A META-ANALYSIS BAYESIAN MODEL FOR CHIP-SEQ DATA

Pablo de Moraes Andrade

THESIS SUBMITTED
TO
INSTITUTE OF MATHEMATICS AND STATISTICS
OF THE
UNIVERSITY OF SÃO PAULO
IN ACCORDANCE WITH THE REQUIREMENTS
FOR THE DEGREE
OF
DOCTOR IN BIOINFORMATICS

Program: Bioinformatics Graduate Program
Advisor: Prof. Dr. Carlos Alberto de Bragança Pereira

The author received financial support from CAPES and FAPESP

São Paulo, December 2016

A META-ANALYSIS BAYESIAN MODEL FOR CHIP-SEQ DATA ANALYSIS

This is the original version of the thesis developed by
the candidate (Pablo de Moraes Andrade), such as
submitted to the Thesis Committee.

Agradecimentos

Agradeço à minha família pelo suporte emocional, sem eles certamente não teria chegado a esse momento. Por me ensinarem a importância da educação; pelo carinho, amor e apoio incondicional em todas as situações.

Aos meus amigos e colegas pelo encorajamento, Danielle Izilda e Davi Toshio que me ajudaram muito durante as fases mais difíceis do doutorado.

Ao meu orientador, Prof. Carlinhos, genuinamente um grande mestre, pelo apoio e ensinamentos. Agradeço por ter acreditado no meu trabalho e me guiar por essa jornada.

Resumo

ANDRADE, P. M. **Modelo Bayesiano de Meta-Análise para dados de ChIP-Seq**. 2016. 107 f. Tese (Doutorado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2016.

Com o desenvolvimento do sequenciamento em larga escala, novas tecnologias surgiram para auxiliar o estudo de sequências de ácidos nucleicos (DNA e cDNA); como consequência, o desenvolvimento de novas ferramentas para analisar o grande volume de dados gerados fez-se necessário. Entre essas novas tecnologias, uma, em particular, chamada *Imunoprecipitação de Cromatina seguida de sequenciamento de DNA em larga escala* ou CHIP-Seq, tem recebido muita atenção nos últimos anos. Este método tornou-se usado amplamente para identificar sítios de ligação de proteínas de interesse no genoma. A análise de dados resultantes de experimentos de ChIP-Seq é desafiadora porque o mapeamento das sequências no genoma apresenta diferentes formas de viés.

Os métodos existentes usados para encontrar picos em dados de ChIP-Seq apresentam limitações relacionadas ao número de amostras de controle e tratamento usadas, e em relação à forma como essas amostras são combinadas. Nessa tese, mostramos que métodos baseados em testes estatísticos de hipótese tendem a encontrar um número muito maior de picos à medida que aumentamos o tamanho da amostra, o que os torna pouco confiáveis para análise de um grande volume de dados.

O presente estudo descreve um método estatístico Bayesiano, que utiliza meta-análise para encontrar sítios de ligação de proteínas de interesse no genoma resultante de experimentos de ChIP-Seq. Esse método foi chamado *Meta-Analysis Bayesian Approach* ou *MABayApp*. Nós mostramos que o nosso método é robusto e pode ser utilizado com diferentes números de amostras de controle e tratamentos, assim como quando comparando amostras provenientes de diferentes tratamentos.

Palavras-chave: ChIP-Seq, Estatística Bayesiana, Meta-Análise.

Abstract

ANDRADE, P. M. **A Meta-Analysis Bayesian Model for ChIP-Seq data**. 2010. 107 p. Thesis (Doctoral) - Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2016.

With the development of high-throughput sequencing, new technologies emerged for the study of nucleic acid sequences (DNA and cDNA) and as a consequence, the necessity for tools to analyse a great volume of data was made necessary. Among these new technologies, one in special *Chromatin Immunoprecipitation followed by massive parallel DNA Sequencing*, or *ChIP-Seq*, has been evidenced during the last years. This method has become a widely used method to identify locations of binding sites for a given protein in the genome. The analysis of data resulting from ChIP-Seq experiments is challenging since it can have different sources of bias during the sequencing and mapping of reads to the genome.

Current methods used to find peaks in this ChIP-Seq have limitations regarding the number of treatment and control samples used and on how these samples should be used together. In this thesis we show that since most of these methods are based on traditional statistical hypothesis tests, by increasing the sample size the number of peaks considered significant changes considerably.

This study describes a Bayesian statistical method using meta-analysis to discover binding sites of a protein of interest based on peaks of reads found in ChIP-Seq data. We call it *Meta-Analysis Bayesian Approach* or *MABayApp*. We show that our method is robust and can be used for different number of control and treatment samples, as well as when comparing samples under different treatments.

Keywords: ChIP-Seq peak calling, Bayesian Model, Meta-Analysis.

Contents

List of abbreviations	ix
List of Symbols	xi
List of Figures	xiii
List of Tables	xv
1 Introduction	1
1.1 Literature review	2
1.2 Objective	3
1.3 Contribution	3
1.4 Document Organization	4
2 Method (Biology)	5
2.1 ChIP-Seq	5
2.1.1 Treatment and Control samples	8
2.1.2 Chip-Seq reads alignment & RNA-Seq data analysis	8
2.2 UCSC genome assembly and annotation	8
3 Statistical background	13
3.1 Definitions and properties of Gamma, Beta and Dirichlet Distributions	13
3.2 Logistic-Normal Distribution	24
3.3 Examples of logistic Normal Approximation	32
4 Model (Statistics)	37
4.1 Categorical Bayesian Model	37
4.1.1 Meta-Analysis	43
5 Workflow (Computer Science)	47
5.1 Algorithm	47
5.2 Peak Smoothing and Identification	48
6 Results and Discussion	53
6.1 MABayApp overview and Model Comparison	54
6.1.1 Bias Correction - Single Treatment Files	54
6.1.2 Meta-analysis – Multiple Treatment Files	55

6.1.3	Model Comparison - Simulation of experiment duplication	56
6.1.4	Model Comparison - Single & Multiple treatment samples	58
6.2	Model Validation	61
6.3	Genome Annotation Analysis	64
7	Conclusion	77
7.1	Future Research	77
A	R Code	79
	Bibliography	89

List of abbreviations

ChIP-Seq	Chromatin immunoprecipitation followed by sequencing
DNA	Deoxyribonucleic acid
RNA	Ribonucleic acid
PCR	Polymerase Chain Reaction
BAC	Bacterial artificial chromosome
IgG	Immunoglobulin G
PolII	RNA polymerase II
bp	Base pair
pdf	Probability distribution function
cdf	Cumulative distribution function
TSS	Transcription Start Site
3UTR	Three prime untranslated region
5UTR	Five prime untranslated region

List of Symbols

Γ	Gamma function
Ψ	Digamma function
Ψ'	Trigamma function
<i>Gamma</i>	Gamma distribution
<i>Beta</i>	Beta distribution
<i>Dir</i>	Dirichlet distribution
$\mathcal{N}$	Normal distribution
$\perp\!\!\!\perp$	Independence
$\sim$	Distributed as
$\dot{\sim}$	Approximate distributed as
$\mathbb{E}$	Expectation
Var	Variance
D_{KL}	Kullback-Leibler Divergence

List of Figures

2.1	ChIP-Seq overview	6
2.2	ChIP-Seq reads aligned to reference genome	7
2.3	Annotation filtering	9
2.4	Karyogram of mouse genome assembly	9
2.5	Genomic features of mouse genome assembly	10
2.6	Lenght distribution of genomic features	11
2.7	Gene type classification	11
3.1	Kullback-Leibler Divergence	30
3.2	Logist-Normal Approximation for different values of α and β	32
3.3	Logist-Normal Approximation for high values of β	33
3.4	Logist-Normal Approximation peak in control sample	34
3.5	Logist-Normal Approximation peak in treatment sample	35
4.1	ChIP-Seq peak alignment example	40
4.2	ChIP-Seq peak coverage example	41
4.3	ChIP-Seq SOX3 peak example	44
4.4	Normal distribution of log-odds	45
4.5	Meta-Analysis	46
5.1	Peak Smoothing	49
5.2	Peak Identification	50
6.1	Chromossomal Bias Correction	55
6.2	Results for Single Treatment Files	56
6.3	Results for Multiple Treatment Files	57
6.4	MABayApp Results for SOX3 ChIP-Seq	57
6.5	MACS Results for experiment duplication	59
6.6	MABayApp Results for experiment duplication	60
6.7	MABayApp vs. MACS for SOX3 ChIP-Seq	61
6.8	MABayApp vs. RNA-Seq Results	63
6.9	MABayApp Results for Genomic Features	64
6.10	Significant regions found by MABayApp	66
6.11	Lenght Distribution of genomic features	67
6.12	Enriched genes according to genomic features	68
6.13	GO: Biological Process (3UTR,5UTR)	71

6.14	GO: Biological Process (TSS)	72
6.15	GO: Cellular Component (3UTR,5UTR)	73
6.16	GO: Cellular Component (TSS)	74
6.17	GO: Molecular Function (3UTR,5UTR)	75
6.18	GO: Molecular Function (TSS)	76

List of Tables

2.1	Gene classification	10
3.1	Kullback-Leibler Divergence	30
6.1	MABayApp Results	55
6.2	MABayApp vs. MACS	58
6.3	Comparison of Peaks called: MABayApp vs. MACS	61
6.4	Comparison of Scores: MABayApp vs. MACS	62
6.5	MABayApp and RNA-Seq SOX Results	65
6.6	Enriched regions of genes from SOX family	69
6.7	List of enriched genes	70

Chapter 1

Introduction

With the development of high-throughput sequencing, new technologies emerged for the study of nucleic acid sequences (DNA and cDNA) and as a consequence, the necessity for tools to analyse a great volume of data was made necessary (Zambelli et al., 2012). Among these new technologies, one in special, described in Park (2009), called Chromatin Immunoprecipitation followed by massive parallel DNA Sequencing, or ChIP-Seq, has been evidenced during the last years.

Chromatin Immunoprecipitation followed by high-throughput Sequencing, or ChIP-Seq, has become a widely used method to map locations of binding sites for a given protein (e.g., transcription factor) in the genome (Jothi et al., 2008). The analysis of data resulting from ChIP-Seq experiments is challenging since it can have different sources of bias during the sequencing and mapping of reads to the genome.

In this method the protein is first linked to the DNA (during a step called *cross-link*), and the genetic material is fragmented. The fragments of DNA that are linked to a protein of interest are then captured using specific antibodies, fragments of a specific length are then sequenced and aligned to the genome. Piles of reads aligned to a specific region of the genome are called *peaks*.

Current methods used to find peaks (Hower et al., 2011; Wu et al., 2015; Zhang et al., 2008) have limitations regarding the number of treatment and control samples used and on how these samples should be used together. The literature review of relevant research is shown in Section 1.1.

For one of the most used methods (the *dominant method* according to the recent reviews Thomas, Thomas, Holloway, and Pollard 2016 and Wilbanks and Facciotti 2010), called *MACS - Model-based Analysis for Chip-Seq* (Zhang et al., 2008), there is no consensus among the researches on how many treatment samples replicates should be used together. Although the software documentation recommends that different replicates should be concatenated in a single file: "*For the experiment with several replicates, it is recommended to concatenate several ChIP-seq treatment files into a single file.*", researches have experienced an unexpected change in the numbers of peaks found when doing so. Some investigators have even suggested that this strategy should be reconsidered and the software should be used on single files, and other strategies should be applied to combine the resulting files afterwards.

In this thesis we show that since most of these methods are based on statistical hypothesis tests (Wu et al., 2015), by increasing the samples (e.g., duplicating both treatment and control samples), the number of peaks considered significant changes considerably for a given threshold; to the extreme of all the peaks in a given chromosome become significant given a large amount of data. This behaviour is known among statisticians as "*increase sample size to reject*" (DeGroot et al., 1986; Stern, 2008). In contrast, our probabilistic approach becomes more assertive regarding the significance of each peak as we increase the sample sizes for both treatment and control samples.

This study describes a Bayesian statistical method using meta-analysis to discover binding sites of a protein of interest based on peaks of reads found in ChIP-Seq data. We call it Meta-Analysis Bayesian Approach or *MABayApp*. The model qualifies peaks found in regions enriched by these reads alignments as significant or non-significant binding site of a specific protein in the genome, using a qualitative measure of probability. The task of identifying peaks in ChIP-Seq data

is commonly known as *Peak Calling* (Hower et al., 2011).

The output of the model is a list of peaks in ascending order of significance based on a probability measure found for each peak, the list also shows the probable binding sites of the DNA-associated protein, discarding the false-positives. By using all the treatment sample replicates and control sample replicates together, the model reduces the chance of sample bias and uses the information regarding the variance of all the replicates to build the final model.

The Bayesian model shown is based on discrete data (counting of reads alignments). The advantages of this meta-analysis Bayesian model include the possibility of combining different studies, adding prior information and retaining the characteristics of the individual distributions of each sample to build the general population distribution. For each study, a Bayesian categorical model is applied, both for treatment and control samples, and the weighted average of these distributions is used as the distribution of the population. Finally, the treatment and control samples are compared, and the probability of each peak to have the same number of alignments in treatment and control samples is found.

We used four samples of ChIP-Seq experiments for the transcription factor SOX3 binding sites in the mouse genome – genome assembly mm10 (Chinwalla et al., 2002) to exemplify the model, three treatment samples resulting from an experiment using antibody to recognize human SOX3, and one control sample without SOX3 antibody. After using our model, we characterized the regions found for each peak, according to the annotation for mouse genome assembly mm10 (Genome Reference Consortium Mouse Build 38, GCA_000001635.2). This annotation is crucial for validating these regions as related to the function of SOXB1 proteins, that play role in the brain development of both human and mouse.

1.1 Literature review

In this section we review some of the state of art algorithms and models used for finding binding in ChIP-Seq data; they are: MACS, SICER, HOMER, T-PEAK and BayesPeak.

Zhang et al. (2008) presented the *Model-based Analysis of ChIP-Seq (MACS)* for analysis of short reads sequences. The model was proposed to address the issues of finding a good tag to distance estimation and predict with higher accuracy the location of binding sites, given the size of sonication.

MACS searches for regions in the genome with enrichment greater than a confidence level enrichment called *mfold*, by sliding a search windows of which size is based on the size of the sonication. This peak detection uses a Poisson distribution *p-value* based on the enrichment of a region, by comparing the Poisson parameter for this region (λ_{local}) against a background Poisson parameter (λ_{BG}).

MACS is currently the most commonly used method to find peaks in data resulting from ChIP-Seq experiments.

In this thesis, we compare our model against MACS and show that, because MACS uses a simple Poisson *p-value* to evaluate candidate peaks, the model is very sensitive to the sample size. We also show that MACS does not address the common situation experienced by many researchers of having ChIP-Seq experiments with multiple control and treatment samples.

Zang et al. (2009) presented a method based on spatial clusters to identify enriched domains from histone modification ChIP-Seq data called SICER.

SICER divides the genome in windows and defines each window as *eligible* if the number of reads aligned to this window is higher than a constant (*count threshold*) l_0 and this count threshold is found based on a *Poisson distribution p-value*. And the significant island are identified with a p-value threshold using Bonferroni correction for multiple significance testing.

Heinz et al. (2010) describes a method called *HOMER (Hypergeometric Optimization of Motif EnRichment)* for analyzing ChIP-Seq data.

HOMER assumes that the local density of tags follows a Poisson distribution. It then uses the expected distribution of peaks and calculates the False Discovery Rate (FDR), or the expected

number of false positives. HOMER then finds peaks with *Poisson p-value* less than the p-value provided by the users and report these peaks as binding sites.

Hower et al. (2011) proposes a method for the identification of statistically significant peaks in ChIP-Seq data based on a *topological data analysis*, called T-PEAK. In this analysis the *height* of each base is found based on the number of read aligned to this base and it incorporates information on the neighbourhood of each site to define the peaks shape.

T-PEAK uses a initial tree based on the number of alignments at each base and uses a topological algorithm called *path excursion* to find the corresponding *root tree*. It then uses a tree shape statistics to find the "peaksness" of each tree. The genome is then divided into regions, T-PEAK identifies possibles peaks in these regions, and finds the *p-value* for each of these trees. Finally it uses a correction for multiple hypothesis testing to remove false-positive peaks.

Spyrou et al. (2009) proposed the statistical algorithm *BayesPeak: Bayesian analysis of ChIP-Seq data*, that uses hidden Markov model to find binding sites in the genome.

BayesPeak uses a hidden Markov model of four states. It also uses a sliding window to search for regions in the genome and it assumes that the dependence between subsequent windows is the same for the whole genome. The states of each window S_t can be 1 or 0 ($S_t = 1$ if there's a binding site in the region t and $S_t = 0$ if there's not a binding site in the region t). The working states Z_t are composed by subsequent windows ($Z_t = (S_t, S_{t+1})$), thus Z_t can assume one of the four states: $(0, 0)$; $(0, 1)$; $(1, 0)$; $(1, 1)$.

BayesPeak assumes a Poisson-Gamma mixture model and uses Markov Chain Monte Carlo (MCMC) algorithms to sample from the Posterior distribution and estimate the parameters of the model. The likelihood expression has no closed form, it is evaluated using a maximization technique of probabilistic functions of Markov Chain.

Although BayesPeak does not uses hypothesis testing, the likelihood and posterior distributions have no closed form and the sampling from the posterior distribution requires advanced statistical methods. The maximization methods used also increases the uncertain of the results. Moreover the method does not allow for multiple control and treatment samples. And according to their experiments, the use of a control sample resulted in a increase number of peaks called (instead of reducing the number of significant peaks), which is surprisingly odd.

1.2 Objective

The main goal of this thesis is to build robust a model, with a strong statistical background to analyse ChIP-Seq data. This model should allow the researches to use as many control and treatment samples as they have available. Nonetheless the increase in number of samples should increase the accuracy of the model, giving more confidence in the results found as the sample size increases.

Together with the model description, a computational tool should be made available for investigators in the area of genetics, and such tool should take as input the genome sequencing data resulting from the technique known as ChIP-Seq, and should output a list of genomic positions (initial and final) of peaks found, ordered by significance. This list should also characterize each peak as probable binding sites of the protein of interest, discarding false-positive peaks. This tool should allow the researchers to input data with several biological and technical replicates, for both treatment and control samples. And the analysis should be performed taking into consideration all the replicates together, thus minimizing the biased results.

1.3 Contribution

The main contributions of this work are:

- construction of a new robust Bayesian model for ChIP-Seq data analysis considering multiple replicates for both treatment and control samples.

- development of a tool to be used mainly at laboratories related to the FAPESP Bioenergy Research Program (FAPESP-BIOEN), and to the Institute of Psychiatry-University of São Paulo, Medical School (FMUSP) in Bioinformatics studies of Chip-Seq data.

1.4 Document Organization

In Chapter 2, we review the biological method used for the experiments studied, ChIP-Seq. Chapter 3 review the Statistical background of the properties and definitions used in this Thesis. In Chapter 4 we introduce the statistical model used for the characterization of significant binding sites of a protein of interest. An overview of the algorithm, from the data collection until the results obtained showing the binding sites found, is shown in Chapter 5, which includes information about the computational methods used. The conclusions found during this study and future research are shown in Chapter 7.

The R code developed for this work is available in Appendix A.

Chapter 2

Method (Biology)

The datasets we analyze are datasets resulting from a method called ChIP-Seq (*Chromatin immunoprecipitation followed by sequencing*). The goal of this method is to find binding sites of a given protein (of interest) in the DNA. Using ChIP-Seq it's possible to identify a set of genes that are active in a given cell at certain time, by analysing the regions of DNA responsible for the regulation of this gene. In order to accomplish this, we use specific antibodies that recognizes the protein of interest, allowing us to know which region of the DNA this protein is bound to (McAninch and Thomas, 2014). This method is discussed next.

2.1 ChIP-Seq

The steps of this method, as shown in Deliard et al. (2013), are described below, and represented in Figure 2.1.

In this method, the protein is first fixed to the DNA (Figure 2.1a), using a chemical process that makes use of formaldehyde and glycine, called *cross-link*. This step is responsible for interrupting the cellular and molecular mechanisms are interrupted, and this state is preserved through freezing using liquid nitrogen followed by storage at -80°C .

The DNA is then fragmented, using a process called Sonication, in which sound waves break the genetic material (Figure 2.1b). Fragments of a specific length can be selected by using either agarose gel electrophoresis or the result of Polymerase Chain Reaction (PCR).

The immunoprecipitation occurs when specific antibodies are connected to small spheres (called *beads*). This antibodies recognize the protein of interest and are bound to them, attaching to the DNA at the protein's binding sites; this binding between the antibody and the protein is reversible.

The protein-DNA complex is then precipitated through a process of centrifugation (Figure 2.1c). After the centrifugation, the genetic material is purified. In this process, the binding between the antibodies and the proteins are reversed, and the DNA is isolated (Figure 2.1d).

The resulting DNA fragments are enriched using the method of Polymerase Chain Reaction (PCR), before DNA sequencing.

Finally these fragments are sequenced, and the resulting *reads* can be aligned to a reference genome (Figure 2.1e). A pile of reads in a given region of the genome is called *peak*, and it is a candidate binding site of the protein of interest.

Figure 2.2 shows the alignment of reads of Sugarcane to a reference genome. The genome used as reference is a Bacterial artificial chromosome (BAC) of Sugarcane. Figures 2.2b, 2.2d and 2.2f show the results of alignment for three runs (three replicated) using the enzyme RNA Polymerase II (treatment sample) and Figures 2.2a, 2.2c and 2.2e show the same results for three replicates using Immunoglobulin G (IgG), commonly used control sample.

Figure 2.1: Sequence of steps of the ChIP-Seq methodology.

(a) *Replicate 1; Control Sample (IgG).* (b) *Replicate 1; Treatment Sample (RNA-PolII).* (c) *Replicate 2; Control Sample (IgG).* (d) *Replicate 2; Treatment Sample (RNA-PolII).* (e) *Replicate 3; Control Sample (IgG).* (f) *Replicate 3; Treatment Sample (RNA-PolII).***Figure 2.2:** *Example of ChIP-Seq reads aligned to Sugarcane BACs.*

2.1.1 Treatment and Control samples

McAninch and Thomas (2014) recently performed ChIP-Seq experiments investigating SOX3 binding sites and enhancers (data accessible at NCBI GEO database, accession GSE57186; Edgar et al., 2002). This experiment consists of three independent DNA libraries resulting from ChIP-Seq experiment using an antibody to recognize human Sox3, and a control sample run without Sox3 antibody.

McAninch and Thomas (2014) compared genome-wide binding sites for Sox2, Sox3 and Sox11, by analysing these transcriptions factors during the neural lineage development. According to them, previous studies have not demonstrated how neural lineage-specific genes expression are selected and later activated during neuronal differentiation. They performed ChIP-Seq, RNA-Seq and microarray experiments investigating SOX2/SOX3/SOX11 binding sites (data accessible at NCBI GEO database, accession GSE33024; Edgar et al., 2002).

We used the samples from these experiments to find significant SOX3 binding sites under the statistical model developed for this study, validate our method and compare against a known method.

2.1.2 Chip-Seq reads alignment & RNA-Seq data analysis

The reads of ChIP-Seq experiments were mapped to mm10 genome assembly using the tool Bowtie version 2.1.0 (Langmead and Salzberg, 2012). We first ran the *bowtie2-build* indexer tool using the mm10 reference genome to create the Bowtie index for mm10 genome assembly; we then ran Bowtie2 with the default parameters. The resulting *.sam* file was converted to *.bam* format using the utilities for the sequence alignment/map format, SAMtools (Li et al., 2009).

The protocol used to analyse the data resulting from RNA-Seq experiments is described in Trapnell et al. (2012). This analysis includes the tools TopHat version 2.0.9 (Kim et al., 2013) and Cufflinks version 2.1.1 (Robertson et al., 2010).

As detailed in the protocol, the reads of each experiment were first mapped to the reference genome using TopHat; we used the parameters "-p 8" to specify the number of 8 processors, "-G genes.gtf" to specify the reference annotation and the Bowtie index for mm10 genome assembly to specify the reference genome.

We then ran Cufflinks for each experiment to assemble the transcripts resulting from TopHat reads alignment. We used the parameter "-p 8" (8 processors), and the *.bam* file resulting from TopHat alignment (*accepted_hits.bam*).

The transcripts assembled by Cufflinks for all the different samples were merged using the tool Cuffmerge. We used the following parameters: "-g genes.gtf" for the reference annotation, "-p 8" for number of processors used, and a text file ("assemblies.txt") with the list of transcript assembly files, one entry per sample (treatment/control sample).

Using the merged transcript assembly found by Cuffmerge tool, we measured the different expression of genes and transcripts using the tool Cuffdiff. We used the parameters "-p 8" to specify that 8 processors should be used, "-L sox3,control" to label the samples in two categories, "Sox3" or "Control", and finding the differential expression between these two categories, for each gene and transcript.

All the tests have been performed on a 64-bits machine with operating system Ubuntu precise release 12.04.5 LTS, with 32GB of memory RAM and 220GB of disk space.

2.2 UCSC genome assembly and annotation

In order to explore the functions associated with the regions of the peaks found by our model, we use an annotation for the mouse genome assembly, the same genome used as input reference to our model. The annotation used is described below.

The annotation for mouse genome assembly mm10 (Genome Reference Consortium Mouse Build 38, GCA_000001635.2) has been downloaded from the UCSC genome browser Rosenbloom et al.

(2015) Schneider and Church (2013). The distribution of the genes and transcripts by chromosome is shown in Fig. 2.5.

Filtering pipeline

Figure 2.3: *Annotation filtering pipeline.*

Figure 2.4: *Stacked karyogram overview of mouse genome assembly mm10.*

A total number of 94,647 ENSEMBL transcripts have been filtered according to the filtering pipeline described in Fig. 2.3. 94,545 transcripts were filtered from random chromosome, from these transcripts, 38,775 genes have been identified. The largest isoform of each gene was selected to define the regions 5'UTR, gene body and 3'UTR of the gene.

The genes found were classified according to their functions, the Fig. 2.7 shows the distribution of genes functions found given the annotation.

The structure classification of the genes annotated is shown in Table 2.1. We compare the length of enriched regions based on its gene structure, in order to find difference in size for the regions for 5'UTR, 3'UTR and gene body.

Figure 2.5: *Genes, transcripts, 5UTR and 3UTR regions distribution by chromossomes for mouse genome assembly mm10.*

Table 2.1: *Classification according to gene region distance.*

Structure classification	Total	Unique regions	Multiple regions
TSS1500	38773	16449	22324
TSS200	38774	16963	21811
5UTR	27550	24341	3209
Body	22415	19702	2713
3UTR	27462	24206	3256
Total	154974	101661	53313

Figure 2.6: Length of genes, transcripts, 5UTR and 3UTR regions by chromosomes for mouse genome assembly mm10.

Figure 2.7: Gene type classification.

Chapter 3

Statistical background

3.1 Definitions and properties of Gamma, Beta and Dirichlet Distributions

These definitions are based on the definition of Dirichlet Distribution and its properties given in [Ferguson \(1973\)](#) and [Frigyik et al. \(2010\)](#).

Theorem 1 (Gamma Function). *The Gamma Function, represented by the letter Γ is defined as follows.*

$$\Gamma(y) = \int_0^{\infty} x^{y-1} e^{-x} dx \quad (3.1)$$

Lemma 1 (Derivative of Gamma Function). *The derivative of the Gamma Function (Γ') is defined as follows.*

$$\Gamma'(y) = \int_0^{\infty} x^{y-1} e^{-x} \ln(x) dx \quad (3.2)$$

Proof.

$$\begin{aligned} \Gamma'(y) &= \frac{d}{dy} \Gamma(y) = \\ &= \frac{d}{dy} \int_0^{\infty} x^{y-1} e^{-x} dx = \\ &= \int_0^{\infty} \frac{d}{dy} e^{y \ln(x)} x^{-1} e^{-x} dx = \\ &= \int_0^{\infty} [x^y \ln(x)] x^{-1} e^{-x} dx = \\ &= \int_0^{\infty} x^{y-1} e^{-x} \ln(x) dx \end{aligned}$$

□

Theorem 2 (Digamma Function). *The Digamma function, represented by the letter Ψ is defined as follows.*

$$\Psi(y) = \frac{d}{dy} \ln(\Gamma(y)) = \frac{\Gamma'(y)}{\Gamma(y)} \quad (3.3)$$

Theorem 3 (Trigamma Function). *The Trigamma function, represented by the Ψ' is defined as*

follows.

$$\Psi'(y) = \frac{d^2}{dy^2} \ln(\Gamma(y)) = \frac{d}{dy} \Psi(y) \quad (3.4)$$

Theorem 4 (Gamma Distribution). *Let X be a random variable distributed according to a Gamma distribution with parameters α and β .*

$$X \sim \text{Gamma}(\alpha, \beta)$$

The probability distribution function of X (X has a Gamma Distribution) is defined as follows.

$$f_X(x|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-x\beta} \quad \text{for } x \geq 0 \text{ and } \alpha, \beta > 0 \quad (3.5)$$

Lemma 2 (Log-gamma distribution). *If X has a Gamma Distribution with parameters (α, β) . The distribution of the variable $Y = \log(X)$ is called Log-gamma Distribution¹ and it's defined as follows.*

$$f_Y(y|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} e^{y\alpha - \beta e^y} \quad (3.6)$$

Proof. If $X \sim \text{Gamma}(\alpha, \beta)$ and $Y = \log(X)$, we have the following relationship between X and Y .

$$\begin{aligned} Y = \log(X) &\implies X = \exp(Y) \\ F_Y(y) = P(Y \leq y) &= P(\log(X) \leq y) = P(X \leq e^y) = F_X(e^y) \end{aligned}$$

The distribution of Y can then be found using the cumulative distribution of X , $F_X(x)$, as follows.

$$f_Y(y) = \frac{d}{dy} F_Y(y) = f_X(e^y) e^y = \frac{\beta^\alpha}{\Gamma(\alpha)} (e^y)^{\alpha-1} e^{-e^y \beta} e^y = \frac{\beta^\alpha}{\Gamma(\alpha)} e^{y\alpha - e^y \beta} \quad -\infty \leq y \leq \infty$$

□

Theorem 5 (Additive property of Gamma Distribution). *Let two variables X_1 e X_2 be independent Gamma distributed variables, with parameters (α_1, β) e (α_2, β) , respectively.*

$$\begin{cases} X_1 \sim \text{Gamma}(\alpha_1, \beta) \\ X_2 \sim \text{Gamma}(\alpha_2, \beta) \\ X_1 \perp\!\!\!\perp X_2 \end{cases}$$

If a third variable $X_{1:2}$ is equal to the sum of X_1 and X_2 . The variable $X_{1:2}$ will also have a Gamma distribution with parameters $(\alpha_1 + \alpha_2, \beta)$.

$$X_{1:2} \sim \text{Gamma}(\alpha_1 + \alpha_2, \beta) \quad \text{where } X_{1:2} = X_1 + X_2 \quad (3.7)$$

Proof. If $X_i \sim \text{Gamma}(\alpha_i, \beta)$, it's distribution is given as follows.

$$f_{X_i}(x_i|\alpha_i, \beta) = \frac{\beta^{\alpha_i}}{\Gamma(\alpha_i)} x_i^{\alpha_i-1} e^{-x_i \beta}$$

¹Sometimes this distribution is called Exponential-Gamma or Distribution.

And the moment-generating function of X_i , can be found as described below.

$$\begin{aligned} M_{X_i}(t) &= \mathbb{E}[e^{tx_i}] = \int_0^\infty \frac{\beta^{\alpha_i}}{\Gamma(\alpha_i)} x_i^{\alpha_i-1} e^{-x_i\beta} e^{tx_i} dx_i \\ &= \int_0^\infty \frac{\beta^{\alpha_i}}{\Gamma(\alpha_i)} x_i^{\alpha_i-1} e^{x_i(t-\beta)} dx_i \end{aligned}$$

Making $y = x_i(t - \beta)$, we have $dy = (t - \beta) dx_i$, and the moment-generating function of the Gamma distribution can be written as follows.

$$\begin{aligned} M_{X_i}(t) &= \int_0^\infty \frac{\beta^{\alpha_i}}{\Gamma(\alpha_i)} x_i^{\alpha_i-1} e^y \frac{dy}{(t - \beta)} \\ &= \frac{\beta^{\alpha_i}}{\Gamma(\alpha_i)} \int_0^\infty \left(\frac{y}{t - \beta}\right)^{\alpha_i-1} e^y \frac{dy}{(t - \beta)} \\ &= \frac{\beta^{\alpha_i}}{\Gamma(\alpha_i)} \int_0^\infty \left(\frac{1}{t - \beta}\right)^{\alpha_i} y^{\alpha_i-1} e^y dy \\ &= \left(\frac{\beta}{t - \beta}\right)^{\alpha_i} \frac{1}{\Gamma(\alpha_i)} \underbrace{\int_0^\infty y^{\alpha_i-1} e^y dy}_{\text{Gamma function, as in Theorem 1}} \\ &= \left(\frac{\beta}{t - \beta}\right)^{\alpha_i} \frac{1}{\Gamma(\alpha_i)} [\Gamma(\alpha_i)] \\ &= \left(\frac{\beta}{t - \beta}\right)^{\alpha_i} \end{aligned}$$

if $X_1 \sim \text{Gamma}(\alpha_1, \beta)$ and $X_2 \sim \text{Gamma}(\alpha_2, \beta)$ are independent random variables, the moment-generating function of $X_1 + X_2$ is defined as follows.

$$\begin{aligned} M_{X_1+X_2}(t) &= \mathbb{E}[e^{t(x_1+x_2)}] \\ &= \mathbb{E}[e^{tx_1} e^{tx_2}] \\ &= \mathbb{E}[e^{tx_1} e^{tx_2}] \\ &= \mathbb{E}[e^{tx_1}] \mathbb{E}[e^{tx_2}] \\ &= M_{X_1}(t) M_{X_2}(t) \\ &= \left(\frac{\beta}{t - \beta}\right)^{\alpha_1} \left(\frac{\beta}{t - \beta}\right)^{\alpha_2} \\ &= \left(\frac{\beta}{t - \beta}\right)^{\alpha_1+\alpha_2} \end{aligned}$$

Which is the moment-generating function of the distribution $\text{Gamma}(\alpha_1 + \alpha_2, \beta)$. □

Theorem 6 (Beta Distribution). *Let X be a random variable distributed according to a Gamma distribution with parameters α_X and β . And a variable Y , independent of X , be also distributed according to a Gamma distribution with parameters α_Y and β , as follows.*

$$\begin{cases} X \sim \text{Gamma}(\alpha_X, \beta) \\ Y \sim \text{Gamma}(\alpha_Y, \beta) \\ X \perp\!\!\!\perp Y \end{cases}$$

Then, $Z = X/(X + Y)$ has a Beta Distribution with parameters (α_X, α_Y) .

$$Z = \frac{X}{X + Y} \sim \text{Beta}(\alpha_X, \alpha_Y) \quad (3.8)$$

Lemma 3 (Probability density function (pdf) of the Beta Distribution). *The pdf of Z (Beta Distribution) is defined as follows.*

$$f_Z(z|\alpha_X, \alpha_Y) = \frac{\Gamma(\alpha_X + \alpha_Y)}{\Gamma(\alpha_X)\Gamma(\alpha_Y)} z^{\alpha_X-1} (1-z)^{\alpha_Y-1} \quad (3.9)$$

Proof. According to Equation 3.8, the variable Z with a Beta Distribution can be defined as follows.

$$Z = \frac{X}{X+Y} \text{ for } \begin{cases} X \sim \text{Gamma}(\alpha_X, \beta) \\ Y \sim \text{Gamma}(\alpha_Y, \beta) \\ X \perp\!\!\!\perp Y \end{cases}$$

The additive property of the Gamma Distribution (Theorem 5) states that the the sum $U = X + Y$ has a Gamma Distribution, as follows.

$$U \sim \text{Gamma}(\alpha_X + \alpha_Y, \beta) \quad \text{where } U = X + Y \quad (3.10)$$

Since X and Y are independent, their joint distribution $f_{X,Y}(x, y)$ is equal to the product of their distributions, as follows.

$$\begin{aligned} f_{X,Y}(x, y) &\stackrel{X \perp\!\!\!\perp Y}{=} f_X(x) f_Y(y) \\ &= \frac{\beta^{\alpha_X}}{\Gamma(\alpha_X)} x^{\alpha_X-1} e^{-x\beta} \frac{\beta^{\alpha_Y}}{\Gamma(\alpha_Y)} y^{\alpha_Y-1} e^{-y\beta} \\ &= \frac{\beta^{(\alpha_X+\alpha_Y)}}{\Gamma(\alpha_X)\Gamma(\alpha_Y)} x^{\alpha_X-1} y^{\alpha_Y-1} e^{-(x+y)\beta} \end{aligned}$$

The transformation from (x, y) to (u, z) , and the corresponding inverse will be the followings.

$$\begin{cases} U = x + y \\ Z = x / (x + y) \end{cases} \implies \begin{cases} X = uz \\ Y = u(1 - z) \end{cases}$$

The Jacobian matrix of this transformation will be the following.

$$\mathbb{J} = \begin{pmatrix} \partial X / \partial u & \partial X / \partial z \\ \partial Y / \partial u & \partial Y / \partial z \end{pmatrix} = \begin{pmatrix} z & u \\ 1 - z & -u \end{pmatrix}$$

The absolute value of it's determinant will be the following.

$$|\det \mathbb{J}| = |-uz - u(1 - z)| = u$$

We can find the joint distribution of U and Z ($f_{U,Z}(u, z)$) using the joint distribution of the variables

X and Y ($f_{X,Y}(x,y)$), as follows.

$$\begin{aligned}
f_{U,Z}(u,z) &= f_{X,Y}(uz, u(1-z))u \\
&= \frac{\beta^{(\alpha_X+\alpha_Y)}}{\Gamma(\alpha_X)\Gamma(\alpha_Y)} (uz)^{\alpha_X-1} (u(1-z))^{\alpha_Y-1} e^{-(uz+u(1-z))\beta} u \\
&= \frac{\beta^{(\alpha_X+\alpha_Y)}}{\Gamma(\alpha_X)\Gamma(\alpha_Y)} u^{\alpha_X-1} z^{\alpha_X-1} u^{\alpha_Y-1} (1-z)^{\alpha_Y-1} e^{-u\beta} u \\
&= \frac{\beta^{(\alpha_X+\alpha_Y)}}{\Gamma(\alpha_X)\Gamma(\alpha_Y)} u^{\alpha_X+\alpha_Y-1} z^{\alpha_X-1} (1-z)^{\alpha_Y-1} e^{-u\beta} \\
&= \underbrace{\frac{\beta^{(\alpha_X+\alpha_Y)}}{\Gamma(\alpha_X+\alpha_Y)} u^{\alpha_X+\alpha_Y-1} e^{-u\beta}}_{f_U(u)=\text{Gamma}(\alpha_X+\alpha_Y, \beta) \text{ as in Equation 3.10}} \frac{\Gamma(\alpha_X+\alpha_Y)}{\Gamma(\alpha_X)\Gamma(\alpha_Y)} z^{\alpha_X-1} (1-z)^{\alpha_Y-1}
\end{aligned}$$

From the equation above, we can see that U and Z are independent, and the distribution of the variable Z is the pdf of the Beta Distribution, as shown in Equation 3.9.

$$f_{U,Z}(u,v) = f_U(u) f_Z(z) \implies f_Z(z) = \frac{\Gamma(\alpha_X+\alpha_Y)}{\Gamma(\alpha_X)\Gamma(\alpha_Y)} z^{\alpha_X-1} (1-z)^{\alpha_Y-1}$$

□

Theorem 7 (Dirichlet Distribution). *Let $X_1, \dots, X_k$ be k independent variables ($k \geq 2$) following a Gamma distribution with parameters $(\alpha_i, 1)$.*

$$\begin{cases} X_i \sim \text{Gamma}(\alpha_i, 1) & 1 \leq i \leq k \\ X_1 \perp\!\!\!\perp X_2 \perp\!\!\!\perp X_3 \perp\!\!\!\perp \dots \perp\!\!\!\perp X_k \end{cases}$$

The distribution of the vector $\boldsymbol{\theta}$ with components θ_i , where $\theta_i = X_i / \left(\sum_{j=1}^k X_j\right)$ is a Dirichlet Distribution, with parameters α_i , as follows.

$$\boldsymbol{\theta} = (\theta_1, \dots, \theta_k) \sim \text{Dir}(\alpha_1, \dots, \alpha_k) \quad \text{where } \theta_i = \frac{X_i}{\sum_{j=1}^k X_j} \quad (3.11)$$

From the definition of Gamma Distribution (Theorem 4), the parameters θ_i must follow the following conditions.

$$\theta_1, \theta_2, \dots, \theta_k > 0 \quad \text{and} \quad \theta_1 + \theta_2 + \dots + \theta_k = 1 \quad (3.12)$$

Lemma 4 (Probability density function (pdf) of the Dirichlet Distribution). *The probability distribution function of $\boldsymbol{\theta}$ (Dirichlet Distribution) is defined as follows.*

$$f_{\boldsymbol{\theta}}(\theta_1, \dots, \theta_k | \alpha_1, \dots, \alpha_k) = \frac{\Gamma\left(\sum_{i=1}^k \alpha_i\right)}{\prod_{i=1}^k \Gamma(\alpha_i)} \prod_{i=1}^k \theta_i^{\alpha_i-1} \quad (3.13)$$

Proof. Since $\sum_{i=1}^k \theta_i = 1$, we can rewrite the pdf of the Dirichlet Distribution as follows.

$$f_{\theta_1, \dots, \theta_{k-1}}(\theta_1, \dots, \theta_{k-1} | \alpha_1, \dots, \alpha_k) = \frac{\Gamma\left(\sum_{i=1}^k \alpha_i\right)}{\prod_{i=1}^k \Gamma(\alpha_i)} \prod_{i=1}^{k-1} \theta_i^{\alpha_i-1} \left(1 - \sum_{j=1}^{k-1} \theta_j\right)^{\alpha_k}$$

Using the definition of Gamma Distribution (Theorem 4), we can define the joint distribution of

the independent variables $X_i \sim \text{Gamma}(\alpha_i, \beta)$, $1 \leq i \leq k$, as the following.

$$f_{X_1, \dots, X_k}(x_1, \dots, x_k) = \prod_{i=1}^k \frac{\beta^{\alpha_i}}{\Gamma(\alpha_i)} x_i^{\alpha_i-1} e^{-x_i \beta} = \frac{\beta^{\sum_{i=1}^k \alpha_i}}{\prod_{i=1}^k \Gamma(\alpha_i)} \prod_{i=1}^k [x_i^{\alpha_i-1}] e^{-\beta \sum_{i=1}^k x_i} \quad (3.14)$$

We define the variables $X_{1:k}$ and θ_j , for $1 \leq j \leq k-1$, as follows.

$$X_{1:k} = \sum_{i=1}^k X_i \quad \theta_j = \frac{X_j}{X_{1:k}} \quad 1 \leq j \leq k-1$$

The variable $X_{1:k}$ is the sum of independent variables $X_i \sim \text{Gamma}(\alpha_i, \beta)$, and according to the Additive property of Gamma Distribution (Theorem 5), $X_{1:k}$ has a Gamma Distribution with parameters $(\sum_{i=1}^k \alpha_i, \beta)$, and it's pdf (Equation 3.5) is defined as follows.

$$f_{X_{1:k}}\left(x_{1:k} \left| \sum_{i=1}^k \alpha_i, \beta\right.\right) = \frac{\beta^{\sum_{i=1}^k \alpha_i}}{\Gamma(\sum_{i=1}^k \alpha_i)} x_{1:k}^{\sum_{i=1}^k \alpha_i - 1} e^{-x_{1:k} \beta} \quad (3.15)$$

The transformation from (X_j, X_k) to $(\theta_j, X_{1:k})$, where $1 \leq j \leq k-1$, and the corresponding inverse will be the followings.

$$\begin{cases} X_{1:k} = \sum_{i=1}^k X_i \\ \theta_j = X_j / X_{1:k} \quad 1 \leq j \leq k-1 \end{cases} \implies \begin{cases} X_j = (\theta_j) (X_{1:k}) \quad 1 \leq j \leq k-1 \\ X_k = X_{1:k} \left(1 - \sum_{j=1}^{k-1} \theta_j\right) \end{cases}$$

The Jacobian matrix of this transformation will be the following.

$$\mathbb{J} = \begin{pmatrix} \frac{\partial X_1}{\partial \theta_1} & \frac{\partial X_1}{\partial \theta_2} & \cdots & \frac{\partial X_1}{\partial \theta_{k-1}} & \frac{\partial X_1}{\partial X_{1:k}} \\ \frac{\partial X_2}{\partial \theta_1} & \frac{\partial X_2}{\partial \theta_2} & \cdots & \frac{\partial X_2}{\partial \theta_{k-1}} & \frac{\partial X_2}{\partial X_{1:k}} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial X_{k-1}}{\partial \theta_1} & \frac{\partial X_{k-1}}{\partial \theta_2} & \cdots & \frac{\partial X_{k-1}}{\partial \theta_{k-1}} & \frac{\partial X_{k-1}}{\partial X_{1:k}} \\ \frac{\partial X_k}{\partial \theta_1} & \frac{\partial X_k}{\partial \theta_2} & \cdots & \frac{\partial X_k}{\partial \theta_{k-1}} & \frac{\partial X_k}{\partial X_{1:k}} \end{pmatrix}$$

$$\det \mathbb{J} = \begin{vmatrix} X_{1:k} & 0 & \cdots & 0 & \theta_1 \\ 0 & X_{1:k} & \cdots & 0 & \theta_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & X_{1:k} & \theta_{k-1} \\ -X_{1:k} & -X_{1:k} & \cdots & -X_{1:k} & 1 - \sum_{j=1}^{k-1} \theta_j \end{vmatrix}$$

Adding the first $k-1$ rows to the last one, we end up with a upper triangular matrix, and we find the determinant multiplying the elements of the diagonal, as follows.

$$\det \mathbb{J} = \begin{vmatrix} X_{1:k} & 0 & \cdots & 0 & \theta_1 \\ 0 & X_{1:k} & \cdots & 0 & \theta_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & X_{1:k} & \theta_{k-1} \\ 0 & 0 & \cdots & 0 & 1 \end{vmatrix} = (X_{1:k})^{k-1}$$

We can find the joint distribution of θ_j , $1 \leq j \leq k-1$ and $X_{1:k}$ $\left(f_{\theta_1, \dots, \theta_{k-1}, X_{1:k}}(\theta_1, \dots, \theta_{k-1}, x_{1:k})\right)$

using the joint distribution of the variables X_i , $1 \leq i \leq k$ (Equation 3.14), as follows.

$$\begin{aligned}
& f_{\theta_1, \dots, \theta_{k-1}, X_{1:k}}(\theta_1, \dots, \theta_{k-1}, x_{1:k}) f_{X_1, \dots, X_k}(\theta_1, \dots, \theta_{k-1}, x_{1:k}) * (x_{1:k})^{k-1} = \\
& \frac{\beta^{\sum_{i=1}^k \alpha_i}}{\prod_{i=1}^k \Gamma(\alpha_i)} \prod_{j=1}^{k-1} [(\theta_j x_{1:k})^{\alpha_j - 1}] \left(x_{1:k} \left(1 - \sum_{j=1}^{k-1} \theta_j \right) \right)^{\alpha_k - 1} e^{-\beta \sum_{i=1}^{k-1} \theta_i} e^{-\beta x_{1:k} (1 - \sum_{j=1}^{k-1} \theta_j)} (x_{1:k})^{k-1} = \\
& \frac{\beta^{\sum_{i=1}^k \alpha_i}}{\prod_{i=1}^k \Gamma(\alpha_i)} x_{1:k}^{\sum_{i=1}^k (\alpha_i - 1) + k - 1} e^{-\beta x_{1:k} (1 - \sum_{j=1}^{k-1} \theta_j)} \prod_{j=1}^{k-1} [(\theta_j)^{\alpha_j - 1}] \left(1 - \sum_{j=1}^{k-1} \theta_j \right)^{\alpha_k - 1} e^{-\beta \sum_{i=1}^{k-1} \theta_i} = \\
& \frac{(\beta^{\sum_{i=1}^k \alpha_i}) \left(x_{1:k}^{\sum_{i=1}^k \alpha_i - 1} \right) (e^{-\beta x_{1:k}})^{k-1} \prod_{j=1}^{k-1} [(\theta_j)^{\alpha_j - 1}] \left(1 - \sum_{j=1}^{k-1} \theta_j \right)^{\alpha_k - 1} e^{-\beta \sum_{i=1}^{k-1} \theta_i} e^{\beta \sum_{j=1}^{k-1} \theta_j}}{\prod_{i=1}^k \Gamma(\alpha_i)} = \\
& \underbrace{\frac{(\beta^{\sum_{i=1}^k \alpha_i}) \left(x_{1:k}^{\sum_{i=1}^k \alpha_i - 1} \right) (e^{-\beta x_{1:k}})}{\Gamma\left(\sum_{i=1}^k \alpha_i\right)}}_{f_{X_{1:k}}(x_{1:k} | \sum_{i=1}^k \alpha_i, \beta) \text{ as in Equation 3.15}} \frac{\prod_{j=1}^{k-1} [(\theta_j)^{\alpha_j - 1}] \left(1 - \sum_{j=1}^{k-1} \theta_j \right)^{\alpha_k - 1}}{\prod_{i=1}^k \Gamma(\alpha_i)}
\end{aligned}$$

From the equation above, we can see that the vector $(\theta_1, \dots, \theta_{k-1})$ and the variable $X_{1:k}$ are independent, and the distribution of the vector of variables $(\theta_1, \dots, \theta_{k-1})$ is the pdf of the Dirichlet Distribution, as shown in Equation 3.13.

$$f_{\theta_1, \dots, \theta_{k-1}, X_{1:k}}(\theta_1, \dots, \theta_{k-1}, x_{1:k}) = f_{X_{1:k}}(x_{1:k}) f_{\theta_1, \dots, \theta_{k-1}}(\theta_1, \dots, \theta_{k-1})$$

$$\therefore f_{\theta_1, \dots, \theta_{k-1}}(\theta_1, \dots, \theta_{k-1}) = \frac{\Gamma\left(\sum_{i=1}^k \alpha_i\right)}{\prod_{i=1}^k \Gamma(\alpha_i)} \prod_{j=1}^{k-1} [(\theta_j)^{\alpha_j - 1}] \left(1 - \sum_{j=1}^{k-1} \theta_j \right)^{\alpha_k - 1}$$

□

Theorem 8 (Additive property of Dirichlet Distribution). *Let $\boldsymbol{\theta} = (\theta_1, \dots, \theta_k)$ be a vector with a Dirichlet Distribution with parameters given by the vector $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_k)$, the following property holds.*

$(\theta_1, \dots, \theta_k) \sim \text{Dir}(\alpha_1, \dots, \alpha_k)$ and $r_1, r_2, \dots, r_n$ integers, such that $1 \leq r_1 \leq \dots \leq r_n = k$, then:

$$\left(\sum_{i=1}^{r_1} [\theta_i], \sum_{i=(r_1+1)}^{r_2} [\theta_i], \dots, \sum_{i=(r_{n-1}+1)}^{r_n} [\theta_i] \right) \sim \text{Dir} \left(\sum_{i=1}^{r_1} [\alpha_i], \sum_{i=(r_1+1)}^{r_2} [\alpha_i], \dots, \sum_{i=(r_{n-1}+1)}^{r_n} [\alpha_i] \right) \quad (3.16)$$

Proof. We prove by induction.

1. The base case: for variables X_1 and X_2
2. Inductive step: if it holds for $X_{1:n} = X_1 + X_2 + \dots + X_n$, then it holds for $X_{1:n+1} = X_n + X_{n+1}$

The proof follows direct from the definition of Dirichlet Distribution given in Theorem 7 and the Additive property of Gamma Distribution given in Theorem 5.

According to the definition of the Dirichlet Distribution (Theorem 7), we can define the vector θ using a vector of Gamma distributed independent random variables $X = (X_1, \dots, X_k)$, as follows.

$$\theta = (\theta_1, \dots, \theta_k) = \left(\frac{X_1}{X_1 + X_2 + \dots + X_k}, \dots, \frac{X_k}{X_1 + X_2 + \dots + X_k} \right),$$

where $X_i \sim \text{Gamma}(\alpha_i, 1)$ for $1 \leq i \leq k$

According to the Theorem 5, the sum of two independent Gamma distributed variables is another Gamma distributed variable as follows.

$$\begin{aligned} \text{If } X_1 \sim \text{Gamma}(\alpha_1, 1), \quad X_2 \sim \text{Gamma}(\alpha_2, 1) \quad \text{and } X_1 \perp\!\!\!\perp X_2, \text{ then} \\ X_1 + X_2 \sim \text{Gamma}(\alpha_1 + \alpha_2, 1) \end{aligned}$$

Using these $k - 1$ independent Gamma distributed variables $([X_1 + X_2], X_3, \dots, X_k)$, we can define another array of variables $([\theta_1 + \theta_2], \theta_3, \dots, \theta_k)$ that will have a Dirichlet Distribution, as follows.

$$\begin{aligned} & \begin{cases} X_1 + X_2 \sim \text{Gamma}(\alpha_1 + \alpha_2, 1) \\ X_j \sim \text{Gamma}(\alpha_j, 1); \quad \text{for } 3 \leq i \leq k \end{cases} \\ & \left(\frac{X_1 + X_2}{\sum_{i=1}^k X_i}, \dots, \frac{X_k}{\sum_{i=1}^k X_i} \right) = \left(\theta_1 + \theta_2, \theta_3, \dots, \theta_k \right) \sim \text{Dir} \left((\alpha_1 + \alpha_2), \alpha_3, \dots, \alpha_k \right) \end{aligned}$$

If the sum of n independent random variables has a Gamma distribution with parameters $\alpha_{1:n}$ and β .

$$X_{1:n} = X_1 + X_2 + \dots + X_n \sim \text{Gamma}(\alpha_1 + \alpha_2 + \dots + \alpha_n, 1)$$

And another random variable is independent of the n variables above, and has a Gamma distribution with parameters α_{n+1} and the same β .

$$X_{n+1} \sim \text{Gamma}(\alpha_{n+1}, 1)$$

According to Theorem 5, the sum of these $n + 1$ variables will also have a Gamma distribution, with parameters $\alpha_{1:n} + \alpha_{n+1}$ and β , as follows.

$$X_{1:n+1} + X_{n+1} \sim \text{Gamma} \left(\sum_{i=1}^n [\alpha_i] + \alpha_{n+1}, 1 \right)$$

Once again, from the definition of Dirichlet Distribution, we can divide each of the $k - n$ independent Gamma distributed variables by their sum, to define a new Dirichlet Distribution, as follows.

$$\begin{aligned} & \begin{cases} X_{1:n+1} \sim \text{Gamma}(\sum_{i=1}^{n+1} [\alpha_i], 1) \\ X_j \sim \text{Gamma}(\alpha_j, 1); \quad \text{for } (n + 2) \leq i \leq k \end{cases} \\ & \left(\frac{\sum_{i=1}^{n+1}}{\sum_{i=1}^k X_i}, \dots, \frac{X_k}{\sum_{i=1}^k X_i} \right) = \left(\sum_{i=1}^{n+1} \theta_i, \theta_{n+2}, \dots, \theta_k \right) \sim \text{Dir} \left(\sum_{i=1}^{n+1} [\alpha_i], \alpha_{n+2}, \dots, \alpha_k \right) \end{aligned}$$

One can repeat this process, starting from integers $r_1, r_2, \dots, r_n$ such that, $1 \leq r_1 \leq \dots \leq r_n = k$ to arrive at the Equation 3.16. $\square$

Theorem 9 (Marginal Distribution of Dirichlet Distribution). *The marginal distributions of a Dirichlet Distribution are Beta Distributions; i.e, if θ is a vector of k random variables θ_i ($k \geq 2$),*

and $\boldsymbol{\theta}$ has a Dirichlet Distribution, then each variable θ_i has a Beta Distribution, as follows.

$$\text{If } \boldsymbol{\theta} = (\theta_1, \dots, \theta_k) \sim \text{Dir}(\alpha_1, \dots, \alpha_k) \quad , \text{ then } \quad \theta_i \sim \text{Beta} \left(\alpha_i, \sum_{j=1}^k [\alpha_j] - \alpha_i \right) \quad (3.17)$$

Proof. The proof follows straightforward from the definition of Dirichlet Distribution given in Theorem 7 and the additive property of Dirichlet Distribution given in Theorem 8.

If $\boldsymbol{\theta} = (\theta_1, \dots, \theta_k)$ has a Dirichlet Distribution with parameters given by the vector $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_k)$

$$\boldsymbol{\theta} = (\theta_1, \dots, \theta_k) \sim \text{Dir}(\alpha_1, \dots, \alpha_k) \quad (3.18)$$

By the Additive Property of the Dirichlet Distribution (Theorem 8), for any parameter θ_j (where $1 \leq j \leq k$), if we sum-up all the remaining parameter in $\boldsymbol{\theta}$ (all θ_i , where $1 \leq i \neq j \leq k$) we will have the following Dirichlet Distribution.

$$\boldsymbol{\theta} = \left(\theta_j, \sum_{\substack{i=1 \\ i \neq j}}^k \theta_i \right) \sim \text{Dir} \left(\alpha_j, \sum_{\substack{i=1 \\ i \neq j}}^k \alpha_i \right)$$

From the conditions given in Equation 3.12, we know that $\sum_{i=1}^k \theta_i = 1$, therefore we can write the distribution above, as follows.

$$\boldsymbol{\theta} = (\theta_j, 1 - \theta_j) \sim \text{Dir} \left(\alpha_j, \sum_{i=1}^k [\alpha_i] - \alpha_j \right)$$

Using the definition of Dirichlet Distribution, given in Equation 3.13, we can write this distribution as follows.

$$f_{\boldsymbol{\theta}} \left(\theta_j, 1 - \theta_j | \alpha_j, \sum_{i=1}^k [\alpha_i] - \alpha_j \right) = \frac{\Gamma \left(\alpha_j + \sum_{i=1}^k [\alpha_i] - \alpha_j \right)}{\Gamma(\alpha_j) \Gamma \left(\sum_{i=1}^k [\alpha_i] - \alpha_j \right)} \theta_j^{\alpha_j} (1 - \theta_j)^{\sum_{i=1}^k [\alpha_i] - \alpha_j}$$

The distribution above is a Beta Distribution (Equation 3.9) with parameters $\left(\alpha_j, \sum_{i=1}^k [\alpha_i] - \alpha_j \right)$. $\square$

Theorem 10 (Neutrality property of Dirichlet Distribution). *A given vector $\boldsymbol{\Theta} = (\theta_1, \dots, \theta_k)$ is said 'neutral' if for any $1 \leq j \leq k$, θ_j is independent of $\frac{1}{\theta_j} \boldsymbol{\Theta}_{-j}$; where $\boldsymbol{\Theta}_{-j}$ is equal to the vector $\boldsymbol{\Theta}$ without the element θ_j . We prove bellow that if $\boldsymbol{\Theta} \sim \text{Dir}(\alpha_1, \dots, \alpha_k)$, $\boldsymbol{\Theta}$ is neutral. Without loss of generality, we use $j = k$.*

The joint distribution of the variables in the array $\boldsymbol{\theta} = ((\theta_1, \dots, \theta_k))$ is the pdf of Dirichlet Distribution, as follows.

$$f_{\boldsymbol{\theta}}(\theta_1, \dots, \theta_k | \alpha_1, \dots, \alpha_k) = \frac{\Gamma \left(\sum_{i=1}^k \alpha_i \right)}{\prod_{i=1}^k \Gamma(\alpha_i)} \prod_{i=1}^k \theta_i^{\alpha_i - 1}$$

Since $\sum_{i=1}^k \theta_i = 1$, we can rewrite the pdf above using $\theta_k = 1 - \sum_{j=1}^{k-1} \theta_j$, as follows.

$$f_{\boldsymbol{\theta}}(\theta_1, \dots, \theta_{k-1} | \alpha_1, \dots, \alpha_k) = \frac{\Gamma \left(\sum_{i=1}^k \alpha_i \right)}{\prod_{i=1}^k \Gamma(\alpha_i)} \prod_{j=1}^{k-1} \theta_j^{\alpha_j - 1} \left(1 - \sum_{j=1}^{k-1} \theta_j \right)^{\alpha_k - 1} \quad (3.19)$$

$$(\theta_1, \dots, \theta_k) \sim Dir(\alpha_1, \dots, \alpha_k) \implies \begin{cases} \theta_k \sim Beta(\alpha_k, \sum_{j=1}^{k-1} \alpha_j) \\ \frac{1}{1-\theta_k}(\theta_1, \dots, \theta_{k-1}) \sim Dir(\alpha_1, \dots, \alpha_{k-1}) \\ \theta_k \perp\!\!\!\perp \frac{1}{1-\theta_k}(\theta_1, \dots, \theta_{k-1}) \end{cases}$$

Proof. The proof of the two first statements above follows straightforwardly from the definitions of Dirichlet Distribution (Theorem 7) and the Marginal Distribution Beta of Dirichlet Distribution (Theorem 9).

$$(\theta_1, \dots, \theta_k) \sim Dir(\alpha_1, \dots, \alpha_k) \implies \theta_i = \frac{X_i}{\sum_{i=1}^k X_i}, \text{ for } \begin{cases} X_i \sim Gamma(\alpha_i, 1) & 1 \leq i \leq k \\ X_1 \perp\!\!\!\perp X_2 \perp\!\!\!\perp \dots \perp\!\!\!\perp X_k \end{cases}$$

$$\text{From Theorem 9, we have: } \theta_k = \frac{X_k}{\sum_{i=1}^k X_i} \sim Beta\left(\alpha_k, \sum_{i=1}^{k-1} \alpha_i\right)$$

$$\text{For } 1 \leq j \leq k-1, \quad \frac{1}{1-\theta_k}(\theta_j) = \frac{1}{\sum_{j=1}^{k-1} \theta_j}(\theta_j) = \frac{\frac{X_j}{\sum_{i=1}^k X_i}}{\left(\frac{\sum_{j=1}^{k-1} X_j}{\sum_{i=1}^k X_i}\right)} = \frac{X_j}{\sum_{j=1}^{k-1} X_j}$$

$$\therefore \frac{1}{\sum_{j=1}^{k-1} \theta_j}(\theta_1, \dots, \theta_{k-1}) = \underbrace{\left(\frac{X_1}{\sum_{j=1}^{k-1} X_j}, \dots, \frac{X_{k-1}}{\sum_{j=1}^{k-1} X_j}\right)}_{\text{Theorem 7}} \sim Dir(\alpha_1, \dots, \alpha_{k-1})$$

In order to prove the last statement, $\theta_k \perp\!\!\!\perp \left(\frac{1}{1-\theta_k}\right)(\theta_1, \dots, \theta_{k-1})$, we use the following transformation.

$$\begin{cases} Q_j = \theta_j / (1 - \theta_k) & 1 \leq j \leq k-2 \\ Q_k = \theta_k \end{cases} \implies \begin{cases} \theta_j = Q_j (1 - Q_k) & 1 \leq j \leq k-2 \\ \theta_k = Q_k \end{cases}$$

The Jacobian matrix of this transformation will be the following.

$$\mathbb{J} = \begin{pmatrix} \frac{\partial \theta_1}{\partial Q_1} & \frac{\partial \theta_1}{\partial Q_2} & \dots & \frac{\partial \theta_1}{\partial Q_{k-2}} & \frac{\partial \theta_1}{\partial Q_k} \\ \frac{\partial \theta_2}{\partial Q_1} & \frac{\partial \theta_2}{\partial Q_2} & \dots & \frac{\partial \theta_2}{\partial Q_{k-2}} & \frac{\partial \theta_2}{\partial Q_k} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial \theta_{k-2}}{\partial Q_1} & \frac{\partial \theta_{k-2}}{\partial Q_2} & \dots & \frac{\partial \theta_{k-2}}{\partial Q_{k-2}} & \frac{\partial \theta_{k-2}}{\partial Q_k} \\ \frac{\partial \theta_k}{\partial Q_1} & \frac{\partial \theta_k}{\partial Q_2} & \dots & \frac{\partial \theta_k}{\partial Q_{k-2}} & \frac{\partial \theta_k}{\partial Q_k} \end{pmatrix}$$

$$\det \mathbb{J} = \begin{vmatrix} 1 - Q_k & 0 & \dots & 0 & -Q_1 \\ 0 & 1 - Q_k & \dots & 0 & -Q_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 - Q_k & -Q_{k-1} \\ 0 & 0 & \dots & 0 & 1 \end{vmatrix} = (1 - Q_k)^{k-2}$$

We can find the joint distribution of $Q_i, 1 \leq i \leq k$ $\left(f_{\mathbf{Q}}(q_1, \dots, q_k)\right)$ using the joint distribution of the variables $\theta_i, i \leq j \leq k$ (Equation 3.19), as follows.

$$\begin{aligned}
f_{\mathbf{Q}}(q_1, \dots, q_k) &= f_{\boldsymbol{\theta}}(q_1, \dots, q_k) \left[(1 - q_k)^{k-2} \right] \\
&= \frac{\Gamma\left(\sum_{i=1}^k \alpha_i\right)}{\prod_{i=1}^k \Gamma(\alpha_i)} \underbrace{\prod_{j=1}^{k-2} (q_j (1 - q_k))^{\alpha_j - 1}}_{\theta_1, \dots, \theta_{k-2}} \underbrace{\left(1 - \sum_{j=1}^{k-2} [q_j (1 - q_k)] - q_k\right)^{\alpha_{k-1} - 1}}_{\theta_{k-1}} \underbrace{\left[(q_k)^{\alpha_k - 1}\right]}_{\theta_k} \left[(1 - q_k)^{k-2} \right] \\
&= \frac{\Gamma\left(\sum_{i=1}^k \alpha_i\right)}{\prod_{i=1}^k \Gamma(\alpha_i)} \prod_{j=1}^{k-2} (q_j (1 - q_k))^{\alpha_j - 1} \left((1 - q_k) \left(1 - \sum_{j=1}^{k-2} q_j\right) \right)^{\alpha_{k-1} - 1} (q_k)^{\alpha_k - 1} \left[(1 - q_k)^{k-2} \right] \\
&= \frac{\Gamma\left(\sum_{i=1}^k \alpha_i\right)}{\prod_{i=1}^k \Gamma(\alpha_i)} \prod_{j=1}^{k-2} [q_j^{\alpha_j - 1}] \left(1 - \sum_{j=1}^{k-2} q_j\right)^{\alpha_{k-1} - 1} (1 - q_k)^{\sum_{j=1}^{k-1} (\alpha_j - 1) + k - 2} (q_k)^{\alpha_k - 1} \\
&= \frac{\Gamma\left(\sum_{i=1}^k \alpha_i\right)}{\prod_{i=1}^k \Gamma(\alpha_i)} \prod_{j=1}^{k-2} [q_j^{\alpha_j - 1}] \left(1 - \sum_{j=1}^{k-2} q_j\right)^{\alpha_{k-1} - 1} (q_k)^{\alpha_k - 1} (1 - q_k)^{\sum_{j=1}^{k-1} \alpha_j - 1} \\
&= \frac{\Gamma\left(\sum_{i=1}^k \alpha_i\right)}{\prod_{i=1}^k \Gamma(\alpha_i)} \frac{\Gamma\left(\sum_{j=1}^{k-1} \alpha_j\right)}{\Gamma\left(\sum_{j=1}^{k-1} \alpha_j\right)} \prod_{j=1}^{k-2} [q_j^{\alpha_j - 1}] \left(1 - \sum_{j=1}^{k-2} q_j\right)^{\alpha_{k-1} - 1} (q_k)^{\alpha_k - 1} (1 - q_k)^{\sum_{j=1}^{k-1} \alpha_j - 1} \\
&= \left[\frac{\Gamma\left(\sum_{j=1}^{k-1} \alpha_j\right)}{\prod_{i=1}^{k-1} \Gamma(\alpha_i)} \prod_{j=1}^{k-2} [q_j^{\alpha_j - 1}] \left(1 - \sum_{j=1}^{k-2} q_j\right)^{\alpha_{k-1} - 1} \right] \underbrace{\left[\frac{\Gamma\left(\alpha_k + \sum_{i=1}^{k-1} \alpha_i\right)}{\Gamma(\alpha_k) \Gamma\left(\sum_{j=1}^{k-1} \alpha_j\right)} (q_k)^{\alpha_k - 1} (1 - q_k)^{\sum_{j=1}^{k-1} \alpha_j - 1} \right]}_{q_k = \theta_k \sim \text{Beta}(\alpha_k, \sum_{j=1}^{k-1} \alpha_j)}
\end{aligned}$$

From the equation above, we can see that the vector $(Q_1, \dots, Q_{k-1}) = \frac{1}{1 - \theta_k} (\theta_1, \dots, \theta_{k-1})$ and the variable Q_k are independent, and the distribution of the vector of variables $\frac{1}{\sum_{j=1}^{k-1} Q_j} (Q_1, \dots, Q_{k-1})$ is the pdf of the Dirichlet Distribution, as follows.

$$f_{\mathbf{Q}}(q_1, \dots, q_k) = f_{Q_k}(q_k) f_{Q_1, \dots, Q_{k-1}}(q_1, \dots, q_{k-1})$$

$$\begin{aligned}
\therefore f_{Q_1, \dots, Q_{k-1}}(q_1, \dots, q_{k-1}) &= \frac{\Gamma\left(\sum_{j=1}^{k-1} \alpha_j\right)}{\prod_{i=1}^{k-1} \Gamma(\alpha_i)} \prod_{j=1}^{k-2} [q_j^{\alpha_j - 1}] \left(1 - \sum_{j=1}^{k-2} q_j\right)^{\alpha_{k-1} - 1} \\
\theta_k &\perp \frac{1}{1 - \theta_k} (\theta_1, \dots, \theta_{k-1}) \quad \text{and} \quad \frac{1}{1 - \theta_k} (\theta_1, \dots, \theta_{k-1}) \sim \text{Dir}(\alpha_1, \dots, \alpha_{k-1})
\end{aligned}$$

□

3.2 Logistic-Normal Distribution

The Logistic-normal Distributions are formally defined in [Atchison and Shen \(1980\)](#) and the substitution of Dirichlet Distribution by Logistic-Normal distribution is also described as the main goal of that work. This substitution of Dirichlet Distribution by the Logistic-Normal Distribution is based on an approximation that is very useful in many applications ([Johnson, 1949](#); [Lindley, 1964](#)). This approximation is shown in the following Thesis [Petri \(2007\)](#); [Rodrigues \(2006\)](#), as well as in [de Bragança Pereira and Stern \(2008\)](#).

Here we define and prove the approximation using a Beta distributed variable and the corresponding univariate Logistic-Normal Distribution. For the multivariate case, see [Atchison and Shen \(1980\)](#).

Theorem 11 (Substitution of Beta Distribution by Logistic-Normal Distribution). *Let X be a random variable having Beta Distribution with parameters α e β , as follows.*

$$X \sim \text{Beta}(\alpha, \beta) \quad \alpha, \beta > 0$$

The distribution of the variable $\theta = \ln(X/(1-X))$ is approximate Normal with mean and variance equal to $\mu_\theta = \Psi(\alpha) - \Psi(\beta)$ and $\sigma_\theta^2 = \Psi'(\alpha) + \Psi'(\beta)$, respectively. Where $\Psi(\alpha)$ is the Digamma function defined in Theorem 2 and $\Psi'(\alpha)$ is the Trigamma function defined in Theorem 3. This approximation is accurate for large values of parameters α and β

$$\theta = \ln(X/(1-X)) \dot{\sim} \mathcal{N}\left(\Psi(\alpha) - \Psi(\beta), \Psi'(\alpha) + \Psi'(\beta)\right) \quad (3.20)$$

We first prove that: (1) θ can be written as the difference of the log of two Gamma distributed variables; (2) the mean of θ is equal to $\Psi(\alpha) - \Psi(\beta)$; (3) the variance of θ is equal to $\Psi(\alpha) - \Psi(\beta)$; (4) the distribution of the log of a variable with Gamma distribution is approximate Normal, and this approximation is accurate for large values of the parameter of the Gamma distribution. We then, put all these results together to show that for large values of α and β , the approximation to the Normal distribution, given in Equation 3.20, is accurate.

Lemma 5 (The logit of the parameter of a Beta Distribution is equal to a difference of independent Gamma distributed variables). *The variable $\theta = \ln(X/(1-X))$ is equal to the difference between two independent Gamma distributed variables, as follows.*

$$\theta = \ln\left(\frac{X}{1-X}\right) = \ln(Y) - \ln(Z), \quad \text{where } \begin{cases} Y \sim \text{Gamma}(\alpha, b) \\ Z \sim \text{Gamma}(\beta, b) \end{cases} \quad b > 0, Y \perp\!\!\!\perp Z$$

Proof. By our definition of Beta Distribution, given in Theorem 6, X can be defined as $Y/(Y+Z)$, where Y and Z and independent Gamma distributed variables, as follows.

$$X = \frac{Y}{Y+Z} \quad \text{for } \begin{cases} Y \sim \text{Gamma}(\alpha, b) \\ Z \sim \text{Gamma}(\beta, b) \end{cases} \quad b > 0, Y \perp\!\!\!\perp Z$$

We can find θ , replacing X by $Y/(Y+Z)$

$$\begin{aligned} \theta &= \ln\left(\frac{X}{1-X}\right) = \ln\left(\left[\frac{Y}{Y+Z}\right] \middle/ \left[1 - \frac{Y}{Y+Z}\right]\right) = \\ &= \ln\left(\left[\frac{Y}{Y+Z}\right] \middle/ \left[\frac{Z}{Y+Z}\right]\right) = \ln\left(\frac{Y}{Z}\right) = \ln(Y) - \ln(Z) \end{aligned} \quad (3.21)$$

□

Lemma 6 (Mean of the logit of the parameter of a Beta Distribution is the difference of Digamma functions). *The mean of θ (i.e., $\mathbb{E}[\theta]$) is equal to $\Psi(\alpha) - \Psi(\beta)$, where Ψ is the Digamma function defined in Theorem 2, as follows.*

$$\mathbb{E}[\theta] = \Psi(\alpha) - \Psi(\beta) \quad (3.22)$$

Proof. As in Equation 3.21, we can define θ as the sum of $\ln$ of two independent Gamma distributed variables Y and Z , as follows.

$$\theta = \ln(Y) - \ln(Z) \quad \text{for } \begin{cases} Y \sim \text{Gamma}(\alpha, b) \\ Z \sim \text{Gamma}(\beta, b) \end{cases} \quad b > 0, Y \perp\!\!\!\perp Z$$

The expected value of θ , can be found as follows.

$$\mathbb{E}[\theta] = \mathbb{E}[\ln(Y) - \ln(Z)] \stackrel{Y \perp\!\!\!\perp Z}{=} \mathbb{E}[\ln(Y)] - \mathbb{E}[\ln(Z)]$$

Using the definition of Gamma distribution (Theorem 3.5), we can find the expected value of $\ln(Y)$ as follows.

$$\begin{aligned} \mathbb{E}[\ln(Y)] &= \int_0^\infty \frac{b^\alpha}{\Gamma(\alpha)} y^{\alpha-1} e^{-yb} \ln(y) dy = \\ &= \frac{1}{\Gamma(\alpha)} \int_0^\infty b^\alpha y^{\alpha-1} e^{-yb} \ln(y) dy \end{aligned}$$

Making $z = yb$, we have $dz = bdy$.

$$\begin{aligned} \mathbb{E}[\ln(Y)] &= \frac{1}{\Gamma(\alpha)} \int_0^\infty b^\alpha \left(\frac{z}{b}\right)^{\alpha-1} e^{-z} [\ln(z) - \ln(b)] \frac{dz}{b} = \\ &= \frac{1}{\Gamma(\alpha)} \int_0^\infty b z^{\alpha-1} e^{-z} [\ln(z) - \ln(b)] \frac{dz}{b} = \\ &= \frac{1}{\Gamma(\alpha)} \int_0^\infty z^{\alpha-1} e^{-z} [\ln(z) - \ln(b)] dz = \\ &= -\frac{\ln(b)}{\Gamma(\alpha)} \underbrace{\int_0^\infty z^{\alpha-1} e^{-z} dz}_{\Gamma(\alpha), \text{ as in Theorem 1}} + \frac{1}{\Gamma(\alpha)} \int_0^\infty z^{\alpha-1} e^{-z} \ln(z) dz = \\ &= -\ln(b) + \frac{1}{\Gamma(\alpha)} \underbrace{\int_0^\infty z^{\alpha-1} e^{-z} \ln(z) dz}_{\Gamma'(\alpha), \text{ as in Equation 3.2}} = \\ &= -\ln(b) + \underbrace{\frac{\Gamma'(\alpha)}{\Gamma(\alpha)}}_{\Psi(\alpha), \text{ as in Theorem 2}} = \Psi(\alpha) - \ln(b) \end{aligned} \quad (3.23)$$

Therefore, the expected value of $\ln(Y)$ is equal to $\Psi(\alpha) - \ln(b)$, and since Z has the same distribution as Y , except for the parameter β , instead of α , the expected value of $\ln(Z)$ is equal to $\Psi(\beta) - \ln(b)$. And we can find the expected value of θ .

$$\mathbb{E}[\theta] = \mathbb{E}[\ln(Y)] - \mathbb{E}[\ln(Z)] = \Psi(\alpha) - \ln(b) - (\Psi(\beta) - \ln(b)) = \Psi(\alpha) - \Psi(\beta)$$

□

Lemma 7 (The Variance of the logit of the parameter of a Beta Distribution is the sum of Trigamma functions). *The variance of θ is equal to $\Psi'(\alpha) - \Psi'(\beta)$, where Ψ' is the Trigamma function defined*

in Theorem 3, as follows.

$$\text{Var}[\theta] = \Psi'(\alpha) + \Psi'(\beta)$$

Proof. We use again the relationship of Equation 3.21, writing θ as the sum of $\ln$ of two independent Gamma distributed variables Y and Z , as follows.

$$\text{Var}[\theta] = \text{Var}[\ln(Y) - \ln(Z)] \stackrel{Y \perp\!\!\!\perp Z}{=} \text{Var}[\ln(Y)] + \text{Var}[\ln(Z)] \quad (3.24)$$

We use the definition of Gamma distribution (Theorem 3.5) to find the variance of $\ln(Y)$ as follows.

$$\text{Var}[\ln(Y)] = \mathbb{E}[\ln(Y)^2] - \left(\mathbb{E}[\ln(Y)]\right)^2$$

We replace the expected value of $\ln(Y)$ with the expression in Equation 3.23 and use the definition of Gamma distribution (Theorem 3.5), as follows.

$$\begin{aligned} \text{Var}[\ln(Y)] &= \mathbb{E}[\ln(Y)^2] - \left(\Psi(\alpha) - \ln(b)\right)^2 = \\ &= \int_0^\infty \frac{b^\alpha}{\Gamma(\alpha)} y^{\alpha-1} e^{-yb} \left(\ln(y)\right)^2 dy - \left(\Psi(\alpha) - \ln(b)\right)^2 \end{aligned} \quad (3.25)$$

Making $z = yb$, we have $dz = bdy$, and we can find $\mathbb{E}[\ln(Y)^2]$ as follows.

$$\begin{aligned} \mathbb{E}[\ln(Y)^2] &= \frac{1}{\Gamma(\alpha)} \int_0^\infty b^\alpha \left(\frac{z}{b}\right)^{\alpha-1} e^{-z} \left[\ln\left(\frac{z}{b}\right)\right]^2 \frac{dz}{b} = \\ &= \frac{1}{\Gamma(\alpha)} \int_0^\infty b^\alpha \left(\frac{z}{b}\right)^{\alpha-1} e^{-z} \left[\ln(z) - \ln(b)\right]^2 \frac{dz}{b} = \\ &= \frac{1}{\Gamma(\alpha)} \int_0^\infty (z)^{\alpha-1} e^{-z} \left[\ln(z) - \ln(b)\right]^2 dz = \\ &= \frac{1}{\Gamma(\alpha)} \int_0^\infty \ln(z)^2 (z)^{\alpha-1} e^{-z} dz - \underbrace{\frac{2 \ln(b)}{\Gamma(\alpha)} \int_0^\infty \ln(z) (z)^{\alpha-1} e^{-z} dz}_{\Gamma'(\alpha) \text{ as in Equation 3.2}} + \underbrace{\frac{\ln(b)^2}{\Gamma(\alpha)} \int_0^\infty (z)^{\alpha-1} e^{-z} dz}_{\Gamma(\alpha) \text{ as in Theorem 1}} = \\ &= \frac{1}{\Gamma(\alpha)} \int_0^\infty \ln(z)^2 (z)^{\alpha-1} e^{-z} dz - 2 \ln(b) \underbrace{\frac{\Gamma'(\alpha)}{\Gamma(\alpha)}}_{\Psi(\alpha) \text{ as in Theorem 2}} + \frac{\ln(b)^2}{\Gamma(\alpha)} \Gamma(\alpha) = \\ &= \frac{1}{\Gamma(\alpha)} \int_0^\infty \ln(z)^2 (z)^{\alpha-1} e^{-z} dz - 2\Psi(\alpha) \ln(b) + \ln(b)^2 = \\ &= \frac{1}{\Gamma(\alpha)} \int_0^\infty \ln(z)^2 (z)^{\alpha-1} e^{-z} dz - \left(\Psi(\alpha)\right)^2 + \left(\Psi(\alpha) - \ln(b)\right)^2 \end{aligned}$$

Taking the derivative of the term $z^{\alpha-1}$ (inside the integral), with respect to the variable α , we have the following result.

$$\begin{aligned} (z)^{\alpha-1} &= e^{\ln((z)^{\alpha-1})} = e^{(\alpha-1)\ln(z)} \\ \frac{d}{d\alpha} (z)^{\alpha-1} &= \left(\ln(z)\right) e^{(\alpha-1)\ln(z)} = \left(\ln(z)\right) (z)^{\alpha-1} \\ \frac{d^2}{d\alpha^2} (z)^{\alpha-1} &= \left(\ln(z)\right)^2 e^{(\alpha-1)\ln(z)} = \left(\ln(z)\right)^2 (z)^{\alpha-1} \end{aligned}$$

We can then replace the term $\left[\ln(z)^2 (z)^{\alpha-1} \right]$ by $\left[\frac{d^2}{d\alpha^2} (z)^{\alpha-1} \right]$ in the integral, as follows.

$$\begin{aligned}
\mathbb{E} [\ln(Y)^2] &= \frac{1}{\Gamma(\alpha)} \int_0^\infty \frac{d^2}{d\alpha^2} (z)^{\alpha-1} e^{-z} dz - \left(\Psi(\alpha) \right)^2 + \left(\Psi(\alpha) - \ln(b) \right)^2 = \\
&\underbrace{\frac{1}{\Gamma(\alpha)} \frac{d^2}{d\alpha^2} \int_0^\infty (z)^{\alpha-1} e^{-z} dz}_{\Gamma(\alpha) \text{ as in Theorem 1}} - \underbrace{\left(\Psi(\alpha) \right)^2}_{\frac{\Gamma'(\alpha)}{\Gamma(\alpha)} \text{ as in Theorem 2}} + \left(\Psi(\alpha) - \ln(b) \right)^2 = \\
&\frac{1}{\Gamma(\alpha)} \frac{d^2}{d\alpha^2} \Gamma(\alpha) - \left(\frac{\Gamma'(\alpha)}{\Gamma(\alpha)} \right)^2 + \left(\Psi(\alpha) - \ln(b) \right)^2 = \\
&\frac{1}{\Gamma(\alpha)} \frac{d}{d\alpha} \Gamma'(\alpha) - \left(\frac{\Gamma'(\alpha)}{\Gamma(\alpha)} \right)^2 + \left(\Psi(\alpha) - \ln(b) \right)^2 = \\
&\underbrace{\left(\frac{d}{d\alpha} \Gamma'(\alpha) \Gamma(\alpha) - \Gamma'(\alpha) \frac{d}{d\alpha} \Gamma(\alpha) \right)}_{\text{quotiente rule of derivative for } \Gamma'(\alpha) / \Gamma(\alpha)} \bigg/ \left(\Gamma(\alpha) \right)^2 + \left(\Psi(\alpha) - \ln(b) \right)^2 = \\
&\frac{d}{d\alpha} \frac{\Gamma'(\alpha)}{\Gamma(\alpha)} + \left(\Psi(\alpha) - \ln(b) \right)^2 = \\
&\underbrace{\frac{d^2}{d\alpha^2} \ln(\Gamma(\alpha))}_{\Psi'(\alpha) \text{ according to Theorem 3}} + \left(\Psi(\alpha) - \ln(b) \right)^2 = \\
&\Psi'(\alpha) + \left(\Psi(\alpha) - \ln(b) \right)^2
\end{aligned}$$

Adding this result of $\mathbb{E} [\ln(Y)^2]$ back to Equation 3.25, we can find $\text{Var} [\ln(Y)]$, as follows.

$$\begin{aligned}
\text{Var} [\ln(Y)] &= \mathbb{E} [\ln(Y)^2] - \left(\Psi(\alpha) - \ln(b) \right)^2 \\
&= \Psi'(\alpha) + \left(\Psi(\alpha) - \ln(b) \right)^2 - \left(\Psi(\alpha) - \ln(b) \right)^2 = \Psi'(\alpha)
\end{aligned} \tag{3.26}$$

Since the variable Z in Equation 3.24 has the same distribution as Y , except for parameter β replacing α , it's easy to show, using the calculations above that the variance of Z is equal to $\Psi'(\beta)$. We can thus find the variance of θ , using Equation 3.24, as follows.

$$\text{Var} [\theta] = \text{Var} [\ln(Y) - \ln(Z)] \stackrel{Y \perp\!\!\!\perp Z}{=} \text{Var} [\ln(Y)] + \text{Var} [\ln(Z)] = \Psi'(\alpha) + \Psi'(\beta)$$

□

Lemma 8 (Approximation of log of a Gamma Distribution). *Let Y be variable having a Gamma distribution with parameters (α, b) . And let $W = \ln(Y)$, according to Equation 3.6 the pdf of L is the following.*

$$f_W(w|\alpha, b) = \frac{b^\alpha}{\Gamma(\alpha)} e^{w\alpha - be^w} \tag{3.27}$$

The mean and variance of $\ln(Y)$ (Equations 3.23 and 3.26) are the followings.

$$\mu_W = \mathbb{E} [\ln(Y)] = \Psi(\alpha) - \ln(b) \qquad \sigma_W^2 = \text{Var} [\ln(Y)] = \Psi'(\alpha)$$

The Normal Distribution with parameters mean and variance given above will have the following pdf.

$$f_W(w|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} e^{w\alpha - \beta e^w}$$

$$f_W(w|\alpha, \beta) \approx \frac{1}{\sqrt{2\Psi'(\alpha)}\pi} \exp \left\{ -\frac{(w - (\Psi(\alpha) - \ln b))^2}{2\Psi'(\alpha)} \right\} \quad (3.28)$$

The Kullback-Leibler divergence between these two distribution can be found as follows.

$$p(w) = \frac{1}{\sqrt{2\Psi'(\alpha)}\pi} \exp \left\{ -\frac{(w - (\Psi(\alpha) - \ln b))^2}{2\Psi'(\alpha)} \right\}$$

$$q(w) = \frac{\beta^\alpha}{\Gamma(\alpha)} e^{w\alpha - \beta e^w}$$

$$D_{KL} = \int_{-\infty}^{\infty} p(w) \ln \left(\frac{p(w)}{q(w)} \right) dw$$

$$= \underbrace{\int_{-\infty}^{\infty} p(w) \ln(p(w)) dw}_{(-) \text{ entropy of } p(w)} - \underbrace{\int_{-\infty}^{\infty} p(w) \ln(q(w)) dw}_{\text{cross-entropy of } p(w) \text{ and } q(w)} \quad (3.29)$$

Lemma 9 (Entropy of Normal Distribution). *The entropy of $p(x)$ is the entropy of a Normal Distribution, and it's defined as follows.*

$$H(p(x)) = - \int_{-\infty}^{\infty} p(w) \ln(p(w)) dw = \frac{1}{2} (1 + \ln(2\Psi'(\alpha)\pi)) \quad (3.30)$$

Proof. Let W be a random variable that follows a Normal Distribution with parameters (μ, σ^2) .

$$W \sim \mathcal{N}(\mu, \sigma^2)$$

$$R(w) = f_W(w|\mu, \sigma^2) = \frac{1}{\sqrt{2\sigma^2\pi}} \exp \left\{ -\frac{(w - \mu)^2}{2\sigma^2} \right\}$$

The entropy of $R(w)$ will be the following.

$$H(R(w)) = - \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\sigma^2\pi}} \exp \left\{ -\frac{(w - \mu)^2}{2\sigma^2} \right\} \ln \left(\frac{1}{\sqrt{2\sigma^2\pi}} \exp \left\{ -\frac{(w - \mu)^2}{2\sigma^2} \right\} \right) dw$$

$$= \frac{1}{2} \ln(2\sigma^2\pi) \underbrace{\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\sigma^2\pi}} \exp \left\{ -\frac{(w - \mu)^2}{2\sigma^2} \right\} dw}_{\int_{-\infty}^{\infty} N(\mu, \sigma^2) = 1} + \frac{1}{2\sigma^2} \underbrace{\int_{-\infty}^{\infty} \frac{(w - \mu)^2}{\sqrt{2\sigma^2\pi}} \exp \left\{ -\frac{(w - \mu)^2}{2\sigma^2} \right\} dw}_{\mathbb{E}[(w - \mu)^2] = \sigma^2}$$

$$= \frac{1}{2} \ln(2\sigma^2\pi) + \frac{1}{2\sigma^2} \sigma^2$$

$$= \frac{1}{2} (1 + \ln(2\sigma^2\pi))$$

□

The cross entropy of $p(w)$ and $q(w)$ is the following.

$$\begin{aligned}
p(w) &= \frac{1}{\sqrt{2\Psi'(\alpha)\pi}} \exp \left\{ -\frac{(w - (\Psi(\alpha) - \ln b))^2}{2\Psi'(\alpha)} \right\} \\
q(w) &= \frac{b^\alpha}{\Gamma(\alpha)} e^{w\alpha - be^w} \implies \ln(q(w)) = \alpha \ln(b) - \ln(\Gamma(\alpha)) + w\alpha - b \exp\{w\} \\
H(p(w), q(w)) &= - \int_{-\infty}^{\infty} p(w) \ln(q(w)) dw \\
&= -\alpha \ln(b) \underbrace{\int_{-\infty}^{\infty} p(w) dw}_{=1} + \ln(\Gamma(\alpha)) \underbrace{\int_{-\infty}^{\infty} p(w) dw}_{=1} - \alpha \underbrace{\int_{-\infty}^{\infty} wp(w) dw}_{\mathbb{E}_{p(w)}[w] = \Psi(\alpha) - \ln b} + b \int_{-\infty}^{\infty} e^w p(w) dw \\
&= -\alpha \ln(b) + \ln(\Gamma(\alpha)) - \alpha(\Psi(\alpha) - \ln b) + b \int_{-\infty}^{\infty} e^w p(w) dw \\
&= \ln(\Gamma(\alpha)) - \alpha\Psi(\alpha) + b \underbrace{\int_{-\infty}^{\infty} e^w p(w) dw}_{\mathbb{E}_{p(w)}[e^w]} \tag{3.31}
\end{aligned}$$

The last term $(\mathbb{E}_{p(w)}[e^w])$ is the following.

$$\mathbb{E}_{p(w)}[e^w] = \exp \left\{ \Psi(\alpha) - \ln b + \frac{\Psi'(\alpha)}{2} \right\}$$

Proof. For $W \sim \mathcal{N}(\mu, \sigma^2)$, we find $\mathbb{E}[e^w]$ as follows.

$$\begin{aligned}
\mathbb{E}[e^w] &= \int_{-\infty}^{\infty} e^w p(w) dw \\
&= \int_{-\infty}^{\infty} e^w \frac{1}{\sqrt{2\sigma^2\pi}} \exp \left\{ -\frac{(w - \mu)^2}{2\sigma^2} \right\} dw \\
&= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\sigma^2\pi}} \exp \left\{ -\frac{(w - \mu)^2}{2\sigma^2} + w \right\} dw \\
&= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\sigma^2\pi}} \exp \left\{ -\frac{w^2 - 2w\mu + \mu^2}{2\sigma^2} + w \right\} dw \\
&= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\sigma^2\pi}} \exp \left\{ -\frac{w^2 - 2w(\mu + \sigma^2) + \mu^2}{2\sigma^2} \right\} dw \\
&= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\sigma^2\pi}} \exp \left\{ -\frac{w^2 - 2w(\mu + \sigma^2) + \mu^2}{2\sigma^2} \right\} dw \\
&= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\sigma^2\pi}} \exp \left\{ -\frac{w^2 - 2w(\mu + \sigma^2) + (\mu + \sigma^2)^2}{2\sigma^2} \right\} dw \exp \left\{ \frac{2\mu\sigma^2 + \sigma^4}{2\sigma^2} \right\} \\
&= \exp \left\{ \mu + \frac{\sigma^2}{2} \right\}
\end{aligned}$$

□

According to Equation 3.31, the cross entropy of $p(x)$ and $q(x)$ will be the following.

$$H(p(w), q(w)) = \ln(\Gamma(\alpha)) - \alpha(\Psi(\alpha)) + be^{\Psi(\alpha) - \ln b + \Psi'(\alpha)/2}$$

Now, using this result of cross entropy, the entropy of $p(x)$ found in Equation 3.30, and the defi-

inition of Kullback-Leibler divergence (Equation 3.29), we can find the divergence between the two distribution with respect to b and α , as follows.

$$\begin{aligned} D_{KL}(p(w), q(w)) &= -H(p(w)) + H(p(w), q(w)) \\ &= -\frac{1}{2} \left(1 + \ln(2\Psi'(\alpha)\pi) \right) + \ln(\Gamma(\alpha)) - \alpha\Psi(\alpha) + be^{\Psi(\alpha) - \ln b + \Psi'(\alpha)/2} \end{aligned}$$

This function is evaluated for different values of parameters α and b , and the results are shown in Figure 3.1. For a fixed $b = 1$, we increase the value of α and find the value of divergences between the two distributions; this result is shown in Table 3.1.

α	$D_{KL}(p(w), q(w))$
1	1.873695e-01
2	6.176739e-02
3	3.599857e-02
4	2.526821e-02
5	1.943434e-02
10	8.991045e-03
20	4.326958e-03
50	1.691923e-03
100	8.396154e-04
200	4.182332e-04
500	1.669169e-04
1000	8.339586e-05

Figure 3.1: The values of Kullback-Leibler Divergence between the Log-Gamma Distribution and the Normal Distribution approximation, for different values of parameters α and b of the Gamma Distribution.

Table 3.1: The values of Kullback-Leibler Divergence between the Log-Gamma Distribution and the Normal Distribution approximation, for different values of the parameter α and a fixed value of the parameter $b = 1$ of the Gamma Distribution.

We can see from Figure 3.1 and Table 3.1 that the approximation of a Log-Gamma Distribution to a Normal Distribution, as given in Equation 3.28, is pretty accurate, according to the measure of Kullback-Leibler Divergence; the approximation becomes more accurate as α increases: for $\alpha \geq 5$, the divergence is lower than 0.02, and decrease even more, approaching 0.0 as the value of α increases.

According to the result above (Lemma 8) the approximation of the Log-Gamma Distributions to Normal Distributions are accurate, as follow.

$$\begin{cases} Y \sim \text{Gamma}(\alpha, b) \\ Z \sim \text{Gamma}(\beta, b) \end{cases} \implies \begin{cases} \ln(Y) \dot{\sim} \mathcal{N}(\Psi(\alpha) - \ln b, \Psi'(\alpha)) \\ \ln(Z) \dot{\sim} \mathcal{N}(\Psi(\beta) - \ln b, \Psi'(\beta)) \end{cases}$$

According to Lemma 5 Distribution of the logit of a Beta distributed variable X is equal to the difference of two independent Gamma distributed variables $\theta = \ln(X/(1-X)) = \ln(Y) - \ln(Z)$.

$$\begin{cases} X \sim \text{Beta}(\alpha, \beta) \\ \theta = \ln(X/(1-X)) \end{cases} \implies \begin{cases} \theta = \ln(Y) - \ln(Z) \\ Y \sim \text{Gamma}(\alpha, b) \\ Z \sim \text{Gamma}(\beta, b) \\ Y \perp\!\!\!\perp Z \end{cases}$$

$$\begin{cases} \ln(Y) \dot{\sim} \mathcal{N}(\Psi(\alpha) - \ln b, \Psi'(\alpha)) \\ \ln(Z) \dot{\sim} \mathcal{N}(\Psi(\beta) - \ln b, \Psi'(\beta)) \\ \theta = \ln(Y) - \ln(Z) \end{cases} \implies \theta \dot{\sim} \mathcal{N}(\Psi(\alpha) - \Psi(\beta), \Psi'(\alpha) + \Psi'(\beta))$$

Proof.

$$\begin{aligned} X &\sim \mathcal{N}(\mu_x, \sigma_x^2) & Y &\sim \mathcal{N}(\mu_y, \sigma_y^2) \\ M_X(t) &= \exp\left\{\mu_X t + \frac{1}{2}\sigma_X^2 t^2\right\} & M_Y(t) &= \exp\left\{\mu_Y t + \frac{1}{2}\sigma_Y^2 t^2\right\} \\ M_{X-Y}(t) &= \exp\left\{\mu_X t + \frac{1}{2}\sigma_X^2 t^2\right\} \exp\left\{-\mu_Y t + \frac{1}{2}\sigma_Y^2 t^2\right\} = \\ &\exp\left\{(\mu_X - \mu_Y)t + \frac{1}{2}(\sigma_X^2 + \sigma_Y^2)t^2\right\} \\ \therefore &\boxed{X - Y \sim \mathcal{N}(\mu_x - \mu_y, \sigma_x^2 + \sigma_y^2)} \end{aligned}$$

□

3.3 Examples of logistic Normal Approximation

Figure 3.2: For $X \sim \text{Beta}(\alpha, \beta)$ and $\theta = \ln(X/(1 - X))$; the distribution of θ is shown in red, and the Normal Distribution $\mathcal{N}(\Psi(\alpha) - \Psi(\beta), \Psi'(\alpha) + \Psi'(\beta))$ is shown in blue; for different values of parameters α and β ($\alpha \in \{1; 10; 100\}$, $\beta \in \{2; 20; 200\}$).

Figure 3.3: For $X \sim \text{Beta}(\alpha, \beta)$ and $\theta = \ln(X/(1-X))$; the distribution of θ is shown in red, and the Normal Distribution $\mathcal{N}(\Psi(\alpha) - \Psi(\beta), \Psi'(\alpha) + \Psi'(\beta))$ is shown in blue; for different values of parameters α and β ($\alpha \in \{100; 500\}$, $\beta \in \{1,000; 10,000\}$).

Figure 3.4: Example of Logist-Normal approximation, for values of α and β from peak found in a control sample of ChIP-Seq dataset.

Figure 3.5: Example of Logist-Normal approximation, for values of α and β from peak found in a treatment sample of ChIP-Seq dataset.

Chapter 4

Model (Statistics)

Consider that for a given chromosomal region of the DNA, with base pairs between the positions b_1 and b_k , we want to decide if a peak found in this region is significant. In other words, if the region of the genome truly represents a binding site of a protein of interest.

In order to accomplish that, we need to verify if the probability of the peak found in this region, for the treatment sample, is significantly different from the probability of occurrence of the same peak (in the same genomic region), when using a control sample. Considering that the occurrence of peaks when using the control sample is random, we can assume that when the probability of the peak given the treatment sample is truly different, the peak in the treatment sample is not random as well.

The Bayesian model described in Section 4.1 was proposed to solve this problem. And to overcome the difficulties of using multiple replicate, we used Meta-Analysis described in Section 4.1.1.

4.1 Categorical Bayesian Model

Consider the set of counts of DNA sequences (*reads*) aligned to a given region of the genome, for base pairs (bps) between b_1 and b_k , to be given by a vector $\mathbf{n}$ defined as follows.

$$\mathbf{n} = [n_1, n_2, \dots, n_k] \quad (4.1)$$

In order to model this dataset, we consider the positions in the genome (i.e., bp in the chromosome), $b_1, b_2, \dots, b_k$, to be independent categories, and the number of alignments, $n_1, n_2, \dots, n_k$, to be the number of *successes* of each category. It is important to mention that the independence assumption here is an approximation, since although the alignment between a bp of the read and a bp of the chromosome (known as a "match") is independent of the alignment of any previous bps, the bps are grouped in sequences (*reads*) and these sequences are aligned to the genome as a whole, instead of each bp aligned independently. But we believe this is a reasonable approximation, and it's necessary to make the solution feasible in a reasonable time, especially for very large genomes.

The probability of obtaining the alignment counts $\mathbf{n}$ (Equation 4.1) between bps b_1 and b_k can be modelled under a Bayesian framework, using a Multinomial distribution as likelihood (probability of obtaining the number of successes equal to $\mathbf{n}$, given the probability of success for bps between b_1 and b_k), as described below.

Consider the probabilities vector $\Theta = [\theta_1, \theta_2, \dots, \theta_k]$, where θ_i represents the probability of having one alignment at position b_i (i.e., probability of success of category b_i). The probability of obtaining the vector of alignment counts $\mathbf{n}$, given this vector Θ is defined as follows.

$$p(\mathbf{n}|\Theta) = \frac{n!}{n_1! \dots n_k!} \theta_1^{n_1} \theta_2^{n_2} \dots \theta_k^{n_k}, \quad \text{for } n = \sum_{i=1}^k n_i \text{ and } \sum_{i=1}^k \theta_i = 1 \quad (4.2)$$

To model the prior knowledge about the probabilities vector Θ , we can use a Dirichlet distribu-

tion [Ferguson \(1973\)](#) as a prior distribution, which is the natural conjugate of the Multinomial distribution [de Bragança Pereira and Stern \(2008\)](#).

For a probability vector Θ , given a known alignment counts vector $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_k]$ (prior knowledge regarding alignment bias in the genome), the Dirichlet distribution is defined as follows.

$$p(\Theta|\alpha) = \frac{\Gamma\left(\sum_{i=1}^k \alpha_i\right)}{\prod_{i=1}^k \Gamma(\alpha_i)} \theta_1^{\alpha_1-1} \theta_2^{\alpha_2-1} \dots \theta_k^{\alpha_k-1}, \quad \text{for } \sum_{i=1}^k \theta_i = 1 \quad (4.3)$$

For the distribution in Equation 4.3 (Dirichlet distribution), $\alpha_i - 1$ represents the number of reads previously mapped to the genome at position i . If we don't have any previous knowledge about these alignments, we can use $\alpha_i = 1$, for all i ; which is equivalent to use the Uniform distribution for all categories $b_1, \dots, b_k$. But if, instead, we have any knowledge a priori about a bias in mapping the reads to certain regions of the genome, we can promptly add this information as the prior distribution in this model.

Finally, the distribution of the probabilities vector Θ given the alignments vector $\mathbf{n}$ (posterior Distribution) will be a Dirichlet distribution as well, which is proportional to the product of the distributions given in Equations 4.3 (priori) and 4.2 (likelihood).

$$p(\Theta|\mathbf{n}) \propto p(\Theta|\alpha) \cdot p(\mathbf{n}|\Theta) \quad (4.4)$$

Therefore, the posterior distribution will be as follows.

$$p(\Theta|\mathbf{n}) = \frac{\Gamma\left(\sum_{i=1}^k [\alpha_i + n_i]\right)}{\prod_{i=1}^k \Gamma(\alpha_i + n_i)} \theta_1^{(\alpha_1+n_1)-1} \theta_2^{(\alpha_2+n_2)-1} \dots \theta_k^{(\alpha_k+n_k)-1} \quad (4.5)$$

The distribution in Equation 4.5 is the joint distribution of the probabilities of the alignment counts found for bps $b_1, b_2, \dots, b_n$; and, for a given peak, we want to compare this distribution between treatment and control samples.

Once we consider the alignments found for a peak in the control sample to be *random* alignments (resulting from alignment biases), when the region is a true binding sites, we expect the peaks for the treatment sample to have very different probabilities. In other words, for a true binding site of the protein of interest the alignments found for treatment sample are *non-random* alignments.

In order to compare these probabilities in *treatment* and *control* samples for a given peak p , we use the measure *log-odds*, which is given by the ratio between a probability θ_p and its complement.

$$\begin{aligned} \text{odds}(p) &= \frac{\theta_p}{1 - \theta_p} \\ \log\text{-odds}(p) &= \log(\text{odds}(p)) = \log\left(\frac{\theta_p}{1 - \theta_p}\right) \end{aligned} \quad (4.6)$$

The log-odds can also be defined as the *logit* function of a probability.

The Dirichlet distribution can be approximated to a Logit-Normal distribution, as described in [Atchison and Shen \(1980\)](#). This approximation is written in terms of Digamma (Ψ) and Trigamma (Ψ') functions as follows.

$$\begin{aligned} \text{If } (\theta_1, \theta_2, \dots, \theta_k) &\sim \text{Dir}(\alpha_1, \alpha_2, \dots, \alpha_k), \\ (\theta_1, \theta_2, \dots, \theta_k) &\simeq N(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ \mu_i &= \Psi(\alpha_i) - \Psi(\alpha_k) \quad i = 1, \dots, k-1 \\ \Sigma_{ii} &= \Psi'(\alpha_i) - \Psi'(\alpha_k) \quad i = 1, \dots, k-1 \\ \Sigma_{ij} &= \Psi'(\alpha_k) \quad i \neq j \end{aligned} \quad (4.7)$$

This approximation becomes more accurate as the parameters α_i increase. In our case, the parameters α_i are the number of alignments in genome positions, and as we describe in the next section, we will use the sum of alignments for large portions of the genome. Therefore, the parameters α_i used in our model are extremely large and this approximation becomes very accurate, as shown in Appendix Section.

Throughout the rest of this paper, we will simplify the notation for the Dirichlet distribution, removing the constant and replacing the equal sign with the mathematical symbol of proportionality, which is a common practice, as follows.

$$p(\Theta|\mathbf{n}) \propto \theta_1^{(\alpha_1+n_1)-1} \theta_2^{(\alpha_2+n_2)-1} \dots \theta_k^{(\alpha_k+n_k)-1} \quad (4.8)$$

Models for treatment and control samples For a given region of length k of the genome DNA sequence (whole chromosomal DNA sequence) with bps between b_1 and b_k , consider the vector of alignment counts and the probability vector, for treatment and control samples, as follows.

	Alignment counts	Parameters (probabilities)
treatment	$\mathbf{n} = [n_1, n_2, \dots, n_k]$	$\Theta = [\theta_1, \theta_2, \dots, \theta_k]$
control	$\mathbf{n}' = [n'_1, n'_2, \dots, n'_k]$	$\Theta' = [\theta'_1, \theta'_2, \dots, \theta'_k]$

$$\text{for } \sum_{i=1}^k \theta_i = \sum_{i=1}^k \theta'_i = 1.$$

Given the parameters vector $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_k]$ of the prior distribution for the *treatment* sample, the posterior distribution for the *treatment* sample will be given as follows.

$$p(\Theta|\mathbf{n}) \propto \theta^{(n_1+\alpha_1)-1} \cdot \theta^{(n_2+\alpha_2)-1} \dots \theta^{(n_k+\alpha_k)-1} \quad (4.9)$$

Given the parameters vector $\alpha' = [\alpha'_1, \alpha'_2, \dots, \alpha'_k]$ of the prior distribution for the *control* sample, the posterior distribution of the *control* sample will be given as follows.

$$p(\Theta'|\mathbf{n}') \propto \theta'^{(n'_1+\alpha'_1)-1} \cdot \theta'^{(n'_2+\alpha'_2)-1} \dots \theta'^{(n'_k+\alpha'_k)-1} \quad (4.10)$$

Now, consider we have a list of candidate peaks found for the treatment sample, and we want to find out if a specific peak p is significant (i.e., if it's a true binding site). This peak p begins at the bp position b_{init} and ends at bp position b_{end} of a chromosome, as shown in Fig. 4.1 and Fig. 4.2.

Since we consider the categories b_i (for $b_i \leq b_i \leq b_k$) to be independent, we can define the probability of this peak, which we will refer to as θ_p , as the sum of the parameters θ_i within this region. We can also define the probability of the remaining peaks, θ_R , and the the probability of the regions with no candidate peaks as $\theta_{\emptyset}$, both as the sum of the parameter θ_i within their respective regions.

Consider the set of all base pairs that fall within regions with no candidate peaks as $S_{\emptyset}$, and the set of all base pairs that fall within the remaining candidate peaks (all peaks except p) as S_R . We define θ_p , θ_R and $\theta_{\emptyset}$ as follows,

$$\theta_p = \sum_{i=b_{init}}^{b_{end}} \theta_i \quad , \quad \theta_R = \sum_{i \in S_R} \theta_i \quad \text{and} \quad \theta_{\emptyset} = \sum_{i \in S_{\emptyset}} \theta_i \quad (4.11)$$

Given that the sum of parameters θ_i , for i from 1 to k i equal to 1, the sum of probabilities θ_R (remaining candidate peaks) and $\theta_{\emptyset}$ (no candidate peak) will be the complement of the probability θ_p .

$$\theta_{\emptyset} + \theta_R = 1 - \theta_p \quad (4.12)$$

Figure 4.1: Example of peak region found between base pair b_{init} and b_{end} .

To define the distributions and relationships between the probabilities θ_p , θ_R and $\theta_\emptyset$, we will use three properties of the Dirichlet distribution. They are: Dirichlet's additive property, Dirichlet's marginal distribution Beta [Ferguson \(1973\)](#), and Dirichlet's neutrality [James and Mosimann \(1980\)](#).

The additive property, neutrality and marginal distribution are detailed in Appendix 1.

The additive property of the Dirichlet distribution says that given a set of variables distributed according to a Dirichlet distribution, if we sum one or more of these parameters, this sum together with the remaining variable are also Dirichlet distributed (Equation 4.13). The marginal distribution of a Dirichlet distribution is a Beta distribution (Equation 4.14).

Additive property of Dirichlet distribution

$$\text{For } (\theta_1, \theta_2, \dots, \theta_k) \sim \text{Dir}(\alpha_1, \alpha_2, \dots, \alpha_k),$$

$$\left(\sum_{i=1}^r \theta_i, \sum_{i=(r+1)}^m \theta_i, \dots, \theta_k \right) \sim \text{Dir} \left(\sum_{i=1}^r \alpha_i, \sum_{i=(r+1)}^m \alpha_i, \alpha_{m+1}, \dots, \alpha_k \right) \quad (4.13)$$

Marginal property of Dirichlet distribution

For $(\theta_1, \theta_2, \dots, \theta_k) \sim \text{Dir}(\alpha_1, \alpha_2, \dots, \alpha_k)$, and $1 \leq i \leq k$,

$$\theta_i \sim \text{Beta} \left(\alpha_i, \left[\sum_{j=1}^k \alpha_j \right] - \alpha_i \right) \quad (4.14)$$

Figure 4.2: Example of peak region found between base pair b_{init} and b_{end} .

Neutrality property of Dirichlet distribution

For $(\theta_1, \theta_2, \dots, \theta_k) \sim Dir(\alpha_1, \alpha_2, \dots, \alpha_k)$, and $1 \leq i \leq k$,

$$\theta_i \perp\!\!\!\perp \left(\frac{\theta_1}{1-\theta_i}, \dots, \frac{\theta_{i-1}}{1-\theta_i}, \frac{\theta_{i+1}}{1-\theta_i}, \dots, \frac{\theta_k}{1-\theta_i} \right) \quad (4.15)$$

From Equation 4.5 (the posterior distribution of the probabilities of alignment for bases between b_1 and b_k), we use the additive property of Dirichlet distribution, followed by the neutrality property and the property of marginal distribution Beta, to find the distributions of the ration $\theta_p/\theta_\emptyset$ and the distribution of $\theta_\emptyset$, as follows.

$$\begin{aligned}
& \theta_1, \dots, \theta_k | \mathbf{n} \sim \text{Dir} \left(\alpha_1 + n_1, \dots, \alpha_k + n_k \right) \\
& \theta_1, \dots, \sum_{b_{init}}^{b_{end}} [\theta_i], \dots, \theta_k | \mathbf{n} \sim \text{Dir} \left(\alpha_1 + n_1, \dots, \sum_{b_{init}}^{b_{end}} [\alpha_i + n_i], \dots, \alpha_k + n_k \right) \\
& \theta_1, \dots, \theta_{\mathbf{p}}, \dots, \theta_k | \mathbf{n} \sim \text{Dir} \left(\alpha_1 + n_1, \dots, \sum_{i=b_{init}}^{b_{end}} [\alpha_i + n_i], \dots, \alpha_k + n_k \right) \\
& \theta_{\mathbf{p}}, \sum_{j \in S_{\emptyset}} [\theta_j], \sum_{l \in S_R} [\theta_l] | \mathbf{n} \sim \text{Dir} \left(\sum_{b_{init}}^{b_{end}} [\alpha_i + n_i], \sum_{j \in S_{\emptyset}} [\alpha_j + n_j], \sum_{l \in S_R} [\alpha_l + n_l], \right) \\
& \theta_{\mathbf{p}}, \theta_{\emptyset}, \theta_R | \mathbf{n} \sim \text{Dir} \left(\sum_{b_{init}}^{b_{end}} [\alpha_i + n_i], \sum_{j \in S_{\emptyset}} [\alpha_j + n_j], \sum_{l \in S_R} [\alpha_l + n_l], \right) \\
& \left(\theta_{\emptyset} \perp \left(\frac{\theta_{\mathbf{p}}}{\theta_{\emptyset}}, \frac{\theta_R}{\theta_{\emptyset}} \right) | \mathbf{n} \right) \\
& \theta_{\emptyset} | \mathbf{n} \sim \text{Beta} \left(\sum_{j \in S_{\emptyset}} [\alpha_j + n_j], \sum_{l \notin S_{\emptyset}} [\alpha_l + n_l] \right) \\
& \frac{\theta_{\mathbf{p}}}{\theta_{\emptyset}} | \mathbf{n} \sim \text{Beta} \left(\sum_{i=b_{init}}^{b_{end}} [\alpha_i + n_i], \sum_{j \in S_R} [\alpha_j + n_j] \right)
\end{aligned} \tag{4.16}$$

Therefore, the probability of the region with no candidate peak ($\theta_{\emptyset}$) will have a Beta distribution, with parameters equal to: (1) sum of the number of alignments found in the region with no candidate peak (plus parameters α_i of the prior distribution in this region); and (2) the sum of the alignments found in the remaining region of the chromosome (plus parameters α_i of the prior distribution in this region).

The ratio θ_p/θ_0 will have a Beta distribution, as well, with parameters: (1) sum of the number of alignments found in the region of the candidate peak p (plus parameters α_i of the prior distribution); and (2) the sum of the alignments found in the region of the remaining peaks, set S_R (plus parameters α_i of the prior distribution).

And the variables $\theta_{\emptyset}$ and θ_p/θ_0 are independent.

Finally, we can define a measure of score for a peak p and the score's distribution using the distributions of $\theta_{\emptyset}$ and $\theta_p/\theta_{\emptyset}$, and the approximated distribution of their log-odds, as described in Equation 4.7.

$$\begin{aligned}
& \text{For } \underbrace{\sum_{j \in S_{\emptyset}} [n_j]}_{\text{total noise}} < \underbrace{\sum_{l \notin S_{\emptyset}} [n_l]}_{\text{all candidate peaks}} \quad \text{in the treatment samples,} \\
& \text{score}(p) = \underbrace{\log\text{-odds} \left(\frac{p}{S_{\emptyset}} \right)}_{\text{odds of peak } p \text{ over noise}} + \underbrace{\log\text{-odds} \left(S_{\emptyset} \right)}_{\text{odds of noise in chromosome}}
\end{aligned} \tag{4.17}$$

Notice that the first term of the equation above controls the odds of the peak with respect to the number of random alignments in the chromosome, and the second term, the odds of the random alignments. In other words, a large value for the first term might be misleading: if $S_{\emptyset}$ is really low,

$p/S_{\emptyset}$ will be large, regardless the value of p ; that is the reason why the second term is so important to identify truly significant peaks. In the Results Section we show how the number of significant peaks can change before and after considering this term.

The condition for using this score is that the total number of alignments in the region $S_{\emptyset}$ is strictly lower than the total alignment in the remaining regions of the chromosome. In other words, the sum of alignments for all the candidate peaks is necessarily greater than the sum of alignments in regions with no candidate peaks. We consider that this assumption is very reasonable since the noise should not be greater than the signal or the results would not be accurate.

The distribution of logg-odds of $p/S_{\emptyset}$ will be given as follows.

$$\begin{aligned} \text{log-odds} \left(\frac{p}{S_{\emptyset}} \right) &= \text{logit} \left(\frac{\theta_p}{\theta_{\emptyset}} | \mathbf{n} \right) \simeq N \left(\mu_{p/S_{\emptyset}}, \sigma_{p/S_{\emptyset}} \right) \\ \mu_{p/S_{\emptyset}} &= \Psi \left(\sum_{b_{init}}^{b_{end}} [\alpha_i + n_i] \right) - \Psi \left(\sum_{j \in S_R} [\alpha_j + n_j] \right) \\ \sigma_{p/S_{\emptyset}} &= \Psi' \left(\sum_{b_{init}}^{b_{end}} [\alpha_i + n_i] \right) + \Psi' \left(\sum_{j \in S_R} [\alpha_j + n_j] \right) \end{aligned} \quad (4.18)$$

And the distribution of the log-odds of $S_{\emptyset}$ will be given as follows.

$$\begin{aligned} \text{log-odds} (S_{\emptyset}) &= \text{logit} (\theta_{\emptyset} | \mathbf{n}) \simeq N (\mu_{S_{\emptyset}}, \sigma_{S_{\emptyset}}) \\ \mu_{S_{\emptyset}} &= \Psi \left(\sum_{j \in S_{\emptyset}} [\alpha_j + n_j] \right) - \Psi \left(\sum_{l \notin S_{\emptyset}} [\alpha_l + n_l] \right) \\ \sigma_{S_{\emptyset}} &= \Psi' \left(\sum_{j \in S_{\emptyset}} [\alpha_j + n_j] \right) + \Psi' \left(\sum_{l \notin S_{\emptyset}} [\alpha_l + n_l] \right) \end{aligned} \quad (4.19)$$

Since $\theta_p/\theta_{\emptyset}$ is independent of $\theta_{\emptyset}$ (Equation 4.16), the distribution of the sum of their logit functions will be given as the distribution of the sum of independent Normal distributed variables, as follows.

$$\boxed{\text{score}(p) = \text{log-odds} \left(\frac{p}{S_{\emptyset}} \right) + \text{log-odds} (S_{\emptyset}) \simeq N \left(\mu_{p/S_{\emptyset}} + \mu_{S_{\emptyset}}, \sigma_{p/S_{\emptyset}} + \sigma_{S_{\emptyset}} \right)} \quad (4.20)$$

We find the distributions of this log-odds based score for each peak p in both treatment and control samples (in the same chromosomal region), and we compare these distributions (for control and treatment samples) in order to decide if the peak p in the treatment sample is significant, as show in the next Section Meta-Analysis.

4.1.1 Meta-Analysis

For each peak found in the treatment sample, we find the distribution of the log-odds based score of the peak, shown in Equation 4.17, for both treatment and control samples, using the same positions in the reference genome.

For each replicate, we have the distribution of the score of the peak, and we use the weighted average of these distributions to compare the scores of each peak in treatment and control samples. This average is weighted using the total number bp alignments for each sample replicate.

Fig. 4.3 shows examples of regions where peaks have been found for treatment and control samples using an antibody to recognize human SOX3, and the respective alignment counts for each replicate. Fig. 4.4 shows the corresponding approximated Normal distribution of the score of this peak, for all the replicates.

Finally, Fig. 4.5 shows the resulting weighted average of the replicates. This distribution averaged across all the replicates is the distribution used to evaluate each peak. Using this distribution, we can find the probability of the score found in the treatment sample to be different from the score of the peak found in the control sample using Equation 4.21.

Consider $score(p)$ the score for the region of a candidate peak p in the treatment sample and $score(p')$ the score for the same region in the control sample.

$$p(score(p) > score(p')) = \int_{-\infty}^{\infty} \int_{-\infty}^{score(p)} f(score(p), score(p')) \partial [score(p')] \partial [score(p)]$$

Since $score(p)$ and $score(p')$ are independent, we have:

$$\begin{aligned} f(score(p), score(p')) &= f(score(p)) \cdot f(score(p')), \text{ thus,} \\ p(score(p) > score(p')) &= \int_{-\infty}^{\infty} \int_{-\infty}^{score(p)} [f(score(p))f(score(p'))] \partial [score(p')] \partial [score(p)] \end{aligned} \quad (4.21)$$

Candidate peaks with values of probability, given by Equation 4.21, close to 1 will be considered significant binding sites (the score of these peaks for the treatment samples are certainly greater than their score for the control sample).

Figure 4.3: Example of peaks found for treatment and control samples of SOX3 experiment aligned to mm10 reference genome.

Figure 4.4: Normal distribution of the log-odds based score of replicates for treatment and control samples aligned to mm10 genome.

Figure 4.5: Meta-analysis: comparison of weighted average of distributions of the score for a peak p , given control and treatment samples.

Chapter 5

Workflow (Computer Science)

After obtaining the data resulting from ChIP-Seq technique (i.e., reads), described in Chapter 2, the first step of the data analysis is the alignment of these reads to a reference genome.

The reads alignment was performed using the tool Bowtie 2 (Langmead and Salzberg, 2012). Bowtie is a tools that executes a fast alignment, even for a very large reference genome, and it allows the user, by specifying parameters used by this tool, to select many execution options. For example, the format of the output file containing the alignments found. Section 5.1 described the algorithm used and section 5.2 a smoothing method for peak identification.

5.1 Algorithm

```
if IsValid(command.line) then
    list.ctrl.path  $\leftarrow$  control samples file names
    list.treatment.path  $\leftarrow$  treatment samples file names
    list.candidate.peaks  $\leftarrow$  list of candidate peaks
else
    Issue(usage.error.msg)
    Exit()
end if

ctrl.alignments  $\leftarrow$  ReadBamByChuks(list.ctrl.path, chunk.size)
treatment.alignments  $\leftarrow$  ReadBamByChuks(list.ctrl.path, chunk.size)
ctrl.coverage  $\leftarrow$  FindCoverage(ctrl.alignments)
treatment.coverage  $\leftarrow$  FindCoverage(treatment.alignments)
     $\triangleright$  Find the mean and variance of noise based on non-candidate peaks region
mu.noise  $\leftarrow$   $\frac{\text{digamma}(\alpha.\text{total}) - \text{digamma}(\alpha.\text{candidate.peaks} + n.\text{candidate.peaks})}{\text{trigamma}(\alpha.\text{total}) + \text{trigamma}(\alpha.\text{all.candidate.peaks} + n.\text{all.candidate.peaks})}$ 
sigma.noise  $\leftarrow$   $\frac{\text{digamma}(\alpha.\text{total}) - \text{digamma}(\alpha.\text{candidate.peaks} + n.\text{candidate.peaks})}{\text{trigamma}(\alpha.\text{total}) + \text{trigamma}(\alpha.\text{all.candidate.peaks} + n.\text{all.candidate.peaks})}$ 
for candidate.peak in list.candidate.peaks do
    mu.candidate.peak.over.noise  $\leftarrow$   $\frac{\text{digamma}(\alpha.\text{candidate.peak} + n.\text{candidate.peak}) - \text{digamma}(\alpha.\text{all.candidate.peaks} + n.\text{all.candidate.peaks})}{\text{trigamma}(\alpha.\text{candidate.peak} + n.\text{candidate.peak}) + \text{trigamma}(\alpha.\text{all.candidate.peaks} + n.\text{all.candidate.peaks})}$ 
    sigma.candidate.peak.over.noise  $\leftarrow$   $\frac{\text{digamma}(\alpha.\text{candidate.peak} + n.\text{candidate.peak}) - \text{digamma}(\alpha.\text{all.candidate.peaks} + n.\text{all.candidate.peaks})}{\text{trigamma}(\alpha.\text{candidate.peak} + n.\text{candidate.peak}) + \text{trigamma}(\alpha.\text{all.candidate.peaks} + n.\text{all.candidate.peaks})}$ 
    mu.score.candidate.peak  $\leftarrow$  mu.candidate.peak.over.noise + mu.noise
    sigma.score.candidate.peak  $\leftarrow$  sigma.candidate.peak.over.noise + sigma.noise
     $\triangleright$  Find the probability of the score in treatment sample greater than score in control sample
    p.peak  $\leftarrow$  p(score.treatment > score.control)
end for
```

5.2 Peak Smoothing and Identification

For the peak identification, we search for local maximum and local minimum, using the partial derivatives of first and second orders. The discrete nature of the number of alignments in each genomic base pair results in abrupt changes in the function of number of alignments per base pairs; therefore leading to a very large number of local maximum and local minimum. In order to overcome this problem, we used a kernel smoothing function, as described below.

Bylsma (2012) compares several continuous functions that approximates the step function. According to their findings, the Hyperbolic Tangent ($\tanh$) function provides the best overall approximation of the step function based on their error criteria. We therefore apply the kernel smoothing function Hyperbolic Tangent ($\tanh$) to approximated the function of number of alignments to a continuous function, prior to differentiating this function.

The kernel smoothing function used is shown in Equation 5.1. This function allows us to control the smoothing using the parameters k (number of steps in the transition between consecutive base pairs), α (shifting factor that defines the switch point of the function) and β (scaling factor that defines the level of smoothing applied). Figure 5.1 exemplify the result found after applying the smoothing function in Equation 5.1 for one step (up and down).

$$y(x, \alpha, \beta) = \frac{k}{2} + \frac{k}{2} \cdot \tanh\left(\frac{x - \alpha}{\beta}\right) \quad (5.1)$$

After applying the smooth function to the data, we can find the derivatives of first and second order, in order to find the local maximum and minimum points. The rules used to define point of local maximum and minimum are the following:

- ◇ Local minimum: the first order derivative is equal to zero (stationary point) and the second order derivative is positive (*concave up* curve); or the first derivative changes sign at this point, from negative to positive (the function is decreasing before this point and increases after this point).
- ◇ Local maximum: the first order derivative is equal to zero (stationary point) and the second order derivative is negative (*concave down* curve); or the first derivative changes sign at this point, from positive to negative (the function is increasing before this point and decreases after this point).

Based on the points of local maximum and local minimum, we can identify the peaks in the data. Figure 5.2 shows the resulting peaks found after applying the smoothing function. The first Subfigure 5.2(a) shows the number of alignments in each position of the genome (i.e., each base pair); Subfigure 5.2(b) shows the smooth function of number of alignments, after applying Equation 5.1 to the original data with parameters: k = number of steps taken, $\alpha = 1$ and $\beta = 1$. In Subfigure 5.2(c) and 5.2(d) we can see the change in number of local maximum and local minimum points found after the application of the kernel smoother function. The original data has a local maximum and local minimum for each step. After applying the kernel smoother based on the hyperbolic tangent function, this number is reduced significantly, reflecting the true maximum and minimum points in the data. The final Subfigure 5.2(e) shows the *start position*, *end position* and the *top* of each peak found on the data.

The last step of the peaks processing is the decision regarding to join, or not, consecutive peaks. To decide if two consecutive peaks should be joint or not, we first find the area under the curve of both peaks in the original data. We then join the peaks, by connecting the *top* position of the peaks and we calculate again the area under the curve. Finally, we split the peaks, by going downhill, smoothly, from the *top* of each peak.

We, then, calculate the distance between the area of the original peaks and two other areas: (1) after joining the peaks; (2) splitting the peaks. We decide for the option with the least distance

Figure 5.1: In (a) and (b) steps up and down found on the original data; (c) and (d) their respective smooth steps resulting from using the function described in Equation 5.1 with parameters: $k = 1$ (a single step), $\beta = 1$ (constant scaling) and α ranging from -5 to $+5$ (changing function shift). In (e) and (f), smooth steps resulting from using the same function with parameter $k = 1$, $\alpha = 1$ (constant shift) and β ranging from 1 to 10 (changing function scaling).

Figure 5.2: (a) Original data (number of alignments in each position of the genome), (b) Smooth data (smooth function of the number of alignments), (c) Local maximum and local minimum points found on the original data (a very large number of local maximum and local minimum points are found on the original data), (d) Local maximum and local minimum found on the smooth data (a reduced number of local maximum and local minimum points are found on the smooth data), (e) Resulting peaks found using the smooth data.

from the original area. If the original area is closer to the area (1), we join the peaks; otherwise, we join the consecutive peaks.

Chapter 6

Results and Discussion

We performed three sets of tests as described below. The results found for each set of tests are shown in the following Sections.

The first set of tests were performed to exemplify our model and compare our model against the publicly available ChIP-Seq peak finding, Model-based Analysis of ChIP-Seq (MACS) [Zhang et al. \(2008\)](#). For this set of tests we used the datasets of ChIP-Seq experiments of *Mus musculus* published by [McAninch and Thomas \(2014\)](#) (data accessible at NCBI GEO database [Barrett et al. \(2013\)](#); [Edgar et al. \(2002\)](#)), accession GSE57186). These datasets include three DNA libraries resulting from ChIP-Seq experiments using an antibody to recognize human SOX3, and a control sample run without SOX3 antibody. The peaks used for these tests were obtained by running MACS with a high p-value threshold ($p\text{-value} \leq 0.1$) in order to obtain a large number of candidate peaks for the analysis. We decided to use the peaks found by MACS in this first set to simplify the direct comparison between both models. For this set of tests we also show that, when using all the treatment and control samples, our model detects a broader range of peaks than MACS, and that the peaks found by our model are significant binding sites of the Sox genes family.

The goal of the second set of tests was to validate the results found by our model (for the whole genome: chromosomes 1 to 19, *X* and *Y*) by comparing them against gene expression resulting from RNA-Seq experiments. For this set of tests, we used the ChIP-Seq and RNA-Seq experiments of *Mus musculus* published by [Bergsland et al. \(2011\)](#) (data accessible at NCBI GEO database [Edgar et al. \(2002\)](#)), accession GSE33024). These experiments include three DNA libraries resulting from ChIP-Seq experiment treated with SOX3 antibody, one DNA library resulting from ChIP-Seq experiment with Control IgG, two RNA-Seq libraries resulting from Sox2/Sox3 genes knock-down and two RNA-Seq control libraries. We used two different lists of peaks for this set of tests, both extracted from mm10 genome assembly annotation. The first list of peaks included the features *CDS*, *exon*, *start codon* and *stop codon*; and it was extracted directly from the annotation GTF file *genes.gtf* from Illumina iGenomes index and annotation package – UCSC Genome Browser [Rosenbloom et al. \(2015\)](#) [Schneider and Church \(2013\)](#). The second list of peaks was extracted from the custom annotation described in Section 2.2, this annotation includes the following genomic features: *3UTR*, *5UTR*, *gene body*, *TSS200* and *TSS1500*.

The third and last set of tests qualifies the peaks found by MABayApp with respect to the current annotation for mm10 mouse assembly genome. For this set, we used the ChIP-Seq experiment from both publication [McAninch and Thomas \(2014\)](#) and [Bergsland et al. \(2011\)](#); and both annotation files mentioned above as candidate peaks (bed extracted from *genes.gtf* and custom annotation described in Section 2.2). We show that our model identifies peaks in different genomic regions, and it is not biased towards specific regions, such as promoter regions; or towards the size of the regions (i.e., the distribution of the sizes in number of bps. of the significant peaks found is similar to the distribution of size in bps of the whole genome annotation). The results found by our method also show that the binding sites of the protein SOX3 are probably shared with other transcription factors of the SOX family, characterizing a possible sequentially acting of these transcription factors, as described in [Bergsland et al. \(2011\)](#).

6.1 MABayApp overview and Model Comparison

For this section, we ran our model using only the reads aligned to *Chromosome Y* of the ChIP-Seq experiments for SOX3 binding sites. The reason to use only the reads aligned to Chromosome Y is that, as described in Section 6.1.3, we performed a large number of tests by simulating experiment duplication and we wanted to focus our attention on a smaller number of candidate peaks when comparing both models. Reducing this way the bias that might be found in a large number of peaks in different chromosomes.

The dataset used was downloaded from NCBI GEO database [Edgar et al. \(2002\)](#) and its accession ID is *GSE57186*. We performed the analysis first using a single treatment sample against the control sample and later all the three treatment samples and we analysed the change in number of peaks found. We then, compared our method with the publicly available ChIP-Seq peak finding, Model-based Analysis of ChIP-Seq (MACS) [Zhang et al. \(2008\)](#).

6.1.1 Bias Correction - Single Treatment Files

We tested the model using single SOX3 ChIP-Seq treatment sample files against a single SOX3 ChIP-Seq control sample file and found the number of peaks called before and after the application of the bias correction. The equation below show the score used by our model $score(p)$ and the score without the bias correction $score'(p)$.

$$score(p) = \log\text{-odds}\left(\frac{p}{S_{\emptyset}}\right) + \underbrace{\log\text{-odds}\left(S_{\emptyset}\right)}_{\text{bias correction}}$$

Test of score without bias correction

$$score'(p) = \log\text{-odds}\left(\frac{p}{S_{\emptyset}}\right)$$

The results found are shown in Figure 6.1 and the number of peaks called (peaks found significant) are shown in Table 6.1. We can see from the results that the number of peaks called change drastically. Although there are many candidate peak regions in the genome where there is a significant difference in enrichment between treatment vs. control samples, this peaks are not significant when compared against the remaining regions (region with no candidate peak). Therefore these peaks are considered false positives.

Figure 6.1: Decreasing in the number of significant peaks after application of chromosomal bias correction. Peak on the right side of the dashed black line have probability equal or greater than 0.9.

Table 6.1: MABayApp Results for Single Treatment Files vs. Control

Samples	Peaks called before bias correction (probability $\geq$ 0.9)	Peaks called by MABayApp (probability $\geq$ 0.9)
Treatment sample 1 vs. Control sample	4,979	1,916
Treatment sample 2 vs. Control sample	4,925	1,283
Treatment sample 3 vs. Control sample	4,339	1,195

6.1.2 Meta-analysis – Multiple Treatment Files

We observed the change in number of peaks called when combining multiple treatment via meta-analysis using our model. Figure. 6.2 shows the probability based scores found by our model for all candidate peaks when comparing each of ChIP-Seq treatment sample separately against the control sample. We sorted the peaks (x-axis) by increasing order of the probability based score (y-axis). The dotted line shows the threshold to consider a candidate peak a true binding site ($score(p) \geq 0.9$). The number of significant binding sites found for each treatment replicate when compared separately against the control sample was: **1,916** sites for replicate 1, **1,283** sites for replicate 2, and **1,195** sites for replicate 3.

In order to show the change in number of peaks called when combining different treatment samples using meta-analysis, we first combined treatment samples 1 and 2, and treatment samples 2 and 3 and ran our model considering all candidate peaks for both samples against the control sample. We then ran the analysis considering all three samples together.

Figure 6.3 shows the results of the tests combining treatment samples 1 and 2, as well as all three treatment samples, with the peak sorted in increasing order of probability based score. Figure 6.4

Figure 6.2: *Single Treatment Files. Probability of score in Treatment sample greater than score in Control sample.*

show the venn diagram with the overview of number of peaks called for all three treatment samples separately, as well as the result after combining the samples two by two, and all together.

We can see from these figures that the number of peaks called reduces when combining the treatment samples. The resulting number of peaks called were (see Figure 6.4): **819** peaks for treatment samples 1 and 2 combined, **863** peaks for treatment samples 1 and 3 combined, **812** peaks for treatment samples 2 and 3 combined, and **575** peaks for treatment samples 1, 2 and 3 all together.

This reduction in number of peaks called is a direct consequence of the nature of the meta-analysis that takes the average distribution of the log-odds of the peaks across all the treatment replicates. Therefore a peak found to be significant when analysing a single treatment versus control sample might be discarded from the list of peaks called, when combining the remaining treatment samples. Because, in this case, the peak loses significance when the average distribution of log-odds is taken across all treatment replicates.

6.1.3 Model Comparison - Simulation of experiment duplication

To compare the score variation for both models when increasing the dataset, we used the reads aligned to Chromosome Y of the ChIP-Seq experiments for SOX3 binding sites and enhancers described in [McAninch and Thomas \(2014\)](#). This dataset was downloaded from NCBI GEO database [Edgar et al. \(2002\)](#), and accession ID is *GSE57186*. We simulated the experiment duplication by concatenating the same ChIP-seq treatment *.bam* file into a single file 2, 4, 8, 16 and 32 times; doing the same for the control *.bam* file. And we used the resulting control and treatment file as input to both models. The details on this simulation are given below.

We first ran MACS against the original dataset (treatment and control sample *.bam* files, before duplication) with a very low threshold (parameter *"-p 0.1"*) allowing the program to find as many

Figure 6.3: Multiple Treatment files - Meta-Analysis.

Figure 6.4: Overview of the number of peaks found by MABayApp for treatment samples of SOX3 ChIP-Seq dataset - chromosome Y.

peaks as possible. All peaks with a score greater or equal than 10 were found, resulting in 9,639 candidate peaks. We then, concatenated the same treatment *.bam* files, using Samtools (Li et al., 2009) command "*samtools merge treatmentX2.bam treatment.bam treatment.bam*" resulting in a larger *.bam* file (*treatmentX2.bam*) with twice the number of alignments; we performed the same operation for the control file. We used the resulting control and treatment files as input to MACS, with parameters to keep the duplicate reads "*-p 0.1 -keep-dup='all' -f 'BAM' -s 50 -verbose 3*", and we observed the score variation for the original 9,639 candidate peaks. We repeated the steps described above, until we had a dataset with 32 times the number of alignments. We used the coordinates of the peaks found by MACS at each step as input to our model, and observed the score variation for the 9,639 candidate peaks in our model as well. The results of this simulation are shown in Figure 6.5 and 6.6 for MACS and MABayApp, respectively.

From these figures, we can see that as we increase the dataset, less peaks are rejected by the statistical test performed by MACS (the green line in Figure 6.5 shows that *all* the peaks are considered significant using the default threshold when the dataset is duplicated 16 times). While our model becomes more decisive on the significance of the peaks (the green line in Figure 6.6 shows that while some of the peaks are considered more significant – *p* approaches 1.0 – other peaks have their score reduced – *p* approaches 0.0). As we increase the data available for our model, the scores of the peaks approaches a step function, giving some of the peaks a score of 1.0 (definitely significant) and the 0.0 (definitely non-significant).

The numbers resulting from this experiments are shown in Table 6.2, where we can see the behaviour of both models, as described above.

Table 6.2: Comparison of scores MABayApp vs. MACS.

Dataset	Peaks called by MACS (score ≥ 50)	Minimum score by MACS	Peaks called by MABayApp (probability ≥ 0.9)	Peaks with probability=0 for MABayApp
Original Dataset	1,305	10.00	1,912	1,751
Dataset $\times 2$	2,742	12.53	2,231	2,299
Dataset $\times 4$	4,883	17.12	2,467	2,498
Dataset $\times 8$	7,525	25.82	2,626	2,605
Dataset $\times 16$	9,617	43.99	2,727	2,737
Dataset $\times 32$	9,639	77.74	2,802	2,299

6.1.4 Model Comparison - Single & Multiple treatment samples

We compared significant peaks found by MABayApp and MACS using each treatment sample against the control sample as well as all treatment samples together against the control sample. Figure 6.7 shows, for each candidate peak, the score found by MACS on the horizontal axis and the score found by MABayApp on the vertical axis, for the test using treatment samples separately. The resulting numbers of common/uncommon peaks found are shown in Table 6.3. The total number of peaks called for individual treatment samples against control, as well as for all treatment samples together against control, are shown in Table 6.4.

The peaks in black are candidate peaks with MACS score lower than 50 and MABayApp score lower than 0.9, which mean both models found them to be not significant. This class of peaks comprehend the greatest number of candidate peaks (see Table 6.3): 7,382 peaks for treatment 1 versus control (**76.53%** of the candidate peaks), 8,156 peaks for treatment 2 versus control (**86.01%** of the candidate peaks), 7,159 peaks for treatment 3 versus control (**85.20%** of the candidate peaks).

Candidates peaks in blue are peaks called by MABayApp, but not called by MACS (MABayApp scores greater or equal than 0.9 and MACS score lower than 50). This class of peaks shows that, for two of the treatment samples, the greatest majority of the peaks called by MABayApp have not been called by MACS (see Table 6.3 for number peaks called only by MABayApp and Table 6.4

Figure 6.5: Score variation (in log scale) for macs when duplicating the SOX3 Chip-Seq dataset for both treatment and control samples, $\times 2$, $\times 4$, $\times 8$, $\times 16$. The dashed black line shows the default threshold $1E-5$.

for the total number of peaks called by MABayApp): 959 peaks for treatment 1 versus control (**50.05%** of the peaks called by MABayApp), 1,250 peaks for treatment 2 versus control (**97.43%** of the peaks called by MABayApp), 1,111 peaks for treatment 3 versus control (**92.97%** of the peaks called by MABayApp).

In yellow, we have the peaks called by MACS, but not called by MABayApp (MABayApp scores lower than 0.9 and MACS score greater or equal than 50). These results show that, for two of the treatment replicates, most of the peaks called by MACS, were also called by MABayApp. The number of peaks called only by MACS were (see Table 6.3 for number peaks called only by MACS and Table 6.4 for the total number of peaks called by MACS): 348 peaks for treatment 1 versus control (**26.67%** of the peaks called by MACS), 44 peaks for treatment 2 versus control (**57.14%** of the peaks called by MACS), 49 peaks for treatment 3 versus control (**36.84%** of the peaks called by MACS).

Finally, in green we have the candidate peaks called by both MACS and MABayApp. According to these results, the percentage of peaks called by MACS that was also called by MABayApp was between **42%** and **73%**: 957 peaks for treatment 1 versus control (**49.95%** of the peaks called by MABayApp and **73.33%** of the peaks called by MACS), 33 peaks for treatment 2 versus control (**2.57%** of the peaks called by MABayApp and **42.86%** of the peaks called by MACS), 84 peaks for treatment 3 versus control (**7.03%** of the peaks called by MABayApp and **63.16%** of the peaks called by MACS).

In order to find the peaks called by MACS when using all the treatment samples together, we concatenated the different replicates in a single *.bam* file, as suggested by the documentation of MACS, and used the resulting file as treatment sample. And in order to find the peaks called by MABayApp, we first ran MACS with a threshold of 0.1 (to find as many candidate peaks as possible) for each treatment sample against the control sample. We then used all the three resulting candidate peaks files (one for each treatment sample) as input of MABayApp, together with the

Figure 6.6: Score variation (in log scale) for our model when duplicating the *SOX3* Chip-Seq dataset for both treatment and control samples, $\times 2$, $\times 4$, $\times 8$, $\times 16$. The dashed black line shows the default threshold $p=0.9$.

Figure 6.7: Peaks called by MACS and MABayApp for *SOX3* ChIP-Seq data: direct comparison. The candidate peaks not called by either MACS or MABayApp are shown in black. Peaks called only by MABayApp are shown in blue. In yellow are all the peaks called by MACS, but not by MABayApp. And in green are the peaks called by both models.

Table 6.3: Comparison of peaks called: MABayApp vs. MACS.

Dataset	Peaks called only by MACS	Peaks called only by MABayApp	Peaks called by both MACS and MABayApp	Peaks called by neither MACS nor MABayApp
Treatment Sample 1 vs. Control	348	959	957	7,382
Treatment Sample 2 vs. Control	44	1,250	33	8,156
Treatment Sample 3 vs. Control	49	1,111	84	7,159

three treatment sample files and the control sample file (totalling four *.bam* files).

The resulting number of peaks found by each model is shown in Table 6.4. As we can see the number of peaks found by the models are very different; MACS found only **13** peaks to be significant (score ≥ 50), while MABayApp found **575** peaks to be significant (probability based score ≥ 0.9). We analysed the genome annotation of this **575** regions of Chromosome Y found by MABayApp in Section 6.3 to show that these peaks are important for researches studying the *SOX* genes family binding sites.

6.2 Model Validation

The goal of this second set of tests was to validate the results found for ChIP-Seq data by our model. In order to validate our results, we compared peaks found by our model in the whole genome (chromosomes 1 to 19, X and Y) and used the data obtained to compare against the results of

Table 6.4: Comparison of scores MABayApp vs. MACS.

Dataset	Peaks called by MACS (score ≥ 50)	Peaks called by MABayApp (probability ≥ 0.9)
Treatment Sample 1 vs. Control	1,305	1,916
Treatment Sample 2 vs. Control	77	1,283
Treatment Sample 3 vs. Control	133	1,195
Treatment Sample 1, 2 and 3 vs. Control	13 (concatenating files)	575 (using all peaks' lists)

RNA-Seq experiments in the whole genome.

The ChIP-Seq and RNA-Seq experiments used for this validation were extracted from the same series of experiments that investigates the acting of Sox transcription factor in neural development of *Mus Musculus* (data accessible at NCBI GEO database, accession GSE33024).

The first test used to validate our model was performed over a list of candidate peaks extracted from the annotation GTF file *genes.gtf* from Illumina iGenomes index and annotation package. We ran MABayApp using the treatment and control samples of ChIP-Seq experiments to find the significance of candidate peaks, which are composed of the following genomic features: *CDS*, *exon*, *start codon* and *stop codon*.

We analysed the gene and transcription differential expression resulting from RNA-Seq experiments using the same annotation file and following the protocol with TopHat version 2.0.9 and Cufflinks version 2.1.1, described in [Trapnell et al. \(2012\)](#).

We first filtered the CDSs from annotation with *log2 fold-change* greater than zero (i.e., Sox3 expression was greater than Control expression) and then compared, for each CDS, the probability based score found by MABayApp and the p-value found by RNA-Seq data. Figure 6.8 shows the results with p-value obtained from RNA-Seq data on the horizontal axis and MABayApp score resulting from ChIP-Seq data on the vertical axis. The 2d density estimated from the data plotted is also shown.

As we can see from this figure, the highest densities are concentrated in regions with the lowest p-values (values lower or equal 0.01 on the x-axis) and highest values of MABayApp score (values between 0.7 and 1.0 on the y-axis). This result confirms that CDSs considered significant in RNA-Seq (low p-values) were also considered significant binding sites according to our models (high values of MABayApp probability based score).

Figure 6.8: *Density 2d estimation for: RNA-Seq CDS p-value $\times$ MABayApp CDS probability. The highest density – for CDS with p-value between 0.00 and 0.05 and log2 fold change greater than 0 – are concentrated in highest value of MABayApp probabilities, confirming that CDSs considered significant during RNA-Seq analysis (p-values closer to 0.0) were also considered significant binding sites according to our model (probabilities closer to 1.0).*

The second test used to validate our model was performed over a list of candidate peaks extracted from the custom annotation described in Section 2.2. We ran MABayApp for ChIP-Seq experiments to find the significance peaks within the genomic regions *3UTR*, *5UTR*, *gene body*, *TSS200* and *TSS1500*. We analysed the RNA-Seq experiments using this same annotation file as reference.

For genes and transcriptions showing Sox3 RNA-Seq expression greater than control (*log2 fold-change* greater than zero), we separated the genomic features *3UTR*, *5UTR*, *TSS200* and *TSS1500* in two groups: significant and non-significant. Genes with p-values lower or equal than 0.05 were considered significant and the remaining, non-significant. The analysis performed using this genomic features shows that our model is capable of detecting peaks in different regions of the genes, and it's not biased towards promoter regions, for example.

We then found, for these four features in the annotation (*3UTR*, *5UTR*, *TSS200* and *TSS1500*), the probability based score found by MABayApp. Figure 6.9 shows the histogram of MABayApp probability score (MABayApp on the horizontal axis and density on the vertical axis) for each genomic feature, separating significant and non-significant genes.

As we can see from this figure the significant genes (in blue) have more features with MABayApp probability score greater or equal than 0.9 (blue histogram is higher than red histogram for highest values on the x-axis). This result shows that the genomic regions *3UTR*, *5UTR*, *TSS200* and *TSS1500* considered significant according to RNA-Seq data analysis (p-value lower or equal than 0.05) had more features considered significant by our model, as well (MABayApp score greater or equal than 0.9).

Figure 6.9: Histogram of MABayApp probabilities found for annotation features (*TSS200*, *TSS1500*, *3UTR* and *5UTR*), comparing significant genes (genes with *p*-value lower or equal than 0.05) and non-significant genes (remaining genes), according to RNA-Seq analysis. The dotted line shows the MABayApp threshold for significant binding sites (probability greater or equal than 0.9). Significant genes had more features found to be binding site according to MABayApp ($p \geq 0.9$).

We also found, for the Sox gene family, the RNA-Seq analysis results for their respective CDS ($\log_2$ fold-change and *p*-value) and MABayApp probability score for the respective features *TSS200*, *TSS1500*, *3UTR* and *5UTR*. The results are shown in Table 6.5 with value of $\log_2$ fold-change greater than 0 in bold font, and value of MABayApp score greater or equal than 0.9 in bold font, as well. These results show that, for Sox family CDS with $\log_2$ fold change greater than 0 (Sox3 expression higher than Control expression), many of the respective features were found significant binding sites (MABayApp equal or greater than 0.9).

6.3 Genome Annotation Analysis

In this section we analysed enriched regions of the genome based on the *UCSC genome assembly and annotation* described in Section 2.2. The regions with MABayApp probability based score equal

Table 6.5: SOX transcription factor family results for RNA-Seq and MABayApp.

gene	RNA-Seq		MABayApp Probability					
	log2 fold change	p-value	3UTR	5UTR	gene body	TSS200	TSS1500	CDS
Sox1	-0.69122	0.00055	—	—	—	—	—	0.29868
Sox2	-0.27152	0.18180	—	—	—	—	—	0.53214
Sox3	-1.49636	5e-05	0.61665	—	0.70828	—	—	0.65139
Sox4	0.28012	0.13675	0.69738	0.9206	0.91384	—	—	0.79458
Sox5	1.72929	1.00000	—	—	—	—	—	1.0000
Sox6	-0.15702	0.77620	0.35761	0.6449	0.61301	—	—	0.89424
Sox7	-1.27658	0.00215	0.21688	0.42205	0.4222	0.87499	0.44284	0.5386
Sox8	-0.43693	0.09710	0.25606	0.67349	0.9354	—	—	0.63634
Sox9	0.43316	0.02980	0.42145	0.93622	0.96064	0.87975	0.92781	0.96644
Sox10	0.1482	1.00000	0.16634	0.73855	0.86635	—	—	0.96616
Sox11	0.34318	0.78200	0.08095	0.91095	0.90962	—	—	0.7912
Sox12	-0.27402	0.14135	0.83505	0.9269	0.784	—	—	0.86437
Sox13	-1.08528	5e-05	0.54161	0.91231	0.71508	—	—	0.92706
Sox14	Inf	1	0.42364	0.62736	0.78954	—	—	0.79983
Sox15	-1.52565	1	0.42146	0.441	0.42753	0.41943	0.26309	0.71639
Sox17	-0.00821	0.97675	0.48921	0.66891	0.76015	—	—	0.88943
Sox18	0.29182	1	0.30093	0.17523	0.59979	—	—	0.80838
Sox21	-0.52945	0.0087	0.97743	0.99364	0.94266	—	—	0.90399
Sox30	0.36272	1	0.17595	0.68305	0.14142	0.98509	0.73542	0.50712

or greater than 0.9 were considered significant binding sites.

We first found the percentage of Gene Body, 3UTR and 5UTR significant regions per chromosomes. We can see from the results shown in Figure 6.10 that the percentage and number of significant regions (3UTR, 5UTR and gene body) have similar distribution across Chromosomes, which might indicate that these regions share genes in common (i.e., the same genes have all three regions, 3UTR, 5UTR and gene body, enriched).

Figure 6.10: *Percentage of significant genomic regions, and number of significant regions found by MABayApp (significant features are those with probability based score ≥ 0.9).*

Next, we check if the MABayApp is biased towards larger regions. In order to check that we analysis the distribution of the length for both significant and non-significant regions. The result is shown in Figure 6.11. As we can see from this Figure, there is no discrepancy in the distribution of the length when comparing significant and non-significant regions, which indicates that MABayApp is not biased toward either larger or smaller regions.

Figure 6.11: Sequence length comparison between significant and non-significant genomic features: 3'UTR, 5'UTR and Gene body. Significant features are those with probability based score ≥ 0.9 .

Since we expected many genes to share significant features, we analysed the number of genes that shared a combination of genomic features. Figure 6.12 shows the Venn Diagram of the number of genes that share combinations of significant features 3'UTR, 5'UTR, TSS1500, TSS200 and gene body. The results shows that, as expected, many genes indeed have more than one significant feature; and many of them have at least three genomic features considered significant by MABayApp.

Figure 6.12: Number of genes with regions enriched. Many genes had enrichment of other regions as well. There are 32 genes which have all the 5 regions enriched: 3'UTR, 5'UTR, gene body, TSS200 and TSS1500.

In order to validate our results, we observed genomic regions enriched for genes of The SOX family. We expected to find many regions of these genes enriched, since the treatment samples had antibodies specific to target genes of this family. As we can see from Table 6.6, the results have confirmed that MABayApp indeed found many features for genes of these families with significant enrichment, which was a very satisfactory result.

We also looked at the genes that had all the 5 genomic features considered significant. Those genes are shown in Table 6.7, together with the gene type information and know role of the gene.

It's interesting to notice that most of the genes have functions associated with the phase of initial development of the mouse, which was expected, since the samples from *Mus Musculus* were taken during neural lineage development. Among the roles played by these genes, the most present ones are: neural development and regulation, axis/skeleton formation, cellular regulation (proliferation, differentiation and apoptosis) and development of immune system.

The Gene Ontology analysis of the enriched regions (Figures 6.13, 6.14, 6.15, 6.16, 6.17, and 6.18) show similar results, with enriched biological processes related to neuronal development and

Table 6.6: SOX transcription factor family - enriched regions.

gene	3UTR	5UTR	gene body	TSS200	TSS1500
Sox2				X	X
Sox3	X				
Sox4	X	X	X		
Sox6	X				
Sox9		X			
Sox10			X		
Sox11	X	X	X		
Sox12		X			
Sox13	X				
Sox15		X	X	X	
Sox17		X	X		
Sox21			X		
Sox30				X	X

differentiation, enriched cellular components related to development of Golgi and skeleton; and enriched molecular functions related to neuronal development.

Table 6.7: Genes which have all the 5 regions enriched: 3'UTR, 5'UTR, gene body, TSS200 and TS1500.

Gene Id	Gene Name	Type	Known Role
ENSMUST00000179781	Bsg	Protein Coding	Reproduction, neural function
ENSMUST00000000127	Wnt3	Protein Coding	Primary axis formation in the mouse
ENSMUST00000026119	Gcgr	Protein Coding	Glucagon receptor
ENSMUST00000000128	Wnt9a	Protein Coding	Regulation of cell fate during embryogenesis
ENSMUST00000108375	Myo18a	Protein Coding	Golgi membrane trafficking
ENSMUST00000147875	Lym9	Protein Coding	Unknown
ENSMUST00000038696	Ppp1r9b	Protein Coding	Receiving signals from central nervous system
ENSMUST00000042779	Zbtb1	Protein Coding	Development of lymphocytes in mice
ENSMUST00000140770	Plekhd1	Protein Coding	Form and maintain the skeleton
ENSMUST00000021674	Fos	Protein Coding	Form and maintain the skeleton
ENSMUST00000075558	Hist1h3f	Protein Coding	Transcription regulation, DNA repair
ENSMUST00000049488	Serinc5	Protein Coding	Inhibiting an early step of viral infection
ENSMUST00000022952	Osr2	Protein Coding	Development of the palate
ENSMUST00000161785	Zfp41	Protein Coding	Meiosis in spermatogenesis
ENSMUST00000127208	Lrrc14	Protein Coding	Formation of protein-protein interaction
ENSMUST00000082439	Selo	Protein Coding	Uncharacterized protein
ENSMUST00000023150	1810013L24Rik	Protein Coding	Unknown function
ENSMUST00000056882	Olig1	Protein Coding	Formation of oligodendrocytes within the brain
ENSMUST00000063344	Lmf1	Protein Coding	Transport through the secretory pathway
ENSMUST00000013706	4833413E03Rik	Protein Coding	Unknown
ENSMUST00000038287	Dusp5	Protein Coding	Cellular proliferation and differentiation
ENSMUST00000066646	Rcor2	Protein Coding	Neurogenesis in the developing mouse brain
ENSMUST00000087215	Rqcd1	Protein Coding	Required for cell differentiation
ENSMUST00000065587	Ackr3	Protein Coding	Unknown
ENSMUST00000043760	Mvk	Protein Coding	Formation of cytoskeleton
ENSMUST00000046999	Abhd11	Protein Coding	Unknown
ENSMUST00000085591	Pdx1	Protein Coding	Necessary for pancreatic development
ENSMUST00000165164	Pcgf1	Protein Coding	Promotes cell progression and proliferation
ENSMUST00000071492	Fam136a	Protein Coding	Development of neurosensory epithelium
ENSMUST00000047621	Ppp1r13l	Protein Coding	Regulation of apoptosis and transcription
ENSMUST00000044111	Ras	Protein Coding	Organization of cytoskeleton
ENSMUST00000112588	Kdm5c	Protein Coding	Transcriptional repression of neuronal genes

Figure 6.13: Number of occurrences of biological processes found only among significant regions 3'UTR, 5'UTR and gene body (MABayApp score ≥ 0.9). These biological processes have not been found on any of the non-significant regions (MABayApp score < 0.9).

Figure 6.14: Number of occurrences of biological processes found only among significant regions TSS200 and TSS1500 (MABayApp score ≥ 0.9). These biological processes have not been found on any of the non-significant regions TSS200 and TSS1500 (MABayApp score < 0.9).

Figure 6.15: Number of occurrences of cellular components found only among significant regions 3'UTR, 5'UTR and gene body (MABayApp score ≥ 0.9). These cellular components have not been found on any of the non-significant regions (MABayApp score < 0.9).

Figure 6.16: Number of occurrences of cellular components found only among significant regions *TSS200* and *TSS1500* (*MABayApp* score ≥ 0.9). These cellular components have not been found on any of the non-significant regions *TSS200* and *TSS1500* (*MABayApp* score < 0.9).

Figure 6.17: Number of occurrences of molecular functions found only among significant regions 3'UTR, 5'UTR and gene body (MABayApp score ≥ 0.9). These molecular functions have not been found on any of the non-significant regions (MABayApp score < 0.9).

Figure 6.18: Number of occurrences of molecular functions found only among significant regions TSS200 and TSS1500 (MABayApp score ≥ 0.9). These molecular functions have not been found on any of the non-significant regions TSS200 and TSS1500 (MABayApp score < 0.9).

Chapter 7

Conclusion

The results of the application of our model showed that it is robust on the detection of significant peaks in ChIP-Seq dataset. The Bayesian method was less sensitive to change in sample size and the meta-analysis corrected the number of significant peaks found by averaging the distribution of the log-odds of each candidate peak over all the samples. Nonetheless the score used also penalizes candidate peaks with enrichment similar to those regions with no candidate peak in the genome, resulting in an extra filter of false-positive peaks. The model has also a very strong statistical background, as detailed in Section 3 which gives a high confidence for its application and future related research.

In order to make this solution available, we developed an application in R language which is available under request. This model should be used when the researcher is interested in validating peaks called for ChIP-Seq datasets with multiple treatment and control samples, where the number of treatment samples are not necessarily equal to the number of control samples. The model can also be used to compare samples with different treatments; in this case, the investigator should select one of the treatment samples as control to find regions with significant enrichment in the remaining treatment samples.

Our validation against other method used for the same data, RNA-Seq, had a very satisfactory result showing that the peaks considered significant by our model were also found significant when using RNA-Seq, as well as regions of genomic features TS200/TS1500 and 3'UTR and 5'UTR with significant enrichment in RNA-Seq also had high MABayApp score. The analysis of RNA-Seq differential expression versus MABayApp scores for the SOX genes family were also satisfactory since SOX genes with high log2 fold change of RNA-Seq expression and low p-values were also found to be significant when using our model. Finally, our model identified several genomic regions enriched for the SOX family genes (SOX2 to Sox30) as expected, since this family of genes was the focus of the investigators how provided the sample (McAninch and Thomas, 2014); and a Ven diagram of the different genomic features enriched showed that for many genes, different genomic features were enriched at the same time (gene body, 3'UTR, 5'UTR, TSS200, TSS1500).

7.1 Future Research

In order to give continuity to this work, we have the following points to be explored:

1. Extrapolate the model to be used with more complex genomes, for example: the hybrid Sugarcane genome
2. Explore different additional method of signal/noise filtering to be added to the model
3. Explore different smoothing methods to find peaks in the dataset, including:
 - 3.1. different parameter values for the smoothing method already defined in Section 5.2
 - 3.2. new kernel functions to define other smoothing methods

We believe this work has achieved its main goals: to develop a robust model with a strong statistical background to find significant peaks in ChIP-Seq datasets, and to make it available for different researchers in the area of genomic studies. It also contains points that could be further explored by other investigator in Bioinformatics and related areas.

Appendix A

R Code

```
1 #!/usr/bin/Rscript
2 rm(list = ls())
3
4 suppressPackageStartupMessages(library(pryr))
5 suppressPackageStartupMessages(library(pracma))
6 suppressPackageStartupMessages(library(Rcpp))
7 suppressPackageStartupMessages(library(inline))
8 suppressPackageStartupMessages(library(GenomicAlignments))
9 suppressPackageStartupMessages(library(GenomicRanges))
10 suppressPackageStartupMessages(library(parallel))
11
12 findalignmentCoverage <- function(chr.coverage, alignments.chunk){
13   chunk.coverage <- coverage(alignments.chunk)
14   if (is.null(chr.coverage)) {
15     chr.coverage <- as.numeric(sum(chunk.coverage))
16   } else {
17     chr.coverage <- chr.coverage + as.numeric(sum(chunk.coverage))
18   }
19   rm(chunk.coverage)
20   return(as.numeric(chr.coverage))
21 }
22
23 filteralignments <- function(alignments.chunk, peaks.gr){
24   return(alignments.chunk[unique(queryHits(findOverlaps(alignments.chunk, peaks.
25     gr)))])
26 }
27
28 empty.alignments <- function(alignments.chunk.list){
29   is.empty <- TRUE
30   for(alignment.chunk.idx in 1:length(alignments.chunk.list))
31     is.empty <- is.empty && (length(alignments.chunk.list[[alignment.chunk.idx
32       ]]) == 0L)
33   return(is.empty)
34 }
35
36 sumX <- "
37   *sum=0.0;
38   for (int i = 0; i < *n; i++)
39     *sum = *sum + x[i];
40 "
41
42 sum.array <- cfunction(signature(n="integer", x="numeric", sum="numeric"), sumX,
43   language = "C", convention = ".C")
44
45 #####
46 # Read arguments (file names)
47 args <- commandArgs(TRUE)
48 pos.c <- which(args=="-c")
49 pos.t <- which(args=="-t")
50 pos.p <- which(args=="-p")
```

```

46 # output file
47 pos.o          <- which(args=="-o")
48
49 #####
50 # BAM and peak files path
51 bam.files.control.path.list <- list()
52 bam.files.treatment.path.list <- list()
53 peak.files.path.list <- list()
54 output.file.name <- ""
55
56 if((length(args)<8) ||
57     (args[1]!="-c") ||
58     (length(pos.t)==0) ||
59     (length(pos.p)==0) ||
60     (length(pos.o)==0))
61 {
62   cat("usage: chipseq-callpeaks.R -c <control BAM file 1> <control BAM file 2> ...
63       -t <treatment BAM file 1> <treatment BAM file 2> ...
64       -p <broadPeak file 1> <broadPeak file 2> ...
65       -o <output file name> \n")
66 } else {
67   cat("\nControl BAM files:\n")
68   for(arg.control.idx in 2:(pos.t-1))
69   {
70     cat(paste(args[arg.control.idx], "\n"))
71     bam.files.control.path.list[[length(bam.files.control.path.list)+1]] <-
72       args[arg.control.idx]
73   }
74   cat("\nTreatment BAM files:\n")
75   for(arg.treatment.idx in (pos.t+1):(pos.p-1))
76   {
77     cat(paste(args[arg.treatment.idx], "\n"))
78     bam.files.treatment.path.list[[length(bam.files.treatment.path.list)+1]] <-
79       args[arg.treatment.idx]
80   }
81   cat("\nPeak files:\n")
82   for(arg.peak.idx in (pos.p+1):(pos.o-1))
83   {
84     cat(paste(args[arg.peak.idx], "\n"))
85     peak.files.path.list[[length(peak.files.path.list)+1]] <-
86       args[arg.peak.idx]
87   }
88   cat("\n")
89   cat("Output file name :\n")
90   output.file.name <- args[pos.o+1]
91   cat(paste(output.file.name, "-peaks.bed", "\n", sep=""))
92   cat("\n")
93
94   cat("Log file name :\n")
95   output.file.name <- args[pos.o+1]
96   cat(paste(output.file.name, "-peaks.bed", "\n", sep=""))
97   cat("\n")
98
99   sink(paste(output.file.name, "-peaks.log", sep=""))
100
101 #####
102 # define BAM files
103 bam.files.control.list <- list()
104 bam.files.treatment.list <- list()
105 for(bam.file.c.idx in 1:length(bam.files.control.path.list)){
106   bam.files.control.list[[bam.file.c.idx]] <- BamFile(bam.files.control.path
107     .list[[bam.file.c.idx]],
108                                           yieldSize=1000000)
109 }
110 for(bam.file.t.idx in 1:length(bam.files.treatment.path.list)){

```

```

110 bam.files.treatment.list[[bam.file.t.idx]] <- BamFile(bam.files.treatment.
111 path.list[[bam.file.t.idx]],
112 yieldSize=1000000)
113 }
114 n.control.samples <- length(bam.files.control.path.list)
115 n.treatment.samples <- length(bam.files.treatment.path.list)
116 cat(paste("n.control.samples: ", n.control.samples, "\n", sep=""))
117 cat(paste("n.treatment.samples: ", n.treatment.samples, "\n", sep=""))
118 #####
119 # get length of chromosomes
120 chr.len <- seqlengths(bam.files.treatment.list[[1]])
121 #####
122 # define coverage as NULL
123 control.chr.coverage.list <- vector("list", n.control.samples)
124 treatment.chr.coverage.list <- vector("list", n.treatment.samples)
125 #####
126 # define peaks coverage as NULL
127 control.chr.peaks.coverage.list <- vector("list", n.control.samples)
128 treatment.chr.peaks.coverage.list <- vector("list", n.treatment.samples)
129 #####
130 ## Peak files and genomic regions
131 cat("Reading peak files ... \n")
132 peak.files.list <- vector("list", length(peak.files.path.list))
133 peak.gr.list <- vector("list", length(peak.files.path.list))
134 for(peak.file.idx in 1:length(peak.files.path.list)){
135 peak.files.list[[peak.file.idx]] <- read.table(peak.files.path.list[[peak.
136 file.idx]], header = TRUE)
137 peak.gr.list[[peak.file.idx]] <- GRanges(seqnames=peak.files.list[[peak.
138 file.idx]][,1],
139 ranges=IRanges(start=peak.files.
140 list[[peak.file.idx]][,2],
141 end=peak.files.
142 list[[peak.
143 file.idx
144 ][,3]])
145 }
146 #####
147 # union of peak genomic regions as parameter for reading BAM file
148 all.peaks.gr <- peak.gr.list[[1]]
149 if(length(peak.gr.list)>1)
150 {
151 for(peak.gr.idx in 2:length(peak.gr.list)){
152 all.peaks.gr <- union(all.peaks.gr, peak.gr.list[[peak.gr.idx]])
153 }
154 }
155 #####
156 # peaks coverage
157 all.peaks.gr.control.cov <- vector("list", n.control.samples)
158 all.peaks.gr.treatment.cov <- vector("list", n.treatment.samples)
159 #####
160 # Find width of all peaks
161 peaks.chr.width <-
162 aggregate(width ~ chr,
163 DataFrame(chr=as.character(seqnames(all.peaks.gr)),
164 width=as.integer(width(all.peaks.gr))),sum)
165 peaks.width <- peaks.chr.width$width
166 peaks.names <- peaks.chr.width$chr
167 peaks.chr.width <- peaks.width
168 names(peaks.chr.width) <- peaks.names

```

```

168 #####
169 # read and coverage by chromosome
170 for( control.sample.idx in 1:n.control.samples){
171   open(bam.files.control.list[[control.sample.idx]])
172 }
173 for( treatment.sample.idx in 1:n.treatment.samples){
174   open(bam.files.treatment.list[[treatment.sample.idx]])
175 }
176 }
177 cat("Reading coverage by chunks from BAM files ... \n")
178 repeat {
179   # read alignment chunks from BAM files
180   control.alignments.chunk.list <- vector("list", n.control.samples)
181   for( control.sample.idx in 1:n.control.samples){
182     control.alignments.chunk.list[[control.sample.idx]] <-
183       readGAlignments(bam.files.control.list[[control.sample.idx]])
184   }
185   treatment.alignments.chunk.list <- vector("list", n.treatment.samples)
186   for( treatment.sample.idx in 1:n.treatment.samples){
187     treatment.alignments.chunk.list[[treatment.sample.idx]] <-
188       readGAlignments(bam.files.treatment.list[[treatment.sample.idx]])
189   }
190
191   # stop reading BAM files if no there's no chunk to be read anymore
192   if((empty.alignments(control.alignments.chunk.list)) &
193      (empty.alignments(treatment.alignments.chunk.list)))
194     break
195
196   # filter peaks alignments from total alignments
197   control.alignments.peaks.chunk.list <- vector("list", n.control.samples)
198   for( control.sample.idx in 1:n.control.samples){
199     control.chr.coverage.list[[control.sample.idx]] <-
200       findalignmentCoverage(control.chr.coverage.list[[control.sample.idx]],
201                             control.alignments.chunk.list[[control.sample.idx]]
202                             ))
203     control.alignments.peaks.chunk.list[[control.sample.idx]] <-
204       filteralignments(control.alignments.chunk.list[[control.sample.idx]],
205                        all.peaks.gr)
206     control.chr.peaks.coverage.list[[control.sample.idx]] <-
207       findalignmentCoverage(control.chr.peaks.coverage.list[[control.sample.idx]]
208                             ,
209                             control.alignments.peaks.chunk.list[[control.sample.idx]]
210                             ))
211     if( is.null( all.peaks.gr.control.cov[[control.sample.idx]] ) ){
212       all.peaks.gr.control.cov[[control.sample.idx]] <-
213         coverage(control.alignments.chunk.list[[control.sample.idx]])
214     } else {
215       all.peaks.gr.control.cov[[control.sample.idx]] <-
216         all.peaks.gr.control.cov[[control.sample.idx]] +
217         coverage(control.alignments.chunk.list[[control.sample.idx]])
218     }
219   }
220
221   treatment.alignments.peaks.chunk.list <- vector("list", n.treatment.samples)
222   for( treatment.sample.idx in 1:n.treatment.samples){
223     treatment.chr.coverage.list[[treatment.sample.idx]] <-
224       findalignmentCoverage(treatment.chr.coverage.list[[treatment.sample.idx]]
225                             ,
226                             treatment.alignments.chunk.list[[treatment.sample.idx]]
227                             ))
228     treatment.alignments.peaks.chunk.list[[treatment.sample.idx]] <-
229       filteralignments(treatment.alignments.chunk.list[[treatment.sample.idx]],
230                        all.peaks.gr)
231     treatment.chr.peaks.coverage.list[[treatment.sample.idx]] <-

```



```

227     findalignmentCoverage(treatment.chr.peaks.coverage.list[[treatment.sample
228         .idx]],
229         treatment.alignments.peaks.chunk.list[[treatment.
230             sample.idx]])
231     if(is.null(all.peaks.gr.treatment.cov[[treatment.sample.idx]])){
232         all.peaks.gr.treatment.cov[[treatment.sample.idx]] <-
233             coverage(treatment.alignments.chunk.list[[treatment.sample.idx]])
234     } else {
235         all.peaks.gr.treatment.cov[[treatment.sample.idx]] <-
236             all.peaks.gr.treatment.cov[[treatment.sample.idx]] +
237             coverage(treatment.alignments.chunk.list[[treatment.sample.idx]])
238     }
239     # remove alignments chunk
240     rm(control.alignments.chunk.list)
241     rm(control.alignments.peaks.chunk.list)
242     rm(treatment.alignments.chunk.list)
243     rm(treatment.alignments.peaks.chunk.list)
244 }
245 cat("End reading coverage by chunks from BAM files ... \n\n")
246 #####
247 # close BAM files and remove variables
248 for(control.sample.idx in 1:n.control.samples){
249     close(bam.files.control.list[[control.sample.idx]])
250 }
251 rm(bam.files.control.list)
252 for(treatment.sample.idx in 1:n.treatment.samples){
253     close(bam.files.treatment.list[[treatment.sample.idx]])
254 }
255 rm(bam.files.treatment.list)
256 cat("End reading coverage by chunks from BAM files ... \n\n")
257 #####
258 control.total.coverage <- 0
259 for(control.sample.idx in 1:n.control.samples){
260     control.total.coverage <-
261         control.total.coverage + control.chr.coverage.list[[control.sample.idx]]
262 }
263 treatment.total.coverage <- 0
264 for(treatment.sample.idx in 1:n.treatment.samples){
265     treatment.total.coverage <-
266         treatment.total.coverage + treatment.chr.coverage.list[[treatment.sample.
267             idx]]
268 }
269 cat("End reading peak files ... \n")
270 #####
271 # Find peaks significance
272 cat("Find peak significance ... \n")
273
274 all.peaks.control.n <- vector("list", n.control.samples)
275 for(control.sample.idx in 1:n.control.samples){
276     all.peaks.control.n[[control.sample.idx]] <-
277         array(0, dim=length(chr.len))
278     names(all.peaks.control.n[[control.sample.idx]]) <- names(chr.len)
279 }
280
281 all.peaks.treatment.n <- vector("list", n.treatment.samples)
282 for(treatment.sample.idx in 1:n.treatment.samples){
283     all.peaks.treatment.n[[treatment.sample.idx]] <-
284         array(0, dim=length(chr.len))
285     names(all.peaks.treatment.n[[treatment.sample.idx]]) <- names(chr.len)
286 }
287
288 #####

```

```

289 # redefine BAM files variables
290 cat("Redefine BAM files variables ... \n")
291 bam.files.control.list <- vector("list", n.control.samples)
292 bam.files.treatment.list <- vector("list", n.treatment.samples)
293 for(control.sample.idx in 1:n.control.samples){
294   bam.files.control.list[[control.sample.idx]] <-
295     BamFile(bam.files.control.path.list[[control.sample.idx]])
296 }
297 for(treatment.sample.idx in 1:n.treatment.samples){
298   bam.files.treatment.list[[treatment.sample.idx]] <-
299     BamFile(bam.files.treatment.path.list[[treatment.sample.idx]])
300 }
301
302 cat("Find noise mean and variance.\n")
303 mu.control.noise.list <-
304   lapply(vector("list", n.control.samples), function(x) x<-vector("list",
305     length(chr.len)))
306 var.control.noise.list <-
307   lapply(vector("list", n.control.samples), function(x) x<-vector("list",
308     length(chr.len)))
309
310 mu.treatment.noise.list <-
311   lapply(vector("list", n.treatment.samples), function(x) x<-vector("list",
312     length(chr.len)))
313 var.treatment.noise.list <-
314   lapply(vector("list", n.treatment.samples), function(x) x<-vector("list",
315     length(chr.len)))
316
317 cat("Find noise mean and variance - control.\n")
318 for(control.sample.idx in 1:n.control.samples){
319   for(chr in names(peaks.chr.width)){
320     chr.idx <- which(names(chr.len) == chr)
321     #####
322     # alpha parameter of Beta distribution
323     alpha.control.prior <-
324       as.numeric(chr.len[chr.idx]) - as.numeric(peaks.chr.width[chr])
325     n.alpha.control.likelihood <-
326       as.numeric(control.chr.coverage.list[[control.sample.idx]][chr.idx]) -
327       as.numeric(control.chr.peaks.coverage.list[[control.sample.idx]][chr.idx]
328       )
329     #####
330     # beta parameter of Beta distribution
331     beta.control.prior <-
332       as.numeric(peaks.chr.width[chr])
333     n.beta.control.likelihood <-
334       control.chr.peaks.coverage.list[[control.sample.idx]][chr.idx]
335     #####
336     # posterior alpha and beta parameters of Beta distribution
337     alpha.control.posterior <-
338       alpha.control.prior + n.alpha.control.likelihood
339     beta.control.posterior <-
340       beta.control.prior + n.beta.control.likelihood
341     mu.control.noise.list[[control.sample.idx]][chr.idx] <-
342       digamma(alpha.control.posterior) -
343       digamma(beta.control.posterior)
344     var.control.noise.list[[control.sample.idx]][chr.idx] <-
345       trigamma(alpha.control.posterior) +
346       trigamma(beta.control.posterior)
347   }
348 }
349
350 cat("Find noise mean and variance - treatment.\n")
351 for(treatment.sample.idx in 1:n.treatment.samples){
352   for(chr in names(peaks.chr.width)){
353     chr.idx <- which(names(chr.len) == chr)

```

```

349 #####
350 # alpha parameter of Beta distribution
351 alpha.treatment.prior <-
352   as.numeric(chr.len[chr.idx]) - as.numeric(peaks.chr.width[chr])
353 n.alpha.treatment.likelihood <-
354   as.numeric(treatment.chr.coverage.list[[treatment.sample.idx]][chr.idx])
355   -
356   as.numeric(treatment.chr.peaks.coverage.list[[treatment.sample.idx]][chr
357     .idx])
358 #####
359 # beta parameter of Beta distribution
360 beta.treatment.prior <-
361   as.numeric(peaks.chr.width[chr])
362 n.beta.treatment.likelihood <-
363   treatment.chr.peaks.coverage.list[[treatment.sample.idx]][chr.idx]
364 #####
365 # posterior alpha and beta parameters of Beta distribution
366 alpha.treatment.posterior <-
367   alpha.treatment.prior + n.alpha.treatment.likelihood
368 beta.treatment.posterior <-
369   beta.treatment.prior + n.beta.treatment.likelihood
370 mu.treatment.noise.list[[treatment.sample.idx]][chr.idx] <-
371   digamma(alpha.treatment.posterior) -
372   digamma(beta.treatment.posterior)
373 var.treatment.noise.list[[treatment.sample.idx]][chr.idx] <-
374   trigamma(alpha.treatment.posterior) +
375   trigamma(beta.treatment.posterior)
376 }
377 }
378 #####
379 cat("chr data - control\n")
380 chr.data.control <- vector("list", n.control.samples)
381 for(control.sample.idx in 1:n.control.samples){
382   chr.data.control[[control.sample.idx]]$all.peaks.gr.control.cov <-
383     all.peaks.gr.control.cov[[control.sample.idx]]
384   chr.data.control[[control.sample.idx]]$control.chr.peaks.coverage.list <-
385     control.chr.peaks.coverage.list[[control.sample.idx]]
386   chr.data.control[[control.sample.idx]]$mu.control.noise.list <-
387     mu.control.noise.list[[control.sample.idx]]
388   chr.data.control[[control.sample.idx]]$var.control.noise.list <-
389     var.control.noise.list[[control.sample.idx]]
390 }
391 cat("chr data - treatment.\n")
392 chr.data.treatment <- vector("list", n.treatment.samples)
393 for(treatment.sample.idx in 1:n.treatment.samples){
394   chr.data.treatment[[treatment.sample.idx]]$all.peaks.gr.treatment.cov <-
395     all.peaks.gr.treatment.cov[[treatment.sample.idx]]
396   chr.data.treatment[[treatment.sample.idx]]$treatment.chr.peaks.coverage.list
397     <-
398     treatment.chr.peaks.coverage.list[[treatment.sample.idx]]
399   chr.data.treatment[[treatment.sample.idx]]$mu.treatment.noise.list <-
400     mu.treatment.noise.list[[treatment.sample.idx]]
401   chr.data.treatment[[treatment.sample.idx]]$var.treatment.noise.list <-
402     var.treatment.noise.list[[treatment.sample.idx]]
403 }
404 cat("Start mclapply ... \n")
405 cat("Number of peaks: ", length(unlist(all.peaks.gr)), "\n")
406 ti <- proc.time()
407 all.peaks.gr.complete <- all.peaks.gr
408 for(chr.current in names(peaks.chr.width)){
409   cat("Chromosome ", chr.current, "\n")

```

```

410 all.peaks.gr <- all.peaks.gr.complete[which(seqnames(all.peaks.gr.complete)==
      chr.current)]
411 cat("Number of peaks for this chromosome: ", length(unlist(all.peaks.gr)), "\n
      ")
412 n.peaks <- length(all.peaks.gr)
413 progress <- 0.0
414 x <- split(all.peaks.gr, ceiling(seq_along(all.peaks.gr)/1000))
415 significant.peaks <- local({
416   f <- fifo(tempfile(), open="w+b", blocking=T)
417   if (inherits(parallel::mcfork(), "masterProcess")) {
418     # Child
419     while (progress < 1 && !isIncomplete(f)) {
420       msg <- readBin(f, "double")
421       progress <- progress + as.numeric(msg)
422       cat(sprintf("Progress: %.7f%%\n", progress * 100))
423     }
424     parallel::mcexit()
425   }
426   result.peaks <- lapply(x, function(peaks.list){
427
428     result <- mclapply(peaks.list, mc.cores = detectCores(), function(peak)
429     {
430       writeBin(1/n.peaks, f)
431       peak.seqname <- as.character(seqnames(peak))
432       peak.range <- ranges(peak)
433       peak.chr.idx <- which((peak.seqname)==names(chr.len))
434       peak.width <- width(peak.range)
435       #####
436       total.peak.control.alignments <-
437         lapply(chr.data.control,
438               function(x){
439           y <- as.vector(x$all.peaks.gr.control.cov[peak][[peak.seqname]])
440           sum.array(length(y), y, 0)$sum)
441       mu.control.peak.list <-
442         mapply(function(x,y) {
443           (digamma(y+peak.width)-
444            digamma((x$control.chr.peaks.coverage.list[peak.chr.idx]-y) +
445                  (as.numeric(peaks.chr.width[peak.seqname]) - peak.width))) +
446           (abs(x$mu.control.noise.list[[peak.chr.idx]]))},
447              x=chr.data.control, y=total.peak.control.alignments)
448       var.control.peak.list <-
449         mapply(function(x,y) {
450           (trigamma(y+peak.width)+
451            trigamma((x$control.chr.peaks.coverage.list[peak.chr.idx]-y) +
452                  (as.numeric(peaks.chr.width[peak.seqname]) - peak.width)))
453           +
454           (x$var.control.noise.list[[peak.chr.idx]])},
455              x=chr.data.control, y=total.peak.control.alignments)
456       control.limits.list <-
457         mapply(function(x, y) {
458           c(l.inf=(x-5*sqrt(y)),
459             l.sup=(x+5*sqrt(y)))},
460              x=mu.control.peak.list, y=var.control.peak.list)
461       #####
462       total.peak.treatment.alignments <-
463         lapply(chr.data.treatment,
464               function(x){
465           y <- as.vector(x$all.peaks.gr.treatment.cov[peak][[peak.seqname
466             ]])
467           sum.array(length(y), y, 0)$sum)
468       mu.treatment.peak.list <-
469         mapply(function(x,y) {
470           (digamma(y+peak.width)-

```

```

471         (as.numeric(peaks.chr.width[peak.seqname]) - peak.width)))
472         +
473         (abs(x$mu.treatment.noise.list[[peak.chr.idx]]))},
474     x=chr.data.treatment, y=total.peak.treatment.alignments)
475 var.treatment.peak.list <-
476     mapply(function(x,y) {
477         (trigamma(y+peak.width)+
478          trigamma((x$treatment.chr.peaks.coverage.list[peak.chr.idx]-y) +
479                  (as.numeric(peaks.chr.width[peak.seqname]) - peak.width))) +
480         (x$var.treatment.noise.list[[peak.chr.idx]])},
481     x=chr.data.treatment, y=total.peak.treatment.alignments)
482 treatment.limits.list <-
483     mapply(function(x, y) {
484         c(l.inf=(x-5*sqrt(y)),
485           l.sup=(x+5*sqrt(y))),
486     x=mu.treatment.peak.list, y=var.treatment.peak.list)
487 #####
488 logodds.limits <- data.frame(min=min(min(treatment.limits.list["l.inf",]),
489                                       min(control.limits.list["l.inf",])),
490                              max=max(max(treatment.limits.list["l.sup",]),
491                                       max(control.limits.list["l.sup",])))
492 #####
493 logodds.values <- seq(logodds.limits$min, logodds.limits$max,
494                      by=(logodds.limits$max-logodds.limits$min)/1000)
495 #####
496 f.logodds.control.sample <- function(logodds.sequence, control.sample.idx){
497     dnorm(logodds.sequence, mean=(mu.control.peak.list[[control.sample.idx]]),
498          sd=sqrt(var.control.peak.list[[control.sample.idx]]))}
499 #####
500 f.logodds.treatment.sample <- function(logodds.sequence, treatment.sample.
501 idx){
502     dnorm(logodds.sequence, mean=(mu.treatment.peak.list[[treatment.sample.idx
503 ]]),
504          sd=sqrt(var.treatment.peak.list[[treatment.sample.idx]]))}
505 #####
506 f.logodds.control <- function(logodds.values) {
507     control.logodds.dist.weighted <-
508     mapply(function(x, y, z) {
509         x[peak.chr.idx]*dnorm(logodds.values, mean=y, sd=sqrt(z))),
510     x=control.chr.coverage.list,
511     y=mu.control.peak.list,
512     z=var.control.peak.list)
513     control.logodds.dist.w.average <-
514     apply(control.logodds.dist.weighted, 1, sum)/control.total.coverage[peak
515 .chr.idx]
516 }
517 #####
518 f.logodds.treatment <- function(logodds.values) {
519     treatment.logodds.dist.weighted <-
520     mapply(function(x, y, z) {
521         x[peak.chr.idx]*dnorm(logodds.values, mean=y, sd=sqrt(z))),
522     x=treatment.chr.coverage.list,
523     y=mu.treatment.peak.list,
524     z=var.treatment.peak.list)
525     treatment.logodds.dist.w.average <-
526     apply(treatment.logodds.dist.weighted, 1, sum)/treatment.total.coverage[
527 peak.chr.idx]
528 }
529 #####
530 product.f.logodds <- function(logodds.treatment.values, logodds.control.
531 values){
532     f.logodds.treatment(logodds.treatment.values)*f.logodds.control(logodds.
533 control.values)}
534 xmax <- function(x) {x}
535 prob.logodds.treatment.gr.control <-

```

```

529     try( integral2( f=product.f.logoods ,
530                   logodds.limits$min,
531                   logodds.limits$max,
532                   logodds.limits$min,
533                   xmax), silent = TRUE)
534   while( inherits( prob.logoods.treatment.gr.control , 'try-error' )){
535     cat( "prob.logoods.t.greater.c | Error!\n")
536     logodds.limits$min <- logodds.limits$min - abs(logodds.limits$min)
537     logodds.limits$max <- logodds.limits$max + abs(logodds.limits$max)
538     prob.logoods.treatment.gr.control <-
539       try( integral2( f=product.f.logoods ,
540                     logodds.limits$min,
541                     logodds.limits$max,
542                     logodds.limits$min,
543                     xmax), silent = TRUE)
544   }
545   #####
546   p <- prob.logoods.treatment.gr.control$Q
547   if (p>1){ p <- 1 }
548   if (p<0){ p <- 0 }
549   #####
550   c(peak.seqname, start(peak.range), end(peak.range), p)
551 })
552 result
553 })
554 close(f)
555 result.peaks
556 })
557
558 tf <- proc.time()
559 cat( "total time: ", sum(tf-ti), "\n")
560 df.significant.peaks <- data.frame( matrix( unlist( significant.peaks), nrow = n.
561                                           peaks, byrow=T)
562                                     file=paste( output.file.name, "-peaks.", chr.current, ".bed", sep=""),
563                                     quote = FALSE,
564                                     sep="\t",
565                                     row.names = FALSE,
566                                     col.names = FALSE)
567 cat( "End ... writing data frame ", chr.current, " ... \n")
568 }
569 sink()
570 }

```

Bibliography

- Atchison, J. and Shen, S. M. Logistic-normal distributions: Some properties and uses. *Biometrika*, 67(2):261–272, 1980. [24](#), [38](#)
- Barrett, T., Wilhite, S. E., Ledoux, P., Evangelista, C., Kim, I. F., Tomashevsky, M., Marshall, K. A., Phillippy, K. H., Sherman, P. M., Holko, M., et al. Ncbi geo: archive for functional genomics data sets-update. *Nucleic acids research*, 41(D1):D991–D995, 2013. [53](#)
- Bergsland, M., Ramsköld, D., Zaouter, C., Klum, S., Sandberg, R., and Muhr, J. Sequentially acting sox transcription factors in neural lineage development. *Genes & development*, 25(23):2453–2464, 2011. [53](#)
- Bylsma, W. Approximating smooth step functions using partial fourier series sums. Technical report, DTIC Document, 2012. [48](#)
- Chinwalla, A. T., Cook, L. L., Delehaunty, K. D., Fewell, G. A., Fulton, L. A., Fulton, R. S., Graves, T. A., Hillier, L. W., Mardis, E. R., McPherson, J. D., et al. Initial sequencing and comparative analysis of the mouse genome. *Nature*, 420(6915):520–562, 2002. [2](#)
- de Bragança Pereira, C. A. and Stern, J. M. Special characterizations of standard discrete models. *REVSTAT-Statistical Journal*, 6(3):199–230, 2008. [24](#), [38](#)
- DeGroot, M. H. M. H. et al. *Probability and statistics*. Number 04; QA273, D4 1986. 1986. [1](#)
- Deliard, S., Zhao, J., Xia, Q., and Grant, S. F. Generation of high quality chromatin immunoprecipitation dna template for high-throughput sequencing (chip-seq). *JoVE (Journal of Visualized Experiments)*, (74):e50286–e50286, 2013. [5](#)
- Edgar, R., Domrachev, M., and Lash, A. E. Gene expression omnibus: Ncbi gene expression and hybridization array data repository. *Nucleic acids research*, 30(1):207–210, 2002. [8](#), [53](#), [54](#), [56](#)
- Ferguson, T. S. A bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230, 1973. [13](#), [38](#), [40](#)
- Frigyik, B., Kapila, A., and Gupta, M. Introduction to the dirichlet distribution and related processes, university of washington technical report. Technical report, UWEETR-2010-0006, 2010. [13](#)
- Heinz, S., Benner, C., Spann, N., Bertolino, E., Lin, Y. C., Laslo, P., Cheng, J. X., Murre, C., Singh, H., and Glass, C. K. Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and b cell identities. *Molecular cell*, 38(4):576–589, 2010. [2](#)
- Hower, V., Evans, S. N., and Pachter, L. Shape-based peak identification for chip-seq. *BMC bioinformatics*, 12(1):15, 2011. [1](#), [2](#), [3](#)
- James, I. R. and Mosimann, J. E. A new characterization of the dirichlet distribution through neutrality. *The Annals of Statistics*, pages 183–189, 1980. [40](#)

- Johnson, N. L. Systems of frequency curves generated by methods of translation. *Biometrika*, 36 (1/2):149–176, 1949. [24](#)
- Jothi, R., Cuddapah, S., Barski, A., Cui, K., and Zhao, K. Genome-wide identification of in vivo protein–dna binding sites from chip-seq data. *Nucleic acids research*, 36(16):5221–5231, 2008. [1](#)
- Kim, D., Pertea, G., Trapnell, C., Pimentel, H., Kelley, R., and Salzberg, S. L. Tophat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome biology*, 14(4):1, 2013. [8](#)
- Langmead, B. and Salzberg, S. L. Fast gapped-read alignment with bowtie 2. *Nature methods*, 9 (4):357–359, 2012. [8](#), [47](#)
- Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R., et al. The sequence alignment/map format and samtools. *Bioinformatics*, 25(16): 2078–2079, 2009. [8](#), [58](#)
- Lindley, D. V. The bayesian analysis of contingency tables. *The Annals of Mathematical Statistics*, pages 1622–1643, 1964. [24](#)
- McAninch, D. and Thomas, P. Identification of highly conserved putative developmental enhancers bound by sox3 in neural progenitors using chip-seq. *PloS one*, 9(11):e113361, 2014. [5](#), [8](#), [53](#), [56](#), [77](#)
- Park, P. J. Chip-seq: advantages and challenges of a maturing technology. *Nature Reviews Genetics*, 10(10):669–680, 2009. [1](#)
- Petri, C. *Relação entre níveis de significância Bayesiano e freqüentista: e-value e p-value em tabelas de contingência*. PhD thesis, Universidade de São Paulo, 2007. [24](#)
- Robertson, G., Schein, J., Chiu, R., Corbett, R., Field, M., Jackman, S. D., Mungall, K., Lee, S., Okada, H. M., Qian, J. Q., et al. De novo assembly and analysis of rna-seq data. *Nature methods*, 7(11):909–912, 2010. [8](#)
- Rodrigues, W. W. *Teste de significância em tabelas de contingência 2x2 usando o modelo logístico-normal*. PhD thesis, 2006. [24](#)
- Rosenbloom, K. R., Armstrong, J., Barber, G. P., Casper, J., Clawson, H., Diekhans, M., Dreszer, T. R., Fujita, P. A., Guruvadoo, L., Haeussler, M., et al. The ucsc genome browser database: 2015 update. *Nucleic acids research*, 43(D1):D670–D681, 2015. [8](#), [53](#)
- Schneider, V. and Church, D. Genome reference consortium. 2013. [9](#), [53](#)
- Spyrou, C., Stark, R., Lynch, A. G., and Tavaré, S. Bayespeak: Bayesian analysis of chip-seq data. *BMC bioinformatics*, 10(1):1, 2009. [3](#)
- Stern, J. M. Cognitive constructivism and the epistemic significance of sharp statistical hypotheses. *Tutorial book for MaxEnt*, pages 6–11, 2008. [1](#)
- Thomas, R., Thomas, S., Holloway, A. K., and Pollard, K. S. Features that define the best chip-seq peak calling algorithms. *Briefings in bioinformatics*, page bbw035, 2016. [1](#)
- Trapnell, C., Roberts, A., Goff, L., Pertea, G., Kim, D., Kelley, D. R., Pimentel, H., Salzberg, S. L., Rinn, J. L., and Pachter, L. Differential gene and transcript expression analysis of rna-seq experiments with tophat and cufflinks. *Nature protocols*, 7(3):562–578, 2012. [8](#), [62](#)
- Wilbanks, E. G. and Facciotti, M. T. Evaluation of algorithm performance in chip-seq peak detection. *PloS one*, 5(7):e11471, 2010. [1](#)

- Wu, Q., Won, K.-J., and Li, H. Nonparametric tests for differential histone enrichment with chip-seq data. *Cancer informatics*, 14(Suppl 1):11, 2015. [1](#)
- Zambelli, F., Pesole, G., and Pavesi, G. Motif discovery and transcription factor binding sites before and after the next-generation sequencing era. *Briefings in bioinformatics*, page bbs016, 2012. [1](#)
- Zang, C., Schones, D. E., Zeng, C., Cui, K., Zhao, K., and Peng, W. A clustering approach for identification of enriched domains from histone modification chip-seq data. *Bioinformatics*, 25(15):1952–1958, 2009. [2](#)
- Zhang, Y., Liu, T., Meyer, C. A., Eeckhoutte, J., Johnson, D. S., Bernstein, B. E., Nusbaum, C., Myers, R. M., Brown, M., Li, W., et al. Model-based analysis of chip-seq (macs). *Genome Biol*, 9(9):R137, 2008. [1](#), [2](#), [53](#), [54](#)