT corresponding to 1887 significant windows. It can be noticed in the figure that all regions declared by ET are subsets from both WT and LRT. In other words, ET, WT and LRT all agree on the significance of these 824 regions. In addition, it can be seen that WT has 7 regions corresponding to 11 windows that are uniquely declared. By checking the p-values of these 11 windows, it is found that the p-values are all close to the threshold

5%, and ET and LRT have p-values just above the threshold.

Figure 5.7 presents the highest and lowest significance windows from the union of the significant results by ET, WT and LRT based on their p-values. In the top panel the most significance window is shown, and in the lower panel is the lowest significance window. From the figure it can be said that the proposed tests are able to detect significant differences either in dense windows or windows with low read counts.

## 5.6 Discussion and conclusion

In this chapter, we have developed three hypothesis tests that are able to detect significant differential binding sites (or regions) in ChIP-Seq experiments. Given two ChIP-Seq samples, test and control, the observed read counts at genomic positions are considered as paired observations. Hence, the difference in read counts between the test and control samples is considered and analysed by the proposed tests. The assumption made on the data is clear, which is the read counts within a genomic window are assumed to follow Poisson distribution with a single and unknown rate parameter. Hence, the difference in read counts within a genomic window follows PD distribution with two unknown rate parameters. An advantage of considering the difference is that the difference observations become more independent, although the read counts in each sample sometimes show weak dependence.

To perform the tests, the two parameters of PD need to be estimated. We seek the mle. Obtaining the mle analytically is not trivial as the derivations involve Bessel terms. Hence, numerical optimisation methods are used instead to obtain the mle. As the two parameters are related, one of them is optimised and the other parameter is estimated by using a constraint. We faced a challenge in the estimation process. That is, in the optimisation process we usually obtain a positive estimate as we use a transformation that grantees that. However, estimating the other parameter by using the constraint sometimes yields a

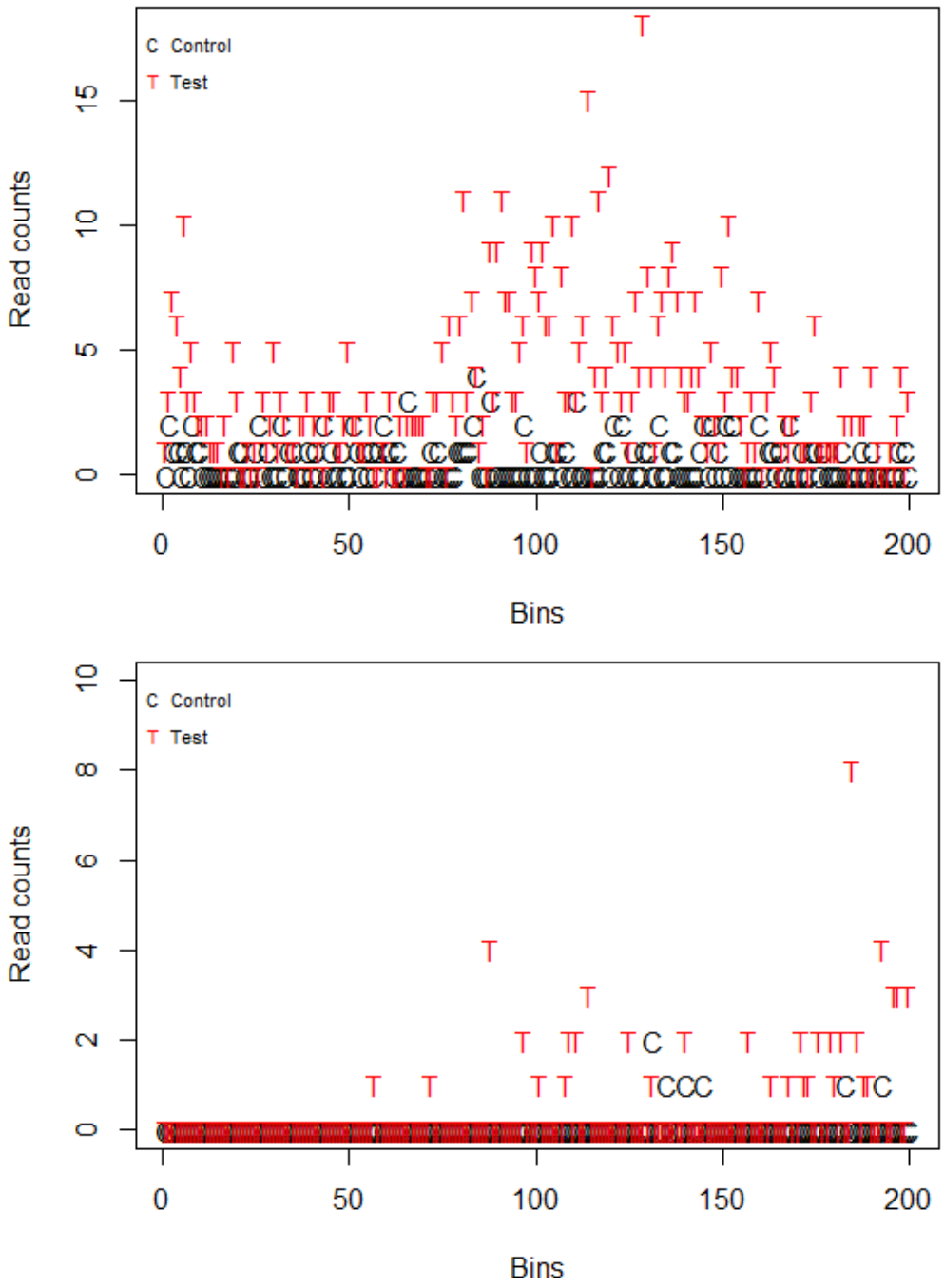



Figure 5.7: Highest (top panel) and lowest (lower panel) significance windows from the union of the significant results by ET, WT and LRT based on their p-values. Horizontal axis represents bins (or positions) and vertical axis represents read counts. The windows are represented by using window size 200 bp, which is optimal and used in the analysis process.

negative estimate. This happens when one of the windows contains no read counts or a very low number of read counts compared to the other window. To solve this issue, we just swap the optimised parameter with the other one, i.e. instead of using Equation (5.15) we use Equation (5.16), and vice-versa.

In PD parameter estimation, we focus on the mle, as they are required in the approximated tests, WT and LRT. However, a moment estimate can be provided as well. That is, for difference data $z$, where $Z \sim \text{PD}(\lambda_x, \lambda_y)$, the moment estimates of the parameters can be shown as $\hat{\lambda}_x = \frac{1}{2}(S^2 + \bar{z})$ and $\hat{\lambda}_y = \frac{1}{2}(S^2 - \bar{z})$, where $S^2$ and $\bar{z}$ are the sample variance and mean of the difference $z$. However, the moment estimates do not exist if the absolute value of the sample mean is larger than the sample variance, i.e. $|\bar{z}| > S^2$. If that happened, then we would obtain a negative estimate for one of the parameters, which is the same issue we faced in mle (but we solve it). Some researchers, like [4], suggest considering the negative estimate zero, and the other one as the absolute value of the sample mean, $|\bar{z}|$. However, if we did consider that suggestion, the probability function of PD, Equation (5.1), would be made redundant. That is, we would not be able to use the function, Equation (5.1), because it would be either $0$ or $\infty$. More details about PD estimators, properties, and asymptotic properties and more are given in [4, 22, 23].

The maximum likelihood estimates are asymptotically unbiased estimates for the parameters of PD. That is, the bias tends to zero as the sample size and the rate parameter go to infinity. However, we notice that large parameters exhibit larger bias when The sample is small with convergence to zero slower than small parameters. This may suggest that we can look for a formula where we can compute the sufficient sample size for a given rate parameter. However, such a formula would not be useful in practice as the actual rate parameters are unknown. On the other hand, the bias would not cause any problem in the testing methods we proposed. That is, the testing we do is all around the expected value under PD, which is the difference between two rates. Hence, the bias would not have much effect when we consider the difference.

The tests we propose involve one exact test (ET) and two approximated tests (WT and LRT). That is, in ET the same distribution under the null assumption is assumed to be the test statistic (which is PD), while in WT and LRT approximated distributions are assumed for the test statistics (which are standard normal and Chi-squared, respectively). In the evaluation of the observed and assumed distributions for the test statistics, we find that all three tests show not well fitting when the rate parameters are very small, which is due to the discreteness. However, the fitting improves quite quickly when the rate parameters increase.

Furthermore, the evaluation of the proposed tests (ET, WT and LRT) based on simulated and real data shows that LRT is the most powerful test when the rate parameters are very small in PD. However, by looking at the false-positive evaluation based on simulated data, we can say that power can be owing to the high level of false-positive rate associated with small rate parameters. On the other hand, ET and WT show very low rates of false positive error evaluation (based on simulated data) associated with low rate parameters, but do well in the power evaluation.

In the real-data-based evaluation, it can be said that ET is the most accurate test compared to WT and LRT. Although LRT in the false positive rate evaluation shows good controlling of the error at $5\%$, the observed variability in ET is much smaller than the observed variability in LRT, Figure 5.5 (top panel). Moreover, by looking at WT's false-positive rate, we can say that WT is doing well in terms of power in the same figure (lower panel).

To sum up, it can be said that the proposed tests (ET, WT and LRT) are able to detect differential regions or binding sites based on the assumptions. Based on the false-positive evaluation at the given level of significance, the tests show reasonable controlling for the false-positive rate, especially when the rate parameters are not very small. Based on the power evaluation, the tests show the ability to detect significant differential regions with quite small observed difference in rate parameters. Based on false-positive and power evaluations, it can be said that ET seems to be the most accurate test, then WT and LRT

in order.

# Chapter 6

# Comparison study

## 6.1 Introduction

In this chapter, we perform a comparison study between the methods of differential binding sites analysis using ChIP-Seq data, which have been discussed in Chapters 4 and 5. These methods are MACS and MAnorm, diffReps, exact test (ET), Wald test (WT) and likelihood ratio test (LRT).

The rest of this chapter is constructed as follows. In Section 6.2, we perform comparisons based on simulated data. We perform comparisons based on real ENCODE data in Section 6.3. We perform a comparison based on RUNX1/ETO in Section 6.4. In Section 6.4 we also perform a comparison based on the gene expression result of RUNX1/ETO. Finally, in Section 6.5 we discuss the findings.

## 6.2 Comparison based on simulated data

In this section, two types of comparisons are performed. First, comparing the ability to control the false-positive rate. Second, comparing the power of the methods under

consideration. We use the same simulated data to evaluate the false-positive rate and power of MACS and MAnorm, diffReps, ET, WT and LRT in 4.3.1 and 5.4.1. Hence, we need to combine the findings and analyse them.

## False-positive rate evaluation

Here is a reminder of how we simulate the data that are used in the false-positive rate evaluation, which is first mentioned in Section 4.3.1.

1. Simulate two samples each of $10,000$ windows (samples) of size $200$ bp from Poi($\lambda$). $\lambda$ ranges from $0.01$ to $0.5$ with step $0.02$. The simulation is made for each value in $\lambda$. By simulating in this setting, we simulate from the null hypothesis, which assumes no difference between the two samples.

2. For each proposed method, calculate the proportion of significant windows out of $10,000$ for each method, where any window with p-value $< 0.05$ is considered significant.

3. Repeat the above steps $50$ times, and consider the average of them as the final results of the false-positive rate of each method.

In Figure 6.1, we represent the false-positive rate evaluation of MACS and MAnorm jointly, MAnorm only, diffReps, ET, WT and LRT. It can be seen in the figure that ET, WT and LRT first start to control the false-positive rate at the $5\%$ level of significance compared to the other methods. It can also be noticed that all methods seem to be stable in controlling the false-positive $\lambda > 0.2$. Note that diffReps is adjusted in the figure by using Equation (4.4), hence we have two lines, before and after the adjustment.

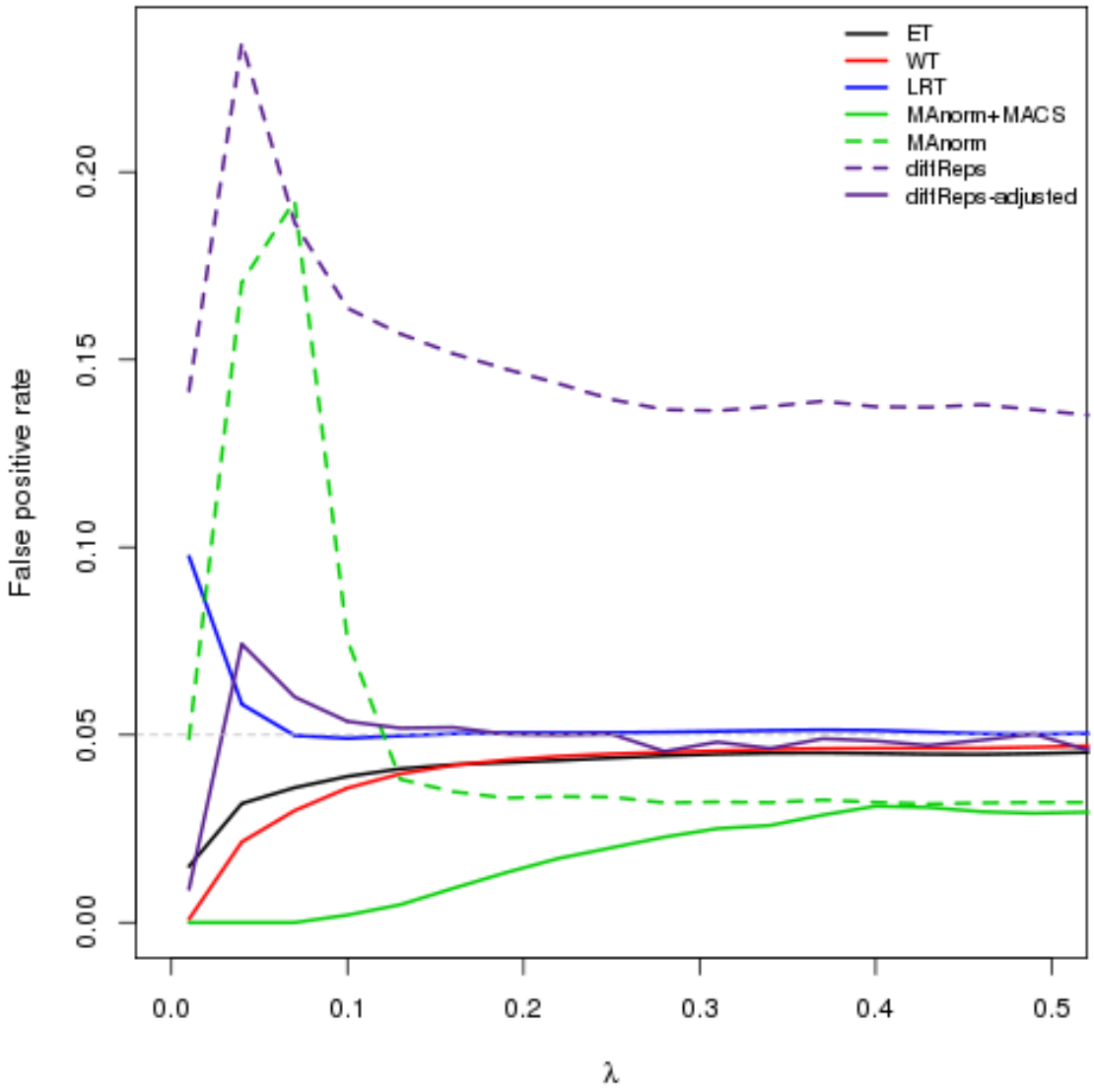



Figure 6.1: The evaluation of the false-positive rate by using MACS and MAnorm jointly, MAnorm only, diffReps, ET, WT and LRT based on simulated data. In the figure, the horizontal axis represents the $\lambda$ used in the simulation and the vertical axis represents the average of 50 false- positive rates for each of the $\lambda$'s.

## Power evaluation

The following is a reminder of how we constructed the simulation to evaluate the power, which is first mentioned in Section 4.3.1.

1. Simulate two samples each of $10,000$ windows of size $200$ bp from Poi$(\lambda_1 = 0.01)$. $\lambda_1$ here represents the low rate read counts, which is most of the genome's regions.

2. Simulate $2000$ windows out of the $10,000$ in both samples at the same locations with Poi$(\lambda_2 = 0.2)$

3. Simulate another $1000$ windows at the same location in each sample by using Poisson with fixed rate $\lambda_2 = 0.02$ and rate $\lambda_3$ ranges from $0.22$ to $1$ by step $0.02$. To keep the difference existing after the normalising process of the methods, in the first sample we simulate the first $500$ windows by using $\lambda_2$ and the second $500$ windows by using $\lambda_3$, and in the second sample the first $500$ from $\lambda_3$ and the second $500$ from $\lambda_2$.

4. Apply the analysis methods, and calculate the proportion of detected significant windows out of $1000$ windows for each of the analysis methods.

5. Repeat the above steps $50$ times, and consider the average of them as the final results of the power for each of the analysis methods.

Note that the choice of $\lambda_2 = 0.2$ in the above simulation is based on the false-positive rate result. That is, to have fair power comparison between the methods, the methods must behave equally (or at least closely) at the desired level of significance, $5\%$, in terms of false-positive rate. If a method exhibited a high false- positive rate, then it would show higher power compared to methods that control the rate at the right level. On the other hand, if a method exhibited a low false- positive rate, then it would show less power compared to other methods that control the rate at the right level. In Figure 6.1, it can

be seen that for $\lambda_2 \geq 0.2$ the methods start to show stable behaviour of controlling the false-positive rate, and their performances are quite close to each other. Hence, we use $\lambda_2 = 0.2$ in the power simulation.

In Figure 6.2, we represent the power of MACS and MAnorm jointly, MAnorm only, diffReps, ET, WT and LRT based on simulated data, as shown above. In the figure, it can be seen that MAnorm itself is the first to detect the significant differences. After MAnorm, we have in order LRT, WT, ET, MACS and MAnorm, and diffReps (after the adjustment). However, LRT, WT and ET reach maximum power first, then MAnorm, diffReps after adjustment, and then MACS and MAnorm.

## 6.3 Comparison based on real data

In this section, we perform two types of comparisons using real ChIP-Seq data from the ENCODE project. First, comparing the control of the false-positive rate, and second, comparing the power. We use the same ENCODE data in the evaluation of the false-positive rate and power in Sections 4.3.2 and 5.4.2. Hence, we need to combine and analyse the findings of Sections 4.3.2 and 5.4.2.

To perform the evaluations, we have three ChIP-Seq experiments from the ENCODE project, each of which has two biological replicates. The experiments are:

1. Cell line Gm12878 with transcription factor ATF2;
2. Cell line Gm12878 with transcription factor BCLAF1;
3. Cell line H1-hESC with transcription factor ATF2.

We combine the first replicate from each experiment, as well as the second replicate. Hence, we end up with one ChIP-Seq sample with two biological replicates. The purpose

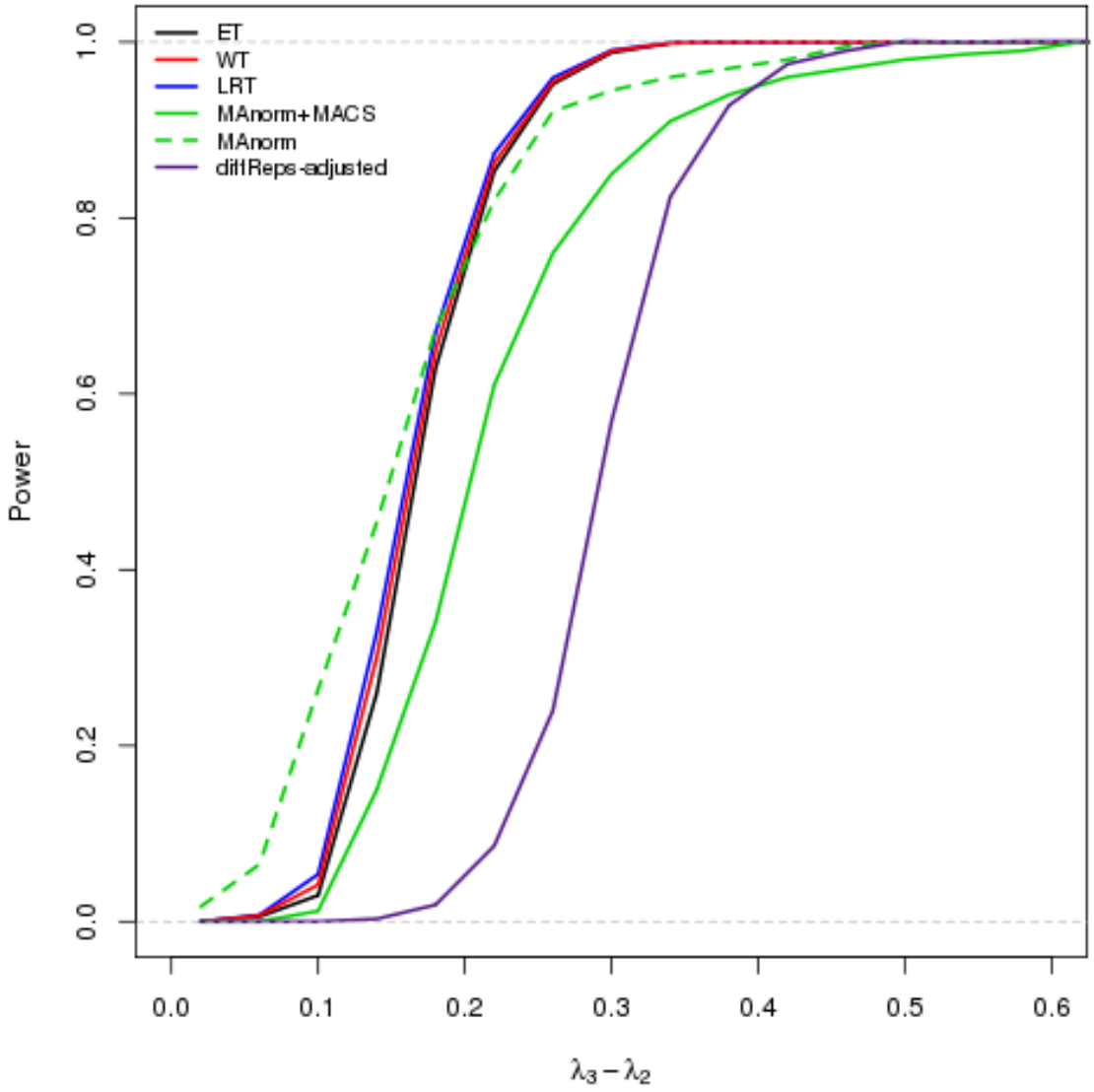



Figure 6.2: The evaluation of the power of MACS and MAnorm jointly, MAnorm only, diffReps, ET, WT and LRT based on simulated data. In the figure, the horizontal axis represents the difference in rates between the two simulated data sets, where we enrich the windows with different rates, and the vertical axis represents the average power of 50 simulations for each difference $\lambda_3 - \lambda_2$.

of doing that is to increase the observed read counts rate, and hence we would have better performance for the analysis methods (full details are given in Section 4.3.2).

## False-positive rate evaluation

Here is a reminder of how we evaluate the false-positive rate based on the real data. We compare the replicates together, where ideally they would not show significant differences. We use the first part of chromosomes 1 to 10, separately (first part means the region before the centromere). Therefore, we would have 10 values of false-positive rate for each of the analysis methods.

In Figure 6.3, we the show the results of false-positive evaluation by using ET, WT, LRT, MACS and MAnorm, MAnorm only and diffReps. From the figure it can be said that ET is the best compared to the other methods in terms of controlling the false positive rate. It can also be seen in the figure that MACS and MAnorm, and MAnorm only exhibit high false-positive rates, and they are adjusted by using Equation (4.4). Moreover, WT shows the lowest false-positive rate compared to the others.

## Power evaluation

Here is a brief reminder of how we evaluate the power based on real data. We consider only the first part of chromosome 10. The chromosome is divided into fixed and non-overlapping windows of size 200 bp. We then consider two copies of that chromosome, say test and control. Hence, the two copies are identical at the first time. Now, the power is evaluated by comparing the test to the control sample, but by shifting one window in the test sample at a time (full details are provided in Section 4.3.2).

Figure 6.4 shows the power evaluation of ET, WT, LRT, MACS and MAnorm, and MAnorm only by using the first part of chromosome 10 as described above. In the figure,

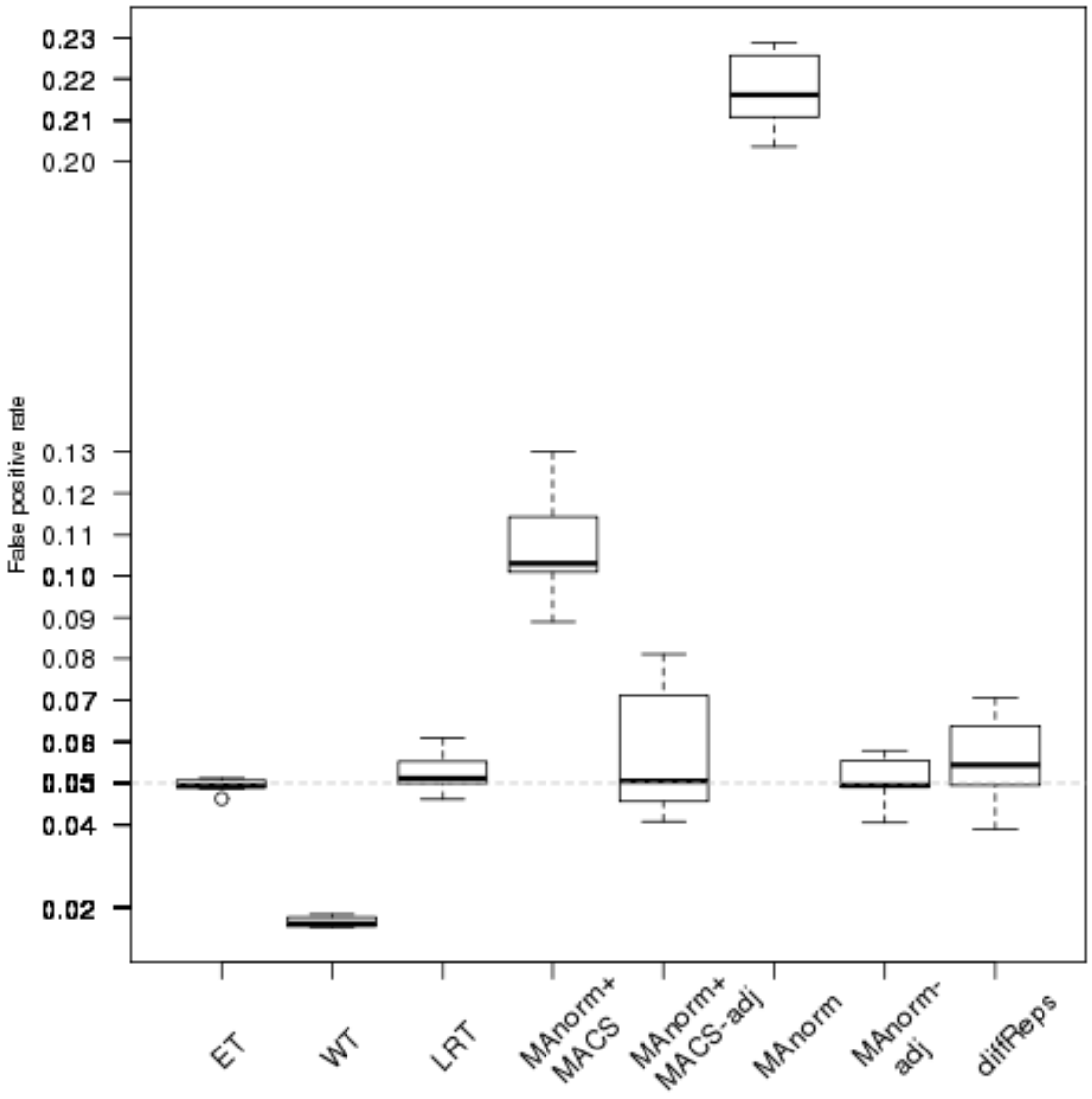



Figure 6.3: The evaluation of the false-positive rate by using MACS and MAnorm jointly, MAnorm only, diffReps, ET, WT and LRT based on real data. In the figure, the horizontal axis represents the analysis methods and the vertical axis represents the false-positive rates. Each point in the plot represents the false-positive rate by using a subsample that is a part of a chromosome, where we use the first part from chromosomes 1 to 10.

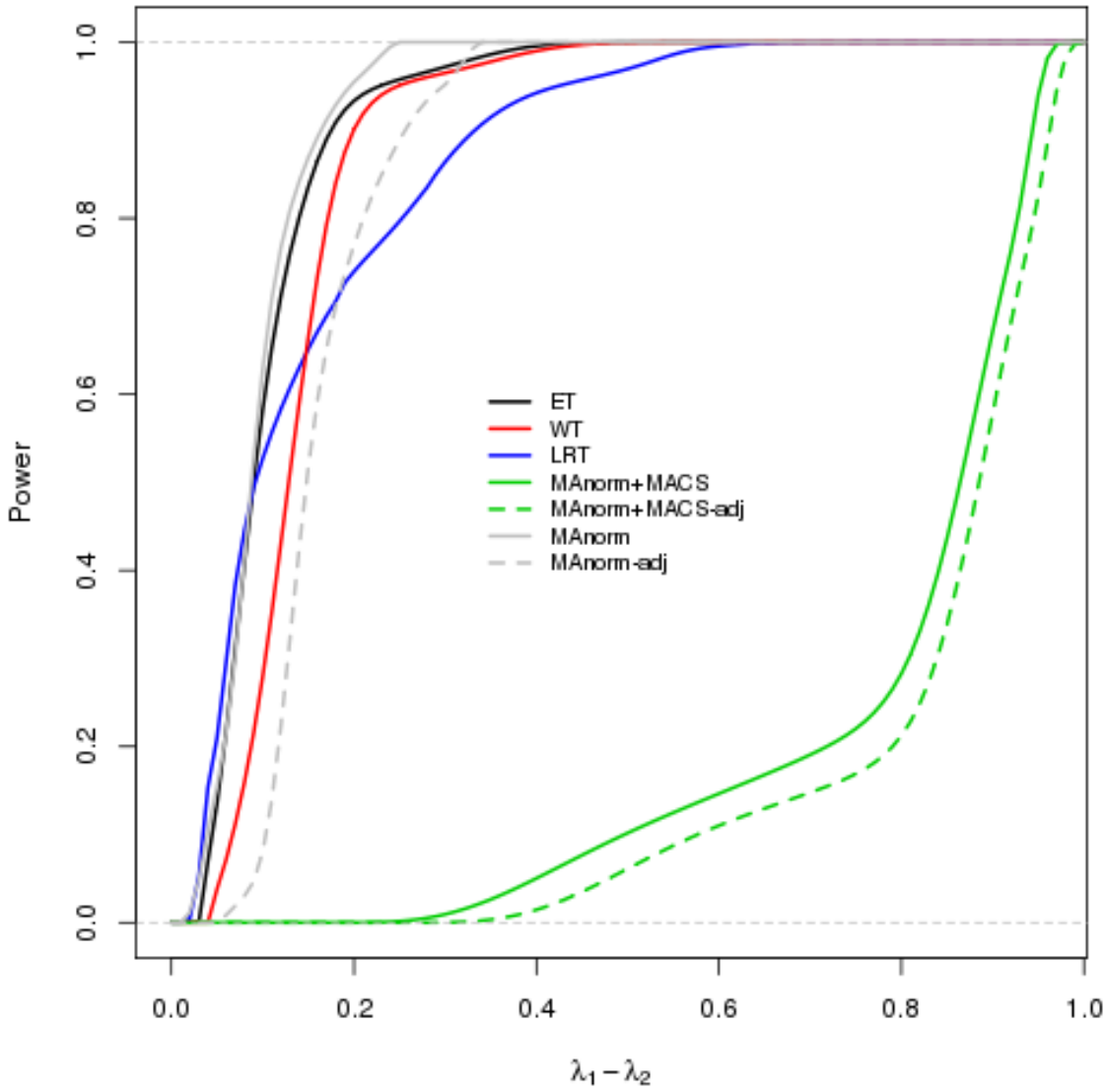



Figure 6.4: The evaluation of the power of ET, WT, LRT, MACS and MAnorm, and MAnorm only based on real data. In the figure, the horizontal axis represents the observed difference in rates between the two samples and the vertical axis represents the average power by using ENCODE data.

it can be seen that unadjusted MAnorm is the first method that starts to detect differences, then LRT, ET and WT, in order. However, after adjusting MAnorm (by using Equation (4.4)), we can see that MAnorm is shifted and shows lower power compared to LRT, ET and WT. On the other hand, MACS and MAnorm power before and after adjustment is far away from the others. It can be seen that MACS and MAnorm start detecting differences around $0.3$, whereas other methods start detecting differences lower than $0.1$. In addition, we can see that ET, WT, LRT and MAnorm reach the maximum power at differences less than $0.6$, while MACS and MAnorm reaches its maximum power at a difference of around $1$.

From Figure 6.4 it can be said that ET, WT and LRT perform better than MACS and MAnorm and MAnorm only. However, it is not easy to declare which of them is the most powerful method. That is, LRT is the first to start detecting differences compared to ET and WT. On the other hand, ET and WT are the first and the second, respectively, to reach full power. However, by recalling the false-positive rate results of ET, WT and LRT, we may say that ET is the most powerful method, and WT is the second most powerful, then LRT. That is because LRT shows a higher false-positive rate compared to ET and WT. Note that diffReps is excluded from the power evaluation as the setting of the evaluation does not work with the working process of diffReps (full details are provided in Section 4.3.2).

## 6.4 Comparison based on RUNX1/ETO

In this section, we perform three related comparisons. First, to compare the significant regions that are identified by ET, WT, LRT, MACS and MAnorm, and diffReps after analysing RUNX1/ETO data. Second, to find and compare the nearest gene to the significant regions that are identified by ET, WT, LRT, MACS and MAnorm. Third, to compare the found gene, which is in the second comparison, with a previous analysis that

has been done for gene expressions of RUNX1/ETO knockdown experiment by [37].

## Significant regions

Recall the setting we use to analyse the data. In RUNX1/ETO data, we find that a window of size $200$ bp is optimal to represent the variability in the data in the test and control samples. Therefore, we divide the data of RUNX1/ETO into non-overlapping windows of fixed size of $200$ bp. Within windows we consider bins of size $1$ bp, i.e., we count the reads at each genomic position. Then, we analyse the data using ET, WT, LRT, MACS and MAnorm, and diffReps. In ET, WT, LRT and diffReps, we use the optimal window size but a shifting process is considered in diffReps with a shift size of $100$ bp. In MACS and MAnorm, the minimum window size in MACS is $300$ bp, and hence we use that minimum window size.

In Figure 6.5 we show four Venn diagrams, which represent the results of analysing RUNX1/ETO by using ET, WT, LRT, MACS and MAnorm, and diffReps. In the Venn diagrams, we show the result of each of the methods, as well as the intersections between them. In panel (a), we represent the results of ET, WT and LRT, which are already shown in Figure 5.6, in terms of genomic regions that may correspond to more than one window. In panel (b), we represent the results of ET, WT, LRT, and MACS and MAnorm, and the intersections between them. In that panel, we can see that most of the regions detected by MACS and MAnorm are intersecting with the detected regions of ET, WT and LRT. Specifically, out of the $509$ significant regions identified by MACS and MAnorm, there are $420$ significant regions that are overlapped with $371$ significant regions identified by all ET, WT and LRT. It can be seen that most of the overlapping regions are in the regions detected by ET, which are detected by WT and LRT as well.

In Figure 6.5, panel (c) represents the results of ET, WT, LRT and diffReps, and the intersections between them. In the panel, it can be seen that there are three regions

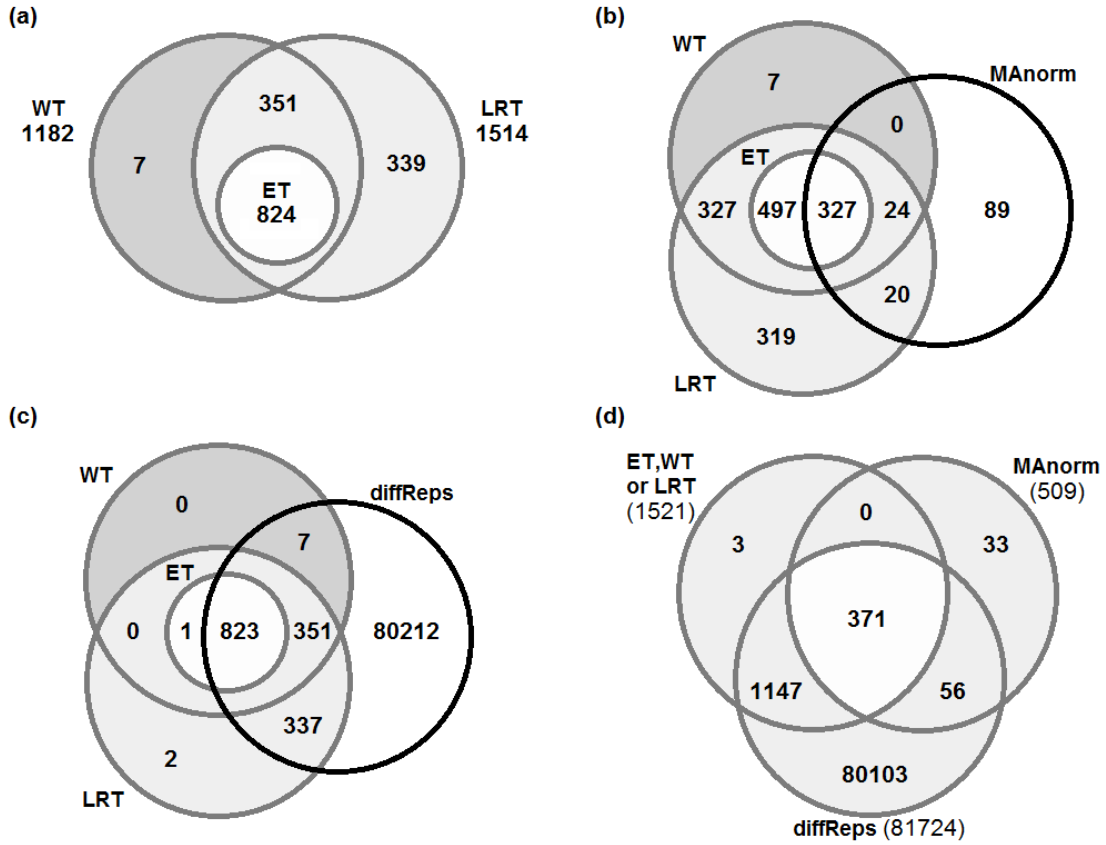



Figure 6.5: Result of RUNX1/ETO analysis by using ET, WT, LRT, MACS and MAnorm, and diffReps methods. The numbers in the Venn diagrams represent the number of the significant genomic regions, which may correspond to more than one genomic window. Panel (a) represents the results of ET, WT and LRT. Panel (b) represents the results of ET, WT, LRT, and MACS and MAnorm, and the intersections between them. Panel (c) represents the results of ET, WT, LRT and diffReps, and the intersections between them. Panel (d) represents the intersections between the union of the proposed tests (ET, WT and LRT), MACS and MAnorm, and diffReps.

identified by either ET, WT or LRT that do not overlap with the regions identified by diffReps. One of these three regions is significant in the three tests (ET, WT and LRT), whereas two regions are significant in LRT only. In panel (d) of the same figure, we represent the intersections between the results of the union of the proposed tests (ET, WT and LRT), MACS and MAnorm, and diffReps. From the panel it can be noticed that most of the significant regions identified by ET, WT, LRT, or MACS and MAnorm overlap with significant regions identified by diffReps. It can be seen that there are $56$ significant regions detected by MACS and MAnorm that overlap with diffReps' regions. These $56$ regions by MACS and MAnorm correspond to $99$ regions in diffReps' findings. This means MACS's and MAnorm's regions are wider than diffReps' regions.

## Significant regions and genes

After identifying the significant regions, the next step is to look for the nearest genes to these regions. We consider the regions that are identified by ET, WT or LRT, and MACS and MAnorm. We do not consider diffReps' regions as it detected a huge number of regions in addition to most of the regions detected by the other methods. Hence, we consider the other methods' regions, which can be described as subset of diffReps' regions, and discard diffReps' unique regions, as we suspect high false-positive results from diffReps.

To identify the nearest genes to the detected regions, we can use annotating software. These software only need information of chromosome and starting and ending positions of the regions of interest to work. There are many available annotating software packages. We use Homer software [54]. The regions we want to look at are the union of ET, WT and LRT, which are $1521$ regions. In addition, we want to look at the identified regions by MACS and MAnorm, which are $509$ regions. Hence we use Homer software with these regions and show the results in Table 6.1.

| | ET, WT or LRT | MACS and MAnorm | Overlaps |
|---|---|---|---|
| Number of genes | 1289 | 406 | 351 |
| Within 20 kbp | 514 | 174 | 132 |

Table 6.1: Number of genes that are mapped closest to the significant regions identified by ET, WT, LRT, and MACS and MAnorm. The first row describes the number of genes, and the second row describes the number of genes located within 20,000 bp from the significant regions.

In Table 6.1 it can be seen that the numbers of genes are much lower than the numbers of significant regions. That is, the union of ET, WT and LRT is $1,521$ regions, whereas the number of closest genes is $1,289$. Furthermore, there were $509$ regions by MACS and MAnorm, while the number of closest genes to these regions is $406$. The reason is that it is observed that we have more than one significant region that corresponds to one gene, which is the closest. Overall, $1,344$ genes are found closest to the identified regions by either the union of ET, WT and LRT, or MACS and MAnorm. Moreover, out of these $1,344$ genes, there $351$ genes that are found by all methods, ET, WT, LRT, and MACS and MAnorm.

The distances between the significant regions and the closest genes range from $13$ bp to $300$ kbp, with median of $19,950$ bp. However, this range of distance is quite wide. Biologically, a gene would be of interest if it fell within $20$ kbp distance of the identified significant regions. Hence, we count the number of genes that fall within the distance of $20$ kbp, and show the results in Table 6.1 in the second row. It can be noticed that the number of genes that fall within $20$ kbp distance is around part of the full number of genes. This can be related to the median, as mentioned before. Within $20$ kbp distance, there are $556$ genes found that are closest to the identified regions by either ET, WT, LRT, or MACS and MAnorm. Out of these genes, $132$ genes are found by all methods.

### Nearest genes and genes' expression

Gene expression study has been done for RUNX1/ETO data [37]. In the study, The fold change of genes is considered, where genes that show more than two-fold changes are reported. The fold change here is $\log_2(x_i/y_i)$, where $x_i$ and $y_i$ are read counts of gene $i$ in the test and control samples, respectively. Hence, we compare these reported genes to the genes we identify closest to the significant regions and show the result in Table 6.2.

| | ET, WT or LRT | MACS and MAnorm | Overlaps |
|---|---|---|---|
| Number of genes | 124 | 37 | 30 |
| Within 20 kbp | 68 | 24 | 20 |

Table 6.2: Number of genes from Table 6.1 that show more than two-fold changes in gene expression.

In Table 6.2 it can be seen that there are $131$ genes that show more than two-fold changes in gene expression by either ET, WT, LRT, or MACS and MAnorm. Out of these $131$ genes, there are $30$ genes identified by all methods. In addition, out of the $131$ genes, there are $72$ genes located within a distance of $20$ kbp from the significant regions. From the $72$ genes, ET, WT, LRT, and MACS and MAnorm methods have $20$ genes in common.

## 6.5 Discussion and conclusion

We have seen that the proposed tests (ET, WT and LRT) show good performance compared to the other methods (MACS and MAnorm, and diffReps). The simulation study shows that ET, WT and LRT have the ability to control the false-positive rate accurately at the $5\%$ nominal level. On the other hand, MACS and MAnorm, and diffReps are not able to accurately control the false-positive rate at the desired level. The accuracy in controlling the false-positive rate is extremely critical when the whole

genome is considered, as multiplicity problems occur. Furthermore, the simulation study shows that ET, WT and LRT are more powerful compared to MACS and MAnorm, and diffReps. In the evaluation, we use MAnorm only to evaluate its performance by giving it the exact regions of interest to test. Although MAnorm behaves better without MACS, in reality what we did cannot be applicable. That is, MAnorm is a peak-finding approach and would not work if the peak regions were not given as inputs.

The simulation findings are reflected in the analysis of RUNX1/ETO data. That is, diffReps shows a high false-positive rate in the simulation study, and hence detects many more significant regions compared to the other methods. However, in the simulation study we are able to adjust the p-values, but in RUNX1/ETO data we cannot adjust them. If there was more than one replicate of test and control samples, then we would be able to adjust the p-values.

Apart from diffReps results, the proposed tests (ET, WT and LRT) detect more significant regions compared to MACS and MAnorm. Moreover, more than three quarters of the significant regions detected by MACS and MAnorm are also detected by either ET, WT or LRT. As a result, the proposed tests identify more genes that are differentially regulated owing to knocking-down RUNX1/ETO.

In the expression study of RUNX1/ETO data, genes that show signs of significant differential expression are provided. In other words, no testing is performed to check the significance of changes in gene expression. The study reported the genes that show fold changes of more than two. Although these are only observed changes (not tested), they can be employed as support to the identified differentially regulated genes by the methods. On the other hand, the identified genes can be used to support the report in terms of differential expression. However, from a biological point of view the relationship between gene expression and gene binding sites has not been fully understood.

The identified differentially regulated genes by the proposed tests are overlapped more with the genes reported from the gene expression study compared to the genes identified

by MACS and MAnorm. One can say that the ratios of overlapping genes are very close in the proposed tests and MACS and MAnorm. That is, from the proposed tests there are $124$ reported genes out of $1289$ identified genes, and this represents $9.6\%$. With MACS and MAnorm, there are $37$ reported genes out of $406$ identified genes, and this represents $9.1\%$. We say that it is not about the ratio only. It is about, first, the information provided, so more matching between the two studies, which are gene expression and gene binding sites, can lead to better understanding of the biology behind. Second, it is about the accuracy of the performance, so identifying the significant differences at the desired level of significance, i.e., if it was about the number of identified genes, then diffReps would stand out. Finally, it is also about the clarity and objectivity of the analysing methods.

In conclusion, it can be said that the proposed tests, which are ET, WT and LRT, perform better than MACS and MAnorm, and diffReps. The tests show good accuracy in controlling the false-positive rate, and are more powerful compared to the other methods. An obvious advantage of the proposed tests is that the tests do not depend on a subjective definition of peaks, as in MACS and MAnorm, or definition of enrichment regions, as in diffReps. where it filters out regions with low read counts.

# Chapter 7

# Conclusion and further work

## 7.1 Introduction and summary

In this chapter we, first, summarise the main ideas and findings that have been shown in the previous chapters. Second, we provide our final conclusion. The rest of the chapter is introducing some ideas for further work.

### Summary

The objective of our research is to establish or develop statistical methods that are able to detect differential binding sites using high-throughput ChIP-Seq data. There are already established approaches for differential binding site analysis, and they can be classified into two classes. First, peak-finding based and, second, non-peak-finding based approaches. Approaches from both classes suffer from major drawbacks. First, they suffer from a subjective definition in defining peaks or enrichment regions, which are regions of high densities of read counts. In peak-finding-based approaches, by using different peak-finding algorithms, we often end up with different results. Also non-peak- finding based approaches do initial filtering step to filter out genomic regions with low density (low

reads counts), and this can be considered as another way of searching for enrichment regions (peaks). Second, both classes of approach suffer from high false-positive rates. Third, in peak-finding-based approaches, the underlying assumptions and models are not clear. This lack of clarity makes the assessment of these approaches problematic.

We address our objective by achieving the following points.

- First, we introduce a novel method based on histogram construction to optimise the presentation of ChIP-Seq data by optimising the window size of the histogram. This enables us to visualise ChIP-Seq data and any counts data in their optimal variabilities. Hence, we are able to see the changes that are occurring in a ChIP-Seq experiment across the whole genome.

- Second, we start by investigating the pattern of RUNX1/ETO data, which is a ChIP-Seq dataset. We find some clear features in the data, such as lengthy gaps of zero counts. This feature is common in ChIP-Seq data, but has different distributions in different ChIP-Seq datasets. Employing this feature in the proposed models for the count data improves the models.

- Third, we start to characterise the ChIP-Seq data within windows. The data are discrete read counts. We assume that read counts within a window are independent and identically distributed as a Poisson random variable with a single unknown rate parameter. We also assume that windows are independent within the same data. These assumptions allow the rate of the read counts to be able to fluctuate between windows.

- Fourth, for two given ChIP-Seq datasets under different biological conditions, we can detect differential binding sites, or differential regions, by comparing windows at the same genomic locations in the two datasets. We assume that the read counts in the two datasets are paired. Hence, the difference in read counts between the datasets is considered in the analysis of differential binding sites. The differences

in read counts within a window are modelled by the Poisson Difference distribution, as we deal with the difference between two Poisson variables.

- Fifth, by using the difference data, we propose three parametric Tests, which are the Exact Test (ET), Wald Test (WT) and Likelihood Ratio Test (LRT), to address our objective, which is to detect differential binding sites.

The proposed tests (ET, WT and LRT) have been evaluated by using simulated and real data. In addition, the tests have been compared to MAnorm and diffReps, which are peak-finding-based and non-peak-finding-based methods, respectively. With MAnorm, we use MACS, which is a peak- finding algorithm. The proposed tests show their ability to accurately control the false- positive rate compared to the other methods. The evaluation study also shows that the proposed tests are more powerful compared to MAnorm based on MACS, and diffReps methods.

## 7.2 Conclusion

We developed a novel objective method to optimise the window size in genetic data, as well as in non-genetic data. By using the method, a researcher can divide the data into windows that show the data in its optimal variation. In addition, we developed three novel tests to detect differential binding sites of transcription factors between two ChIP-Seq datasets. A clear assumption is made on the read counts, which is that they follow the Poisson distribution. Detecting differences between the two datasets is done by considering the difference in read counts, which follows Poisson Difference distribution. Based on simulated and real data, the tests are accurate and powerful compared to some current methods. The proposed tests are not applicable exclusively to detect differential binding sites in ChIP-Seq data. However, they can be applied to cases where a researcher wants to investigate differences between counts of two paired datasets.

## 7.3 Further work

### R package

We have written an R package to perform the proposed testing methods ET, WT and LRT. For two given ChIP-Seq datasets and window sizes, the package does the analysis and returns the resulting p-values for each of the windows across the whole genome. The package can also do all proposed tests or some of them, depending of the user choice.

### Mixture model

We have shown that a mixture model of three Poisson components is able to simulate the observed behaviour of RUNX1/ETO data in terms of lengths of consecutive zeros and correlation. However, we were not able to obtain the maximum likelihood estimates of the parameters in the model as the behaviour of the likelihood function was not regular.

An idea is that instead of looking for a mixture model for each of the samples, we could consider the difference between the samples. Hence, we could look for a mixture model of Poisson Difference components.

### Biological replicates

It is quite common to have several biological replicates for the same ChIP-Seq experiment. That is, instead of having single samples of test and control, we can have test-1, test-2, control-1 and control-2, or even more. From a biological point of view, having biological replicates can help in controlling the systematic variation, which is caused by machines, and biological variation.

Currently, the proposed tests are designed for only one replicate of each of the conditions. To our knowledge, none of the current approaches that are designed to analyse differential binding sites are designed to consider more than one replicate under each condition. They instead borrow some algorithms that were originally proposed and designed to analyse gene expression when the experiment includes biological replicates, except DiffBind [60]. For instance, DESeq [5] and EdgeR [35], which were originally designed for gene expression analysis. However, the analysis of differential binding sites is different from the analysis of gene expression. That is, the data in gene expression is a single number for each gene, and that single number represents the observed read count of that gene. This number of that gene is compared to the number of the same gene in the other sample. On the other hand, the data in differential binding sites analysis is read counts within windows, and the read counts are compared in the same windows in the two samples. Hence by using gene expression methods to analyse differential binding sites, the windows are considered as genes, and then represented as a single number, which is not correct.

Hence, including biological replicates in the analysis of the proposed tests is still an open challenge.

# Bibliography

[1] Abramowitz, M. and Stegun, I., *Handbook of mathematical functions*, vol. 1, Dover New York, 1972.

[2] Akaike, H., *Factor analysis and AIC*, Psychometrika **52** (1987), no. 3, 317–332.

[3] Akaike, H., *Information theory and an extension of the maximum likelihood principle*, Breakthroughs in Statistics (1992), 610–624.

[4] Alzaid, A. and Omair, M., *On the Poisson difference distribution inference and applications*, Bulletin of the Malaysian Mathematical Sciences Society **8** (2010), no. 33, 17–45.

[5] Anders, S., *Analysing RNA-Seq data with the DESeq package*, Mol Biol (2010), 1–17.

[6] Audic, S. and Claverie, J., *The significance of digital gene expression profiles*, Genome Research **7** (1997), no. 10, 986–995.

[7] de-Boer, B., van-Duijvenboden, K., van-den-Boogaard, M., Christoffels, V., Barnett, P. and Ruijter, J., *OccuPeak: ChIP-Seq Peak Calling Based on Internal Background Modelling*, PloS one **9** (2014), no. 6, e99844.

[8] Benjamini, Y. and Hochberg, Y., *Controlling the false discovery rate: a practical and powerful approach to multiple testing*, Journal of the Royal Statistical Society: Series B (Methodological) (1995), 289–300.

[9] Brent, R., *Algorithms for minimization without derivatives*, Courier Corporation, 2013.

[10] Browne, M., *Cross-validation methods*, Journal of Mathematical Psychology **44** (2000), no. 1, 108–132.

[11] Dudoit, S., Yang, Y., Callow, M. and Speed, T., *Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments*, Statistica Sinica (2002), 111–139.

[12] ENCODE Project Consortium et al., *The ENCODE (ENCyclopedia Of DNA Elements) project*, Science **306** (2004), no. 5696, 636–640.

[13] KKruczyk, M., Umer, H., Enroth, S. and Komorowski, J., *Peak Finder Metaserver-a novel application for finding peaks in ChIP-seq data*, BMC Bioinformatics **14** (2013), no. 1, 280.

[14] Shen, L., Shao, N., Liu, X., Maze, I., Feng, J. and Nestler, E., *diffreps: detecting differential chromatin modification sites from chip-seq data with biological replicates*, PLoS ONE **8** (2013), no. 6, e65598.

[15] Fernández, C. and Green, P., *Modelling spatially correlated data via mixtures: a Bayesian approach*, Journal of the Royal Statistical Society: Series B (Statistical Methodology) **64** (2002), no. 4, 805–826.

[16] de-Magalhães, J., Finch, C. and Janssens, G, *Next-generation sequencing in aging research: emerging applications, problems, pitfalls and possible solutions*, Ageing research reviews **9** (2010), no. 3, 315–323.

[17] Fletcher, R. and Reeves, C., *Function minimization by conjugate gradients*, The Computer Journal **7** (1964), no. 2, 149–154.

[18] Furey, T., *hIP–seq and beyond: new and improved methodologies to detect and characterize protein–DNA interactions*, Nature Reviews Genetics **13** (2012), no. 12, 840–852.

[19] Hartzog, G. and Winston, F., *Nucleosomes and transcription: recent lessons from genetics*, Current Opinion in Genetics & Development **7** (1997), no. 2, 192–198.

[20] Ibrahim, M. and Lacadie, S., *JAMM: a peak finder for joint analysis of NGS replicates*, Bioinformatics **31** (2015), no. 1, 48–55.

[21] Irwin, J., *The frequency distribution of the difference between two independent variates following the same Poisson distribution*, Journal of the Royal Statistical Society (1937), 415–416.

[22] Karlis, D. and Ntzoufras, I., *Distributions based on Poisson differences with applications in sports*, Tech. report, Technical Report 101, Department of Statistics, Athens University of Economics, 2000.

[23] Karlis, D. and Ntzoufras, I., *Bayesian analysis of the differences of count data*, Statistics in Medicine **25** (2006), no. 11, 1885–1905.

[24] Xi, R., Kim, T. and Park, P., *Detecting structural variations in the human genome using next generation sequencing*, Briefings in Functional Genomics (2011), elq025.

[25] Asp, P., Blum, R., Vethantham, V., Parisi, F., Micsinai, M., Cheng, J., Bowman, C., Kluger, Y. and Dynlacht, B., *Genome-wide remodeling of the epigenetic landscape during myogenic differentiation*, Proceedings of the National Academy of Sciences **108** (2011), no. 22, E149–E158.

[26] Mikkelsen, T., Ku, M., Jaffe, D., Issac, B., Lieberman, E., Giannoukos, G., Alvarez, P., Brockman, W., Kim, T., Koche, R. and Lee, W., *Genome-wide maps of chromatin state in pluripotent and lineage-committed cells*, Nature **448** (2007), no. 7153, 553–560.

[27] Kuha, J., *AIC and BIC comparisons of assumptions and performance*, Sociological Methods & Research **33** (2004), no. 2, 188–229.

[28] Latchman, D., *Transcription factors: an overview*, The International Journal of Biochemistry & Cell Biology **29** (1997), no. 12, 1305–1312.

[29] Leleu, M., Lefebvre, G. and Rougemont, J., *Processing and analyzing ChIP-seq data: from short reads to regulatory interactions*, Briefings in Functional Genomics (2010), elq022.

[30] Li, H. and Durbin, R., *Fast and accurate short read alignment with Burrows–Wheeler transform*, Bioinformatics **25** (2009), no. 14, 1754–1760.

[31] Zhang, Y., Liu, T., Meyer, C., Eeckhoute, J., Johnson, D., Bernstein, B., Nusbaum, C., Myers, R., Brown, M., Li, W. and Liu, X., *Model-based analysis of ChIP-Seq (MACS)*, Genome Biol **9** (2008), no. 9, R137.

[32] Xu, H., Wei, C., Lin, F. and Sung, W., *An HMM approach to genome-wide identification of differential histone modification sites from ChIP-seq data*, Bioinformatics **24** (2008), no. 20, 2344–2349.

[33] Zhu, C.,Byrd, R., Lu, P. and Nocedal, J., *Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization*, ACM Transactions on Mathematical Software (TOMS) **23** (1997), no. 4, 550–560.

[34] Cairns, J., Spyrou, C., Stark, R., Smith, M., Lynch, A. and Tavaré, S., *BayesPeak: R package for analysing ChIP-seq data*, Bioinformatics **27** (2011), no. 5, 713–714.

[35] Robinson, M., McCarthy, D. and Smyth, G., *edgeR: a Bioconductor package for differential expression analysis of digital gene expression data*, Bioinformatics **26** (2010), no. 1, 139–140.

[36] McKean, J., *Robust analysis of linear models*, Statistical Science (2004), 562–570.

[37] Ptasinska, A., Assi, S., Mannari, D., James, S., Williamson, D., Dunne, J., Hoogenkamp, M., Wu, M., Care, M., McNeill, H. and Cauchy, P., *Depletion of RUNX1/ETO in t (8; 21) AML cells leads to genome-wide changes in chromatin structure and transcription factor binding*, Leukemia **26** (2012), no. 8, 1829–1841.

[38] Johnson, D., Mortazavi, A., Myers, R. and Wold, B., *Genome-wide mapping of in vivo protein-DNA interactions*, Science **316** (2007), no. 5830, 1497–1502.

[39] Valouev, A., Johnson, David S., Sundquist, A., Medina, C., Anton, E., Batzoglou, S., Myers, R. and Sidow, A., *Genome-wide analysis of transcription factor binding sites based on ChIP-Seq data*, Nature Methods **5** (2008), no. 9, 829–834.

[40] Nelder, J. and Mead, R., *A simplex method for function minimization*, The Computer Journal **7** (1965), no. 4, 308–313.

[41] Neyman, J. and Pearson, E., *On the problem of the most efficient tests of statistical hypotheses*, Springer, 1992.

[42] National Institutes of Health, *National Human Genome Research Institute*, 2015.

[43] Shao, Z., Zhang, Y., Yuan, G., Orkin, S. and Waxman, D., *Manorm: a robust model for quantitative comparison of ChIP-Seq data sets*, Genome Biology **13** (2012), no. 3, R16.

[44] Pawitan, Y., *In all likelihood: statistical modelling and inference using likelihood*, Oxford University Press, 2001.

[45] Giles, R., Peters, D. and Breuning, M., *Conjunction dysfunction: CBP/p300 in human disease*, Trends in Genetics **14** (1998), no. 5, 178–183.

[46] Gusnanto, A., Taylor, C., Nafisah, I., Wood, H., Rabbitts, P. and Berri, S., *Estimating optimal window size for analysis of low-coverage next-generation sequence data*, Bioinformatics **30** (2014), no. 13, 1823–1829.

[47] Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K. and Walter, P., *Molecular Biology of the Cell*, Garland Science, 2008.

[48] Castle, J.C., Biery, M., Bouzek, H., Xie, T., Chen, R., Misura, K., Jackson, S., Armour, C., Johnson, J., Rohl, C. and Raymond, C., *DNA copy number, including telomeres and mitochondria, assayed using next-generation sequencing*, BMC Genomics **11** (2010), no. 1, 244.

[49] De-Kok, Y., van-der-Maarel, S., Bitner-Glindzicz, M., Huber, I., Monaco, A., Malcolm, S., Pembrey, M., Ropers, H. and Cremers, F., *Association between X-linked mixed deafness and mutations in the POU domain gene POU3F4*, Science **267** (1995), no. 5198, 685–688.

[50] Scott, D., *On optimal and data-based histograms*, Biometrika **66** (1979), no. 3, 605–610.

[51] Sham, P., *Statistics in Human Genetics*, Arnold London, 1998.

[52] Shanno, D., *Conditioning of quasi-Newton methods for function minimization*, Mathematics of Computation **24** (1970), no. 111, 647–656.

[53] Shendure, J. and Ji, H., *Next-generation DNA sequencing*, Nature Biotechnology **26** (2008), no. 10, 1135–1145.

[54] Heinz, S., Benner, C., Spann, N., Bertolino, E., Lin, Y., Laslo, P., Cheng, J., Murre, C., Singh, H. and Glass, C., *Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and B cell identities*, Molecular Cell **38** (2010), no. 4, 576–589.

[55] Skellam, J., *The frequency distribution of the difference between two Poisson variates belonging to different populations.*, Journal of the Royal Statistical Society: Series A (General), no. Pt 3, 296–296.

[56] Bro, R., Kjeldahl, K., Smilde, A. and Kiers, H., *Cross-validation of component models: a critical look at current methods*, Analytical and Bioanalytical Chemistry **390** (2008), no. 5, 1241–1251.

[57] Bhinge, A., Kim, J., Euskirchen, G., Snyder, M. and Iyer, V., *Mapping the chromosomal targets of STAT1 by Sequence Tag Analysis of Genomic Enrichment (STAGE)*, Genome Research **17** (2007), no. 6, 910–916.

[58] Sokal, R. and Rohlf, F., *Biometry*, NY: WH Freeman & Co (1995).

[59] Soneson, C. and Delorenzi, M., *A comparison of methods for differential expression analysis of RNA-seq data*, BMC Bioinformatics **14** (2013), no. 1, 91.

[60] Stark, R., and Brown, G., *DiffBind: differential binding analysis of ChIP-Seq peak data*, R package version **100** (2011).

[61] Rashid, N., Giresi, P., Ibrahim, J., Sun, W. and Lieb, J., *ZINBA integrates local covariates with DNA-seq data to identify broad and narrow regions of enrichment, even within amplified genomic regions*, Genome Biol **12** (2011), no. 7, R67.

[62] Bailey, T., Krajewski, P., Ladunga, I., Lefebvre, C., Li, Q., Liu, T., Madrigal, P., Taslim, C. and Zhang, J., *Practical guidelines for the comprehensive analysis of ChIP-seq data*, PLoS Comput Biol **9** (2013), no. 11, e1003326.

[63] Taylor, C., *Akaike's information criterion and the histogram*, Biometrika **74** (1987), no. 3, 636–639.

[64] Kharchenko, P., Tolstorukov, M. and Park, P., *Design and analysis of ChIP-seq experiments for DNA-binding proteins*, Nature Biotechnology **26** (2008), no. 12, 1351–1359.

[65] Trapnell, C. and Salzberg, S., *How to map billions of short reads onto genomes*, Nature Biotechnology **27** (2009), no. 5, 455–457.

[66] Villard, J., *Transcription regulation and human diseases*, Swiss Medical Weekly **134** (2004), 571–579.

[67] Wald, A., *Tests of statistical hypotheses concerning several parameters when the number of observations is large*, Transactions of the American Mathematical Society **54** (1943), no. 3, 426–482.

[68] Wilbanks, E. and Facciotti, M., *Evaluation of algorithm performance in ChIP-seq peak detection*, PloS ONE **5** (2010), no. 7, e11471.

[69] Wilks, S., *The large-sample distribution of the likelihood ratio for testing composite hypotheses*, The Annals of Mathematical Statistics **9** (1938), no. 1, 60–62.

[70] Xie, C. and Tammi, M., *CNV-seq, a new method to detect copy number variation using high-throughput sequencing*, BMC Bioinformatics **10** (2009), no. 1, 80.

[71] Yates, F., *Contingency tables involving small numbers and the $\chi$ 2 test*, Supplement to the Journal of the Royal Statistical Society (1934), 217–235.

[72] Yoon, S., Xuan, Z., Makarov, V., Ye, K. and Sebat, J., *Sensitive and accurate detection of copy number variants using read depth of coverage*, Genome Research **19** (2009), no. 9, 1586–1592.

[73] Qin, Z., Yu, J., Shen, J., Maher, C., Hu, M., Kalyana-Sundaram, S., Yu, J. and Chinnaiyan, A., *HPeak: an HMM-based algorithm for defining read-enriched regions in ChIP-Seq data*, BMC Bioinformatics **11** (2010), no. 1, 369.

[74] Yu, S., *Hidden semi-Markov models*, Artificial Intelligence **174** (2010), no. 2, 215–243.

[75] Allhoff, M., Seré, K., Chauvistr, H., Lin, Q., Zenke, M. and Costa, I., *Detecting differential peaks in ChIP-seq signals with ODIN*, Bioinformatics **30** (2014), no. 24, 3467–3475.