LARGE SCALE COMPUTATIONS IN GENOMIC AND EPIGENOMIC ANALYSIS

by

Chandler Zuo

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy
(Statistics)

at the

UNIVERSITY OF WISCONSIN–MADISON
2015

Date of final oral examination: 5/29/2015

The dissertation is approved by the following members of the Final Oral Committee:

Sündüz Keleş, Professor, Statistics
Emery H. Bresnick, Professor, Cell and Regenerative Biology
Michael A. Newton, Professor, Statistics
Karl Rohe, Assistant Professor, Statistics
Kam-Wah Tsui, Professor, Statistics

LARGE SCALE COMPUTATIONS IN GENOMIC AND EPIGENOMIC ANALYSIS

Chandler Zuo

Under the supervision of Professor Sündüz Keleş

At the University of Wisconsin-Madison

Genomic and epigenomic studies aim to elucidate genomic regulatory mechanisms under various biological conditions. The next-generation sequencing technology has been widely applied in this area to generate vast data from different organisms, cell types and experiments. The availability of these data has motivated me to develop several computational algorithms with data scalability and time efficiency.

Chapter 2 introduces an empirical Bayesian framework, **C**hIP-**S**eq **S**tatistical **P**ower (CSSP), for calculating the required sequencing depth for ChIP-seq experiments. ChIP-seq is the state-of-the-art technology to study transcription factor binding and protein interactions. The sequencing depth of such an experiment determines the power of detecting interacting genome regions with the protein. By predicting statistical power with multiple testing adjustment, CSSP facilitates the experimental design using low-sequenced pilot experiments.

Chapter 3 introduces a software package, atSNP (**a**ffinity **t**esting for **S**ingle **N**ucleotide **P**olymorphism), a highly scalable computational tool to identify putative regulatory SNPs using transcription factor binding motifs. atSNP implements innovative algorithms using the importance sampling technique. It easily scales up to analyses involving millions of SNP-motif pairs, which can not be achieved using the existing tools.

Chapter 4 and 5 studies the integrative modeling for general genomic and epigenomic data. Chapter 4 introduces the MBASIC framework (**M**atrix **B**ased **A**nalysis for **S**tate-space **I**nference and **C**lustering), a unified approach to analyze data from different types of experiments, including but not restricted to transcription factor binding, gene expression and allele-specific binding. I have also developed an Expectation and Maximization algorithm to jointly estimate all parameters in the hierarchical model. In Chapter 5, I cast the MBASIC framework in a Bayesian setting to develop a MAD-Bayes algorithm. This algorithm is derived under the small-variance asymptotic

view of the K-means algorithm. It shows an order-of-magnitude decrease in time costs compared to the Expectation and Maximization algorithm.

ACKNOWLEDGMENTS

This could not have been done without the guidance of my PhD advisor, Professor Sündüz Keleş. I want to thank her for providing exciting work, excellent advice, and ample funding. Her guidance has shaped me into a more rigorous thinker, critical writer, and skilled researcher. I consider myself incredibly lucky to have the chance to learn from her, a point that could not possibly be overstated. I also owe my gratitude to all current and previous members in Professor Keleş's research group. They have nourished an active and constructive research environment that made my past five-year experience life memorable.

I would also like to thank Professor Kam Tsui, who has introduced to me many statistical topics that are related and tremendously valuable to my work. He always peppered our conversations with humour and life wisdom, and have continuously instilled an interest in research.

Professor Michael Newton and Doctor Karl Rohe deserve thanks, as they have made very insightful comments on my work. The time they have spent for my benefit is sincerely appreciated. Thanks to Professor Emery Bresnick for providing the real data to apply my models and evaluate their real life significance.

Finally, I want to thank my parents, Yanyan Zuo and Yuying Fang, for their love and support. I owe them so much and I know that I am very lucky to have them. They were certainly the most important people in shaping who I am today.

TABLE OF CONTENTS

Page

Appendix

Page

LIST OF TABLES

LIST OF FIGURES

Appendix
Figure Page

Appendix
Figure Page

Appendix
Figure Page

Appendix
Figure Page

Appendix
Figure Page

ABSTRACT

Genomic and epigenomic studies aim to elucidate genomic regulatory mechanisms under various biological conditions. The next-generation sequencing technology has been widely applied in this area to generate vast data from different organisms, cell types and experiments. The availability of these data has motivated me to develop several computational algorithms with data scalability and time efficiency.

Chapter 2 introduces an empirical Bayesian framework, **C**hIP-**S**eq **S**tatistical **P**ower (CSSP), for calculating the required sequencing depth for ChIP-seq experiments. ChIP-seq is the state-of-the-art technology to study transcription factor binding and protein interactions. The sequencing depth of such an experiment determines the power of detecting interacting genome regions with the protein. By predicting statistical power with multiple testing adjustment, CSSP facilitates the experimental design using low-sequenced pilot experiments.

Chapter 3 introduces a software package, atSNP (**a**ffinity **t**esting for **S**ingle **N**ucleotide **P**olymorphism), a highly scalable computational tool to identify putative regulatory SNPs using transcription factor binding motifs. atSNP implements innovative algorithms using the importance sampling technique. It easily scales up to analyses involving millions of SNP-motif pairs, which can not be achieved using the existing tools.

Chapter 4 and 5 studies the integrative modeling for general genomic and epigenomic data. Chapter 4 introduces the MBASIC framework (**M**atrix **B**ased **A**nalysis for **S**tate-space **I**nference and **C**lustering), a unified approach to analyze data from different types of experiments, including but not restricted to transcription factor binding, gene expression and allele-specific binding. I

have also developed an Expectation and Maximization algorithm to jointly estimate all parameters in the hierarchical model. In Chapter 5, I cast the MBASIC framework in a Bayesian setting to develop a MAD-Bayes algorithm. This algorithm is derived under the small-variance asymptotic view of the K-means algorithm. It shows an order-of-magnitude decrease in time costs compared to the Expectation and Maximization algorithm.

Chapter 1

Introduction

1.1 Background

Modern genomic and epigenomic studies focus on the global detection of the genome regions that interact with biochemical factors and regulate gene activities. Compared to the genetics and epigenetic research that investigate the roles and the functions of single genome regions, genomics and epigenomics primarily focus on the global functional system of the genome through discovery-based approaches. The investigation for the whole genome network has been made possible by the advances in the next-generation sequencing technologies, which can produce millions of DNA sequences emitted from all genome regions to reflect their initial association with the biochemical factors. One example is the ChIP-seq (Chromatin Immunoprecipitation followed by Sequencing) experiments, which uses the next-generation sequencing to produce precipitated DNA sequences around protein binding regions in order to analyze DNA-protein interactions. It is believed that determining how DNA-protein interactions affect the gene expression is essential to understand many biological processes and disease states. The ENCODE consortium now curates ChIP-seq data sets from hundreds of transcription factor experiments from a standardized pipeline to help decipher the whole genome transcription network. Given the structures and the scales of such data, there are several challenges in applying the conventional statistical models.

First, a typical sequencing data set involves hundreds of thousands of observations, an order of magnitude beyond the scope of genetic and epigenetic analysis. As a result, any statistical models designed for such data should emphasize on the time cost of model estimation. Depending on the objectives of such analyses, it might be desirable to trade off the model complexity for efficiency,

and formulate models that can be solved via parallel as opposed to sequential procedures. The computational performance is strongly emphasized throughout all chapters in this thesis. Designing models with efficient algorithms while maintaining their generality to different data structures sometimes leads to a dilemmatic situation which requires considerable efforts to resolve.

Second, given the data generation mechanisms in the sequencing experiments, the conventional experimental design framework sometimes needs be modified. For the ChIP-seq experiment, because the state of a genome location is estimated from the number of DNA sequences piling there, the accuracy or the statistical power of inferring such states is more influenced by the total number of DNA sequences obtained throughout the genome, typically referred to as the "sequencing depth" of such an experiment. On the other hand, experimental replicates are generated in a small number (usually 2 to 3), with the main purpose of experimental validation. As a result, to ensure the statistical power in the designing stage, the concept of "sample size" should refer to the sequencing depth rather than the number of replicates as in a conventional setting. Consequently, for "sample size calculation" we would require modeling the process for the sequence accumulation at individual locations, and the classical methods derived under the analysis of variance framework are not applicable. This problem is discussed in Chapter 2, where I model the power of such an analysis as the function of the sequencing depth to guide the experimental design.

Third, for the discovery-based approaches on the whole genome scale, false discovery rate (FDR) control becomes an necessary step to guard against the Type-I errors. Its impact on a state prediction model can be usually addressed by a post-processing step on the p-values, which is straightforward. For other types of models, its impact may be indirect and nested with the model design. For example, for the power analysis, we need redefine the power as subject to the FDR control. For algorithms to compute large sets of p-values, we should ensure the accuracy in obtaining those small p-values that can survive the FDR adjustment. Such consideration is taken in both the power analysis model in Chapter 2, and the p-value computation algorithm in Chapter 3.

Fourth, one common target of all sequencing experiments is to detect regions in particular association states with a biochemical factor. Mapping the actual data from different experiments to

a homogeneous state space requires models that can adapt to different distributional assumptions and robust to their violations. Most of the existing models were designed for specific experiments or data types, neglecting the commonalities underlying their model structures. As the sequencing technology evolves, new experimental protocols are introduced, and the demand of integrating different data types increase, these models are unlikely to be applicable in future research. In Chapter 4 and 5, I introduce generalized models and algorithms that address the common modeling structures while adapting to different experimental and distributional settings.

Such challenges have triggered the development of new statistical models and algorithms. In this thesis, I introduce four newly developed statistical tools in this area.

1.2 Overview of the Chapters

In Chapter 2, I discuss the "sample size design" problem in the ChIP-seq experiments, and present a statistical framework named CSSP (**C**hIP-**S**eq **S**tatistical **P**ower) for power calculations in ChIP-seq experiments by considering a local Poisson model which is commonly adopted by many peak callers. Evaluations with simulations and data-driven computational experiments demonstrate that this framework can reliably estimate the power of a ChIP-seq experiment at different sequencing depths based on pilot data. Furthermore, it provides an analytical approach for calculating the required depth for a targeted power while controlling the false discovery rate at a user-specified level. Hence, our results enable researchers to utilize their own or publicly available data for determining required sequencing depths of their ChIP-seq experiments and potentially make better use of the multiplexing functionality of the sequencers.

In Chapter 3, I describe atSNP (**a**ffinity **t**esting for regulatory **SNP**s), a computationally efficient `R` package for identifying rSNPs *in silico*. Regulatory SNPs (rSNPs) are such SNPs that affect gene regulation by changing transcription factor (TF) binding affinities to genomic sequences. The current *In silico* methods that evaluate the impact of SNPs on TF binding affinities are not scalable for large-scale analysis. atSNP implements an importance sampling algorithm coupled with a first-order Markov model for the background nucleotide sequences to test the significance of affinity scores and SNP-driven changes in these scores. Application of atSNP with $>$20K SNPs indicates

that atSNP is the only available tool for such a large-scale task. Evaluations of atSNP with known rSNP-TF interactions indicates that rSNP is able to prioritize motifs for a given set of SNPs with high accuracy.

Chapter 4 and 5 discuss the integrative analysis of multiple genomic and epigenomic data sets.In recent years, a large number of genomic and epigenomic studies have been focusing on the integrative analysis of multiple experimental datasets measured over a large number of observational units. The objectives of such studies include not only inferring a hidden state of activity for each unit over individual experiments, but also detecting highly associated clusters of units based on their inferred states. In Chapter 4, I develop the MBASIC (**M**atrix **B**ased **A**nalysis for **S**tate-space **I**nference and **C**lustering) framework. MBASIC consists of two parts: state-space mapping and state-space clustering. In state-space mapping, it maps observations onto a finite state-space, representing the activation states of units across conditions. In state-space clustering, MBASIC incorporates a finite mixture model to cluster the units based on their inferred state-space profiles across all conditions. Both the state-space mapping and clustering can be simultaneously estimated through an Expectation and Maximization algorithm. MBASIC flexibly adapts to a large number of parametric distributions for the observed data, as well as the heterogeneity in replicate experiments. It allows for imposing structural assumptions on each cluster, and enables model selection using information criterion. A number of numerical studies using both synthetic and real data demonstrate that MBASIC is a unified framework applicable for a wide range of problems such as transcription factor binding network, differential expression, allele-specific binding.

In Chapter 5, I propose a MAD-Bayes algorithm for the state-space inference and clustering in joint analysis of multiple data sets. The MAD-Bayes framework was proposed by [6] as a general procedure to derive fast-converging K-means algorithm from a small-variance-asymptotic view. The MAD-Bayes algorithm is developed on a MBASIC model with the Bayesian structure. I also develop methods for model initialization and tuning parameter selection. A number of numerical experiments are presented which shows that my algorithm is robust against model assumptions and the local optima problem. Most importantly, it shows an order-of-magnitude decrease in time costs compared to the MBASIC's Expectation and Maximization algorithm.

Chapter 2

A Statistical Framework for Power Calculations in ChIP-seq Experiments [1]

2.1 Introduction

Next-generation sequencing technologies produce tens of millions of sequence reads during each instrument run and are employed to answer questions central to human diseases. Multiple NIH consortia (ENCODE, modENCODE, 1000 Genomes, Roadmap Epigenome) are pursuing mapping of transcription factor (TF) binding and epigenome in multiple tissues and developmental stages with ChIP-seq applications ([28, 16, 46, 58, 48]). Analysis of ChIP-seq experiments involves comparing sequence reads from a ChIP sample to an appropriate control sample (e.g., chromatin input) to identify genomic loci/regions that exhibit enrichment in the ChIP sample compared to the control sample. Although there are more than 30 algorithms for analyzing data from ChIP-seq experiments (reviewed in [74]), there has been little and mostly empirically driven efforts on the design of these experiments ([20]). Identification of biologically interesting genomic regions can be hindered by background noise. Detection of these regions can be improved by sequencing more reads. The total number of reads from a sequencing experiment is referred to as the sequencing depth. Sequencing depths so far have been set empirically due to lack of a formal statistical framework, e.g., the ENCODE Consortium suggested using a minimum sequencing depth of 20 million (M) mapped reads for sequence-specific TFs ([48]); however, [8] recently concluded with empirical studies that the regularly adopted sequencing depth of 15-20 M reads in humans may not be high enough.

[1] The manuscript for this chapter is published in [85]. Method in this chapter is implemented in the `R` package `CSSP` and is freely available at `http://bioconductor.org/packages/release/bioc/html/CSSP.html`.

[29] and [60] explored the impact of sequencing depths of ChIP samples using saturation analysis. This analysis evaluated the effect of sequencing depth on the number of peaks discovered by identifying peaks from reads sub-sampled at varying proportions from the original ChIP sample. The proportion of sub-sample peaks that overlap the peaks from the full set is plotted against the sub-sample depth. When this curve reaches a horizontal asymptote, it indicates that the set of detected enrichment sites has stabilized at the current depth. Although this computational approach is useful for evaluating the available sequencing depth of a ChIP sample, it has three major drawbacks: (1) it is not suited for addressing how many more reads are needed if saturation has not been reached at the available depth (e.g., in a recent ENCODE publication ([48]), RNA Pol II, which mainly interacts with DNA across genes, exhibited a nearly linear gain in the number of peaks through 50 M reads with no indication of how many more reads are needed for saturation); (2) it only evaluates saturation based on the ChIP sample and discards the control sample; (3) it only allows investigating saturation from the point of either a minimum fold-enrichment or false discovery rate (FDR), but not both.

Addressing the question of sequencing depth requires (i) defining a statistical criterion that can quantify the information loss of an experiment due to its apparent sequencing depth and (ii) determining the sequencing depth needed to control the information loss based on a pilot, possibly under-sequenced, dataset. From a statistical point of view, ChIP-seq peak calling procedures can be cast as multiple testing problems because they aim to assess whether data for each candidate locus is supported by the background noise distribution or the ChIP signal. Therefore, the information loss is naturally connected to the concept of the testing power. As a result, both of the above issues can be considered within a power calculation framework where the sequencing depth plays the role of sample size.

Power computations require modelling distribution of both the background reads and ChIP signal in a way that reflects the stochastic nature of read accumulation at each genomic locus as a function of sequencing depth. Although a number of models were proposed for locus-specific read counts, none of them explicitly accounted for read accumulation. [83] and [25] considered models with locally Poisson distributed background and did not model ChIP signal. [33] proposed a

flexible model taking into account the genome structure and over-dispersion. However, this model utilized the control sample as a covariate and did not explicitly parametrize the model in terms of sequencing depths. [82] proposed a hierarchical Bayesian t-mixture model to identify local concentration of directional reads, but did not consider the relationship between read accumulation and sequencing depth. [77] adopted a signal-to-noise model, parameters of which followed some arbitrary prior to account for intrinsic read bias. Although such a prior distribution, if estimated, could be utilized to model the background distribution at varying sequencing depths, the work of [77] exclusively focused on the normalization aspect of ChIP-seq analysis.

I developed CSSP (**C**hIP-**s**eq **S**tatistical **P**ower) framework for statistical power calculation by considering a local Poisson model for the read generation process. I assume that background reads in the ChIP and the control samples are generated by local Poisson processes with shared Gamma prior distributions. The corresponding Gamma parameters are modelled as functions of the local genome structure, including mappability and GC content. The local Poisson parameters for the enrichment signals follow convolution of Gamma distributions. This model preserves the local structure of the [77] model while keeping the Negative Binomial distribution as the marginal signal distribution as in [33]. Such a local structure is key for capturing dynamics of the counting process for individual genomic locus as a result of increasing sequencing depths. I introduce a conditional power definition that employs the practically used notion of fold change of ChIP signal over the control sample. I show with data-driven computational experiments that my approach can be used to determine (i) the apparent conditional power for a given sequencing depth; (ii) the required sequencing depth to achieve a target power while controlling the false discovery rate at a specified level. Simulation experiments based on a deeply sequenced *E. coli* dataset indicate that power predictions of my model agree well with the observed empirical power. Utilizing data from pilot studies, I can reliably estimate power for larger sequencing depths; thus the CSSP framework has significant implications for designing ChIP-seq experiments with the multiplexing functionality. Finally, I study the power of multiple ENCODE datasets with varying sequencing depths. My results illustrate that, although the power varies considerably with the signal-to-noise ratios of the datasets, the current sequencing depths have high power for protein-DNA interactions with

large effect sizes and are generally adequate for smaller effect sizes. My calculations are further supported by the data quality metrics proposed by the ENCODE project ([19]).

2.2 The CSSP Framework

2.2.1 The Hierarchical Model

My CSSP framework models read counts from ChIP-seq data as Poisson processes with Gamma prior distributions. I assume that the uniquely mapping reads of both the ChIP and the control samples are preprocessed by the commonly adopted method of extension to the average fragment length provided by the experimental design ([60, 33]). For modeling purposes, I divide the reference genome into n non-overlapping intervals, e.g., bins as in [25], with sizes set to the average fragment length. Let X_i and Y_i denote the number of extended control and ChIP sample reads overlapping the i-th bin, respectively. Let N_x and N_y denote the sequencing depths for control and ChIP samples. I assume that X_i and Y_i follow Poisson distributions [83, 77]:

$$(X_i \mid \lambda_i^x) \sim Pois(\lambda_i^x N_x), \quad (Y_i \mid \lambda_i^y) \sim Pois(\lambda_i^y N_y), \tag{2.1}$$

where λ_i^x and λ_i^y are bin-specific rate parameters for the control and ChIP samples, satisfying $E[\sum_i \lambda_i^x] = 1$ and $E[\sum_i \lambda_i^y] = 1$, where the expectations are with respect to the prior distributions that I introduce below. This formulation models bin counts as Poisson processes with fixed intensities. Let Z_i be the vector containing local genomic information such as mappability and GC-content as in [33] and [55] for the i-th bin. I consider the following bin-specific prior distributions for local Poisson intensities of the control sample:

$$\begin{aligned} (\lambda_i^x | Z_i = z_i) &\sim \Gamma\left(a, \frac{a}{\mu(z_i)}\right), \\ \mu(z_i) &= \exp\{\gamma_0 + f_\gamma(z_i)\}, \end{aligned} \tag{2.2}$$

where γ_0 is a normalization constant such that $\sum_{i=1}^n \mu(z_i) = 1$ and $f_\gamma(.)$ is a function of local genomic information. I adopt the flexible smoothing spline framework as in [33] for capturing the effect of mappability and GC by $f_\gamma(.)$ on the control read counts.

For the ChIP sample, I define an unobserved variable B_i to indicate enrichment state of bin i, e.g., $B_i = 0$, for background bins. For enriched bins, I allow J different states to reflect levels of enrichment strengths (e.g., $J = 2$ broadly captures low and high affinity binding for TFs), and correspondingly $B_i = j$, $j = 1, \cdots, J$. The prior distributions for each state are:

$$\begin{aligned}(\lambda_i^y | Z_i = z_i, B_i = 0) &\sim \Gamma\left(b, \frac{b}{e_0 \mu_i}\right), \\ (\lambda_i^y | Z_i = z_i, B_i = j) &\sim \Gamma\left(b^j, \frac{b^j}{\nu^j}\right), \quad j = 1, \cdots, J,\end{aligned} \tag{2.3}$$

where $e_0 \in (0, 1)$ is a normalizing factor reflecting the proportion of background reads in the ChIP sample (77; 38). For brevity, I denote $\mu(z_i)$ by μ_i by suppressing its dependence on z_i, which is fixed for a genome at given read and fragment lengths and bin size. Under this model specification, the marginal distributions of X_i and Y_i are Negative Binomials given by:

$$\begin{aligned}X_i &\sim NB\left(a, \frac{a}{\mu_i N_x}\right), \\ (Y_i | B_i = 0) &\sim NB\left(b, \frac{b}{e_0 \mu_i N_y}\right), \\ (Y_i | B_i = j) &\sim NB\left(b^j, \frac{b^j}{\nu^j N_y}\right), \quad j = 1, \cdots, J.\end{aligned} \tag{2.4}$$

In contrast to the [33] model, marginal distributions in my model arise from two levels of hierarchy: a local Poisson distribution and a prior distribution. The local Poisson structure is critical for modeling counts for each bin on a process level as sequencing depths increase. The prior distribution models the intrinsic read biases which are supported by arguments in [60] and [77]. In the resulting model, although the number of local Poisson parameters λ_i^ys is the same as the number of observations, inference is possible through Bayesian analysis where the posterior distribution of the ChIP counts for each bin given the bin count y_i and enrichment state $B_i = j$ is given by:

$$(\lambda_i^y | Z_i = z_i, B_i = 0, Y_i = y_i) \sim \Gamma\left(b + y_i, \tfrac{b}{e_0 \mu_i} + N_y\right), \tag{2.5}$$

$$(\lambda_i^y | Z_i = z_i, B_i = j, Y_i = y_i) \sim \Gamma\left(b^j + y_i, \tfrac{b^j}{\nu^j} + N_y\right), \tag{2.6}$$

$j = 1, 2, \cdots, J$. Determining the number of components J is a model selection problem within the CSSP framework. In practice, I recommend setting J to 2. [33] observed that when modeling the

ChIP signal as a mixture of Negative Binomial distributions, two components adequately captured both the low and high affinity binding. My `R` package enables using larger values of J and users can apply model selection criterion such as Bayesian Information Criterion ([62]) to control model complexity. I used $J = 2$ for the examples presented in this paper.

2.2.2 A Multiple Testing Procedure and Power Evaluation

There is a plethora of algorithms for assessing whether individual bins are enriched in the ChIP sample compared to control sample ([74]). My CSSP framework naturally lands itself into a multiple testing framework. For unenriched bin i, the ChIP count originates from Negative Binomial distribution, $Y_i \sim NB(b, b/(e_0\mu_i N_y))$, and ChIP counts for enriched bins have larger values. Therefore, I consider one-sided testing against the null $H_0 : Y_i \sim NB(b, b/(e_0\mu_i N_y))$. Under the prior distribution for the local Poisson rate, the decision rule based on the marginal distribution achieves optimal Bayes risk. The p-value for $Y_i = y_i$ is thus $pval(y_i) = P\{Y_i \geq y_i | Y_i \sim NB(b, b/(e_0\mu_i N_y))\}$. Suppose I control the overall Type-I error at q and reject the null hypothesis when $pval(y_i) < \alpha_q$. The corresponding rejection region for bin R_i is $(Q_i(\alpha_q), \infty)$ where $Q_i(\alpha_q)$ is the $(1 - \alpha_q)$-th percentile of the null distribution. In order to control the false discovery rate, I set α_q using the Benjamini-Hochberg procedure ([5]). The mean power of the above testing procedure across all enriched bins is given by:

$$Pow'_q(N_y) = \frac{\sum_{i:B_i \neq 0} P\{Y_i > Q_i(\alpha_q) | B_i \neq 0\}}{\#\{i : B_i \neq 0\}}.$$

This definition of power nominally considers all enriched regions regardless of their actual enrichment levels, i.e., effect sizes. However, in practice investigators expect true protein-DNA interaction regions to not only exhibit statistically higher read counts in the ChIP sample compared to the control sample but also achieve a pre-determined enrichment level. Regions failing to achieve this are usually filtered by peak callers (i.e., SPP, MACS) through post-processing. The statistical implication of such a practice is that in testing the observed ChIP count Y_i against the background distribution, I should impose restriction on the effect size in addition to a p-value threshold. As a

result, when evaluating the power across all enriched regions, my attention should be restricted to only the enriched regions with sufficiently large effect sizes.

I introduce quality thresholds including a fold change threshold r and a minimum intensity threshold τ to accommodate this practical issue. As a result, the enrichment detection procedure requires that read counts exceed not only the corresponding percentile $Q_i(\alpha_q)$ required for FDR control, but are also r fold of the prior mean $e_0\mu_i N_y$ and exceed minimum intensity of τN_y. Hence, the peak calling threshold for bin i is

$$T_i(\alpha_q, r, \tau) = Q_i(\alpha_q) \vee re_0\mu_i N_y \vee \tau N_y,$$

where $x \vee y = \max(x, y)$. As a result, I establish the following conditional power function:

$$\begin{aligned} &Pow_{q,r,\tau}(N_y) = \\ &\frac{\sum_{i\in A} P\{Y_i > T_i(\alpha_q, r, \tau) | \lambda_i^y > re_0\mu_i \vee \tau, B_i \neq 0\}}{\#A}, \end{aligned} \tag{2.7}$$

where $A = \{i : \lambda_i^y > re_0\mu_i \vee \tau, B_i \neq 0\}$ and $\#A$ denotes the size of set A. This definition depends on local Poisson parameters which are usually not estimable. I propose the following conditional posterior power function by plugging in Bayesian estimators of the numerator and denominator of Eqn. (2.14), respectively.

$$\begin{aligned} &Pow^B_{q,r,\tau}(N_y) = \\ &\frac{\sum_{i=1}^n w_i E\left[P\{Y_i > T_i(\alpha_q, r, \tau) | \lambda_i^y > re_0\mu_i \vee \tau, B_i \neq 0\} \mid Y_i = y_i\right]}{\sum_{i=1}^n w_i}, \end{aligned} \tag{2.8}$$

where $w_i = P\{\lambda_i^y > re_0\mu_i \vee \tau, B_i \neq 0 | Y_i = y_i\}$.

A numerical algorithm to compute (2.8) is as follows. This calculation first requires the determination of the significance level α_q which ensures that overall FDR across all the bins is controlled at level q. I achieve this by a computationally fast simulation-based approach.

Recall that the posterior distribution of the ChIP sample rate parameter λ_i^y given the bin count y_i and enrichment state $B_i = j$ is given by Eqns. (2.5) and (2.6). Let p_i^j denote the posterior probability of $B_i = j$ given the observed data as outputted by the generalized Expectation-Maximization (EM) algorithm described in Section 2.2.3. Then, I calculate the power for a given sequencing depth N_y and FDR level q as follows:

A. Estimate α_q to control the overall FDR at level q by simulation:

A.1 At simulation k, $k = 1, \cdots, K = 10$:

A.1.1 For bin i, $i = 1, \cdots, n$:

A.1.1.1 Simulate B_i^k as Bernoulli(p_i^j).

A.1.1.2 Simulate $\lambda_i^{y(k)}$ given $Y_i = y_i$ and B_i^k according to Eqns. (2.5) and (2.6).

A.1.1.3 Simulate $Y_i^{(k)} \sim Pois(\lambda_i^{y(k)} N_y)$.

A.1.1.4 Compute p-values $pval_i^{(k)} = P\{Y_i \geq Y_i^{(k)} \mid Y_i \sim NB(b, b/(e_0\mu_i N_y))\}$.

A.1.2 Estimate $\alpha_q^{(k)}$ with the Benjamini-Hochberg (BH) FDR adjustment procedure ([5]). Let $pval_{(1)}^{(k)}, \cdots, pval_{(n)}^{(k)}$ denote ordered p-values. Then, $\alpha_q^{(k)} = \max_i\{pval_{(i)}^{(k)} : pval_{(i)}^{(k)} \leq \frac{i}{n}q\}$.

A.2 Set $\alpha_q = \sum \alpha_q^{(k)}/K$.

B. Compute $w_i = \sum_j p_i^j Pr\{\lambda_i^y > re_0\mu_i \vee \tau | B_i = j, Y_i = y_i\}$, $i = 1, \cdots, n$, where $Pr\{\lambda_i^y > re_0\mu_i \vee \tau | B_i = j, Y_i = y_i\}$ is evaluated according to Eqn. (2.6).

C. Compute $E[P\{Y_i > Q_i(\alpha_q) | \lambda_i^y > re_0\mu_i \vee \tau, B_i \neq 0\} | Y_i = y_i]$, $i = 1, \cdots, n$ by simulation:

C.1 For bin i, $i = 1, \cdots, n$:

C.1.1 At simulation k, $k = 1, 2, \cdots, K = 10$:

C.1.1.1 Generate multinomial random variable B_i^k where $P\{B_i^k = j\} = p_i^j/(1 - p_i^0)$, $j = 1, \cdots, J$. Thus, $B_i^k \sim (B_i | Y_i = y_i, B_i \neq 0)$.

C.1.1.2 Generate Unif$(0, 1)$ random variable Z_i^k. Set $\lambda_i^{y(k)}$ as the $1 - Z_i^k P\{\lambda_i^y > re_0\mu_i \vee \tau | B_i = B_i^k, Y_i = y_i\}$-th percentile of distribution in Eqn. (2.5). This guarantees that $P\{\lambda_i^{y(k)} > re_0\mu_i \vee \tau | B_i = B_i^k, Y_i = y_i\} = 1$.

C.1.1.3 Generate $Y_i^k \sim Pois(\lambda_i^{y(k)} N_y)$.

C.1.2 Estimate $E[P\{Y_i > Q_i(\alpha_q) | \lambda_i^y > re_0\mu_i \vee \tau, B_i \neq 0\} | Y_i = y_i]$ by $\sum_{k=1}^{K} I\{Y_i^k > Q_i(\alpha_q)\}/K$, where $Q_i(\alpha_q)$ is the $(1-\alpha_q)$-th percentile of the background read distribution for bin i and $I(.)$ denotes the indicator function.

2.2.3 Estimating the Model Parameters

I decouple estimation of the parameters for the background read distribution and the signal distribution. As a result, the estimation procedure consists of three consecutive steps:

1. *Generalized linear model (GLM) fit.* Estimate μ_i (equivalently $f_\gamma(.)$) and the shape parameter a by fitting the control model given in Eqn. (4) of the main manuscript on control data with the `glm.nb` `R` function.

2. *Minimum distance estimation (MDE).* Estimate e_0, π_0, and b by minimizing the following function:

$$\min_{e_0,b,\pi_0} \sup_{x>c} |F_n(x) - 1 + \pi_0 - x\pi_0|, \tag{2.9}$$

   where $F_n(x)$ is the distribution of the list of continuity corrected p-values given by $1-(1-Z_i)F_{NB(b,b/(e_0\mu_i N_y))}(y_i-1) - Z_i F_{NB(b,b/(e_0\mu_i N_y))}(y_i)$ (Lemma 2.1 below), Z_is are independent Uniform random variables on the unit interval and are independent of Y, and c is a tuning parameter such that majority of the bins with $B_i \neq 0$ have p-values smaller than c.

3. *Generalized EM algorithm.* Estimate ν^j and b^j, $j = 1, \cdots, J$, using a generalized EM algorithm.

First, I estimate hyper parameters in $f_\gamma(.)$ by fitting a Negative Binomial regression model to the control sample. This is easily carried out with the `glm.nb()` function in `R`. Second, I estimate the normalization factor e_0, the proportion of background bins π_0, and the dispersion parameter b from the ChIP sample. These parameters normalize the ChIP sample against the control sample and are critical for the downstream power evaluation. I observed that the conventional estimating

methods, e.g., maximum likelihood and method of moments, lead to poor estimators of π_0 and b (Section 2.3.2). Therefore, I propose the following minimum distance estimator which is motivated by the minimum distance and robust estimation framework in [53] as an alternative.

2.2.3.1 Minimum distance estimation of e_0, π_0, and b

The validity of many multiple hypotheses testing procedures, including the BH procedure ([5]) for FDR control relies on the assumption that p-values are uniformly distributed under the null hypotheses. This assumption is violated when the distribution of p-values is discrete. Therefore, I first propose a continuity correction for the p-values derived from the Negative Binomial null distribution in my multiple testing framework.

Theorem 2.1. *Let Y be a integer-valued random variable and let F denote its distribution function. For $Z \sim$ Unif$(0,1)$, $X = (1-Z)F(Y-1) + ZF(Y)$ is Unif$(0,1)$.*

Proof. Let $y_x = \inf\{t \in \mathbb{Z} : F(t) \geq x\}$.

$$\begin{aligned} P\{X \leq x\} &= P\{Y \leq y_x - 1\} + P\left\{Y = y_x, Z \leq \frac{x - F(y_x - 1)}{F(y_x) - F(y_x - 1)}\right\} \\ &= F(y_x - 1) + P\{Y = y_x\} P\left\{Z \leq \frac{x - F(y_x - 1)}{F(y_x) - F(y_x - 1)}\right\} \\ &= F(y_x - 1) + [F(y_x) - F(y_x - 1)] \frac{x - F(y_x - 1)}{F(y_x) - F(y_x - 1)} \\ &= x. \end{aligned}$$

□

Hence, the continuity corrected p-values can be computed by

$$pval_i = 1 - (1 - Z_i) F_{NB(b, b/(e_0 \mu_i N_y))}(y_i - 1) - Z_i F_{NB(b, b/(e_0 \mu_i N_y))}(y_i). \tag{2.10}$$

As a result, $pval_i$, $i = 1, \cdots, n$ are uniformly distributed when Y_is are from the background read distribution. For enriched regions, $pval_i$ is expected to be very small. Therefore, p-values which are larger than a chosen threshold c are uniformly distributed on the unit interval, i.e., for $pval_i > c$,

$P(pval_i \leq x) = 1 - \pi_0 + \pi_0 x$. This motivates the objective function in Eqn. (2.9) and I develop the following algorithm to minimize this function.

A Set iteration number $t = 1$ and initialize the parameters as $e_0^{(1)} = e_0.init = 0.9$ and $b^{(1)} = a$. Initialize the step size parameters for updating e_0 and b estimates as $e_{step}^{(1)} = 0.01$ and $b_{step}^{(1)} = 1$. Initialize the set of parameters for monitoring convergence as: $flag.e^{(1)} = 1$, $flag.b^{(1)} = 1$, and $dis = 1$, $tol = 0.001$. Set $c = 0.5$.

B While $dis > tol$:

B.1 Using $e_0^{(t)}$ and $b^{(t)}$, compute Eqn. (2.10) to obtain p-values. Order the p-values as $pval_{(1)} \leq \cdots \leq pval_{(m)} \leq c < pval_{(m+1)} \leq \cdots \leq pval_{(n)}$.

B.2 Update $\pi_0^{(t)}$ as $\pi_0^{(t)} = \frac{1-m/n}{1-pval_{(m)}}$ by using the fact that $1 - \pi_0 + \pi_0 pval_{(m)} = m/n$.

B.3 Evaluate Eqn. (2.9) for $i = m, m+1, \cdots, n$ by computing $r_i = i/n - 1 + \pi_0^{(t)} - \pi_0^{(t)} pval_{(i)}$.

B.4 Update e_0:

If $\sum_{i=m}^{n} r_i > 0$, set $e_0^{(t+1)} = e_0^{(t)} - e_{step}^{(t)}$, $flag.e^{(t+1)} = -1$.

Otherwise, set $e_0^{(t+1)} = e_0^{(t)} + e_{step}^{(t)}$, $flag.e^{(t+1)} = 1$.

B.5 Update b: Set $k = \lfloor (m+n)/2 \rfloor$, where $\lfloor x \rfloor$ denotes the largest integer smaller than x.

If $\{\#\text{ of } i : r_i < 0, m \leq i \leq k\} < \{\#\text{ of } i : r_i < 0, k < i \leq n\}$, $b^{(t+1)} = b^{(t)} - b_{step}^{(t)}$, $flag.b^{(t+1)} = -1$.

Otherwise, $b^{(t+1)} = b^{(t)} + b_{step}^{(t)}$, $flag.b^{(t+1)} = 1$.

B.6 Update step sizes:

If $flag.e^{(t+1)} flag.e^{(t)} = -1$, $e_{step}^{(t+1)} = e_{step}^{(t)}/2$.

If $flag.b^{(t+1)} flag.b^{(t)} = -1$, $b_{step}^{(t+1)} = b_{step}^{(t)}/2$.

B.7 Set $dis = \max_{m \leq i \leq n} |r_i|$ and increase t by 1.

In the iteration steps, I update e_0 and b consecutively with a coordinate descent strategy, which is more efficient than a grid search method. I avoid differentiating the objective function when determining the direction of parameter updates. I determine the derivative by analyzing the "residuals", r_i's, which are obtained by evaluating the the difference between the empirical cumulative distribution function ($F_n(x)$) and the expected cumulative distribution function ($1-\pi_0+\pi_0 x$). For choosing a direction for e_0, I utilize the fact that $\sum r_i > 0$ indicates that p-values calculated against the estimated background are smaller than expected. This implies that the estimated background mean is too large, therefore e_0 is reduced subsequently. For choosing a direction for b, I utilize the fact that when $\{\#\text{ of } i : r_i < 0, m \le i \le k\} > \{\#\text{ of } i : r_i < 0, k < i \le n\}$ holds, it indicates that the empirical distribution of p-values from background is "S" shaped, with less points concentrated at tails than expected. This implies that the estimated background variance $\mu(1+\mu/b)$ is too large, thus b is increased subsequently.

2.2.3.2 Estimating the parameters of the signal component from the ChIP sample

I use a generalized EM algorithm as in [33] for this step. Since I have $(Y_i \mid Z_i = z_i, B_i = j) \sim NB\left(b^j, \frac{b^j}{\nu^j N_y}\right)$, $j = 1, \cdots, J$, b^j and ν^j are the additional parameters that need to be estimated. Let $P\{B_i = j\} = \pi_j$, $j = 0, 1, \cdots, J$. In the M-step, I update the estimates $\hat{b}^j$ and $\hat{\nu}^j$ by the method of moments. Let $p_i^{j(t+1)}$ denote the estimate of $P\{B_i = j \mid Z_i = z_i, Y_i = y_i\}$ based on parameters from the t-th iteration. Then the E- and M-steps are as follows.

E-step:

$$p_i^{j(t+1)} = \frac{\hat{\pi}_j^{(t)} dNB(y_i|B_i = j)}{\sum_{k=0}^{J} \hat{\pi}_k^{(t)} dNB(y_i|B_i = k)}, \quad j = 0, 1, \cdots J,$$

where $dNB(y_i|B_i = j)$ denotes the posterior marginal density when $B_i = j$.

M-step:

$$\hat{\pi}_j^{(t+1)} = \sum_i p_i^{j(t+1)}/n.$$

Let $\mu_j^{(t)} = \frac{\sum_{i=1}^n p_i^{j(t+1)} y_i}{\sum_{i=1}^n p_i^{j(t+1)}}$ and $\sigma_j^{(t)} = \frac{\sum_{i=1}^n p_i^{j(t+1)} (y_i - \mu_j^{(t)})^2}{\sum_{i=1}^n p_i^{j(t+1)}}$ denote the mean and variance estimates of the signal component j at iteration t. Then, $\nu^{j(t+1)}$ and $b^{j(t+1)}$ are estimated by:

$$\begin{aligned} \hat{\nu}^{j(t+1)} &= \mu_j^{(t)}/N_y, \\ \hat{b}^{j(t+1)} &= \frac{(\hat{\nu}^{j(t+1)} N_y)^2}{\hat{\sigma}_j^{(t)} - \hat{\nu}^{j(t+1)} N_y}. \end{aligned}$$

I evaluated this three step estimation scheme with extensive simulations and established its estimation consistency in Section 2.3.1. Furthermore, I illustrated that it performs better than a more conventional generalized EM algorithm which estimates all the model parameters simultaneously in Section 2.3.2.

2.3 Numerical Studies

I first evaluated my CSSP framework in a simulation study to assess the consistency of my parameter estimates, power, and FDR control (Section 2.3.1). Then I performed sub-sampling experiments based on two deeply sequenced datasets (*E. coli* FNR ChIP-seq dataset of [47] and mouse GATA1 ChIP-seq dataset of [76]) to demonstrate the consistency and power of my CSSP framework. I utilized multiple human CTCF ChIP-seq datasets from the ENCODE consortium to evaluate the impact of lab and lab-specific batch effects on power estimation. Finally, I investigated eight ENCODE datasets to assess the power of currently available typical ChIP-seq studies.

2.3.1 Simulation Study Based on the *E. coli* FNR Dataset

2.3.1.1 Simulation Set-up

I utilized the parameter estimates from the FNR fit (Section 2.3.4) to generate simulated data for evaluating various aspects of my power calculation framework. I first simulated local Poisson intensities for the control data based on:

$$\begin{aligned} (\lambda_i^x | Z_i = z_i) &\sim \Gamma\left(a, \frac{a}{\mu(z_i)}\right), \\ \mu(z_i) &= \exp\{\gamma_0 + f_\gamma(z_i)\}. \end{aligned} \tag{2.11}$$

Then, I generated a set of enriched bins by sampling from the estimated bin-level posterior probabilities of the FNR data and simulated local Poisson intensities for ChIP data based on:

$$\begin{aligned}(\lambda_i^y|Z_i = z_i, B_i = 0) &\sim \Gamma\left(b, \frac{b}{e_0\mu_i}\right),\\ (\lambda_i^y|Z_i = z_i, B_i = j) &\sim \Gamma\left(b^j, \frac{b^j}{\nu^j}\right), \quad j = 1, 2.\end{aligned} \tag{2.12}$$

I fixed the values of the local parameters λ_i^x and λ_i^y and simulated bin-level counts at multiple sequencing depths N_x and N_y based on the local Poisson models:

$$(X_i \mid \lambda_i^x) \sim Pois(\lambda_i^x N_x), \quad (Y_i \mid \lambda_i^y) \sim Pois(\lambda_i^y N_y). \tag{2.13}$$

I varied N_x and N_y as $f \in (0, 1)$ times the sequencing depths of the full control and ChIP samples. In what follows, all the calculations are based on a false discovery rate of 0.05 unless specified otherwise.

2.3.1.2 Consistency of Parameter Estimates

I first evaluated the performance of my estimation procedure presented in Section 2.2.3 in detail. I set $f = 0.02, 0.04, 0.06, 0.08$ and generated 100 independent datasets at each of these sequencing depths to assess how well the parameters were estimated for low sequencing depths. Table 2.1 displays mean squared errors [2] of the key parameters, namely e_0, π_0, b, μ, ν^j, b^j, $j = 1, 2$ relative to their true values and indicates that my minimum distance estimation results in remarkably accurate estimates of all the parameters. In this table, each row corresponds to sequencing depth of $f\times$(full data sequencing depth) for the ChIP and the control samples. To further evaluate the parameter estimation and the multiple testing procedure my peak calling relies on, I plot the empirical cumulative distribution of the p-values obtained under the fitted background model in Figure 2.1. As evidenced in this figure, empirical distribution of the p-values obtained with an accurate background model exhibits an expected mixture of a Uniform distribution on the unit interval and a spike around 0. I observed that, for more realistic settings presented in the data-driven

[2] Relative mean squared error for parameter s is computed as $(1/100)\sum_{t=1}^{100}(\hat{s}_t - s)^2/s^2$, where s denotes the true value of the parameter and $\hat{s}_t$ its estimate on the t-th simulated dataset.

computational experiments of the paper, a conventional maximum likelihood procedure based on a generalized EM algorithm failed to capture this expected pattern of the distribution of the p-values. I discuss these observations further in Section 2.3.2.

Table 2.1 Relative mean squared error ($\times 10^{-4}$) of the estimated parameters across 100 simulated FNR datasets at varying sequencing depths.

f	e_0	π_0	b	μ	ν^1	ν^2	b^1	b^2
0.02	98.5	4.9	12.6	0	5.1	3.5	73.8	4.6
0.04	17.6	3.1	12.7	0	0.9	0.7	8.0	0.5
0.06	6.2	1.6	12.9	0	1.1	1.2	9.8	2.3
0.08	5.1	0.9	10.7	0	1.0	0.9	5.5	1.0

2.3.1.3 Power Estimation from Pilot Data

I next evaluated the accuracy of the power curve estimated based on pilot data using the FNR simulation set-up. This is critical from a practical point of view because ChIP-seq studies are typically under-powered and it is of interest to assess what depth is needed to achieve various levels of power. I first generated an oracle power curve by using the actual simulation parameters. Then, I simulated 100 low depth samples ($f = 0.06$) and estimated the power at varying depths ($f = 0.02, 0.04, \cdots, 0.2, 1$) based on the fits from these low depth samples. Figure 2.2(a) depicts boxplots of the resulting power estimates. In this figure, x-axis represents the percentage of full sequencing depth, and box-plots represent power estimates based on under-sequenced data (6% of the full dataset) over 100 data-driven simulations. Although the power estimates for lower depths exhibit some bias, overall they track the oracle power curve well.

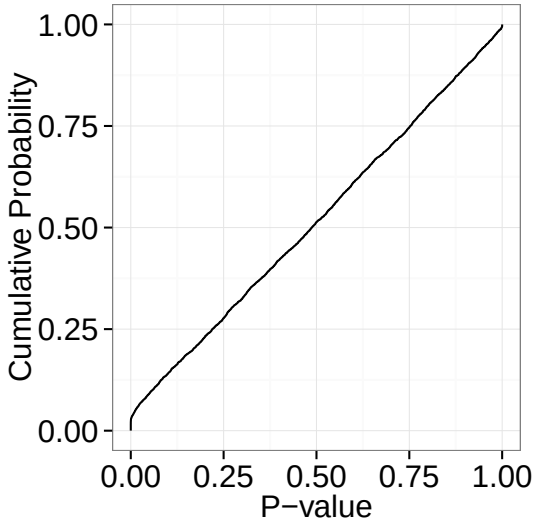


Figure 2.1 Cumulative probability plot of the p-values from a simulated FNR dataset.

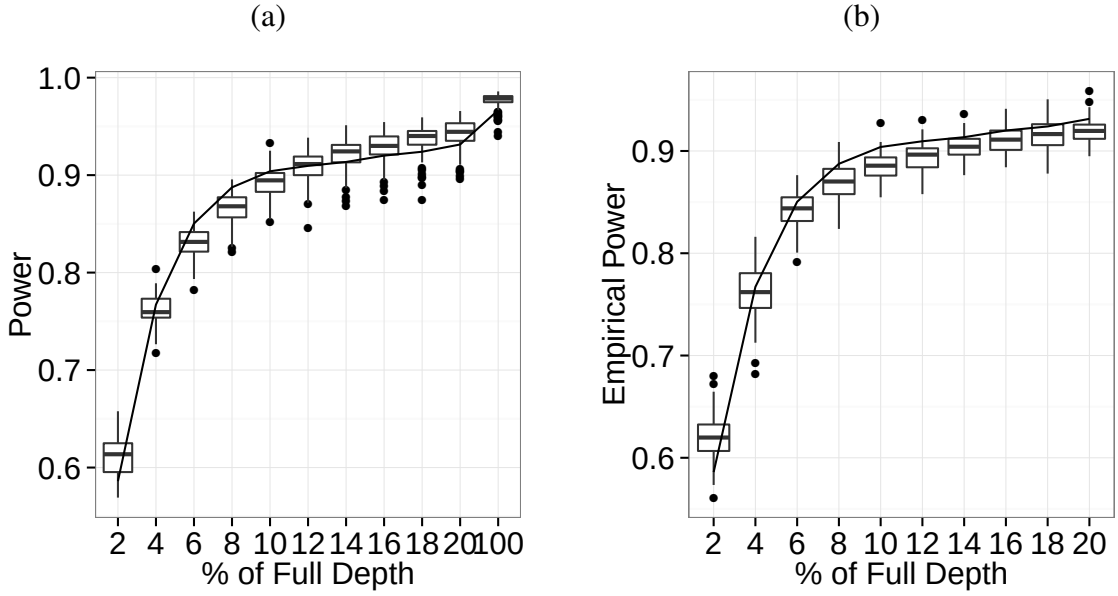


Figure 2.2 (a) Accuracy of pilot data-based power estimation. (b) Comparison of the estimated power with the empirical power.

2.3.1.4 Comparing CSSP Power Estimates with Empirical Power

I evaluated how the power predictions at various sequencing depths compared with the empirical power achievable at these depths. I define the empirical power as the proportion of gold standard peaks that are identified from under-sequenced data. The gold standard peak set is obtained by simulating data at full depth using the actual simulation parameters. To impose conditional power, I refined the gold standard peak set as those peaks with enrichment intensities $\lambda_i^y > re_0\mu_i$. Figure 2.2(b) compares boxplots of empirical power from 100 simulated datasets at each sequencing depth to an oracle power curve obtained by using the actual parameter values. In this figure, box-plots of empirical power (proportion of gold standard peaks identified with under-sequenced dataset) observed at under-sequenced datasets ranging from 2% to 20% of the full sequencing depth from 100 simulations. The solid lines in both panels represent the oracle CSSP power curves based on parameters estimated at full depth. I observe that the proportion of under-sequenced sample peaks overlapping the gold standard peaks can be predicted well by my model at every sequencing depth. This shows that the Bayesian estimate, Eqn. (2.8), accurately estimates the true power defined by:

$$Pow_{q,r,\tau}(N_y) = \frac{\sum_{i\in A} P\{Y_i > T_i(\alpha_q, r, \tau) | \lambda_i^y > re_0\mu_i \vee \tau, B_i \neq 0\}}{\#A},$$

where $A = \{i : \lambda_i^y > re_0\mu_i \vee \tau, B_i \neq 0\}$ and $\#A$ denotes the size of set A.

I also evaluated the FDR control within my power calculations by comparing the empirical FDR to the target level of 0.05. The empirical FDR is computed as the proportion of identified enriched regions that are not part of the gold standard peak set. Figure 2.3 displays the boxplots of the empirical FDR across 100 simulated datasets and shows that FDR is well controlled at all but the lowest depth with $f = 0.02$.

2.3.2 Comparing the Minimum Distance Estimator with the Conventional Generalized EM

My model fitting strategy decouples estimation of e_0, b, and π_0 from the estimation of signal component parameters . A more conventional way of estimating these parameters is to adopt a

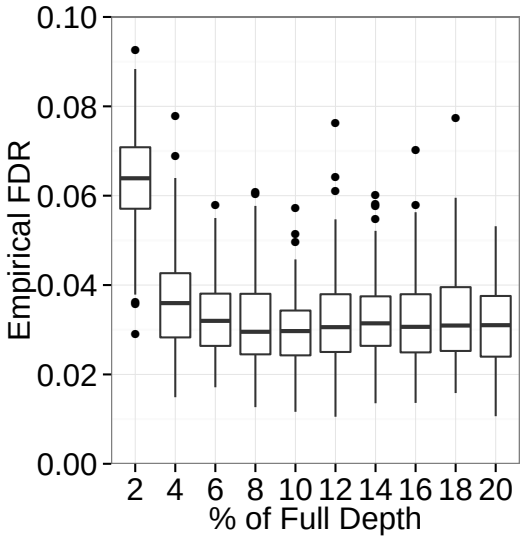


Figure 2.3 Boxplots of empirical FDR for under-sequenced datasets simulated at 2% to 20% of the full dataset.

generalized EM algorithm (GEM) and update the background and signal component parameters simultaneously in the M-step. In the `R` package that supplements this paper, I implemented this as an alternative algorithm. Overall, I observed that although the generalized EM algorithm is more efficient in simulations, decoupling of the estimation for the two sets of parameters behave more stably with real data. I next provide more details on this observation.

I first compared the two estimation schemes using the FNR simulation set-up as described in Section 2.3.1 and compared the relative mean squared errors for the key parameters. This reflects an ideal situation where the model is correctly specified. The generalized EM estimates are known to be consistent when there is no model misspecification and, therefore, can be used as gold standard. I simulated 50 datasets according to the FNR simulation set-up at 6% of the sequencing depth ($f = 0.06$). The relative bias and mean relative squared error for the key parameters are listed in Table 2.2. In this table, for parameter s, the relative values of bias and mean squared error are computed as $1/50 \sum_{t=1}^{50}(\hat{s}_t - s)/s$ and $1/50 \sum_{t=1}^{50}[(\hat{s}_t - s)/s]^2$, where s denotes the true value of the parameter and $\hat{s}_t$ its estimate based on the t-th simulated dataset. Although the minimum distance estimators are less efficient for some parameters such as b and e_0 compared to generalized EM estimators, they have smaller relative biases and mean relative squared errors for parameters including ν^j and π_0.

Table 2.2 Relative bias ($\times 10^{-2}$) and mean relative squared error ($\times 10^{-4}$) for estimated model parameters using data simulated from the FNR fit for the conventional Generalized EM (GEM) and the Minimum Distance Estimation (MDE).

Method	b	e_0	π_0	ν^1	ν^2
GEM	4.8, 28.3	-0.026, 0.01	-2.8, 7.87	-7.8, 61.7	4.9, 25.2
MDE	8.6, 99.9	3.1, 9.73	1.09, 1.69	0.34, 9.12	1.5, 3.69

Next I compared the two estimation approaches on real data using sub-sampling experiments. I generated 20 sub-sampled FNR datasets at 2% to 20% of the original sequencing depths and estimated the model parameters with each of the approaches. The estimates for π_0 are plotted as a function of the sequencing depth fraction f in Figures 2.4(a) and (d) for the minimum distance and

generalized EM approaches, respectively. The horizontal solid lines represent the π_0 estimates of each approach from the full dataset. As the sequencing depth increases, generalized EM estimators of π_0 are not consistent with each other, i.e., estimates of π_0 vary between ~ 0.4 and ~ 0.8 as the sequencing depth varies. In contrast, the minimum distance estimation results in consistent estimators of π_0 across different depths. A similar conclusion applies to the estimates of b (Figures 2.4(b) and (e)). I also evaluated the empirical cumulative distribution of the p-values obtained by the resulting estimates of the two approaches (Figures 2.4(c) and (f)). The generalized EM estimates at the full data ($\hat{\pi}_0 = 0.38$) fail to capture the expected mixture pattern of a Uniform distribution on the unit interval and a point mass at 0 for the empirical distribution of the p-values. Overall, the conventional generalized EM estimates are not robust against model misspecification while my three step estimation scheme using minimum distance estimation is able to estimate the model parameters with high accuracy.

2.3.3 Power Prediction with Misspecified Background Distribution

One of my fundamental assumptions for the CSSP framework is that the background distributions for the ChIP and control samples are the same. This assumption may not hold in practice when I have ChIP and control samples generated from different experimental protocols. For example, when the ChIP and control samples have different read lengths or fragment lengths, the genomic features for the control sample (i.e., mappability and GC-content) are different from those for the ChIP sample. This corresponds to misspecifying the prior distribution for the mean parameter of the ChIP background model.

In order to investigate whether CSSP is robust against such misspecified ChIP background models, I conducted the following computational experiments. For both the FNR and GATA1 datasets, I extended each control read to a fragment length different from that of the ChIP sample and regenerated bin-level data for these control samples. I extended FNR control reads to 250 bp and GATA1 control reads to 200 bp, while the fragment lengths for the ChIP samples remained at their experimental value of 150 bp and 250 bp for FNR and GATA1, respectively. I then repeated the sub-sampling experiment of Section 2.3.5 using 6% and 20% as the sampling percentages

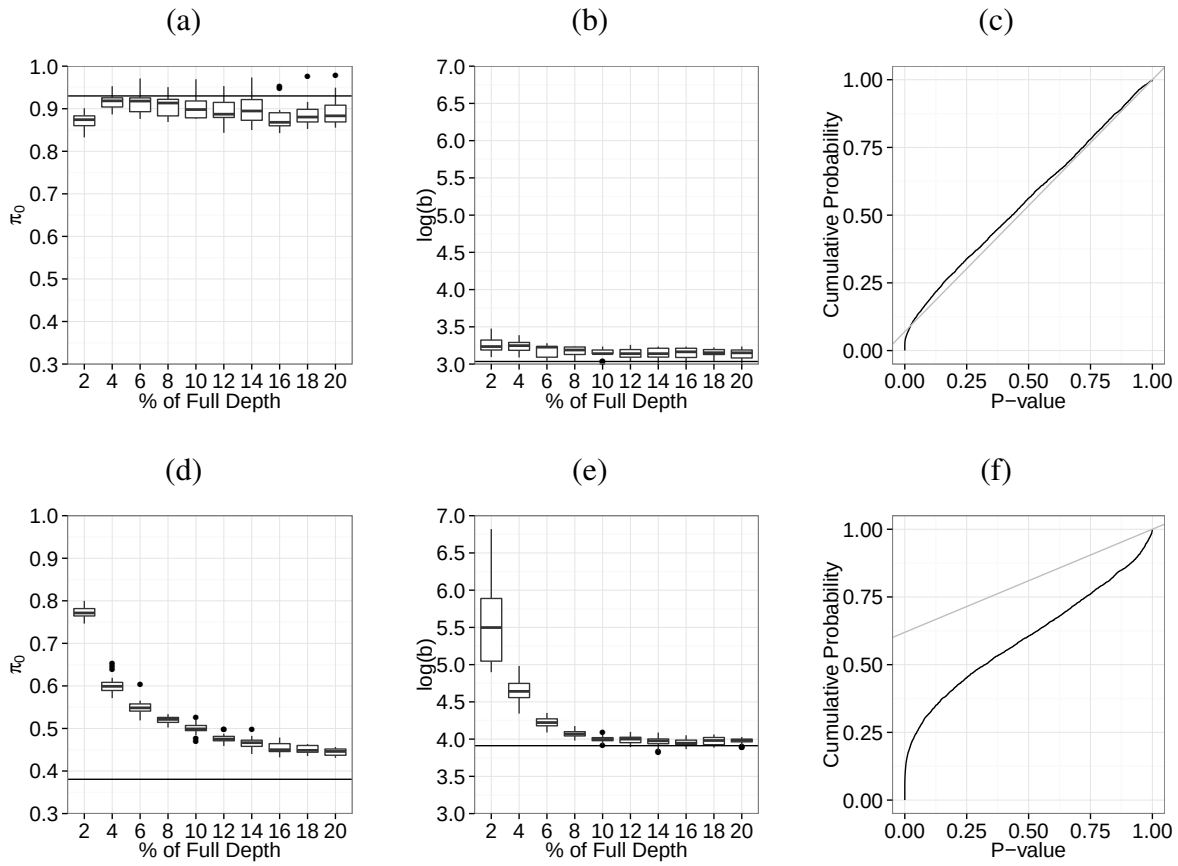


Figure 2.4 Comparison of the GEM and MDE methods at varying sequencing depths. (a, d) Boxplots of estimated values of π_0 as a function of sequencing depth based on 20 sub-sampled datasets at each depth using (a) minimum distance and (d) generalized EM methods. The horizontal line depicts the π_0 estimate based on the full data. (b, e) Boxplots of the logarithm of b estimates as a function of sequencing depth based on 20 sub-sampled datasets at each depth using b) minimum distance and e) generalized EM methods. The horizontal lines depict the b estimate on the log scale based on the full data. (c, f) Cumulative probability distribution plot of the p-values computed using (c) minimum distance and (f) generalized EM methods. The straight lines depict the Uniform distribution of the p-values due to non-enriched regions based on estimated π_0.

for the FNR and GATA1 datasets, respectively. Figures 2.5(a) and (b) compare estimated power based on these biased control samples with the oracle power curves and illustrates robustness of CSSP for this type of background model misspecification.In this figure, the x-axis displays the percentage of full sequencing depth, and the box-plots represent power estimates based on under-sequenced data (6% of the full FNR data, 20% of the full GATA1 data) over 20 sub-samples with control reads extended to different average fragment lengths from those of the ChIP samples (250 bp for FNR, 200 bp for GATA1). The solid lines indicate the oracle power curves estimated from the full datasets with control and ChIP reads both extended to the average fragment lengths (150 bp for FNR, 250 bp for GATA1) set up by the experimental protocol. The dashed lines indicate the power curves estimated using full datasets with control reads extended to misspecified average fragment lengths of 250 bp for FNR and 200 bp for GATA1. I observe that background distributions misspecified due to fragment length, overall, have negligible effects on the power estimated based on full data.

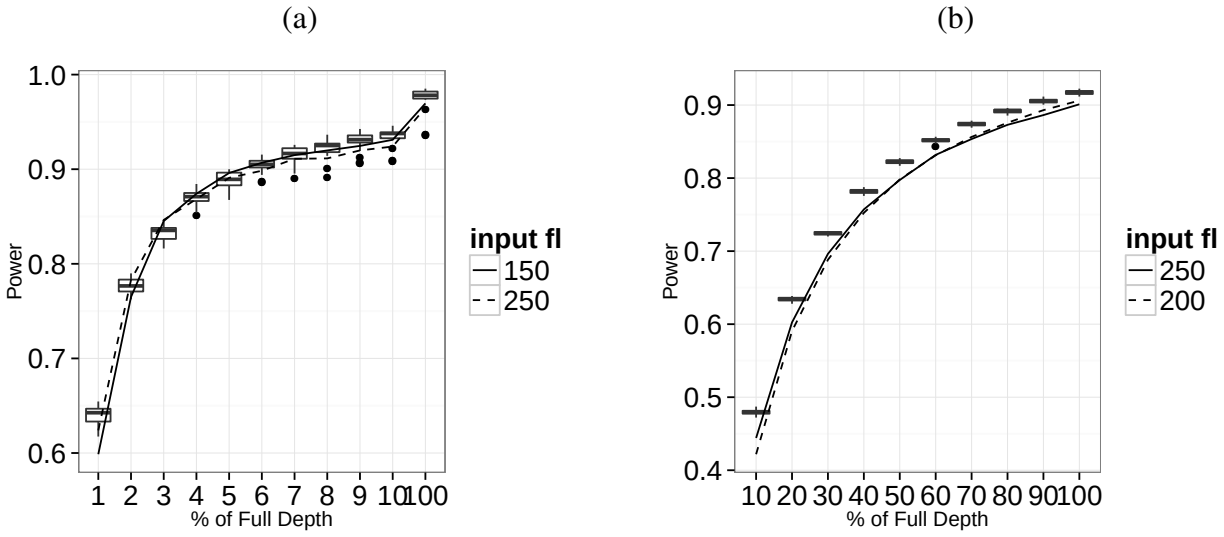


Figure 2.5 Robustness against misspecified prior mean of the ChIP background distribution for (a) FNR and (b) GATA1 datasets.

2.3.4 Model Fit for Deeply Sequenced Data

The *E. coli* FNR dataset consists of 9.07 M 32mer single-end ChIP and 6.45 M control reads. These sequencing depths approximately correspond to 4.9 and 3.5 billion reads for the mappable human genome ([60]). The GATA1 dataset from G1E-ER4+E2 cell line is also deeply sequenced compared to many available mammalian ChIP-seq datasets and has 106.4 M 55mer and 15.6 M 36mer reads for the ChIP and control samples, respectively. For both of the datasets, I created bin level data by extending aligned reads to the average fragment length provided by their experimental protocols (150 bp for FNR and 250 bp for GATA1) and counting the number of reads overlapping every bin. Fitted probabilities of bin level counts are compared to their empirical frequencies in Figure 2.6(a) and 2.6(b), for FNR and GATA1 ChIP samples, respectively. Both figures show very good agreement between the fitted and empirical frequencies. In addition, I plot the empirical cumulative distribution of the p-values obtained under the fitted background distributions in Figure 2.6(c) and 2.6(d). Both of the empirical distributions exhibit an expected mixture pattern where the majority of the p-values follow a Uniform distribution between 0 and 1. I note here that, for the GATA1 dataset, the ChIP and control samples have different read lengths. This indicates that my assumption of the same background prior distribution for the ChIP and control samples may not hold. However, the resulting model fits, as well as the computational experiments of the latter sections, suggest that the power estimation is robust against this type of misspecified background estimation. This is partly because GC score is not affected by the read length and the mappability remains the same for majority of the bins ($\approx 95\%$) between read lengths 36 bp and 55 bp. I have discussed misspecifications in background estimation in more details in Section 2.3.3.

2.3.5 Evaluating the Accuracy of the Power Curve Estimated Based on Pilot Data

Next, I evaluated the consistency of CSSP power estimation when only low sequenced pilot data is available. For both the FNR and GATA1 datasets, I generated a power curve at various sequencing depths using parameters estimated from the full data. I set the quality thresholds as $r = 2$ and $\tau = 0$. Because both datasets are deeply sequenced, their corresponding power curves

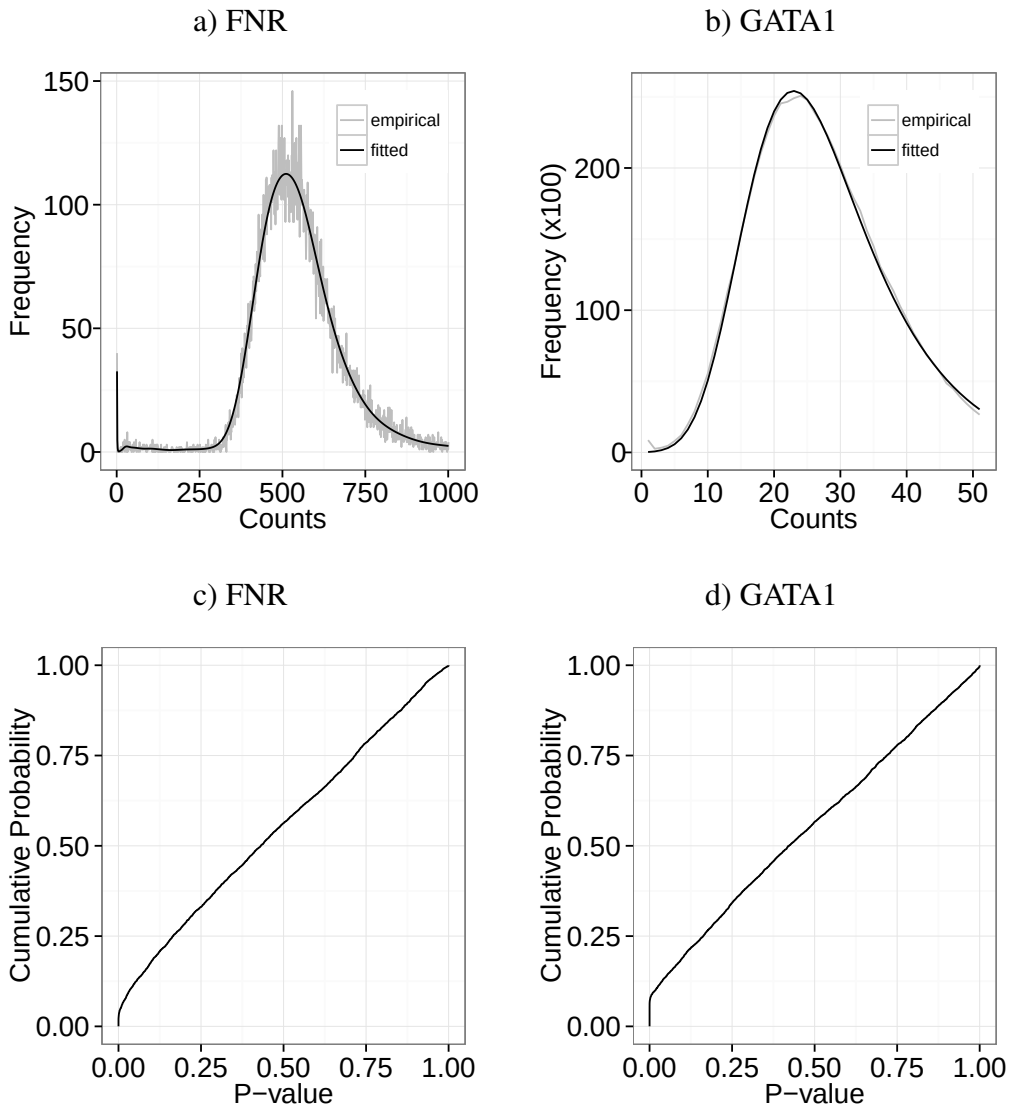


Figure 2.6 *Evaluating CSSP model fits.* (a, b) Goodness of fit plots for the FNR (a) and GATA1 (b) datasets. (c, d) Cumulative probability plots of the p-values for the FNR (c) and GATA1 (d) datasets.

can be viewed as oracle or gold-standard curves. To simulate low sequenced pilot datasets, I sampled 0.5%, 2%, 6% (FNR) and 20%, 40%, 60% (GATA1) of the available ChIP and control data and refitted the models. The lowest sampling percentages of 0.5% and 20% were chosen because sub-samples with depths lower than these resulted in non-convergent parameter estimators in the CSSP model. Both of these low depth sub-samples had an average bin-level ChIP count of 2.5. I generated 20 independent sub-samples at each sampling percentage for both datasets. The power estimates from sub-sampled data are compared to the oracle power estimates in Figure 2.7a and 2.7b. I observed that my power estimates based on under-sequenced GATA1 data agreed well with the oracle power curve. For the FNR dataset, when I sub-sampled below 6% of the full dataset, the predicted power was biased at low sequencing depths and agreed well with the oracle curve as the sequencing depth got larger. The mean biases of the power estimates were 0.009 and 0.015 for the 6% FNR and 20% GATA1 sub-sample datasets, respectively. The overall implications of these experiments are significant since they indicate that my power framework is capable of reliably estimating the required depth for a target power when only under-sequenced data is available.

2.3.6 Predicted Power versus Empirical Power

The above comparisons thus far relied on theoretical calculations of the power based on my model fit. I next compared my theoretical power predictions with the empirical power observed in under-sequenced datasets. This corresponds to assessing whether my Bayesian estimator of power in Eqn. (2.8) is a consistent estimator for power defined in Eqn. (2.14). Although the true set of enriched regions required by Eqn. (2.14) is unknown, I obtained gold-standard peak sets for both datasets using the full dataset with FDR of 0.05, $r = 2$, $\tau = 0$ and considered these gold-standard peak sets as proxies for the true set of enriched regions. Then, I generated 20 independent sub-samples at varying proportions of the full sequencing depth (2% to 20% for FNR and 20% to 90% for GATA1). For each sub-sample, I generated the list of enriched regions that were significant at FDR adjusted significance levels and had at least 2 fold enrichment against the estimated background mean ($r = 2$). I then calculated the proportions of the gold-standard peak sets that overlapped with the sub-sample-based peak sets. I report the empirical power by

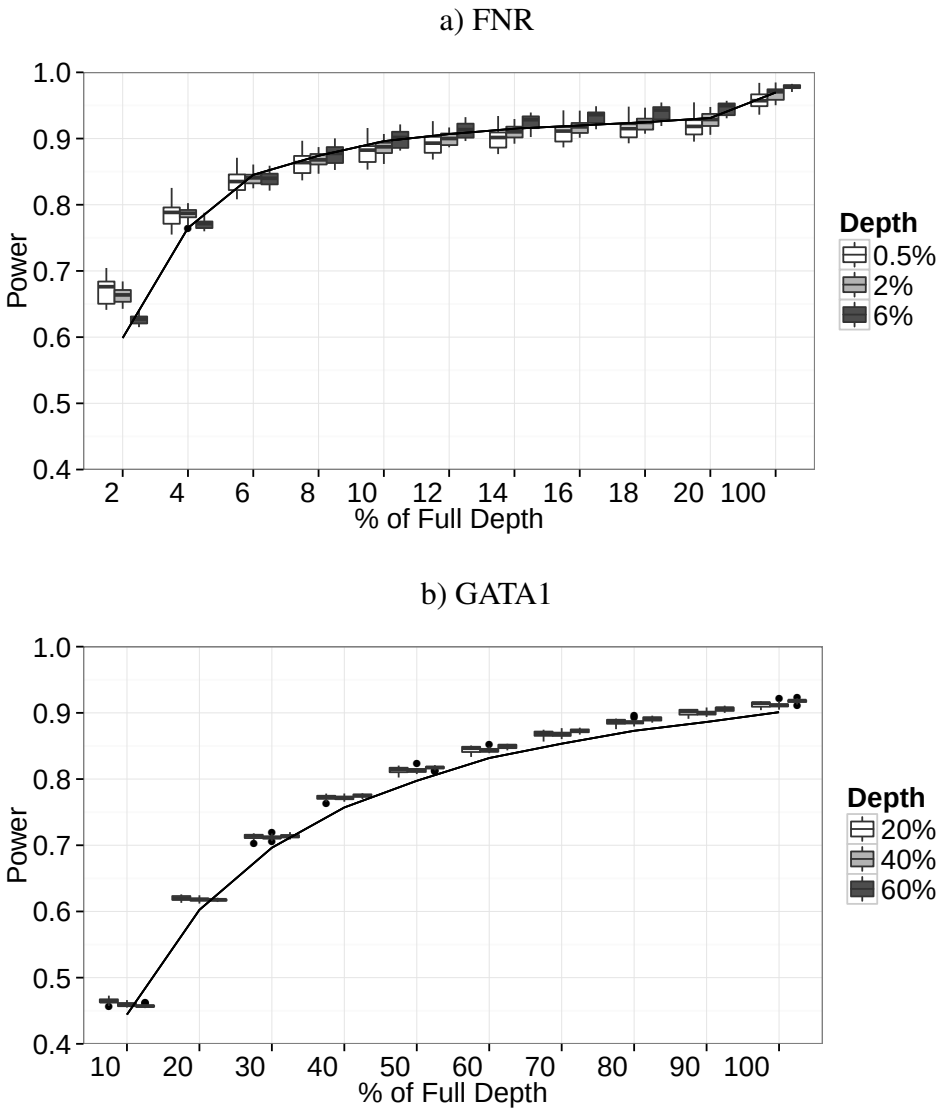


Figure 2.7 *Accuracy of power estimation based on pilot data.* x-axis represents the percentage of full sequencing depth. a) Boxplots represent power estimates based on sub-sampled (a) FNR (0.5%, 2%, 6% of the full dataset) and (b) GATA1 (20%, 40%, 60% of the full dataset) datasets over 20 sub-samples. The solid lines indicate the oracle power curve based on the parameters estimated from the full dataset.

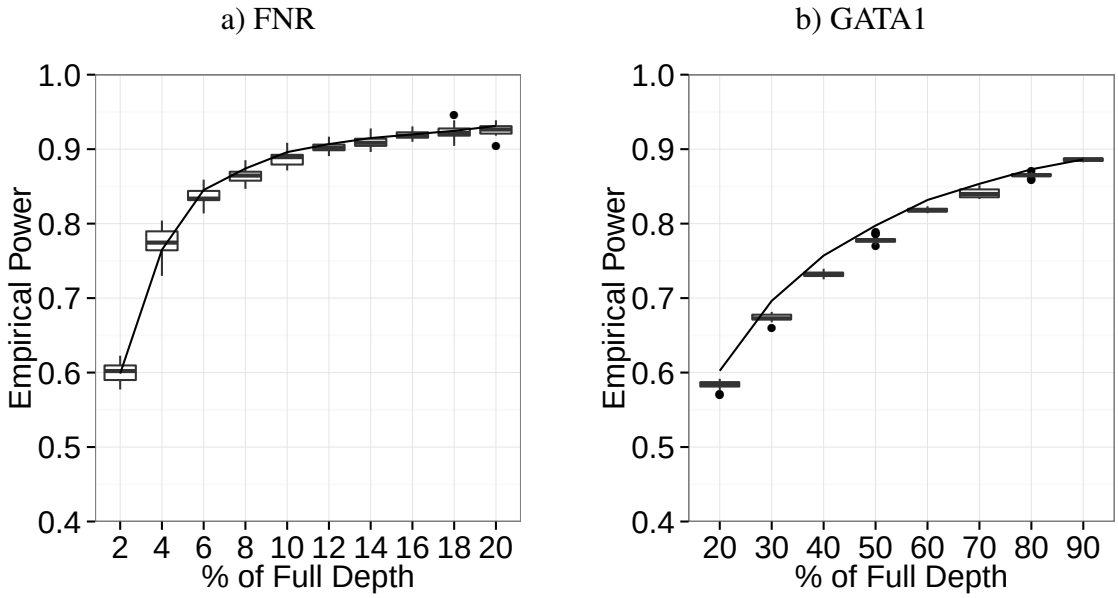


Figure 2.8 *Predicted power versus empirical power.* Solid lines represent the model-based prediction of the power curve. Boxplots represent empirical power observed at sub-sampled (a) FNR and (b) GATA1 datasets. A total of 20 datasets are sub-sampled at every sequencing depth.

multiplying these proportions with the power at the full sequencing depths since the full depth power was used as a proxy for Eqn. (2.14).

Figure 2.8a and 2.8b display the boxplots of empirical power as a function of sequencing depth and compare the empirical power with the oracle power curve. In both cases the empirical power follows the CSSP power estimates very closely.

2.3.7 Impact of the Control Sample on Power Calculations

The sequencing depth of the control library is an important factor that influences the power of ChIP-seq experiments. My computational experiments thus far varied ChIP and the control samples simultaneously. [20] observed identification of more peaks when a ChIP-seq dataset was normalized against a more deeply sequenced control dataset. Furthermore, this study also observed that deeply sequenced control datasets correlated well with the GC content. [8] concluded that deeper sequencing of the control sample led to better detection specificity. These studies also established that the dependence on the sequencing depth of the control sample varied substantially between different algorithms. For example, MACS ([83]) achieved best performance when the ChIP and the control samples were balanced in terms of sequencing depths, whereas USeq ([50]) performed better when more control was sequenced compared to ChIP sample.

In order to assess how the CSSP power estimates are influenced by the sequencing depth of the control sample, I evaluated the power from FNR sub-samples by varying the control sequencing depth as a percentage of the full depth at three levels of 0.4%, 2%, and 20% and fixing the ChIP sample depth at 6%. In Figure 2.10, I compare the resulting estimated power curves to the oracle power curve. I performed a similar experiment with the GATA1 dataset where I used 1%, 2%, and 4% of the control sample and 20% of the ChIP sample. I observed that varying depths of the control sample had little effect on my power calculations as long as the model parameters are reliably estimated. This is due to the fact that the control sample is only used to estimate prior mean for the background intensity while estimation of the parameters regarding the ChIP signal intensity (ν^j) and normalizing effects (e_0, b) mostly rely on the ChIP sample. I observed that the estimation algorithm encountered convergence problems at extremely small depths which would

be considered as low quality data by the currently employed ChIP-seq data quality standards. My analysis suggest that if the normalization is done in a similarly efficient fashion for other peak callers, the effect of control on their performances might also be minimized since the background distribution alone can be captured using lower depth control samples. To illustrate this point, I compared the set of enriched bins identified at sub-sampled ChIP and control data at different combinations of depths at an FDR of 0.05 with the quality thresholds set as $r = 2$ and $\tau = 0$. For fixed ChIP samples, the set of identified enriched regions remained consistent at varying control depths (Figure 2.9). Overall, this supports that increasing the sequencing depth of the control beyond what is required for estimating the background parameters does not lead to power gain.

2.3.8 Impact of Lab and Batch Effects

Table 2.3 Summary for the CTCF ChIP-seq datasets in GM12878 cell line. *: Non-convergent CSSP fit.

Lab	Rep	# of Uni-reads	SPOT	PBC	NSC
BroadHistone*	1	1.01M	0.2586	0.81	2.21
BroadHistone	2	1.75M	0.1873	0.93	1.75
OpenChrome*	1	1.28M	0.3944	0.91	7.34
OpenChrome	2	0.67M	0.2992	0.95	5.13
OpenChrome	3	0.55M	0.3577	0.77	5.47
SydhTfbs	1	1.01M	0.5067	0.85	8.02
SydhTfbs	2	2.05M	0.3187	0.93	4.20
UwTfbs	1	1.42M	0.2673	0.61	3.13
UwTfbs	2	0.65M	0.2597	0.95	3.19

My computational experiments thus far focused on the effect of increasing the sequencing depth while keeping other experimental factors, i.e., lab and batch effects, fixed. Such effects almost always exist when pilot data are used for designing future experiments. In order to investigate

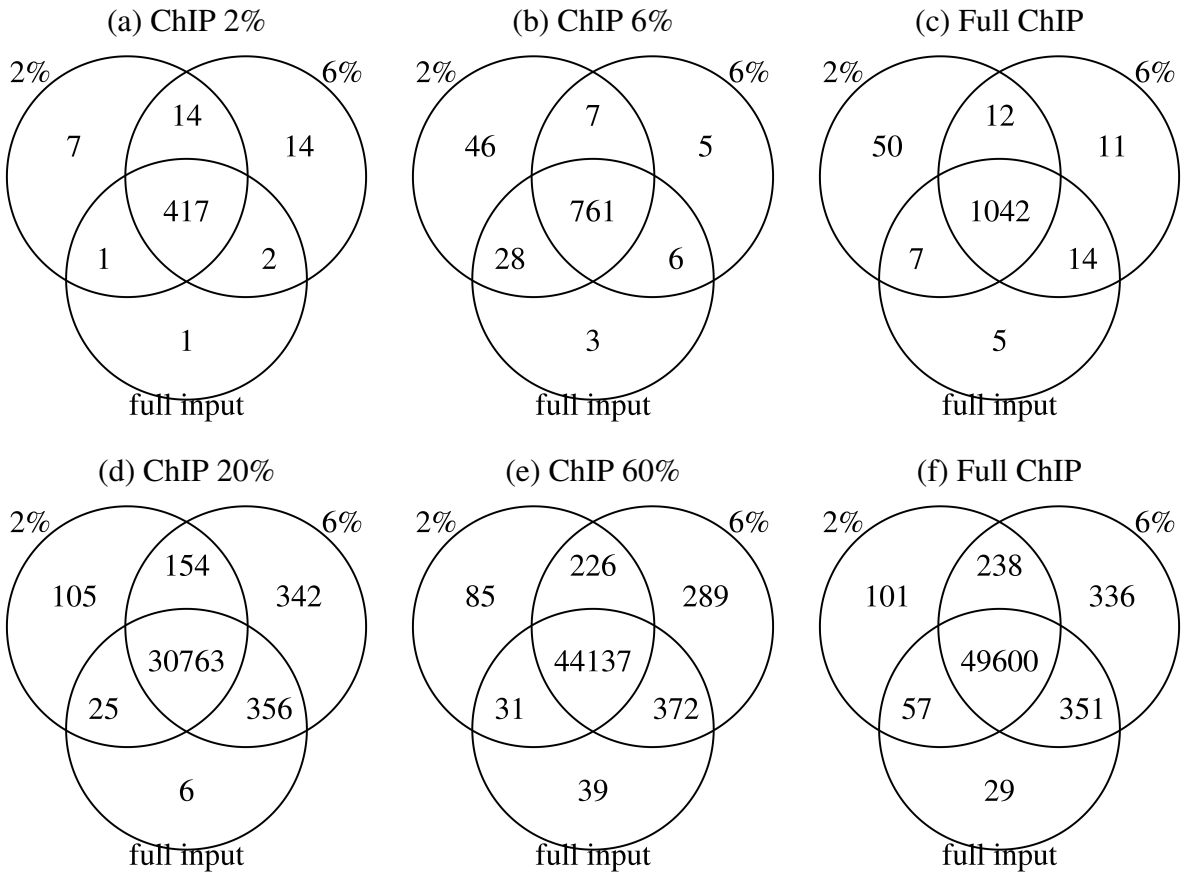


Figure 2.9 Comparison of the number of enriched regions identified by varying the control depths while fixing the ChIP depths for FNR (a, b, c) and GATA1 (d, e, f) datasets.

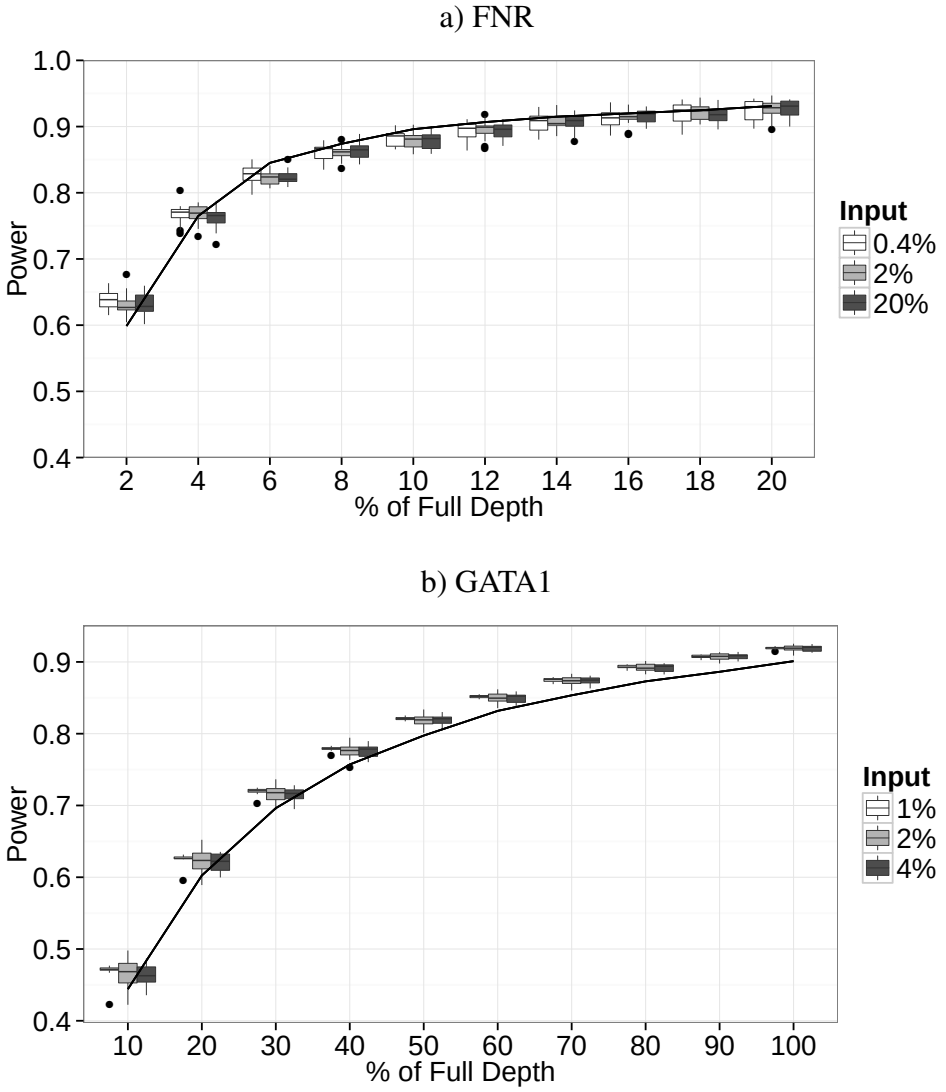


Figure 2.10 *Comparison of power estimation at varying control sequencing depths.* ChIP sample depths are fixed at 6% and 20% of the full sample depth for the (a) FNR and (b) GATA1 samples, respectively. The oracle power based on parameters estimated from full ChIP and control data are displayed by the solid curves.

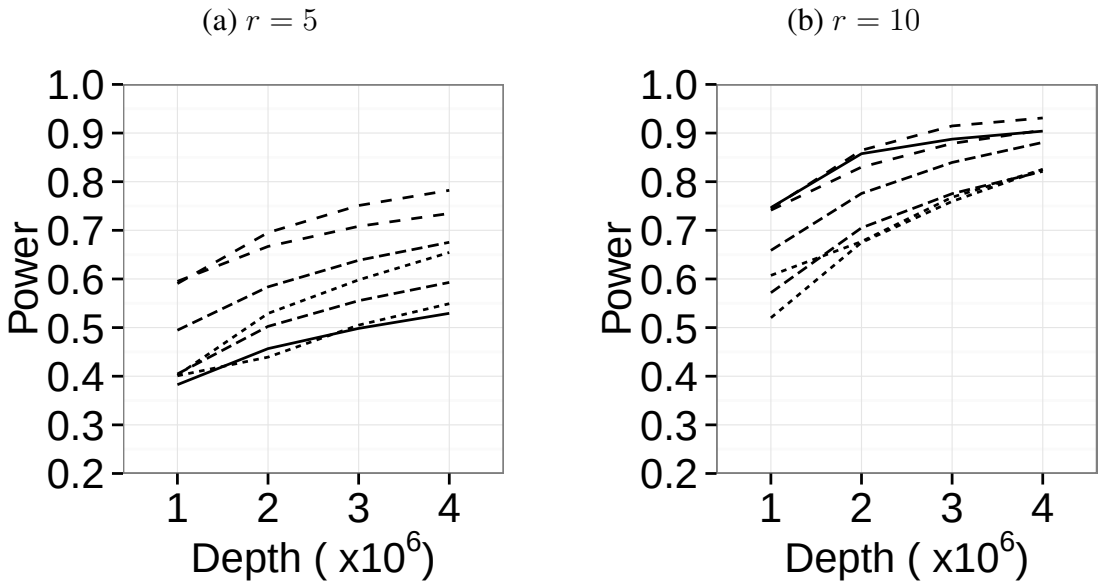


Figure 2.11 *Power prediction within and between labs.* Estimated power curves for CTCF binding in GM12878 cell line using datasets from four different labs, evaluated with fold change thresholds at (a) 5 and (b) 10.

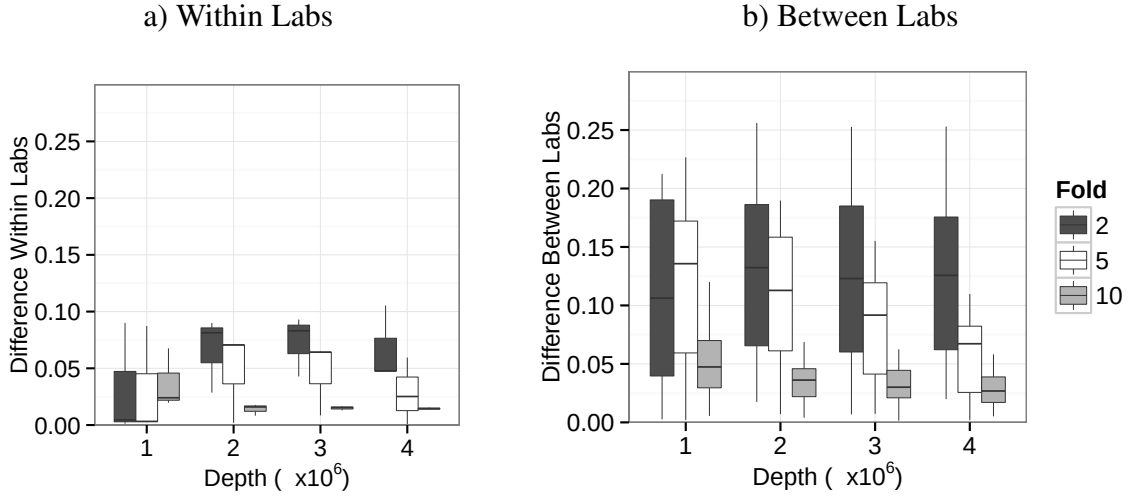


Figure 2.12 *Comparison of power prediction within and across labs.* Absolute differences in predicted power are computed at various sequencing depths for CTCF ChIP-seq samples from (a) same lab and (b) different labs.

their effects on power prediction, I analyzed seven CTCF ChIP-seq datasets from the GM12878 cell line. These datasets were generated by four different labs within the ENCODE consortium ([67]). Each lab sequenced 2-3 individual cultures of GM12878. Samples within a lab differed in one or more of the following aspects: person and/or date for preparation of the cell cultures, cross-linked DNA, and Illumina sequencing libraries; sequencing machine; date of sequencing. The total number of reads as well as the data quality metrics for each dataset are summarized in Table 2.3 [3]. To shorten computation time, I focused on chromosome 1. I sampled ChIP reads from each original dataset so that each pilot dataset had the same number of reads as the lowest sequenced dataset. Then, I extended the reads to average fragment lengths estimated by the `R` package SPP ([29]) and mapped them to bins of size 100 bp. Finally, I paired each ChIP sample to its matching control and fitted the CSSP model. I generated power curves with quality thresholds of $r = 5, 10$ and $\tau = 0$ at sequencing depths ranging from 10^6 to 4×10^6 (Figure 2.11). I computed the differences in predicted power between samples within the same and across different labs at each sequencing depth. Figure 2.12 illustrates that lab effects are more prominent than batch effects within the same lab, and both effects decrease when the fold change threshold increases. Although my study of the lab and batch effects is hampered by the lack of designed experiments for specifically investigating these effects, this limited study on CTCF confirms that while batch effects are difficult to control, controlling the lab effects by using pilot data from the same lab should result in better power prediction.

2.3.9 Power Estimation for a Set of Recent ChIP-seq Datasets

I applied the CSSP framework to evaluate the power of eight ChIP-seq experiments generated by the ENCODE projects ([67]). Datasets included ChIP-seq profiling of GATA2, cFos, cJun, and Pol II in Huvec and K562 cell lines. I merged replicates within experiments and removed abnormal reads using the `R` package SPP ([29]). The remaining uniquely mapping reads were extended to 200 bp and mapped to bins of size 200 bp. I note that these datasets have much lower sequencing

[3] The data quality metrics (SPOT, PBC, NSC) are provided at `http://encodeproject.org/ENCODE/qualityMetrics.html` by the ENCODE consortium. SPOT: proportion of reads mapping to read-enriched regions; PBC: proportion of non-redundant reads; NSC: normalized strand cross correlation coefficient.

Table 2.4 Summary for Huvec and K562 datasets.

	Huvec			
	GATA2	cFos	cJun	Pol II
# of Uni-Reads	38.2M	44.1M	25.4M	23.0M
# of Filtered Reads	30.5M	36.7M	25.1M	22.3M
# of SPP Peaks	66777	87295	59881	57183
# of Peaks, 2-fold	54947	58344	62813	35553
# of Peaks, 5-fold	49346	57223	44828	28654
# of Peaks, 10-fold	26292	38740	23105	13976
	K562			
	GATA2	cFos	cJun	Pol II
# of Uni-Reads	16.0M	9.9M	11.1M	12.6M
# of Filtered Reads	15.9M	9.3M	10.7M	12.3M
# of SPP Peaks	77317	35064	83473	167491
# of Peaks, 2-fold	79978	62935	64408	40454
# of Peaks, 5-fold	52603	62293	64408	40454
# of Peaks, 10-fold	15961	25403	35319	28382

Table 2.5 Mean squared errors (MSE) between the empirical and the estimated power of the simulation experiments ($\times 10^{-4}$).

Sequencing Depth	50%	100%	200%
$r = 2$	0.8	9.9	6.4
$r = 5$	8.2	0.8	0.6
$r = 10$	7.6	0.4	2.2

Table 2.6 Quality metrics for the ENCODE experiments.

Cell line	Factor	UniRead	SPOT	PBC	NSC	Low S/N ratio
Huvec						
	GATA2	18.5M	0.0837	0.82	1.34	N
	GATA2	16.3M	0.21	0.62	1.97	N
	cFos	10.8M	0.1075	0.73	1.37	N
	cFos	33.4M	0.1691	0.76	1.34	N
	cJun	10.7M	0.0587	0.97	1.3	N
	cJun	18.3M	0.1271	0.97	1.46	N
	Pol II	9M	0.0979	0.92	1.37	Y
	Pol II	9.4M	0.0746	0.98	1.21	Y
	Pol II	8.1M	0.133	0.97	1.39	N
K562						
	cFos	4.1M	0.1433	0.93	3.19	N
	cFos	3.8M	0.0833	0.9	2.49	N
	cFos	4.1M	0.1652	0.96	1.81	N
	cJun	7.7M	0.1311	0.94	1.77	N
	cJun	6M	0.0544	0.94	1.29	N
	Pol II	7.6M	0.365	0.94	2.4	N
	Pol II	7.2M	0.334	0.94	2.26	N

depths than the FNR and GATA1 experiments, with an average bin count of 5. For computational reasons, I fitted the CSSP model for each chromosome separately. To restrict our attention on high quality peaks, I set the quality threshold τ to be equivalent to the 99-th percentile of the ChIP bin count intensity λ_i^y estimated using the posterior distributions in Eqns. (2.5) and (2.6), i.e., determined by:

$$\frac{1}{n}\sum_i Pr\{\lambda_i^y \leq \tau | Y_i = y_i, Z_i = z_i\} = 0.99.$$

The estimated τN_y ranged between 10 and 20 counts across all experiments. I set the fold change threshold r at different levels of 2, 5, 10. The numbers of final set of utilized reads, as well as the genome wide power are shown in Table 2.7. Further results on these analysis are available in Table 2.4. In this table, the first two rows denote the number of uni-reads based on Bowtie alignments ([36]) with default parameters and the number of reads after `SPP` filtering ([29]); the third row denotes the number of peaks identified by `SPP` at FDR of 0.05 and the last three rows are the numbers of CSSP peaks at FDR of 0.05 and r values of 2, 5, and 10.

Table 2.7 Estimated power of selected ENCODE datasets.

Cell line	Factor	# of Usable Reads	Power		
			2-fold	5-fold	10-fold
Huvec	GATA2	30.4M	0.894	0.922	0.987
	cFos	36.7M	0.925	0.932	0.987
	cJun	25.1M	0.849	0.959	0.992
	Pol II	22.2M	0.848	0.926	0.987
K562	GATA2	15.9M	0.785	0.808	0.869
	cFos	9.3M	0.759	0.794	0.844
	cJun	10.7M	0.847	0.846	0.869
	Pol II	12.2M	0.887	0.886	0.911

The estimated powers were generally above 80% indicating that more than 80% of true enriched regions that meet our fold change and minimum intensity thresholds were identified. I evaluated robustness of these power calculations with simulations in Table 2.5. MSEs in this table

are computed across 50 simulation samples based on the fit of ENCODE cFos ChIP-seq experiment in Huvec cell line. The estimated power, in general, should increase as sequencing depth increases. However, when comparing power estimates across different experiments, quality of the individual datasets should be considered. I investigated the sequencing quality metrics of these datasets provided by the ENCODE consortium (Table 2.6 [4]) and corroborated implications of these metrics with our power results. For cFos and cJun, SPOT and PBC values are comparable for both cell lines, reflecting comparable data quality. As a result, power estimates for the deeper sequenced Huvec samples are higher. For Pol II, SPOT values of Huvec, which are reflective of the signal-noise ratio, are a lot smaller than those of K562. Hence, for Pol II, although the sequencing depth of the Huvec dataset is almost twice of the K562, K562 sample has higher or comparable power.

2.3.10 Power Implications for Other Peak Callers

Table 2.8 Implications of the CSSP estimated power for SPP.

	Huvec		K562	
Factor	Power	Overlap	Power	Overlap
GATA2	0.410	0.920	0.585	0.759
cFos	0.737	0.964	0.677	0.931
cJun	0.728	0.951	0.645	0.855
PolII	0.656	0.862	0.574	0.566

Although the CSSP framework employs a specific peak calling procedure based on testing against background read distribution and quality thresholding, power estimated by CSSP has implications for other peak callers. To illustrate this, I considered one of the commonly used peak callers SPP ([29]), which is also adopted by the ENCODE projects. The key to adapting CSSP power estimation to SPP is to identify quality thresholds r and τ that would correspond to the

[4] SPOT: proportion of reads mapping to read-enriched regions; PBC: proportion of non-redundant reads; NSC: normalized strand cross correlation coefficient; Low S/N ratio: Low signal to noise ratio (Yes (Y) or No (N)). Information for K562 GATA2 was not available. Source: `http://encodeproject.org/ENCODE/qualityMetrics.html`.

analysis generated by SPP at the same FDR level. To enable this comparison on the ENCODE datasets, I utilized the enriched regions identified by SPP and set the r and τ parameters based on data from these regions. Specifically, to set τ, I mapped the filtered reads to 200 bp regions surrounding the binding sites identified by SPP, and then set τN_y as the minimum ChIP count across these bins. Similarly, I set rN_x/N_y to the minimum ChIP to input count ratio of these bins to estimate r.

I then applied the CSSP model with these r and τ estimates driven by the SPP analysis. I evaluated how well the set of enriched regions from CSSP and SPP agreed with the idea that, for datasets with good agreement, the CSSP power would yield an upper bound for the SPP power (Table 2.8). In this table, overlap proportion was calculated as the proportion of SPP peaks that are among the CSSP peaks, and the SPP peaks were constructed by extending each peak of SPP by the estimated "half window size" ([29]) in both of the $5'$ and $3'$ directions. In addition to imposing the fold change and minimum count thresholds, SPP further filters the set of enriched regions based on the symmetry of the read distributions around each enrichment site; therefore, it is more conservative than CSSP. For the four datasets where the overlap percentages between CSSP and SPP exceeded 90%, CSSP estimated power presents upper bounds for the SPP power. For the other four experiments, more than 15% of SPP peaks are not captured by the CSSP model. I noticed that these four experiments either have lower data quality or sequencing depths, and the discrepancy between the two peak callers might due to low signal-to-noise ratios or different FDR control procedures and requires further investigation.

2.4 Conclusions and Discussion

The sequencing depths of most, if not all, initial published experiments have been limited by practical considerations such as cost or instrument availability. With decreasing sequencing costs, considerations are shifting from how many sequences should be obtained for a single experiment, to how many experiments one can perform in a single lane. Therefore, power calculations are extremely important for ChIP-seq experiments. I have developed the CSSP framework to enable such power calculations. This framework can be applied to compute power at a wide range of

sequencing depths with varying fold change and minimum intensity thresholds. My extensive computational experiments demonstrated the consistency in predicting power from pilot data and its practical implications. To the best of our knowledge, this is the first model that enables power analysis for ChIP-seq data through an analytical approach.

It is worth noting that although our calculations mostly emphasize the sequencing depth N_y, other parameters including e_0 and ν^j, which indicate the signal-to-noise ratio of the data, as well as the data quality are also important factors of the power analysis. These parameters are fixed when comparing datasets obtained under the same experimental conditions. However, for comparing datasets with different experimental conditions such as TF and cell line, effects of data quality and strengths of enrichment signals should bear equal emphasis. My limited investigation of the lab and batch effects indicated that lab effects are larger than batch effects within a lab and that pilot data from the same lab would yield more unbiased power prediction than pilot data from another lab.

While the analytical calculations in the CSSP framework depend on the peak calling procedure implied by our model, the power estimation has broad applications for other peak callers. In general, if the peaks identified by the peak caller can also be identified by CSSP at the same FDR level and at certain fold change and minimum enrichment thresholds, then the power evaluated at these thresholds can serve as the upper bound for that peak caller. When peaks identified by a peak caller are vastly different than that of the CSSP, our power results can not directly be related to that peak caller. Analyzing the power for an arbitrary peak caller requires specialization of our algorithm. Overall, since the CSSP peak calling procedure is simpler than most existing peak callers, our power estimation can serve as a benchmark for other peak callers.

The CSSP framework also enables the investigation of the impact of input sequencing depth on power. My computational experiments indicate that increasing the input depth does not increase the peak calling power or the accuracy of power prediction. The impact of input depth is, to a large extent, determined by how the ChIP read counts are normalized against input read counts, a procedure that highly varies among peak callers. Overall, our results suggest that if the ChIP

sample is normalized efficiently against the input data, the dependence of power on input depths may be reduced.

Chapter 3

atSNP: Transcription Factor Binding Affinity Testing for Regulatory SNP Detection[1]

3.1 Introduction

Genome-wide association studies have been instrumental in identifying single nucleotide polymorphisms (SNPs) associated with large numbers of phenotypes. The vast majority of association SNPs are in non-coding regions suggesting that they may have regulatory roles in deriving the phenotype ([45]). In particular, regulatory SNPs which alter binding affinity of transcription factors and affect gene expression constitute an important class of such SNPs ([52]). A standard *in silico* approach for identifying rSNPs is by evaluating how the SNP-driven nucleotide change impacts binding affinity of TFs to the region surrounding the SNP ([42, 56, 69, 3]). Specifically, the DNA sequences around each SNP are scored against a library of TF motifs with both the reference and the SNP alleles using position weight matrices (PWMs) ([65]) of the motifs. SNPs with significantly different scores between the reference and SNP alleles are then hypothesized as rSNPs.

I describe atSNP, an `R` package that carries out the following tasks for every SNP-motif combination of the input data after extracting genome sequences of small windows (± 30 bps) around the SNP positions: (1) computing affinity scores for both alleles; (2) statistical testing for allele-specific affinity scores; (3) statistical testing for changes in affinity scores between alleles. A few existing tools can perform various subsets of these tasks (Table 3.1). The most distinctive feature of atSNP is its ability to accommodate large scale analysis (e.g., over $>$ 20K SNPs). is-rSNP has the most similar functionality to atSNP; however, is-rSNP (both 1.0 and 2.0) can only analyze at most

[1]The manuscript for this chapter [86] is accepted by the Bioinformatics journal. Method in this chapter is implemented in the `R` package `atSNP` and is freely available at `http://github.com/chandlerzuo/atsnp`.

Method	Allele-specific scores	P-values for allele-specific scores	Between-allele scores	P-values for between-allele scores	User specified motif library	Scalability to >20K SNPs	Visualization of SNP effects	Open source code
atSNP	✓	✓	✓	✓	✓	✓	✓	✓
is-rSNP ([42])	✓	✓	✓	✓	✓			
RAVEN ([3])	✓		✓					
rSNP-MAPPER ([56])	✓		✓					
TRAP* ([69])	✓	✓	✓					
FIMO** ([17])	✓	✓			✓			✓

Table 3.1 Comparison of existing *in-silico* rSNP detection tools. *: TRAP takes as input only one SNP at a time. **: FIMO scans sequences for occurrences of motifs and is not readily a rSNP tool.

20 SNPs at a time. Similarly, TRAP takes as input only one SNP. Although rSNP-mapper can take as input larger number of SNPs, it lacks critical calculations such as the significance of SNP-driven affinity change. FIMO is not designed for evaluating SNP impact on affinity scores; however, it enables p-value computation for affinity scores and can be used to compare scores under different alleles. Moreover, due to computational reasons, FIMO can only accommodate outputting results thresholded by a small pre-specified significance level for large SNP sets. In my hands with a 24 AMD Opteron 2.2 GHz processor, a FIMO run for 26,100 SNPs against a single PWM without thresholding could not finish within 24 hours whereas atSNP required less than 5 minutes. The main computational burden of both FIMO and is-rSNP is the computation of the exact p-values by enumerating all possible sequences and computing their score under the null hypothesis. atSNP utilizes an importance sampling technique to overcome this challenge.

Figure 3.1 summarizes the main inputs and outputs of atSNP. The input SNP file contains the reference (a1) and the SNP (a2) alleles; however, when only dbSNP IDs are provided, atSNP acquires the necessary location and allele information using the R package `rsnps` (`http://cran.`

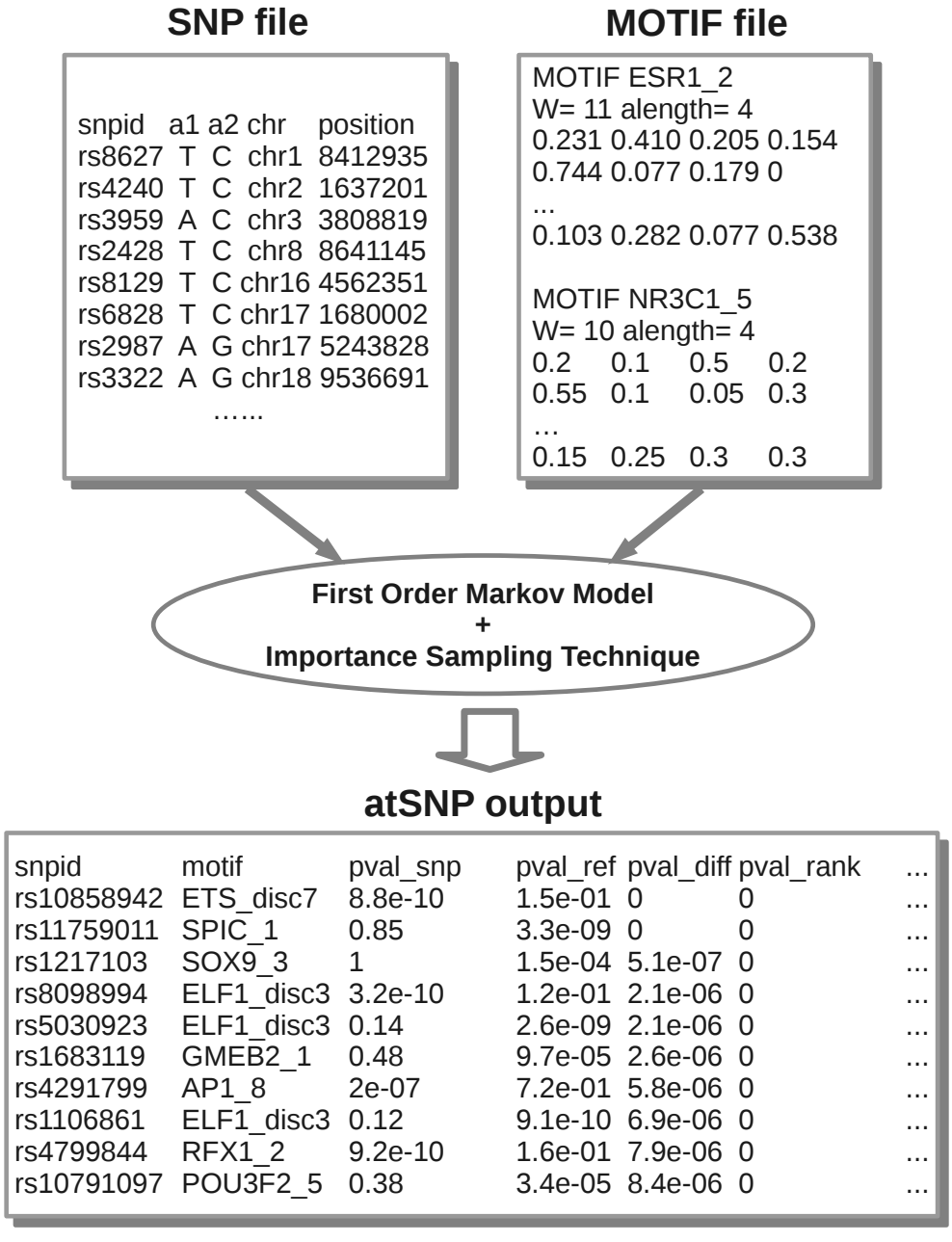


Figure 3.1 A flow chart describing atSNP analysis.

r-project.org/web/packages/rsnps/rsnps.pdf). The motif file is in MEME motif format, one of the several allowed formats. atSNP uses a first order Markov model for generating random background sequences and importance sampling techniques for efficient p-value calculation.

atSNP includes a motif library of 2065 PWMs from the ENCODE project ([30]) and the JASPAR core motif library ([43]). In addition, it allows user-defined motif libraries in a variety of formats, e.g., MEME format ([17]) or other PWM libraries from the JASPAR database ([43]). atSNP accesses genome data of the input organism through the Bioconductor BSGenome package ([51]) and thus can analyze data from a variety of organisms. It computes the binding affinity score for each subsequence overlapping the SNP position in either strand and reports the maximum of these as the affinity score of the sequence. In order to evaluate the significance of these scores, atSNP first estimates a null distribution for the scores by a first-order Markov model using the sub-sequences surrounding the SNP positions (default ± 30 bps of the SNP position). P-value computations for both the allele-specific scores and between-allele score differences are carried out using importance sampling algorithms adapted from [7] (Section 5.3). I compared the p-values computed by atSNP with those computed by FIMO ([17]) and illustrated that the importance sampling method drastically improves computational time without sacrificing accuracy (Section 3.3.1).

atSNP produces as output a `data.table` listing the affinity scores, p-values, and allele-specific matching positions for each SNP-motif pair. This `R` data structure provides powerful functionality for querying and integrating additional data sources. The atSNP output table in Figure 3.1 contains the SNPs with the most significant affinity score changes for my example in Section 3.3.4. Each row provides in-depth SNP-motif pair information such as SNP ID (snpid), motif name (motif), p-value for the binding affinity with the SNP and the reference alleles (pval_snp, pval_ref), and the p-value for binding affinity change based on log-likelihood ratio and log-rank ratio (pval_diff, pval_rank).

Furthermore, atSNP provides composite logo plots for directly visualizing the SNP effects on motif matches as in Figure 3.2. In Figure 3.2, the SNP location is within the dashed box. The p-values for the binding affinity of the best matches with the SNP and reference alleles are 2.29e-3 and 4.9e-4, respectively. The p-value for the affinity change is 0.058 (ranked 1450th among all the

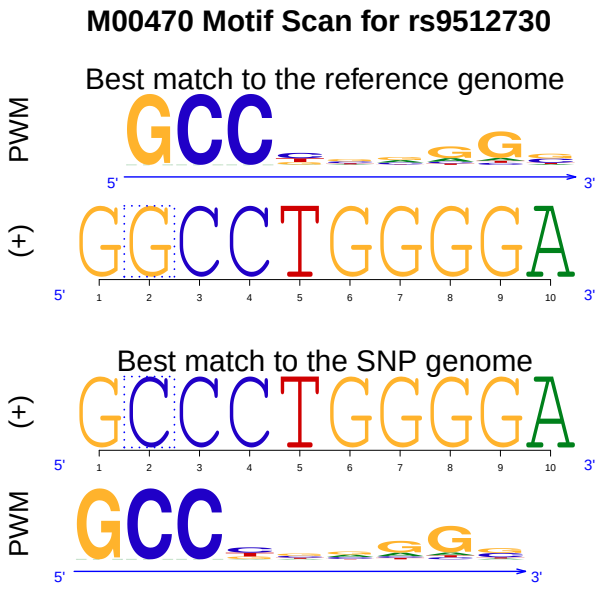


Figure 3.2 A composite logo plot for rs9512730-M00470 (TFAP2) pair from atSNP.

26100 SNPs). Notice that Figure 3.2 also clarifies an important feature of atSNP. If we compare the binding affinities of the reference and SNP allele sequence based on the matching position on the reference allele only, there is a big score change induced by the SNP. This is likely to be a false positive, because shifting 1 bp to the left results in a matching subsequence with the SNP allele. In Figure 3.2, atSNP allows the matching positions on both alleles to be different and thereby avoids such potential false positives.

3.2 The Importance Sampling Algorithms

In this section, I describe the algorithms for computing and testing the affinity scores for each allele and change in affinity scores between the alleles. I code the four nucleotides by 'A'-1, 'C'-2, 'G'-3, and 'T'-4. The reverse complement of nucleotide i is obtained by $5 - i$ in this coding scheme. Let W denote the $4 \times L$ position weight matrix for a motif of length L and $W(i, l)$ the entry for nucleotide i at position l with $\sum_{i=1}^{4} W(i, l) = 1$.

The affinity score calculation requires considering all possible nucleotide sequences of length L that overlaps the SNP position. Such a sequence must be located within a window of size $2L - 1$

around the SNP position. Let $\mathbf{x} = (x_1, x_2, \cdots, x_{2L-1})$ denote the nucleotides in this window. The binding affinity score of a subsequence $(x_s, x_{s+1}, \cdots, x_{s+L-1})$ is given by

$$C(\mathbf{x}, s) = \sum_{l=1}^{L} \log W(x_{l+s-1}, l). \tag{3.1}$$

Then, the affinity score of $\mathbf{x}$ is the maximum of the scores across all subsequences from both strands given by

$$C(\mathbf{x}) = \max\{C(T(\mathbf{x}), s) : T \in \{I, R\}, s = 1, 2, \cdots, L\},$$

where I and R are two strand operators with $I(\mathbf{x}) = \mathbf{x}$, $R(\mathbf{x}) = (5-x_{2L-1}, 5-x_{2L-2}, \cdots, 5-x_1)$, i.e., the reverse complement sequence.

The binding affinity score definition in Eqn. (3.1) assumes that W describes a motif with a product multinomial distribution as in [17, 7], i.e., $W(i, l) \in [0, 1]$, and, therefore, $C(\mathbf{x}, s)$ represents the log-likelihood of the subsequence starting at position s under this model. If W is already a transformed version of the product multinomial model parameters, e.g., $W(i, l) \in \mathbb{R}$, then the affinity score simply corresponds to

$$C(\mathbf{x}, s) = \sum_{l=1}^{L} W(x_{l+s-1}, l). \tag{3.2}$$

The affinity tests of atSNP are based on Eqn. (3.1) by default; however, they can be modified to adapt Eqn. (3.2) by an exponential transformation of the entries of the PWM, i.e., by replacing $W(i, l)$ with $\exp(W(i, l))$. In the subsequent sections, I describe the p-value computation algorithms based on Eqn. (3.1). These algorithms readily provide the tests for Eqn. (3.2) once I apply the exponential transformation.

3.2.1 Computing and Testing Allele-specific Binding Affinity Scores

I assume that, under the null hypothesis that a subsequence overlapping the SNP comes from a genomic background distribution, the nucleotide sequences follow a stationary reversible first order Markov model with distribution $P(X_l = k) = \pi(k)$, $k = 1, \cdots, 4$, and transition probabilities

$P(X_{l+1} = n|X_l = k) = p(k,n)$, $k,n = 1,\cdots,4$. Under this model, the joint probability for sequence $\mathbf{x}$ is given by

$$f_{\mathcal{H}_0}(\mathbf{x}) = \pi(x_1) \prod_{l=1}^{2L-2} p(x_l, x_{l+1}). \tag{3.3}$$

Given an observed sequence $\mathbf{x}_0$, either from the reference or the SNP allele, atSNP computes the allele-specific p-value defined as the probability that affinity score of a sequence from the null background model is at least as large as $C(\mathbf{x}_0)$:

$$pval(\mathbf{x}_0) = P\{C(\mathbf{X}) \geq C(\mathbf{x}_0)|\mathbf{X} \sim f_{\mathcal{H}_0}\}, \tag{3.4}$$

where $\mathbf{X}$ is the random variable denoting the sequence of length $2L-1$ overlapping the SNP. Note that this p-value corresponds to the whole sequence of length $2L-1$ which includes all subsequences of length L that can overlap the SNP position. Another useful quantity is the so-called *conditional p-value* that can be calculated for a fixed subsequence of length L. The traditional algorithms, such as FIMO, that scan a sequence with PWMs calculate such p-values for each subsequence. Formally, I define the conditional p-values as follows. Given the observed sequence $\mathbf{x}_0$, I first find the location of the subsequence that best matches the PWM: $(T_0, s_0) = \arg\max\{C(T(\mathbf{x}_0), s) : T \in \{I, R\},\ 1 \leq s \leq L\}$. The conditional p-value is the probability for the score of a random sequence evaluated at this fixed location to be as large as the observed score $C(\mathbf{x}_0)$. This can be formulated as:

$$pval'(\mathbf{x}_0) = P\{C(T_0(\mathbf{X}), s_0) \geq C(\mathbf{x}_0)|\mathbf{X} \sim f_{\mathcal{H}_0}\}. \tag{3.5}$$

Before I describe the estimation algorithm for the p-values, I will discuss the differences between these two p-value types in Figure 3.3. Both quantities compare $C(\mathbf{x}_0)$ with the affinity scores from sequences randomly generated under the null model. Given a null sequence, the conditional p-value calculates the sample affinity score based on a fixed strand and location, while the p-value calculates the maximum affinity score based on all subsequences from both strands. As a result, the sample affinity scores corresponding to the p-values are at least as large as those for the conditional p-values. Therefore, p-values are always larger than the conditional p-values; however, they directly reflect the significance of the maximum affinity score for the observed sequence. I argue

that, because I do not know the location of the subsequences that best match to the motifs, Eqn. (3.4) is more appropriate for calculating allele-specific significance. atSNP provides computation of both the p-values and the conditional p-values, thereby allows us to compare its accuracy with the conditional p-values from FIMO (Section 3.3.1).

Next, I describe the estimation algorithm. If I can simulate B sequences, $\mathbf{x}_1, \cdots, \mathbf{x}_B$, under the null distribution $f_{\mathcal{H}_0}$, an empirical estimator for the p-value is $\sum_{b=1}^{B} 1\{C(\mathbf{x}_b) \geq C(\mathbf{x}_0)\}/B$. I note that the p-value is just the probability of the event $\{C(\mathbf{X}) \geq C(\mathbf{x}_0)\}$. This is a rare event when p-values are small, and the naive Monte-Carlo simulation method requires a large number of simulations. The importance sampling technique addresses this problem by the following insight:

$$\begin{aligned} pval(\mathbf{x}_0) =& E[1\{C(\mathbf{X}) \geq C(\mathbf{x}_0)\} | \mathbf{X} \sim f_{\mathcal{H}_0}] \\ =& E[1\{C(\mathbf{X}) \geq C(\mathbf{x}_0)\} \frac{f_{\mathcal{H}_0}(\mathbf{X})}{h(\mathbf{X})} | \mathbf{X} \sim h], \end{aligned} \tag{3.6}$$

where h is a sampling distribution under which the event $\{C(\mathbf{X}) \geq C(\mathbf{x}_0)\}$ occurs more often compared to $f_{\mathcal{H}_0}$. Motivated by the idea from [7], I consider a sampling distribution by adding the exponents of the affinity score as weights to $f_{\mathcal{H}_0}$. First, I consider sampling a random sequence $\mathbf{X}$ and a motif matching position S from the following distribution:

$$g_\theta(\mathbf{x}, s) = \frac{f_{\mathcal{H}_0}(\mathbf{x}) \exp(\theta C(\mathbf{x}, s))}{H(\theta)}.$$

Here, θ is a tilting parameter and $H(\theta)$ is the normalizing constant. Because I put a weight of $\exp(\theta C(\mathbf{x}, s))$, when $\theta > 0$, I am more likely to get sequences with large affinity scores. Then, the sampling distribution for the sequence $\mathbf{X}$ is given by

$$h_\theta(\mathbf{x}) = \frac{\sum_{s=1}^{L} f_{\mathcal{H}_0}(\mathbf{x}) \exp(\theta C(\mathbf{x}, s))}{H(\theta)}. \tag{3.7}$$

A useful property for g_θ is:

$$E(C(\mathbf{X}, S) | (\mathbf{X}, S) \sim g_\theta) = \frac{d}{d\theta} \log H(\theta).$$

Since, under g_θ, a random sequence $\mathbf{x}$ tends to have a large affinity score for a subsequence starting from s, it is very likely that $C(\mathbf{x}, s) = C(\mathbf{x})$. In other words, if I simulate sequences under g_θ,

(a)

(b)

PWM=

A	0.25	1e-10	0.98	1e-10	0.47
C	0.25	0.09	1e-10	1e-10	0.13
G	0.34	0.91	0.01	1e-10	0.28
T	0.17	1e-10	1e-10	1	0.11

Random sequence: GTTGTCTAA

Sequence	Score	Rev Score
GTTGTCTAA	**-72.3**	-28.8
G**TTGTC**TAA	-30.8	-70.9
GT**TGTCT**AA	-50.12	**-25.3**
GTT**GTCTA**A	-47.88	-53.8
GTTG**TCTAA**	-50.93	-48.6

Conditional sample score: -72.3
Unconditional sample score: -25.3

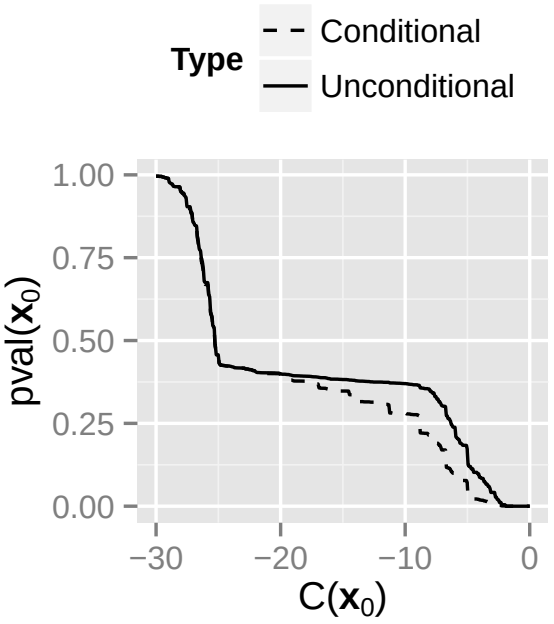


Figure 3.3 Difference between the p-value and the conditional p-value. (a) Computing the sample affinity scores based on a random sequence generated from the null hypothesis. (b) The p-values and conditional p-values at different affinity scores.

then the expected value of $C(\mathbf{X})$ is approximately $\frac{d}{d\theta}\log H(\theta)$. [7] suggested choosing θ such that $E(C(\mathbf{X})|g_\theta) \approx C(\mathbf{x}_0)$ for estimating p-value at $C(\mathbf{x}_0)$. Following this suggestion, I first group the scores from all SNPs into multiple ranges, and then for each range where the scores are close to c, I use the sampling distribution g_θ with θ set by solving $\frac{d}{d\theta}\log H(\theta) = c$.

Finally, the p-values can be estimated by

$$\widehat{pval}(\mathbf{x}_0) = \frac{1}{T}\sum_{t=1}^{T} 1\{C(\mathbf{x}_t) \geq C(\mathbf{x}_0)\}\frac{H(\theta)}{\sum_{s=1}^{L}\exp(\theta C(\mathbf{x}_t, s))}. \tag{3.8}$$

Similarly, the conditional p-value can be estimated using the same sampling distribution by

$$\widehat{pval'}(\mathbf{x}_0) = \frac{1}{B}\sum_{b=1}^{B} 1\{C(T(\mathbf{x}_b), s_b) \geq C(\mathbf{x}_0)\}\frac{H(\theta)}{\exp(\theta C(\mathbf{x}_b, s_b))}. \tag{3.9}$$

In summary, the p-values for all SNPs for a given PWM can be computed by this importance sampling scheme as follows:

1. Group the affinity scores into different sets $\mathcal{G}_1, \cdots, \mathcal{G}_K$ such that the scores within each set are close to each other and to c_k. K and representative score value c_k for each set k, $k = 1, \cdots, K$ are set as follows.

   (a) Denote the number of SNPs by N. If $N \leq 20$, then each $\mathcal{G}_k$ is the singleton set of one score, and $K = N$.

   (b) If $N > 20$, set $K = 20$, and $p_k = 1 - N^{-k(k+1)/[K(K+1)]}$ for $1 \leq k \leq K$.

   (c) Set c_k as the $100 \times p_k$-th percentile of the observed scores of all SNPs across both alleles.

   (d) Set $\mathcal{G}_k$ as the set of the scores in the interval $((c_{k-1}+c_k)/2, (c_k+c_{k+1})/2]$ for $2 \leq k \leq K-1$, $\mathcal{G}_1$ as the set of the scores in the interval $(-\infty, (c_1+c_2)/2]$, $\mathcal{G}_K$ as the set of the scores in the interval $((c_{K-1}+c_K)/2, \infty)$. Set c_k as the representative score for $\mathcal{G}_k$.

2. For each set of scores $\{C(\mathbf{x}_0^i)\}$ in $\mathcal{G}_k$, $k = 1, \cdots, K$ with representative score c_k:

   (a) Set $\theta : \frac{d}{d\theta} \log H(\theta) = c_k$. Calculate $H(\theta)$.

   (b) To set the Monte-Carlo sample size B, first calculate B' as the integer part of $100(1 - p_k)/p_k$. If $B' > 10^5$, set $B = 10^5$; if $B' < 2000$, set $B = 2000$; otherwise, set $B = B'$.

   (c) Simulate B Monte-Carlo samples $(\mathbf{x}_b, s_b)$ from the distribution g_θ. Compute $C(\mathbf{x}_b)$ and $\sum_{s=1}^{L} C(\mathbf{x}_b, s)$. Let $(T_b, s_b) = \arg\max\{C(T(\mathbf{x}_b), s) : T \in \{I, R\},\ 1 \le s \le L\}$.

   (d) Estimate the p-value and the conditional p-value for each $\mathbf{x}_0^i \in \mathcal{G}_k$ by Eqns. (3.8) and (3.9).

The details for computing $H(\theta)$ and sampling from g_θ are discussed in Section 3.2.3.

3.2.2 Computing and Testing Binding Affinity Score Change Between Alleles

I assume that the sequence of the reference allele $\mathbf{X}$ under the null distribution follows the first order Markov model in Eqn. (3.3). The SNP allele sequence differs from the reference allele sequence only by nucleotide x_L. I let $\mathbf{x}^a = (x_1, \cdots, x_{L-1}, x_L^a, x_{L+1}, \cdots, x_{2L-1})$ denote the sequence with the SNP allele and assume that

$$P(X_L^a = x_L^a | X_L = x_l) = \frac{1\{x_L^a \neq x_L\}}{3}.$$

Then, the joint distribution of $\mathbf{x}$ and $\mathbf{x}^a$ is given by

$$f^a(\mathbf{x}, \mathbf{x}^a) = \frac{1\{x_L \neq x_L^a\}}{3} \pi(x_1) \prod_{l=2}^{2L-1} p(x_{l-1}, x_l) 1\{x_L \neq x_L^a\}.$$

For a given SNP-PWM pair, atSNP evaluates whether the SNP allele impacts the match to PWM significantly, either by disrupting a subsequence overlapping the SNP position with good binding affinity score or generating a subsequence with even better score. It computes two types of p-values corresponding to different test statistics. The first p-value, denoted by $pval_d$, assesses whether the change in the binding affinity scores of the two alleles is significantly different than what would be

expected by chance and is given by

$$pval_d(\mathbf{x}_0, \mathbf{x}_0^a) = P\{|C(\mathbf{X}) - C(\mathbf{X}^a)| \geq |C(\mathbf{x}_0) - C(\mathbf{x}_0^a)|| \\ (\mathbf{X}, \mathbf{X}^a) \sim f^a\}.$$

The second p-value, denoted by $pval_r$ assesses whether the change in the ranks of the PWM matches of the subsequences with the reference and SNP alleles is significantly different than what would be expected by chance and is given by

$$pval_r(\mathbf{x}_0, \mathbf{x}_0^a) = P\{|\log(pval(\mathbf{X})) - \log(pval(\mathbf{X}^a))| \geq \\ |\log(pval(\mathbf{x}_0)) - \log(pval(\mathbf{x}_0^a))|| \\ (\mathbf{X}, \mathbf{X}^a) \sim f^a\},$$

where $pval$ follows the definition in Eqn. (3.4). $pval_d$, which compares the log of the likelihoods of the best subsequence matches to the PWM with the reference and SNP allele, is motivated by the likelihood ratio test framework and is easier to compute. However, I observed that since the binding affinity score difference is bounded, if the PWM has multiple highly conserved bases with probability of the relevant nucleotide occurrence close to 1, the p-value at the maximum score difference can still be insignificant. The rank test attenuates this problem since the maximum log rank ratio, $|\log(pval(\mathbf{X})) - \log(pval(\mathbf{X}^a))|$, is essentially unbounded. Figure 3.4 provides an example illustrating this difference.

Next, I introduce a few additional quantities to derive the importance sampling distribution. Let IW, a $4 \times L$ matrix, denote induced PWM with entries

$$IW(i,l) = \frac{W(i,l) + 1/4}{2}.$$

Let D denote another $4 \times L$ matrix with entries

$$D(i,l) = \exp\left(\frac{\sum_{j \neq i}(\log W(i,l) - \log W(j,l))}{3}\right).$$

In the sampling distribution, I assume that in addition to a subsequence of length L in the center, subsequences on the two ends follow the Markov model. With a slight abuse of the notation as

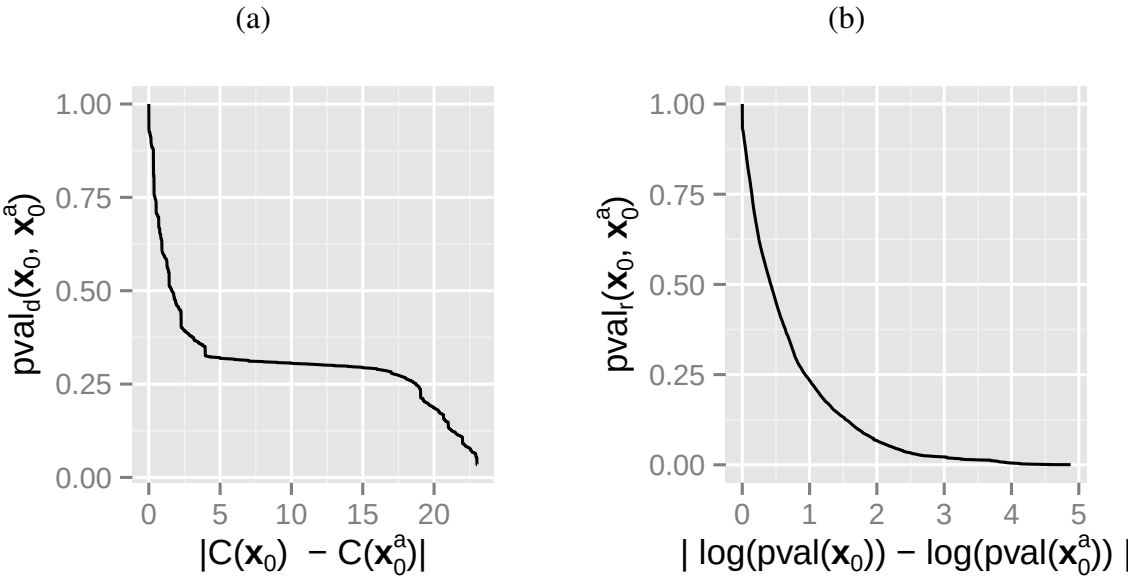


Figure 3.4 Comparison between the score statistic- ($pval_d$) and rank-based ($pval_r$) p-values. (a) The score test p-values at varying score changes between the reference and the SNP allele. The p-value at the maximum possible score change is 0.0312. (b) The rank test p-values at varying log rank ratios.

$f(x_m, \cdots, x_n) = \pi(x_m)p(x_m, x_{m+1}) \cdots p(x_{n-1}, x_n)$, I have

$$\begin{aligned} h^a_\theta(\mathbf{x}, \mathbf{x}^a) =& \frac{1\{x^a_L \neq x_L\}}{3H^a(\theta)} \sum_{s=1}^{L} \Bigg\{ f(x_1, \cdots, x_{s-1}) \\ & \Big[\prod_{1 \leq l \leq L, l \neq L} IW(x_{s+l-1}, l) \Big] D(x_L, L-s+1)^\theta \\ & f(x_{L+s}, \cdots, x_{2L-1}) \Bigg\}, \end{aligned}$$

which marginalizes g^a_θ over s:

$$\begin{aligned} g^a_\theta(\mathbf{x}, \mathbf{x}_0, s) =& \frac{1\{x^a_L \neq x_L\}}{3H^a(\theta)} f(x_1, \cdots, x_{s-1}) \\ & \Big[\prod_{1 \leq l \leq L, l \neq L} IW(x_{s+l-1}, l) \Big] D(x_L, L-s+1)^\theta \\ & f(x_{L+s}, \cdots, x_{2L-1}). \end{aligned}$$

The key points when simulating sequences for calculating change in binding affinity scores are (1) the sequence should have a subsequence matching to the PWM and (2) a change at base x_L of the SNP position will result in a large change in the affinity score. In g^a_θ, $\big[\prod_{l=1}^{L} IW(x_{s+l-1}, l)\big]$ weighs a length L subsequence starting from s, and $\log D(x_L, L-s+1)$ is the expected change in affinity score for this subsequence when x_L is changed. I also have

$$E_{g^a_\theta}[C(\mathbf{x}, s) - C(\mathbf{x}^a, s)] = \frac{d}{d\theta} \log H^a(\theta).$$

Therefore, to compute the p-value for an observed score change $|C(\mathbf{x}_0) - C(\mathbf{x}^a_0)|$, I can pick a value Δc close to $|C(\mathbf{x}_0) - C(\mathbf{x}^a_0)|$, and set θ by solving $\Delta c = \frac{d}{d\theta} \log H^a(\theta)$. For computing p-values for score changes at all SNPs, I implement the following algorithm:

1. Group the difference in affinity scores into different sets, $\mathcal{G}_1, \cdots, \mathcal{G}_K$, such that the scores within each set is close to each other and Δc_k, $k = 1, \cdots, K$.

   (a) Set $p_k = 0.1, \cdots, 0.9, 0.91, 0.92, \cdots, 0.99$ for $1 \leq k \leq K = 18$.

   (b) Set Δc_k as the $100 \times p_k$ percentile of the observed scored differences across all SNPs.

(c) Set $\mathcal{G}_k$ as the score set in the range $((\Delta c_{k-1} + \Delta c_k)/2, (\Delta c_k + \Delta c_{k+1})/2]$ for $2 \leq k \leq K-1$, $\mathcal{G}_1$ as the score set in the range $(-\infty, (\Delta c_1 + \Delta c_2)/2]$, $\mathcal{G}_K$ as the score set in the range $((\Delta c_{K-1} + \Delta c_K)/2, \infty)$.

2. For each set of score differences $\{|C(\mathbf{x}_0^i) - C(\mathbf{x}_0^{ai})|\}$ in $\mathcal{G}_k$, $k = 1, \cdots, K$:

(a) Set $\theta : \frac{d}{d\theta} \log H^a(\theta) = \Delta c_k$. Calculate $H^a(\theta)$.

(b) To set the Monte-Carlo sample size B, first, calculate B' as the integer part of $100(1 - p_k)/p_k$. If $B' > 10^5$, set $B = 10^5$; if $B' < 2000$, set $B = 2000$; otherwise, set $B = B'$.

(c) Simulate B Monte-Carlo samples $(\mathbf{x}_b, \mathbf{x}_b^a, s_b)$ from distribution g_θ^a. Compute $C(\mathbf{x}_b)$, $C(\mathbf{x}_b^a)$.

(d) Estimate the score test p-value for each pair $(\mathbf{x}_0^i,\ \mathbf{x}_0^{ai})$ by

$$\widehat{pval}_d(\mathbf{x}_0^i, \mathbf{x}_0^{ai}) = \frac{1}{B}\sum_{b=1}^{B} 1\{|C(\mathbf{x}_b) - C(\mathbf{x}_b^a)| \geq |C(\mathbf{x}_0^i) - C(\mathbf{x}_0^{ai})|\} \cdot \frac{h_\theta^a(\mathbf{x}_b, \mathbf{x}_b^a)}{f^a(\mathbf{x}_b, \mathbf{x}_b^a)}.$$

(e) For the rank tests p-values, first compute the allele-specific p-value for each Monte-Carlo observation by

$$pval_b = \frac{1}{B}\sum_{b'=1}^{T} 1\{C(\mathbf{x}_{b'}) \geq C(\mathbf{x}_b)\} \cdot \frac{h_\theta^a(\mathbf{x}_b, \mathbf{x}_b^a)}{f^a(\mathbf{x}_b, \mathbf{x}_b^a)}.$$

$$pval_b^a = \frac{1}{B}\sum_{b'=1}^{T} 1\{C(\mathbf{x}_{b'}) \geq C(\mathbf{x}_b^a)\} \cdot \frac{h_\theta^a(\mathbf{x}_b, \mathbf{x}_b^a)}{f^a(\mathbf{x}_b, \mathbf{x}_b^a)}.$$

(f) Estimate the rank test p-value for each pair $\mathbf{x}_0^i$, $\mathbf{x}_0^{ai}$ by

$$\widehat{pval_r}(\mathbf{x}_0^i, \mathbf{x}_0^{ai})) = \frac{1}{B}\sum_{b=1}^{B} 1\Bigg[\Big\{\big|\log(pval_b) - \log(pval_b^a)\big| \geq \big|\log(\widehat{pval}(C(\mathbf{x}_0^i))) - \log(\widehat{pval}(C(\mathbf{x}_0^{ai})))\big|\Big\} \cdot \frac{h_\theta^a(\mathbf{x}_b, \mathbf{x}_b^a)}{f^a(\mathbf{x}_b, \mathbf{x}_b^a)}\Bigg].$$

3.2.3 Computational Details

3.2.3.1 Details for Allele-specific Tests

To compute $H(\theta)$, I first note that $H(\theta) = \sum_{s=1}^{L} H_s(\theta)$, where

$$H_s(\theta) = \sum_{\mathbf{x}\in\{1,2,3,4\}^{2L-1}} \pi(x_1) \prod_{l=1}^{2L-2} p(x_l, x_{l+1}) \prod_{l=1}^{L} W(x_{l+s-1}, l)^\theta.$$

I use the recursive algorithm in [7] to compute $H_s(\theta)$. Let V be a $4 \times (2L-1)$ matrix, with $V(i,l) = W(i, l-s+1)^\theta$ for $l = s, \cdots, s+L-1$ and the rest of the entries set as 1. Then,

$$H_s(\theta) = \sum_{\mathbf{x}\in\{1,2,3,4\}^{2L-1}} \pi(x_1) \prod_{l=1}^{2L-2} p(x_l, x_{l+1}) \prod_{l=1}^{2L-1} V(x_l, l), \tag{3.10}$$

can be computed by the following recursion:

$$Q_s(i, 2L-1)) = V(i, 2L-1),\ 1 \leq i \leq 4; \tag{3.11}$$

$$Q_s(i,l) = V(i,l) \sum_{j=1}^{4} p(i,j) Q_s(j, l+1), \quad 1 \leq l \leq 2L-2,\ 1 \leq i \leq 4; \tag{3.12}$$

$$H_s(\theta) = \sum_{i=1}^{4} \pi(i) Q_s(i,1). \tag{3.13}$$

Finally, $(\mathbf{X}, S) \sim g_\theta$ can be simulated as follows:

$$P(S=s) = \frac{H_s(\theta)}{H(\theta)}; \tag{3.14}$$

$$P(X_1=x_1|S=s) = \frac{\pi(x_1)Q_s(x_1,1)}{H_s(\theta)}; \tag{3.15}$$

$$P(X_l=x_l|X_{l-1}=x_{l-1},S=s) = \frac{p(x_{l-1},x_l)Q_s(x_l,l)}{Q_s(x_{l-1},l-1)}, \quad 2\le l\le 2L-1. \tag{3.16}$$

3.2.3.2 Details for Tests of Change in Affinity Scores Between the Alleles

To compute $H^a(\theta)$, I first note that $H^a(\theta)=\sum_{s=1}^{L} H_s^a(\theta)$, where

$$\begin{aligned}
H_s^a(\theta) = & \sum_{\mathbf{x}\in\{1,2,3,4\}^{2L-1}} \Bigg\{ f(x_1,\cdots,x_{s-1}) f(x_{s+L},\cdots,x_{2L-1}) \\
& \Big[\prod_{1\le l\le L, l\ne L} IW(x_{l+s-1},l)\Big] D(x_L, L-s+1)^\theta \Bigg\} \\
= & \sum_{\{x_s,\cdots,x_{s+L-1}\}\in\{1,2,3,4\}^L} \Bigg\{ D(x_L,L-s+1)^\theta \\
& \Big[\prod_{1\le l\le L, l\ne L} IW(x_{l+s-1},l)\Big] \Bigg\} \\
= & \left\{ \prod_{1\le l\le L, l\ne L-s+1} \left[\sum_{i=1}^{4} IW(i,l)\right] \right\} \left[\sum_{i=1}^{4} D(i,L-s+1)^\theta\right] \\
= & \sum_{i=1}^{4} D(i,L-s+1)^\theta.
\end{aligned}$$

A sequence following g_θ^a can be simulated as follows.

$$P(S = s) = \frac{H_s^a(\theta)}{H^a(\theta)}, \tag{3.17}$$

$$P(X_l = x_l | S = s) = \pi(x_l), \tag{3.18}$$

$$\text{for} \quad l = 1, s + L, \tag{3.19}$$

$$P(X_l = x_l | S = s, X_{l-1} = x_{l-1}) = p(x_{l-1}, x_l) \tag{3.20}$$

$$\text{for} \quad l = 2, \cdots, s - 1, s + L + 1, \cdots, 2L - 1, \tag{3.21}$$

$$P(X_l = x_l) = IW(x_l, l - s + 1) \tag{3.22}$$

$$\text{for} \quad l = s, \cdots, L - 1, L + 1, \cdots, s + L - 1, \tag{3.23}$$

$$P(X_L = x_L) = \frac{D(x_L, L - s + 1)^\theta}{H_s^a(\theta)}. \tag{3.24}$$

3.3 Numerical Evaluations

In this section, I first compare the conditional p-values from atSNP with the p-values from FIMO ([17]) to evaluate the accuracy of atSNP p-values that are based on importance sampling. Next, I compare the results for the evaluation of the binding affinity changes from atSNP and is-rSNP. I then apply atSNP's between allele affinity score change test to a set of rSNPs with known SNP-TF interactions from the ORegAnno database ([18]). All the analysis are based on hg19 version of the human genome.

3.3.1 Comparison with FIMO

To assess the computation accuracy of atSNP, I compared atSNP's conditional p-values with FIMO's p-values using the set of 26,100 SNPs from the Psychiatric Genomics Consortium (`http://www.med.unc.edu/pgc`) and the ENCODE-derived PWM for an arbitrarily chosen TF ATF3 [2] ([30]). Figure 3.5(a) compares FIMO p-values of all SNPs with a p-value less than 1e-4 (default threshold of FIMO [3]) with the conditional p-values from atSNP and indicates that the two sets of p-values agree well. Furthermore, for the SNPs with FIMO p-values larger than 1e-4, conditional

[2] ATF3_GM12878_encode-Myers_seq_hsa_r1:MDscan#1#Intergenic.

[3] FIMO run without any thresholding did not complete within 24 hours.

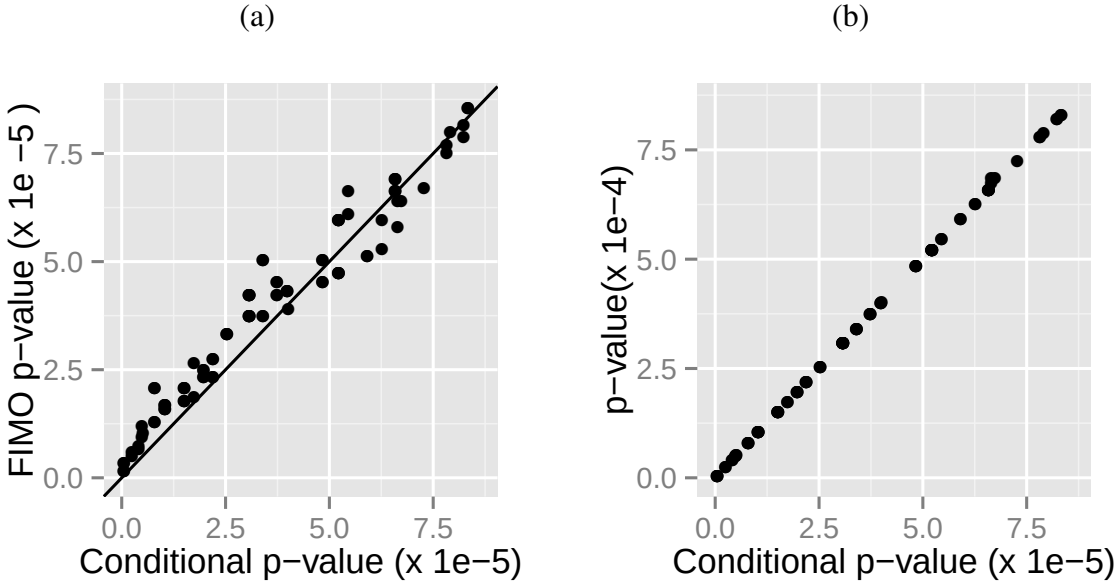


Figure 3.5 (a) Comparison between FIMO's p-values and atSNP's conditional p-values. (b) Comparison between atSNP's conditional p-values and p-values.

p-values from atSNP were also larger than 1e-4. This suggests that my importance sampling algorithm is indeed speeding up the computations without sacrificing accuracy. Similar conclusions are obtained when I utilize other TFs instead of ATF3. Because the allele-specific affinity tests are an intermediate step in is-rSNP and are not included in the output, I was not able to compare their results with my conditional p-values.

I also compared atSNP p-values with its conditional p-values in Figure 3.5(b). I observe that the difference between the two p-value types are quite apparent at large affinity score values.

3.3.2 Comparison with is-rSNP

I used comparison with FIMO as a way of validating the accuracy of my importance sampling algorithm. Next, I compared the tests for affinity score changes between atSNP and is-rSNP. Since is-rSNP does not support batch processing large SNP sets[4], I compared atSNP and is-rSNP using one SNP from Section 3.3.1, namely rs9909429, as a representative case and utilized the PWMs from the JASPAR database[5]. is-rSNP reported the p-values adjusted by the Benjamini-Hochberg procedure ([5]). As a comparison, I adopted the same procedure to adjust rank test p-values and thresholded the adjusted p-values at 0.05 for both methods.

Table 3.2 lists the 16 motifs identified by atSNP and/or is-rSNP. In this table, '$pval_r$-BH' reports the rank test p-values of atSNP adjusted by the Benjamini-Hochberg (BH) procedure ([5]) whereas '$pval_{adj}$' reports the BH adjusted p-values from is-rSNP. Five of these motifs are identified as significantly affected by the SNP by both atSNP and is-rSNP. atSNP and is-rSNP assigns different significance on the effect of the SNP for the other 11 motifs. These discrepancies can be attributed to multiple factors. First, is-rSNP seems to compute the affinity score using Eqn. (3.2) even when the entries of the PWM are in the form of nucleotide probabilities, while atSNP applies the definition in Eqn. (3.1), which corresponds to log likelihood of the sequence when the PWM is in the form of nucleotide probabilities. Second, is-rSNP evaluates the change in the binding affinity by first scoring the reference and SNP allele versions of the subsequences overlapping the SNP and

[4]All versions of is-rSNP (1.0 and 2.0) can only analyze at most 20 SNPs at a time

[5]I used the latest is-rSNP version 2.0. and found that is-rSNP uses 2010 freeze of the JASPAR database. In order to make my results comparable, I also used this version of the JASPAR database in Sections 3.3.2 and 3.3.3.

identifying the subsequence with the maximum affinity score, i.e., best match might be achieved with the reference or the SNP allele. Then, it compares the affinity scores of this subsequence with both the reference and the SNP allele. This approach overlooks the possibility that both the reference and SNP alleles may provide equally good matches to the PWM, albeit with subsequences starting at different positions. Figure 1 of the main text provides an illustrative example of this scenario. Here, is-rSNP chooses the subsequence starting at the 2nd position in the reference genome as the best match to the PWM. Then, it evaluates the binding affinity of this subsequence with the SNP allele and obtains a big change in the affinity score. However, as is visible from the logo, there is an almost equally good match to the PWM starting at the 1st position of the sequence with the SNP allele. Clustered degenerate binding sites are especially susceptible to these types of potential false positives ([81]). A third source of discrepancy is that is-rSNP assumes an independent multinomial model for the background distribution whereas atSNP accommodates dependency between consecutive positions motivated by the fact that modeling dependency between the positions of the background sequences improve motif detection ([68]).

I present the composite sequence logo plots comparing the reference and SNP alleles for all the commonly identified SNP-PWM pairs in Section A.1, for SNP-PWM pairs only identified by atSNP in Section A.2, and by is-rSNP in Section A.3. I observe that all the commonly identified motifs have very good matching subsequences with either the reference or the SNP allele, and the SNP significantly impacts the binding affinity. Motifs prioritized only by one method typically have a number of mismatches to the motif consensus in their best matching subsequence around the SNP in addition to the mismatch at the SNP position. Overall, motifs prioritized by atSNP seem to have slightly better matching subsequences to the motif with either the reference or the SNP allele. On average, the proportions of positions that do not agree with the motif consensus are 0.14 and 0.27 for atSNP and is-rSNP, respectively. These proportions are obtained by counting the mismatches between the best matching subsequences and the most likely consensus sequences from the PWMs by discarding the degenearte positions that are on either edges of the PWM.

I further observe that many of the significant PWMs are very similar to each other (e.g., CN0007.1, CN0002.1, MA0139.1, PF0045.1 are variants of Ctcf PWM) indicating that the hypotheses evaluated within the multiple testing framework are far from independent. This suggests that the classical multiple testing procedures adopted by FIMO and is-rSNP, i.e., Benjamini-Hochberg FDR procedure ([5]) and Storey's q-value ([64]), can be overly conservative. One possible remedy for this is to adopt group false discovery rate procedure proposed by [24]. However, its implementation requires additional considerations such as appropriate grouping within the PWM libraries. For these reasons, atSNP currently does not support a built-in multiple testing adjustment method. I suggest using the commonly adapted conservative procedures already available by R functions `stats::p.adjust` and `qvalue::qvalue`.

Motif	Motif Info	$pval_r$-BH	$pval_{adj}$
rSNPs identified by both atSNP and is-rSNP			
CN0007.1	LM7	2.4e-4	1.8e-6
CN0002.1	LM2	3.6e-4	4e-6
MA0139.1	CTCF	4.5e-4	2.1e-5
PF0045.1	CCANNAGRKGGC	1.6e-3	5.7e-5
MA0055.1	MYF	0.044	9.9e-4
rSNPs identified only by atSNP			
PF0057.1	ACCTGTTG	0	1
CN0023.1	LM23	5.8e-4	1
PL0011.1	HLH-2::HLH-4	1.9e-3	1
PL0002.1	HLH-2::HLH-3	2.1e-3	1
CN0146.1	LM146	0.0032	0.63
CN0047.1	LM47	4.4e-3	0.383
rSNPs identified only by is-rSNP			
MA0322.1	INO4	0.128	0.042
PL0017.1	HLH-2::HLH-10	0.22	5.3e-3
CN0049.1	LM49	0.252	9.4e-3
CN0194.1	LM194	0.267	8.1e-3
CN0169.1	LM169	0.482	0.028

Table 3.2 rSNP interactions of SNP rs9909429 identified by atSNP and is-rSNP.

ORegAnno-reported		Top motif in the ORegAnno-reported TF family				Top motif in the JASPAR library			Dist
SNP ID	TF	Motif	Motif Info	$pval_r$	Rank	Motif	Motif Info	$pval_r$	
rs2569190	SP FAMILY	PB0075.1	SP4	8.6e-06	2	MA0381.1	SKN7	0	0.4436
rs763110	CEBPB	MA0102.1	CEBPA	2.4e-04	4	MA0327.1	MATA1	0	0.3223
rs12720461	ETS	MA0062.2	GABPA	3.7e-04	4	MA0275.1	ASG1	0	0.4372
rs28095	SP1/SP3	MA0079.1	SP1	6.6e-04	4	POL010.1	DCE_S_III	0	0.2911
rs712829	SP1	PB0075.1	SP4	7.1e-04	4	MA0373.1	RPN4	2.0e-04	0.427
rs13434811	YY1	PB0097.1	ZFP410	7.4e-04	9	MA0035.2	GATA1	2.3e-05	0.3081
rs16998970	YY1	PB0097.1	ZFP410	7.6e-04	10	CN0095.1	LM95	3.7e-05	0.2976
rs1800775	SP1/SP3	PB0025.1	GLIS2	0.0012	2	PF0056.1	GGGTGGRR	7.9e-04	0.4139
rs243865	SP1	MA0039.2	KLF4	0.0018	4	PF0082.1	CTGYNNCTYTAA	6.2e-04	0.3789
rs934345	TP53	MA0106.1	TP53	0.0022	11	MA0217.1	CAUP	0	0.4177
rs2333227	SP1	PB0096.1	ZFP281	0.0027	15	PL0002.1	HLH-2::HLH-3	4.1e-04	0.4154
rs213045	E2F2	MA0024.1	E2F1	0.0028	5	MA0334.1	MET32	3.6e-04	0.3927
rs1800590	SP1/SP3	PB0051.1	PLAGL1	0.0029	8	MA0381.1	SKN7	0	0.425
rs1862513	SP1/SP3	PB0096.1	ZFP281	0.0031	8	MA0366.1	RGM1	0	0.4312
rs2838769	TP53	MA0106.1	TP53	0.0034	11	MA0233.1	MIRR	0	0.4013
rs27646	SP1	MA0163.1	PLAG1	0.0039	12	MA0410.1	UGA3	0	0.3745
rs2227306	CEBPB	MA0102.2	CEBPA	0.0133	16	PF0172.1	TTGCWCAAY	0.0023	0.2776
rs1658728	TP53	MA0106.1	TP53	0.0164	41	PB0040.1	MAFB	4.4e-04	0.4148
rs2251746	GATA1	MA0037.1	GATA3	0.0259	35	MA0408.1	TOS8	9.4e-05	0.3306
rs2279744	SP1	MA0146.1	ZFX	0.0316	38	MA0185.1	DEAF1	0	0.3567
rs3761624	TP53	MA0106.1	TP53	0.0687	99	PF0106.1	CCGNMNNTNACG	3.3e-04	0.3426
rs268682	TP53	MA0106.1	TP53	0.0812	66	MA0260.1	CHE-1	0	0.4267
rs2232945	TP53	MA0106.1	TP53	0.0846	110	PF0134.1	CATRRAGC	0	0.3478
rs11836625	CREB1	MA0414.1	XBP1	0.1724	185	MA0130.1	ZNF354C	0	0.2894

Table 3.3 Affinity score change tests for the curated rSNP-TF pairs in the ORegAnno database ([18]) by atSNP.

3.3.3 Validation Using Known rSNP-TF Interactions

The ORegAnno database ([18]) lists 36 known rSNP-TF pairs. I analyzed each SNP with the PWMs of the JASPAR motifs ([44]) that came from the TF family of each TF in the rSNP-TF pair. I used the TF families defined as the homolog clusters according to ([13]). For 9 of the SNPs, neither the reference nor the SNP allele matched the allele listed at the SNP location in hg19 version of the genome and, hence, I discarded these from the analysis. I further discarded 3 rSNPs for which the associated TF families were not included among the JASPAR motifs. For each of the remaining 24 SNPs, I ran atSNP against the JASPAR PWMs to identify the motif with the most significant regulatory effect both among the whole set of motifs and among motifs from the associated TF family (Table 3.3). In Table 3.3, Columns 3-6 correspond to the top significant motif in the ORegAnno-reported TF family, while Columns 7-9 correspond to the top significant motif among all the 1192 motifs in the JASPAR database. 'Rank' is the rank of $pval_r$ for the top motif in the TF family across the whole motif library. 'Dist' is the L^2 distance between the top motifs in the TF family and in the whole library, normalized by the matrix size [6]. atSNP successfully identified motifs from the ORegAnno-reported TF family for 20 of the SNPs based on the rank test p-value (at significance level of 0.05).

Table 3.3 shows the ranks for the top motifs from the ORegAnno-reported TF family among the entire set of JASPAR motifs. Because I evaluated the SNPs against the whole set of JASPAR motifs, I was able to calculate the ranking of the motifs from ORegAnno-reported TF families. For all the known 20 rSNPs, these motifs fall in the top 5% among the 1192 JASPAR motifs. However, none of the top ranked motifs is from the PWMs of the reported TF family. This does not indicate that atSNP results are inconsistent with the ORegAnno database; it is possible that these more significant SNP-TF interactions may not have been studied experimentally or cannot be further discriminated based on sequence alone. For example, for the rs2251746-GATA1 pair, the most significant score change within the GATA family is obtained with GATA3 PWM and ranked as the 35th most significant change among the whole set of JASPAR motifs (Figure A.35(a)); however,

[6] R function implementing this distance is available at `http://www.stat.wisc.edu/~keles/Software/motif_distance.R`. When the two PWM have different sizes, 'Dist' is based on the submatrix of the larger PWM that minimizes the distance to the smaller PWM.

SNP ID	ORegAnno-reported TF	Top motif in the ORegAnno-reported TF family			
		Motif	Motif Info	p-value	Rank
rs28095	SP1	MA0079.1	SP1	0	1
rs934345	TP53	MA0106.1	TP53	0	4
rs1800775	Glis2_1	PB0025.1	GLIS2	0	3
rs27646	Zfx	MA0146.1	ZFX	0	4
rs2569190	PLAG1	MA0163.1	PLAG1	1e-04	4
rs1862513	Osr2_1	PB0051.1	PLAGL1	1e-04	5
rs2333227	Zfp281_1	PB0097.1	ZFP410	1e-04	17
rs213045	E2F1	MA0024.1	E2F1	2e-04	5
rs243865	Egr1_1	PB0010.1	EGR1	2e-04	9
rs268682	TP53	MA0106.1	TP53	2e-04	7
rs13434811	Zfp410_1	PB0098.1	ZFP691	3e-04	20
rs2279744	PLAG1	MA0163.1	PLAG1	3e-04	14
rs2227306	Cebpa	MA0102.1	CEBPA	4e-04	20
rs1800590	Hic1_1	PB0029.1	HIC1	7e-04	21
rs12720461	ELF5	MA0136.1	ELF5	8e-04	28
rs712829	Zfp281_1	PB0097.1	ZFP410	9e-04	22
rs2251746	GATA2	MA0036.1	GATA2	0.0013	46
rs1658728	TP53	MA0106.1	TP53	0.0013	64
rs16998970	Zfp410_1	PB0098.1	ZFP691	0.0013	77
rs763110	XBP1	MA0414.1	XBP1	0.0038	181
rs2838769	TP53	MA0106.1	TP53	0.0059	204
rs2232945	TP53	MA0106.1	TP53	0.0065	221
rs11836625	CREB1	MA0018.2	CREB1	0.0066	161
rs3761624	TP53	MA0106.1	TP53	0.0095	290

Table 3.4 Affinity score change tests for the curated rSNP-TF pairs in the ORegAnno database ([18]) using is-rSNP.

when rs2251746 is evaluated against the whole set of PWMs in the JASPAR library, PWM for TOS8 is reported as exhibiting the most significant change in the affinity score (Figure A.35(b)). When I visualize the sequence logo plots for these two PWMs, I observe that both changes seem significant, and rs2251746 is disrupting a match to the longer TOS8 motif.

atSNP provides a way to prioritize putative SNP-TF interactions and these interactions can further be filtered by other functional data such as ChIP-seq data of transcription factors from ENCODE or other consortia projects. I display the composite sequence logo plots for the SNP-TF pairs in Table 3.3. These plots directly illustrate how each SNP affects the binding pattern of the corresponding motif. For each SNP, the top motif in the library always has an almost perfect match to a sequence around the SNP location, while the SNP location is matched to a nucleotide that significantly changes the affinity score. Such patterns indicate strong *in silico* evidence for the regulatory effects.

I also analyzed these set of SNPs with is-rSNP (Table 3.4). In this table, 'Rank' is the rank of 'p-value' for the top motif in the TF family across the whole motif library. Figure 3.6 displays the atSNP and is-rSNP ranks of the top motifs in the ORegAnno-reported TF family for each SNP across all the JASPAR PWMs. Overall, the median rank of the highest ranked motifs in the ORegAnno-reported TF family is 10.5 for atSNP and 20 for is-rSNP across all the JASPAR PWMs.

3.3.4 Run-time Comparisons

Table 3.5 presents an illustrative summary of run time comparisons.

3.4 Conclusions and Discussion

Genome-wide association studies revealed that most disease-associated single nucleotide polymoprhisms (SNPs) are located in regulatory regions within introns or in regions between genes. Regulatory SNPs (rSNPs) are such SNPs that affect gene regulation by changing transcription factor (TF) binding affinities to genomic sequences. Identifying potential rSNPs is crucial for understanding disease mechanisms. *In silico* methods that evaluate the impact of SNPs on TF binding affinities are not scalable for large-scale analysis.

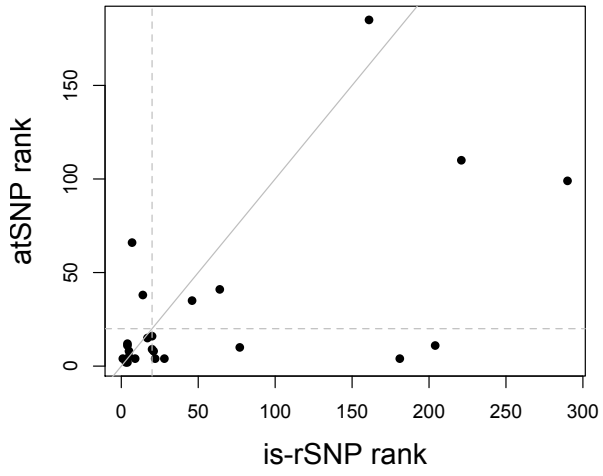


Figure 3.6 atSNP and is-rSNP ranks of the top motifs in the ORegAnno-reported TF family for each SNP across all the JASPAR PWMs. Horizontal and vertical dashed lines mark rank 20.

Method	# of SNPs	# of PWMs	# of cores	Total time	Time for reading in data	Time for writing data
atSNP	26,100	1	1	3m8s	26s	41s
atSNP	26,100	10	10	7m15s	25s	3m13s
atSNP	26,100	10	1	23m4s	25s	3m13s
FIMO	26,100	10	1	2h30m*		
atSNP	500	2,065	30	2h2m	2s	25s
atSNP	26,100	2,065	30	5h48m	26s	18m35s

Table 3.5 Run time evaluations of atSNP. *: only outputs results with p-value ≤ 0.1.

I describe atSNP (**a**ffinity **t**esting for regulatory **SNP**s), a computationally efficient `R` package for identifying rSNPs *in silico*. atSNP implements an importance sampling algorithm coupled with a first-order Markov model for the background nucleotide sequences to test the significance of affinity scores and SNP-driven changes in these scores. Application of atSNP with $>$20K SNPs indicates that atSNP is the only available tool for such a large-scale task. atSNP provides user-friendly output in the form of both tables and composite logo plots for visualizing SNP-motif interactions. Evaluations of atSNP with known rSNP-TF interactions indicates that rSNP is able to prioritize motifs for a given set of SNPs with high accuracy.

Chapter 4

A Hierarchical Framework for State-Space Matrix Inference and Clustering[1]

4.1 Introduction

This chapter is motivated by a number of genomic and epigenomic studies that aim to elucidate genome regulatory mechanisms across multiple biological conditions. A large number and wide variety of experiments are performed on different organisms to study multiple aspects of genome regulation. Some examples of data types from these experiments include transcription factor occupancy, gene expression, methylation, and histone modification data. Computational and statistical analysis of these data often involve identifying genomic loci that show significant signal, i.e., enrichment, compared to background noise in the experimental measurements.

Improvements in the next-generation sequencing technology further accelerated rapid generation of these types of data. In return, the vast availability of such data has revolutionized the scope of genome regulation studies. Previous analyses had been restricted to detecting regions of genome that were associated with one particular factor in one single organism. Many recent studies focus on detecting more complex functional patterns that integrate data from multiple organisms under multiple conditions. Namely, the associations between DNA elements and how they change across biological and/or experimental conditions have been the focus of many integrative modeling approaches. Examples of these studies include:

[1]The manuscript for this chapter [84] is submitted to the Annals of Applied Statistics. Method in this chapter is implemented in the `R` package `MBASIC` and is freely available at `http://github.com/chandlerzuo/mbasic`.

Differential binding analysis among multiple ChIP-seq data. Gene expression is, to a large extent, regulated by the differential activities of transcription factors and epigenetic modifications. Currently, chromatin immunoprecipitation followed by high throughput sequencing (ChIP-seq) is the state-of-the-art method for profiling transcription factor occupancy and histone modifications genome-wide. The study of gene regulation often involves comparing transcription factor occupancy and histone modifications across multiple biological conditions. Such conditions can be different treatment levels, time points of measurements, or different dosage levels ([39], [2], [26], [73]).

Transcription factor regulatory network analysis. The combinatorial nature of transcription factor regulation underlies the large diversity observed in eukaryotic gene control. The large-scale data from the ENCODE project ([67]) now enable joint analyses of over one hundred human transcription factors across multiple cell types. Such analyses are posed to reveal a great amount of information about co-association patterns between different TFs, hierarchical network organizations, and systems-level integration of complex cellular signals ([49], [15], [9], [80]). While the large number of TFs makes it computationally formidable to exhaust all possible combinatorial associations for such analyses, it is important to detect the most significant combinatorial patterns that preserve global regulatory dynamics.

Comparative functional genomic studies across different species. Functional genomics analysis compares gene expressions or TF occupancy profiles between multiple species. The main task is to identify divergent and conserved functional modules that are central to evolutionary relationships (e.g., [35], [61]). Existing methods, that build on hidden Markov models ([59]) or biclustering ([70]), implicitly assume that the functional modules should at least have similar signal profiles (i.e., expression, occupancy) among some subsets of the species under consideration. For these analyses, it is also important to identify functional modules that are fully divergent across species. These regions play an equally important role in understanding connectivity among species over the evolutionary history.

Although the types of data for these different studies vary, the underlying statistical principles are largely shared. Therefore, I propose a unified framework for the analysis of such data by formalizing the shared aspects. I formulate the underlying statistical problem as follows. Suppose a dataset $\{Y_{ik}\}$ is collected over a set of observational units (e.g., loci in genomic experiments) $i = 1, 2, \cdots, I$ under conditions $k = 1, 2, \cdots, K$. Inferring the association patterns within a single experiment involves mapping the corresponding set of observations $\{Y_{ik} : i = 1, 2, \cdots, I\}$ to a finite discrete state-space, $\mathscr{S} = \{1, 2, \cdots, S\}$. This space contains different levels of association (e.g., enrichment/non-enrichment indicating the status of occupancy in ChIP-seq experiments, expressed/not expressed in RNA-seq gene expression experiments). This falls under the classical finite-mixture modeling framework, where a latent state variable $\theta_{ik} \in \mathscr{S}$ is inferred for each observational unit Y_{ik}. A higher level of modeling on the matrix $\Theta = (\theta_{ik})_{1\leq i\leq I, 1\leq k\leq K}$ is required for integrating the association patterns under different conditions. I call this matrix the *state-space matrix* since it describes the latent states of individual observations.

I propose the following framework to model the state-space matrix Θ. I assume that rows of Θ can be partitioned into $J+1$ subsets: $\{1, \cdots, I\} = \mathscr{C}_0 \cup \mathscr{C}_1 \cup \cdots \cup \mathscr{C}_J$. Rows of Θ within partition $\mathscr{C}_j$, $j \geq 1$, are generated by the same distribution parametrized by $w_{j\cdot} = (w_{jk})_{1\leq k\leq K}$:

$$\theta_{ik} \sim g(\cdot|w_{jk}), \quad i \in \mathscr{C}_j,$$

while the rows of $\mathscr{C}_0$, which denotes the group of "singleton" units, i.e., units that do not cluster in any of the J groups, are generated by row specific distributions. The goal of this model is thus to estimate a partitioning that best characterizes the row associations of state-space matrix Θ.

I refer to the proposed framework as **M**atrix **B**ased **A**nalysis for **S**tate-space **I**nference and **C**lustering (**MBASIC**). MBASIC is related to classical factor analysis which considers the problem of projecting one dimension (either row or column) of large noisy matrices into low-dimensional spaces. MBASIC has two distinguished features compared to the existing literature in these areas. First, MBASIC deals with matrices with discrete entries, while most existing methods are designed for matrices on continuous scales. Second, MBASIC estimates the low-dimensional projection by grouping the rows of the original matrix as compared to the Principle Component Analysis (PCA) approaches (e.g., [26], [37]). This is motivated by the following arguments:

1. In MBASIC, each factor estimate $w_{j\cdot}$ characterizes the commonality of a group of rows and is easily interpretable in practice. Such interpretability can further be enhanced by imposing structural restrictions on the $w_{j\cdot}$ vector for practical purposes. Examples of such constraints are described in Section 4.3.3;

2. PCA for high dimensional matrices are often accompanied by regularization techniques, which are computationally prohibitive for many epigenetic datasets. In contrast, clustering the matrix rows can be implemented very efficiently and in a straightforward manner.

The hierarchical structure of MBASIC is similar to two other recently proposed statistical models: iASeq [72] and Cormotif [73]. Both these models incorporate a state-space clustering structure similar to MBASIC. MBASIC extends these models in several critically essential directions. First, MBASIC is developed for general purposes and can be easily implemented for a wide range of parametric distributions, while Cormotif and iASeq operate with specific distributions targeting the problems of differential expression and allele-specific binding. Second, neither of these models include a group of singletons with idiosyncratic state-space profiles. When I am agnostic about the "true" clustering structure in applications, separating the singletons can reduce their influence on the estimation of clustering parameters. Third, both iASeq and Cormotif separate estimation for the distributional parameters from the clustering structure, while MBASIC can jointly fit all model parameters. One caveat for MBASIC compared to these models is that MBASIC does not allow the distributional parameters within the same state to be heterogeneous. To analyze such data, a preprocessing step that accounts for the the heterogeneity can be a possible remedy. I evaluate and discuss all of these features with extensive simulation studies in this chapter.

This chapter is organized as follows. I start with a formal description of MBASIC in Section 5.2, followed by model estimation and selection methods in Section 4.3. I also investigate general features of MBASIC compared to iASeq and Cormotif with extensive simulations in this section. Section 4.5 presents results from several real data examples.

4.2 The Hierarchical Mixture Model Framework

Consider a dataset with observations from I different *observational units* under K different *conditions*. For each condition $k \in \{1, 2, \cdots, K\}$, there are n_k replicate experiments, indexed by $l = 1, 2, \cdots, n_k$. I use Y_{ikl} to denote the observation for the l-th replicate of unit i under condition k. For each condition k at unit i, there exists a hidden state variable $\theta_{ik} \in \mathscr{S} = \{1, 2, \cdots, S\}$. The MBASIC model consists of the following components:

1. **State-space Mapping**:
$$Y_{ikl}|\theta_{ik} = s \overset{ind.}{\sim} f_s(\cdot|\mu_{kls}, \sigma_{kls}, \gamma_{ikls}). \tag{4.1}$$

2. **State-space Clustering**: θ_{ik}'s are independently sampled from $\mathscr{S}$ with the sampling probability:
$$P(\theta_{ik} = s) = \zeta p_{is} + (1 - \zeta) \sum_{j=1}^{J} \pi_j w_{jks}. \tag{4.2}$$

In (4.1), μ_{kls} and σ_{kls} are the parameters related to the mean and dispersion for the s-th state for replicate l under condition k, and γ_{ikls} is the covariate encoding known information for unit i. In (5.2), p_{is}, ζ, π_j, and w_{jks} are additional non-negative parameters subject to restrictions:

$$0 \leq \zeta \leq 1; \quad \sum_{j=1}^{J} \pi_j = 1; \quad \sum_{s=1}^{S} w_{jks} = 1, \forall j, k; \quad \sum_{s=1}^{S} p_{is} = 1, \forall i.$$

I further discuss these parameters in Section 4.2.2.

4.2.1 State-space Mapping

Equation 4.1 partitions observational units $i = 1, \cdots, I$ into S subsets according to their hidden states. Within the same replicate, data from the same hidden state follow the same distribution $f_s(\cdot|\mu_{kls}, \sigma_{kls}, \gamma_{ikls})$. MBASIC assumes that the hidden states θ_{ik}'s are independent of the replicate index l, which means all replicates under the same condition have the same set of hidden states. However, distributional parameters for a given state can be different among replicates. Such a setting allows for the flexibility of modeling the heterogeneity in replicate experiments.

The density function f can be from an arbitrary parametric distribution. I consider three fundamental families of distributions commonly used for genomic data analysis:

- *Log-normal Distribution.* $LN(\mu_{kls}\gamma_{ikls}, \sigma_{kls})$ with a density function:

$$f_s(y|\mu_{kls}, \sigma_{kls}, \gamma_{ikls}) = \frac{1}{\sqrt{2\pi}\sigma_{kls}} \exp\left\{-\frac{(\log(y+1) - \mu_{kls}\gamma_{ikls})^2}{2\sigma_{kls}^2}\right\}. \tag{4.3}$$

- *Negative Binomial Distribution.* $NB(\mu_{kls}\gamma_{ikls}, \sigma_{kls})$ with a density function:

$$f_s(y|\mu_{kls}, \sigma_{kls}, \gamma_{ikls}) = \frac{\Gamma(y+\sigma_{kls})}{\Gamma(\sigma_{kls})\Gamma(y)} \frac{(\mu_{kls}\gamma_{ikls})^y \sigma_{kls}^{\sigma_{kls}}}{(\mu_{kls}\gamma_{ikls} + \sigma_{kls})^{y+\sigma_{kls}}}. \tag{4.4}$$

- *Binomial Distribution.* $Binom(\gamma_{ikls}, \mu_{kls})$ with a density function:

$$f_s(y|\mu_{kls}, \gamma_{ikls}) = \binom{\gamma_{ikls}}{y} \mu_{kls}^y (1-\mu_{kls})^{\gamma_{ikls}-y}. \tag{4.5}$$

In these three examples, γ_{ikls} represents the known heterogeneity across loci, while μ_{kls} and σ_{kls} are parameters to be estimated. For example, when using Eqn. (4.3) or (4.4) in a ChIP-seq analysis with $S = 2$ states, we can estimate γ_{ikl1} using data from the control samples so that the ChIP sample read counts scale with the control sample data, and assume $\gamma_{ikl2} = 1$ for the enriched states. Eqn. (4.5) can be used to analyzie allele-specific binding data, where γ_{ikls} is the total read counts from both paternal and maternal alleles and is uniform across s. I require for model identification that for each k and l, $\mu_{kls}\sum_{i=1}^{I}\gamma_{ikls}$ is strictly increasing in s.

The MBASIC can be easily extended to other classes of parametric distributions and estimation for these distributions follows the same Expectation-Maximization skeleton. While Section 4.3 relies on these three distributions to describe the model and the estimation algorithms, the second real data example in Section 4.5 utilizes a more complex parametrization, which demonstrates the wide applicability of the MBASIC framework. Furthermore, I consider the following degenerate distribution:

$$f_s(y|\mu_{kls}, \sigma_{kls}, \gamma_{ikls}) = I(y = s), \tag{4.6}$$

where I(.) denotes the indicator function. This degenerate form corresponds to the situation where the states, θ_{ik}'s, are directly observed rather than inferred from Y_{ikl}'s. I utilize this parametrization

for comparing MBASIC to alternative two-step analysis approaches in Section 4.4. Parameter estimation for this case follows a slightly modified procedure from the non-degenerate cases, which is described in Section 4.3.

4.2.2 State-space Clustering

Equation (5.2) models the distribution of θ_{ik} as a mixture of multiple distributions. To illustrate this model I introduce additional variables. The goal is to identify J clusters from the set of observation units $1 \leq i \leq I$. Let $b_i = I(\text{unit } i \text{ does not belong to any cluster})$ and $z_{ij} = I(\text{unit } i \text{ belongs to cluster } j)$. The b_i variables entertain the possibility that some observations are "singletons", i.e., they do not cluster with any other observational units. With these additional variables, the distribution in Equation (5.2) can be hierarchically decomposed as follows:

- $b_i \overset{\text{i.i.d.}}{\sim} \text{Bernoulli}(\zeta)$;
- $(z_{i1},\ z_{i2}, \cdots,\ z_{iJ}) \overset{\text{i.i.d.}}{\sim} \text{MultiNom}(1, (\pi_1,\ \pi_2, \cdots,\ \pi_J))$;
- Conditional on b_i and z_{ij}, θ_{ik}'s are independent samples from $\mathscr{S}$, with sampling probabilities $P(\theta_{ik} = s|b_i = 1) = p_{is}$, $P(\theta_{ik} = s|b_i = 0, z_{ij} = 1) = w_{jks}$.

It is worth noting that this hierarchical structure essentially seeks a low-rank representation for the matrix $\Theta = (\theta_{ik})_{1\leq i\leq I, 1\leq k\leq K}$. To illustrate this, I introduce additional matrices $\Theta_s = (I(\theta_{ik} = s))_{1\leq i\leq I, 1\leq k\leq K}$, $W_s = (w_{jks})_{1\leq j\leq J, 1\leq k\leq K}$, $Z = (z_{ij})_{1\leq i\leq I, 1\leq j\leq J}$ and vectors $p_s = (p_{is})_{1\leq i\leq I}$, $B = (b_i)_{1\leq i\leq I}$. Then, the conditional expectation of Θ_s is:

$$E(\Theta_s|Z, B) = (ZW_s) \circ ((1 - B)1_K^T) + (p_s \circ B)1_K^T, \tag{4.7}$$

where "$\circ$" denotes the Hadamard product. I note that $E(\Theta_s|Z, B)$ is a matrix of rank $J + 1$, which is usually much smaller than the dimension of the matrix Θ_s. Similar models for low-rank representation of discrete matrices were considered in [37], and turned out to be challenging both theoretically and computationally. The row-clustering structure for the matrices $E(\Theta_s|Z, B)$ in MBASIC is more restrictive than the general low-rank structure. Such additional restrictions not

only reduce the difficulty in parameter estimation but also enable the flexibility in many useful ways. For example, while [37] can only estimate one matrix at a time and thus is only applicable when $S = 2$, MBASIC can be applied to arbitrary values of S.

4.3 Model Estimation and Selection

4.3.1 Likelihood Functions

In the MBASIC model, the likelihood function for both the observed random variables Y_{ikl}'s and the unobserved θ_{ik}'s, z_{ij}'s, b_i's, i.e., full data likelihood, is given by:

$$\begin{aligned} &l(\mu, \sigma, \pi, p, \zeta, w; y, \theta, z, b) = \\ &\prod_{i=1}^{I} \zeta^{b_i} (1-\zeta)^{1-b_i} \cdot \prod_{i=1}^{I} \prod_{k=1}^{K} \prod_{s=1}^{S} p_{is}^{I(\theta_{ik}=s) b_i} \cdot \prod_{i=1}^{I} \prod_{j=1}^{J} \pi_j^{z_{ij}} \\ &\cdot \prod_{i=1}^{I} \prod_{k=1}^{K} \prod_{s=1}^{S} \left[\prod_{l=1}^{n_k} f_s(y_{ikl} | \mu_{kls}, \sigma_{kls}, \gamma_{ikls}) \right]^{I(\theta_{ik}=s)} \cdot \prod_{i=1}^{I} \prod_{j=1}^{J} \prod_{k=1}^{K} \prod_{s=1}^{S} w_{jks}^{I(\theta_{ik}=s)(1-b_i) z_{ij}}. \end{aligned} \tag{4.8}$$

For non-degenerate distributions, I can show that the marginal likelihood is:

$$\begin{aligned} l(\mu, \sigma, \pi, p, \zeta, w; y) = \prod_{i=1}^{I} \Bigg\{ &\zeta \prod_{k=1}^{K} \left[\sum_{s=1}^{S} p_{is} \prod_{l=1}^{n_k} f_s(y_{ikl} | \mu_{kls}, \sigma_{kls}, \gamma_{ikls}) \right] \\ &+ (1-\zeta) \sum_{j=1}^{J} \pi_j \prod_{k=1}^{K} \left[\sum_{s=1}^{S} w_{jks} \prod_{l=1}^{n_k} f_s(y_{ikl} | \mu_{kls}, \sigma_{kls}, \gamma_{ikls}) \right] \Bigg\}. \end{aligned} \tag{4.9}$$

Equation (4.9) is easily interpretable. Conditional on b_i and z_{ij}, the joint distribution for each $Y_{ikl},\ 1 \le l \le n_k$ is a mixture of S components, where the weight on the s-th component is either p_{is} (when $b_i = 1$) or w_{jks} (when $b_i = 0$ and $z_{ij} = 1$). This yields the expressions in the square brackets. Integrating out b_i and z_{ij}, the joint distribution for Y_{ikl} of fixed i is a mixture of $J + 1$ components, with probability ζ of being a singleton and probability $(1 - \zeta)\pi_j$ of belonging to cluster j.

For the degenerate case, by substituting (4.6) into (4.9), it can be shown that the marginal likelihood is:

$$l(\mu,\sigma,\pi,p,\zeta,w;\theta)=\prod_{i=1}^{I}\left\{\zeta\prod_{k=1}^{K}\prod_{s=1}^{S}p_{is}^{I(\theta_{ik}=s)}+(1-\zeta)\sum_{j=1}^{J}\pi_j\prod_{k=1}^{K}\prod_{s=1}^{S}w_{jks}^{I(\theta_{ik}=s)}\right\}. \tag{4.10}$$

4.3.2 An Expectation and Maximization (E-M) Algorithm

The hierarchical structure of MBASIC naturally fits in the Expectation-Maximization algorithm [10], which maximizes the marginal likelihood (equations (4.9) or (4.10)) by iteratively maximizing the complete data log-likelihood function. I let ϕ to denote a vector including all unknown parameters μ, σ, π, p, ζ, w, and $\hat{\phi}^{(t)}$ to denote the parameter estimates at the t-th iteration. The complete data log-likelihood function is:

$$\begin{aligned}
Q(\phi|\hat{\phi}^{(t-1)}) =& \sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{s=1}^{S}\left[\sum_{l=1}^{n_k}\log f_s(y_{ikl}|\mu_{kls},\sigma_{kls},\gamma_{ikls})\right]E[I(\theta_{ik}=s)|\hat{\phi}^{(t-1)}]\\
&+\sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{s=1}^{S}\log p_{is}E[I(\theta_{ik}=s)b_i|\hat{\phi}^{(t-1)}]+\sum_{i=1}^{I}\sum_{j=1}^{J}\log\pi_j E[z_{ij}(1-b_i)|\hat{\phi}^{(t-1)}]\\
&+\sum_{i=1}^{I}\{\log\zeta E[b_i|\hat{\phi}^{(t-1)}]+\log(1-\zeta)(1-E[b_i|\hat{\phi}^{(t-1)}])\}\\
&+\sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{j=1}^{J}\sum_{s=1}^{S}E[I(\theta_{ik}=s)z_{ij}(1-b_i)|\hat{\phi}^{(t-1)}]\log w_{jks}
\end{aligned} \tag{4.11}$$

The E-M algorithm for MBASIC is outlined by Algorithm 4.1. E-step updates are listed in equations (4.12)-(4.15) and their derivations are provided in Section 4.3.5.1.

$$E(b_i|\hat{\phi}^{(t-1)})=\frac{\hat{\zeta}^{(t-1)}\prod_{k=1}^{K}\left(\sum_{s=1}^{S}\hat{f}_{iks}^{(t-1)}\hat{p}_{is}^{(t-1)}\right)}{(1-\hat{\zeta}^{(t-1)})\sum_{j=1}^{J}\hat{\pi}_j^{(t-1)}\prod_{k=1}^{K}\left(\sum_{s=1}^{S}\hat{f}_{iks}^{(t-1)}\hat{w}_{jks}^{(t-1)}\right)+\hat{\zeta}^{(t-1)}\prod_{k=1}^{K}\left(\sum_{s=1}^{S}\hat{f}_{iks}^{(t-1)}\hat{p}_{is}^{(t-1)}\right)}, \tag{4.12}$$

Algorithm 4.1 Expectation-Maximization (EM)

for $t = 1,\ 2, \cdots$ until convergence **do**

Expectation Step: Compute the conditional expectations $E[I(\theta_{ik} = s)|\hat{\phi}^{(t-1)}]$, $E[b_i|\hat{\phi}^{(t-1)}]$, $E[I(\theta_{ik} = s)b_i|\hat{\phi}^{(t-1)}]$, $E[z_{ij}(1-b_i)|\hat{\phi}^{(t-1)}]$, $E[I(\theta_{ik} = s)z_{ij}(1-b_i)|\hat{\phi}^{(t-1)}]$;

Maximization Step: Update estimates for parameters μ_{kls}, σ_{kls}, ζ, π_j, w_{jks}, p_{is} as maximizers for (4.11) .

end for

$$E(z_{ij}(1-b_i)|\hat{\phi}^{(t-1)}) = \frac{\hat{\pi}_j^{(t-1)} \prod_{k=1}^{K} \left(\sum_{s=1}^{S} \hat{f}_{iks}^{(t-1)} \hat{w}_{jks}^{(t-1)}\right)}{\sum_{j=1}^{J} \hat{\pi}_j^{(t-1)} \prod_{k=1}^{K} \left(\sum_{s=1}^{S} \hat{f}_{iks}^{(t-1)} \hat{w}_{jks}^{(t-1)}\right)} [1 - E(b_i|\hat{\phi}^{(t-1)})], \tag{4.13}$$

$$\begin{aligned} &E(I(\theta_{ik} = s)z_{ij}(1-b_i)|\hat{\phi}^{(t-1)}) = \\ &[1 - E(b_i|\hat{\phi}^{(t-1)})] \frac{\hat{\pi}_j^{(t-1)} \prod_{k=1}^{K} \left(\sum_{s=1}^{S} \hat{f}_{iks}^{(t-1)} \hat{w}_{jks}^{(t-1)}\right)}{\sum_{j=1}^{J} \hat{\pi}_j^{(t-1)} \prod_{k=1}^{K} \left(\sum_{s=1}^{S} \hat{f}_{iks}^{(t-1)} \hat{w}_{jks}^{(t-1)}\right)} \cdot \frac{\hat{f}_{iks}^{(t-1)} \hat{w}_{jks}^{(t-1)}}{\sum_{s=1}^{S} \hat{f}_{iks}^{(t-1)} \hat{w}_{jks}^{(t-1)}}, \end{aligned} \tag{4.14}$$

$$E(I(\theta_{ik} = s)b_i|\hat{\phi}^{(t-1)}) = E(b_i|\hat{\phi}^{(t-1)}) \cdot \frac{\hat{f}_{iks}^{(t-1)} \hat{p}_{is}^{(t-1)}}{\sum_{s=1}^{S} \hat{f}_{iks}^{(t-1)} \hat{p}_{is}^{(t-1)}}, \tag{4.15}$$

where $\hat{f}_{iks}^{(t-1)} = \prod_{l=1}^{n_k} f(y_{ikl}|\hat{\mu}_{kls}^{(t-1)}, \hat{\sigma}_{kls}^{(t-1)}, \gamma_{ikls})$. Given these results from the E-step, updates of ζ, π_j, w_{jks}, p_{is} in the M-step are straight forward as in equations (4.16), (4.17), (4.18), and (4.19).

$$\hat{\zeta}^{(t)} = \frac{\sum_{i=1}^{I} E[b_i|\hat{\phi}^{(t-1)}]}{I}, \tag{4.16}$$

$$\hat{\pi}_j^{(t)} = \frac{\sum_{i=1}^{I} E[z_{ij}(1-b_i)|\hat{\phi}^{(t-1)}]}{\sum_{i=1}^{I} \sum_{j=1}^{J} E[z_{ij}(1-b_i)|\hat{\phi}^{(t-1)}]}, \tag{4.17}$$

$$\hat{p}_{is}^{(t)} = \frac{\sum_{k=1}^{K} E[I(\theta_{ik} = s)b_i|\hat{\phi}^{(t-1)}]}{\sum_{s=1}^{S} \sum_{k=1}^{K} E[I(\theta_{ik} = s)b_i|\hat{\phi}^{(t-1)}]}, \tag{4.18}$$

$$\hat{w}_{jks}^{(t)} = \frac{\sum_{i=1}^{I} E[I(\theta_{ik} = s)z_{ij}(1-b_i)|\hat{\phi}^{(t-1)}]}{\sum_{s=1}^{S} \sum_{i=1}^{I} E[I(\theta_{ik} = s)z_{ij}(1-b_i)|\hat{\phi}^{(t-1)}]}. \tag{4.19}$$

Updates for μ_{kls} and σ_{kls} have to be treated according to the specific distributions. For the log-normal distributions (4.3), I have:

$$\hat{\mu}_{kls}^{(t)} = \frac{\sum_{i=1}^{I} \log(y_{ikl}+1) P[\theta_{ik}=s|\hat{\phi}^{(t-1)}]}{\sum_{i=1}^{I} \gamma_{ikls} P[\theta_{ik}=s|\hat{\phi}^{(t-1)}]}, \tag{4.20}$$

$$\hat{\sigma}_{kls}^{(t)2} = \frac{\sum_{i=1}^{I} P[\theta_{ik}=s|\hat{\phi}^{(t-1)}][\log(y_{ikl}+1) - \hat{\mu}_{kls}^{(t)}\gamma_{ikls}]^2}{\sum_{i=1}^{I} P[\theta_{ik}=s|\hat{\phi}^{(t-1)}]}. \tag{4.21}$$

For the binomial distributions (4.5), I have:

$$\hat{\mu}_{kls}^{(t)} = \frac{\sum_{i=1}^{I} y_{ikl} P[\theta_{ik}=s|\hat{\phi}^{(t-1)}]}{\sum_{i=1}^{I} \gamma_{ikls} P[\theta_{ik}=s|\hat{\phi}^{(t-1)}]}. \tag{4.22}$$

Closed form maximizers of μ and σ do not exist for the negative binomial distribution (4.4). I adopt the method of moment estimates as in [33, 85], where the updated values $\hat{\mu}_{kls}^{(t)}$ and $\hat{\sigma}_{kls}^{(t)}$ are the solutions of the following equations:

$$\begin{aligned}
\hat{\mu}_{kls}^{(t)} \sum_{i=1}^{I} \gamma_{ikls} P[\theta_{ik}=s|\hat{\phi}^{(t-1)}] &= \sum_{i=1}^{I} y_{ikl} P[\theta_{ik}=s|\hat{\phi}^{(t-1)}], \\
\sum_{i=1}^{I} [\hat{\mu}_{kls}^{(t)2}\gamma_{ikls}^2(1+\frac{1}{\hat{\sigma}_{kls}^{(t)}}) + \hat{\mu}_{kls}^{(t)}\gamma_{ikls}] P[\theta_{ik}=s|\hat{\phi}^{(t-1)}] &= \sum_{i=1}^{I} y_{ikl}^2 P[\theta_{ik}=s|\hat{\phi}^{(t-1)}].
\end{aligned}$$

For the degenerate distributions as in (4.6), θ_{ik}'s are directly observed. Therefore, the E-M algorithm for this case requires slight modifications: I skip the estimation for $E[I(\theta_{ik}=s)|\hat{\phi}^{(t-1)}]$ in the E-step and for μ, σ in the M-step.

4.3.3 Estimating Structured Clusters

In integrative functional genomics studies, the set of experimental conditions usually consists of interactions of multiple experimental factors; hence, it is often important to identify clusters, states of which are homogeneous across the levels of one or more factors. For example, in a typical transcription factor network analysis, experimental conditions include the combination of different cell types and TFs. It is often desirable to identify clusters of loci whose states are homogeneous

within each cell type across different TFs. I refer to such a cluster as *TF-homogeneous*. Another example is encountered in comparative functional genomics studies across different species, where experimental conditions range across both species and TFs. Clusters of loci, states of which are homogeneous across species conditional on each TF, constitute conserved functional modules. The *TF-homogeneous* clusters in this context represent the marginal effect of the species factor, and play a central role in understanding the evolutionary relationships.

To estimate a cluster with homogeneity for a particular experimental factor, MBASIC allows structural constraints on its state-space parameters. Recall that the parameters of cluster j are represented by $w_{j.s} = (w_{j1s}, w_{j2s}, \cdots, w_{jKs})$. Marginalizing the effect of this factor, the K experimental conditions can be partitioned into M sets, $\{1, 2, \cdots, K\} = T_1 \cup T_2 \cup \cdots \cup T_M$, where conditions within each set differ only in the levels of this factor. The parameters of this cluster satisfy the following constraints:

$$w_{jk_1s} = w_{jk_2s}, \quad \text{if} \quad \exists\, m \quad s.t. \quad k_1, k_2 \in T_m. \tag{4.23}$$

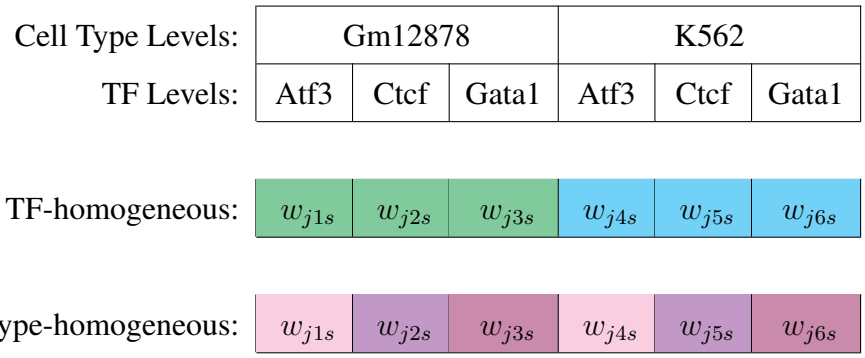


Figure 4.1 A graphical description for a parametrization with structural constraints. Interactions of 2 cell types and 3 TFs result in six experimental conditions. Parameters with homogeneous values are shaded by the same color.

A pictorial depiction with six experimental conditions due to full interaction between 2 cell types and 3 TFs is depicted in Figure 4.1. Estimating structured clustering models follows the previous E-M algorithm with a modification in Equation (4.19). A constrained maximizer for w_{jks}

subject to constraint (4.23) is computed as:

$$\hat{w}_{jks}^{(t)} = \frac{\sum_{k':k'\in T_m}\sum_{i=1}^{I} E[I(\theta_{ik'}=s)z_{ij}(1-b_i)|\hat{\phi}^{(t-1)}]}{\#\{T_m\}\sum_{s=1}^{S}\sum_{i=1}^{I} E[I(\theta_{ik}=s)z_{ij}(1-b_i)|\hat{\phi}^{(t-1)}]}, \;\; k\in T_m.$$

MBASIC requires that such structural constraints must be specified a priori and remain fixed during model fitting. MBASIC incorporates a model selection procedure to compare models with different hypothesized structural constraints and numbers of clusters. I next describe the details of this model selection procedure.

4.3.4 Model Selection

The MBASIC framework so far assumes that the total number of clusters J is known a priori. In practice, models with varying values of J need to be fitted independently and compared with each other according to some information criterion to determine the best value of J. Since the E-M algorithm aims to maximize the data likelihood function, AIC and BIC criteria can be utilized with MBASIC. The degrees of freedom for a model with J clusters is $df = F_1 S\sum_{k=1}^{K} n_k + (S-1)I + J + F_2$, where $F_1 = 2$ for distributions (4.3) and (4.4), $F_1 = 1$ for (4.5), and F_2 is the total number of free variables among w_{jks}'s. If there are no structured clusters, I have $F_2 = JK(S-1)$.

When there is no prior information available, both the total number of clusters and the number of clusters following structural constraints have to be determined. This results in a prohibitively large number of candidate models, and computing the information criterion for each of them is not practical. In such cases I incorporate the following two-phase strategy to limit the number of candidate models:

1. Evaluate models with varying total number of clusters without any structural constraints. Select J_{opt} according to the minimal AIC or BIC value.

2. Evaluate models with the fixed number of J_{opt} clusters while varying the number of clusters following each structural constraint. Select the number of clusters following each structural constraint based on the minimal AIC or BIC value.

I acknowledge that the above two-step strategy is only a practical compromise to restrict the space of candidate models and does not guarantee finding the best model that globally minimizes the information criterion. However, I have conducted extensive simulation studies which illustrated that the proposed two-phase strategy performs well in a wide variety of settings.

4.3.5 Details of the Expectation-Maximization (EM) Algorithms

4.3.5.1 Derivation for the E-step

I derive the expressions for the E-step updates of my algorithm in Eqns. (4.12), (4.13), (4.14), (4.15) as well as the marginal likelihood in Eqns. (4.9) and (4.10). In what follows, I let θ_{iks} denote $1\{\theta_{ik} = s\}$. The joint density of (z, b, θ, Y) is given by:

$$f(z,b,\theta,y) = \prod_{i=1}^{I} \zeta^{b_i}(1-\zeta)^{1-b_i} \cdot \prod_{i=1}^{I}\prod_{j=1}^{J} \pi_j^{z_{ij}} \cdot \prod_{i=1}^{I}\prod_{k=1}^{K}\prod_{s=1}^{S} \left[f_{iks} (\prod_{j=1}^{J} w_{jks}^{z_{ij}})^{1-b_i} p_{is}^{b_i} \right]^{\theta_{iks}}, \quad (4.24)$$

where $f_{iks} = \prod_{l=1}^{n_k} f(y_{ikl}|\mu_{kls}, \sigma_{kls}, \gamma_{ikls})$. The following elementary equality is used repeatedly throughout the rest of the derivations in this section.

$$\sum_{\sum_j a_{ij}=1, a_{ij}\in\{0,1\}} \prod_i \prod_j b_{ij}^{a_{ij}} = \prod_i \left(\sum_j b_{ij} \right).$$

The joint density of (z, b, Y) can be calculated from Eqn. (4.24):

$$\begin{aligned} f(z,b,y) &= \sum_{\sum_s \theta_{iks}=1} f(z,b,\theta,y) \\ &= \prod_{i=1}^{I} \zeta^{b_i}(1-\zeta)^{1-b_i} \cdot \prod_{i=1}^{I}\prod_{j=1}^{J} \pi_j^{z_{ij}} \cdot \sum_{\sum_s \theta_{iks}=1} \prod_{i=1}^{I}\prod_{k=1}^{K}\prod_{s=1}^{S} \left[f_{iks} \left(\prod_{j=1}^{J} w_{jks}^{z_{ij}} \right)^{1-b_i} p_{is}^{b_i} \right]^{\theta_{iks}} \\ &= \prod_{i=1}^{I} \zeta^{b_i}(1-\zeta)^{1-b_i} \cdot \prod_{i=1}^{I}\prod_{j=1}^{J} \pi_j^{z_{ij}} \cdot \prod_{i=1}^{I}\prod_{k=1}^{K} \left[\sum_{s=1}^{S} f_{iks} \left(\prod_{j=1}^{J} w_{jks}^{z_{ij}} \right)^{1-b_i} p_{is}^{b_i} \right]. \end{aligned} \quad (4.25)$$

Since

$$\sum_{s=1}^{S} f_{iks} \left(\prod_{j=1}^{J} w_{jks}^{z_{ij}} \right)^{1-b_i} p_{is}^{b_i} = \prod_{j=1}^{J} \left[\sum_{s=1}^{S} f_{iks} w_{jks}^{1-b_i} p_{is}^{b_i} \right]^{z_{ij}},$$

Eqn. (4.25) can be rewritten as:

$$f(z,b,y) = \prod_{i=1}^{I} \zeta^{b_i}(1-\zeta)^{1-b_i} \cdot \prod_{i=1}^{I}\prod_{j=1}^{J}\left[\pi_j \prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks} w_{jks}^{1-b_i} p_{is}^{b_i}\right)\right]^{z_{ij}}. \tag{4.26}$$

The joint distribution of (b, Y) can be calculated from Eqn. (4.26):

$$\begin{aligned} f(b,y) &= \sum_{\sum_j z_{ij}=1} f(z,b,y) \\ &= \prod_{i=1}^{I} \zeta^{b_i}(1-\zeta)^{1-b_i} \cdot \sum_{\sum_j z_{ij}=1} \prod_{i=1}^{I}\prod_{j=1}^{J}\left[\pi_j \prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks} w_{jks}^{1-b_i} p_{is}^{b_i}\right)\right]^{z_{ij}} \\ &= \prod_{i=1}^{I} \zeta^{b_i}(1-\zeta)^{1-b_i} \cdot \prod_{i=1}^{I}\left[\sum_{j=1}^{J}\pi_j \prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks} w_{jks}^{1-b_i} p_{is}^{b_i}\right)\right]. \end{aligned} \tag{4.27}$$

I note that

$$\sum_{j=1}^{J}\pi_j \prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks} w_{jks}^{1-b_i} p_{is}^{b_i}\right) = \left[\sum_{j=1}^{J}\pi_j \prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks} w_{jks}\right)\right]^{1-b_i} \left[\prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks} p_{is}\right)\right]^{b_i}.$$

Then, Eqn. (4.27) can be rewritten as:

$$f(b,y) = \prod_{i=1}^{I}\left[(1-\zeta)\sum_{j=1}^{J}\pi_j \prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks} w_{jks}\right)\right]^{1-b_i} \left[\zeta \prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks} p_{is}\right)\right]^{b_i}. \tag{4.28}$$

I can calculate the marginal density of Y, given in Eqn. (4.9), from Eqn. (4.28) as:

$$f(y) = \sum_{b_i \in \{0,1\}} f(b,y) = \prod_{i=1}^{I}\left[(1-\zeta)\sum_{j=1}^{J}\pi_j \prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks} w_{jks}\right) + \zeta \prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks} p_{is}\right)\right]. \tag{4.29}$$

Eqn. (4.10) can be obtained similarly. Moreover, I can rewrite (4.25) as

$$f(z,b,y) = \prod_{i=1}^{I}\left[\zeta \prod_{k=1}^{K}\sum_{s=1}^{S} f_{iks} p_{is}\right]^{b_i} \left[(1-\zeta)\prod_{j=1}^{J}\prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks} w_{jks}\right)^{z_{ij}}\right]^{1-b_i} \cdot \prod_{i=1}^{I}\prod_{j=1}^{J}\pi_j^{z_{ij}} \tag{4.30}$$

by using

$$\sum_{s=1}^{S} f_{iks}\left(\prod_{j=1}^{J} w_{jks}^{z_{ij}}\right)^{1-b_i} p_{is}^{b_i} = \left[\prod_{j=1}^{J}\left(\sum_{s=1}^{S} f_{iks} w_{jks}\right)^{z_{ij}}\right]^{1-b_i} \left[\sum_{s=1}^{S} f_{iks} p_{is}\right]^{b_i}.$$

Thus, the density of (z, Y) can be calculated as:

$$\begin{aligned} f(z,y) =& \sum_{b_i\in\{0,1\}} \prod_{i=1}^{I} \left[\zeta \prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks}p_{is}\right)\right]^{b_i} \left[(1-\zeta)\prod_{j=1}^{J}\prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks}w_{jks}\right)^{z_{ij}}\right]^{1-b_i} \cdot \prod_{i=1}^{I}\prod_{j=1}^{J}\pi_j^{z_{ij}} \\ =& \prod_{i=1}^{I}\left[\zeta \prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks}p_{is}\right) + (1-\zeta)\prod_{j=1}^{J}\prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks}w_{jks}\right)^{z_{ij}}\right] \cdot \prod_{i=1}^{I}\prod_{j=1}^{J}\pi_j^{z_{ij}} \\ =& \prod_{i=1}^{I}\prod_{j=1}^{J}\left[\pi_j\zeta \prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks}p_{is}\right) + \pi_j(1-\zeta)\prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks}w_{jks}\right)\right]^{z_{ij}}. \end{aligned} \tag{4.31}$$

Using Eqns. (4.28 and (4.29), I obtain Eqn. (4.12) as

$$E[b_i|Y] = \frac{\zeta \prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks}p_{is}\right)}{(1-\zeta)\sum_{j=1}^{J}\pi_j \prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks}w_{jks}\right) + \zeta \prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks}p_{is}\right)}. \tag{4.32}$$

Similarly, using Eqns. (4.31) and (4.29), I have

$$E[z_{ij}|Y] = \frac{\pi_j\zeta \prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks}p_{is}\right) + (1-\zeta)\pi_j \prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks}w_{jks}\right)}{\zeta \prod_{k=1}^{K}\sum_{s=1}^{S} f_{iks}p_{is} + (1-\zeta)\sum_{j=1}^{J}\pi_j \prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks}w_{jks}\right)}. \tag{4.33}$$

Using Eqns. (4.26) and (4.27), I have

$$E[z_{ij}|b, Y] = \frac{\pi_j \prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks}w_{jks}^{1-b_i}p_{is}^{b_i}\right)}{\sum_{j=1}^{J}\pi_j \prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks}w_{jks}^{1-b_i}p_{is}^{b_i}\right)}. \tag{4.34}$$

Eqns. (4.34) and (4.32) together results in Eqn. (4.13). Using Eqns. (4.24) and (4.25), I have

$$E[\theta_{iks}|z, b, Y] = \frac{f_{iks}\left(\prod_{j=1}^{J} w_{jks}^{z_{ij}}\right)^{1-b_i} p_{is}^{b_i}}{\sum_{s=1}^{S} f_{iks}\left(\prod_{j=1}^{J} w_{jks}^{z_{ij}}\right)^{1-b_i} p_{is}^{b_i}}. \tag{4.35}$$

Therefore, I obtain Eqn. (4.14) by using Eqns. (4.27), (4.34), and (4.35):

$$\begin{aligned} E[\theta_{iks}z_{ij}(1-b_i)|Y] =& [1-E(b_i|Y)]E(z_{ij}|b_i=0,Y)E(\theta_{iks}|b_i=0,z_{ij}=1,Y) \\ =& \frac{(1-\zeta)\sum_{j=1}^{J}\pi_j \prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks}w_{jks}\right)}{(1-\zeta)\sum_{j=1}^{J}\pi_j \prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks}w_{jks}\right) + \zeta \prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks}p_{is}\right)} \cdot \\ & \frac{\pi_j \prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks}w_{jks}\right)}{\sum_{j=1}^{J}\pi_j \prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks}w_{jks}\right)} \cdot \frac{f_{iks}w_{jks}}{\sum_{s=1}^{S} f_{iks}w_{jks}}. \end{aligned} \tag{4.36}$$

Finally, I obtain Eqn. (4.15) using Eqns. (4.35) and (4.27):

$$\begin{aligned} E(\theta_{iks}b_i|Y) =& E(b_i|Y)E(\theta_{iks}|b_i=1,Y) \\ =& \frac{\zeta\prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks}p_{is}\right)}{(1-\zeta)\sum_{j=1}^{J}\pi_j\prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks}w_{jks}\right)+\zeta\prod_{k=1}^{K}\left(\sum_{s=1}^{S} f_{iks}p_{is}\right)}\cdot\frac{f_{iks}p_{is}}{\sum_{s=1}^{S} f_{iks}p_{is}}. \end{aligned} \tag{4.37}$$

4.3.5.2 EM Algorithm with Mixture Data Distributions

An important extension of the MBASIC model is to allow multiple mixture components within each state. For example, my model in Section 4.5.2 models the data from state $s=2$ as a mixture of two negative binomial distributions following the well motivated model of [33]:

$$Y_{ikl}-3|\theta_{ik}=2 \sim \nu_{ikl}NB(\mu_{kl2},\sigma_{kl2})+(1-\nu_{ikl})NB(\mu_{kl3},\sigma_{kl3}),$$

$$\nu_{ikl} \sim Bernoulli(v_{kl}),$$

where the constant 3 denotes the minimum number of reads required to be in state $\theta=2$. In this section, I describe the general algorithm for such extensions. I assume that data from state s has a distribution of m_s components:

$$Y_{ikl}|\theta_{iks}=1 \sim \sum_{r=1}^{m_s} v_{klsr}f_{sr}(\cdot|\mu_{klsr},\sigma_{klsr},\gamma_{iklsr}).$$

This can be written in a hierarchical form, using ν_{iklsr} as the hidden variable indicating the mixture component within the state:

$$(\nu_{iklsr})_{1\le r\le m_s} \sim \text{Multinom}(1,(v_{klsr})_{1\le r\le m_s}),\ Y_{ikl}|\theta_{iks}=1,\nu_{iklsr}=1 \ \sim\ f_{sr}(\cdot|\mu_{klsr},\sigma_{klsr},\gamma_{iklsr}). \tag{4.38}$$

Here, I allow the distribution parameters μ and σ as well as the prior data γ to depend on the component. Let $f_{iklsr}=f_{sr}(y_{ikl}|\mu_{klsr},\sigma_{klsr},\gamma_{iklsr})$. The joint density for this model is:

$$
\begin{aligned}
f(z,b,\theta,\nu,y) = & \prod_{i=1}^{I} \zeta^{b_i}(1-\zeta)^{1-b_i} \cdot \prod_{i=1}^{I}\prod_{j=1}^{J} \pi_j^{z_{ij}} \cdot \prod_{i=1}^{I}\prod_{k=1}^{K}\prod_{l=1}^{n_k}\prod_{s=1}^{S}\prod_{r=1}^{m_s} v_{iklsr}^{\nu_{klsr}} \\
& \cdot \prod_{i=1}^{I}\prod_{k=1}^{K}\prod_{s=1}^{S} \left[\left(\prod_{l=1}^{n_k}\prod_{r=1}^{m_s} f_{iklsr}^{\nu_{iklsr}} \right) \left(\prod_{j=1}^{J} w_{jks}^{z_{ij}} \right)^{1-b_i} p_{is}^{b_i} \right]^{\theta_{iks}}.
\end{aligned} \tag{4.39}
$$

Let $f_{iks} = \prod_{l=1}^{n_k}(\sum_{r=1}^{m_s} v_{klsr} f_{iklsr})$, then the joint density for z, b, θ, Y can be expressed exactly the same as Eqn. (4.24). Therefore, the M-step updates for W, P, ζ and π are not changed, with the related E-step quantities computed as Eqns. (4.12), (4.13), (4.14), (4.15). I only need to modify the algorithm to estimate variables that depend on the component index r: μ, σ, and v.

The related quantities that need to be computed are $E[\nu_{iklsr}|Y]$ and $E[\theta_{iks}\nu_{iklsr}|Y]$. By Eqn. (4.39), I have

$$
P(\nu_{iklsr}=1|\theta_{iks}=0) = v_{klsr}, \;\; P(\nu_{iklsr}=1|\theta_{iks}=1) = \frac{v_{klsr} f_{iklsr}}{\sum_{r=1}^{m_s} v_{klsr} f_{iklsr}}.
$$

Therefore, I have

$$
E[\theta_{iks}\nu_{iklsr}|Y] = E[\theta_{iks}|Y] \frac{v_{klsr} f_{iklsr}}{\sum_{r=1}^{m_s} v_{klsr} f_{iklsr}}, \tag{4.40}
$$

where $E[\theta_{iks}|Y] = \sum_{j=1}^{J} E[\theta_{iks}(1-b_i)z_{ij}|Y] + E[\theta_{iks}b_i|Y]$ and can be computed by Eqns. (4.36) and (4.37). As a result,

$$
E[\nu_{iklsr}|Y] = (1 - E[\theta_{iks}|Y])v_{klsr} + E[\theta_{iks}\nu_{iklsr}|Y]. \tag{4.41}
$$

Given Eqns. (4.40) and (4.41), the M-step update for v_{klsr} is:

$$
v_{klsr}^{(t)} = \frac{\sum_{i=1}^{I} E[\nu_{iklsr}|\hat{\phi}^{(t-1)}]}{I}.
$$

The M-step updates for μ_{klsr}, σ_{klsr} can be derived using Eqn. (4.40). For the negative binomial distribution as in Section 4.5.2,

Table 4.1 Design of the simulation studies. S: size of the state-space; I: number of units; J: number of clusters; K: number of experimental conditions.

Study	Distribution	S	ζ	I	(J, K)	Model Selection
1	LN, NB, Bin	2, 3, 4	0, 0.1, 0.4	4000	(20, 30)	No
2	LN, NB, Bin	2	0.1, 0.4	4000	(20, 30)	Yes
3	iASeq	3	0, 0.1, 0.4	4000	(10, 20), (20, 30)	Yes
4	Cormotif	2	0	10,000	(4, 4), (5, 8), (5, 10)	Yes
5	Cormotif	2	0, 0.1, 0.4	4000	(10, 20)	Yes
6	LN	2	0, 0.33	4120, 4600, 6120	(8, 30)	Yes

$$\hat{\mu}^{(t)}_{klsr} \sum_{i=1}^{I} \gamma_{ikls} E[\theta_{iks} v_{iklsr} | \hat{\phi}^{(t-1)}] = \sum_{i=1}^{I} E[\theta_{iks} \nu_{iklsr} | \hat{\phi}^{(t-1)}](Y_{ikl} - 3),$$

$$\sum_{i=1}^{I} E[\theta_{iks} v_{iklsr} | \hat{\phi}^{(t-1)}][\hat{\mu}^{(t)2}_{klsr} \gamma^2_{ikls} \left(1 + \frac{1}{\hat{\sigma}^{(t)}_{klsr}}\right) + \hat{\mu}^{(t)}_{klsr} \gamma_{ikls}] = \sum_{i=1}^{I} E[\theta_{iks} v_{iklsr} | \hat{\phi}^{(t-1)}](Y_{ikl} - 3)^2.$$

4.4 Simulation Studies

I conducted 6 model-based simulation studies to investigate the performance of MBASIC in various settings as summarized in Table 4.1. Each simulation study has multiple settings that vary the distributional assumptions, size of the state-space (S), number of units (I), number of clusters (J), and number of conditions (K).

4.4.1 Simulation Study 1

The first simulation study investigated the performance of MBASIC when the true value of J was known and there were no structured clusters. I set the number of observational units as $I = 4000$ and the number of clusters as $J = 20$. The number of conditions was set to $K = 30$,

and within each condition the numbers of replicates varied as $n_k = 1, 2, 3$, each with probability 0.3, 0.5, and 0.2. The size of the hidden state space was varied at three levels: $S = 2, 3, 4$. I simulated data under three distributional families: log-normal (LN) (4.3), negative binomial (NB) (4.4), and binomial (Bin) (4.5). I also varied the proportion of singleton units ζ at 0, 0.1 and 0.4. For simplicity, I set $\gamma_{ikls} = 1$ in all distributions.

4.4.1.1 Parameter Settings

Parameters w_{jks}'s and p_{is}'s generate the hidden state variables θ_{ik}'s. I set them as follows. For different values of k, j, and i, the vectors $w_{jk\cdot} = (w_{jks} : 1 \leq s \leq S)$ and $p_{i\cdot} = (p_{is} : 1 \leq s \leq S)$ were simulated independently, each following an S-dimensional Dirichlet distribution $Dir(\alpha, \cdots, \alpha)$. I chose a uniform concentration parameter of $\alpha = 0.2$ for all dimensions to ensure that for each vector $w_{jk\cdot}$ or $p_{i\cdot}$, the probability mass tended to concentrate on one component. This controlled the conditional variance of $(\theta_{ik}|b_i, z_{ij})$. An increased value of α would increase the conditional variance of θ_{ik}, thus make it more difficult to recover w_{jks}'s and p_{is}'s.

The settings for parameters μ_{kls}'s and σ_{kls}'s were important. These parameters connected hidden states θ_{ik}'s to the observed values Y_{ikl}'s. In general, recovering hidden states from the observed data is more difficult if: (1) differences of the mean values μ_{kls}'s between the states are small; (2) variances of the distributions within each state are large. To control these two aspects at reasonable levels, I set these parameters as follows:

- For log-normal distributions (4.3), I set $\xi_s = 2 + \log(4s - 3)$, simulated $\mu_{kls} \sim N(\xi_s, 0.05^2)$, and set $\sigma_{kls} = 0.5$;
- For negative binomial distributions (4.4), I set $\xi_s = 8s - 6$, simulated $\mu_{kls} \sim N(\xi_s, 0.5^2)$, and set $\sigma_{kl1} = 2.82$, $\sigma_{kls} = 5$ for $s = 2,\ 3,\ 4$;
- For binomial distributions (4.5), I simulated $\mu_{kls} \sim Beta(3s, 3(S + 1 - s))$, and $\gamma_{ikl1} = \gamma_{ikl2} = \cdots = \gamma_{iklS} \sim Pois(10)$.

Figure 4.2 displays the histograms of $Y_{1,i,l}, \quad 1 \leq i \leq I$ from one of the simulated data sets for all the three distributions with $S = 4$ components. For comparison, I also present the histogram

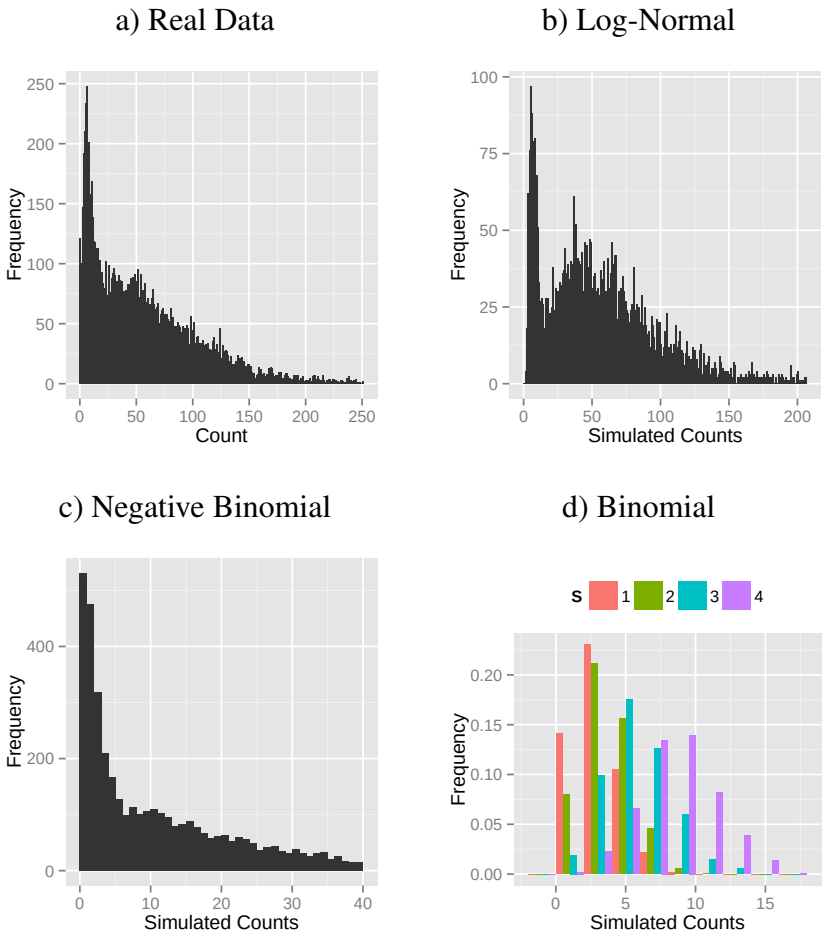


Figure 4.2 Histograms for a) a real data set from a K562 Pol2 replicate in Section 4.5.1; and simulated data from one condition based on one simulation for the b) Log-Normal, c) Negative Binomial and d) Binomial distribution with S=4 states.

of an actual data set from the analysis in Section 4.5. I observe that the mixture distribution of my simulated data with log-normal or negative-binomial distributions are very similar to the real data.

4.4.1.2 Alternative Approaches for Benchmarking MBASIC

The MBASIC algorithm can be summarized as Algorithm 4.2:

Algorithm 4.2 MBASIC

for $t = 1,\ 2, \cdots$ until convergence **do**

Expectation-Step: Compute the conditional values of $E[I(\theta_{ik} = s)|\hat{\phi}^{(t-1)}]$, $E[b_i|\hat{\phi}^{(t-1)}]$, $E[I(\theta_{ik} = s)b_i|\hat{\phi}^{(t-1)}]$, $E[z_{ij}(1-b_i)|\hat{\phi}^{(t-1)}]$, $E[I(\theta_{ik} = s)z_{ij}(1-b_i)|\hat{\phi}^{(t-1)}]$;

Maximization-Step: Update estimates for parameters μ_{kls}, σ_{kls}, ζ, π_j, w_{jks}, p_{is}.

end for

To the best of my knowledge, there are currently no existing methods suited for the general setup of MBASIC. There are, however, algorithms tailored for analyzing specific data types with hierarchical state-space models similar to MBASIC. These algorithms largely fall into two categories. In the first category, estimation for the state-space variables are separated from state-space clustering. This approach is followed by [9], [15], [26], [49]. In the second category, distributional parameters for each experimental replicate are estimated first. These parameters are then fixed, and conditional on their estimates, the state-space variables and the clustering structure are estimated jointly. Such an approach is followed by [72] [80], and [73].

To compare the general implementation of MBASIC as in Algorithm 4.2 with these existing model fitting ideas, I designed six benchmark algorithms. Table 4.2 provides a summary of these algorithms. Two of these algorithms, SE-HC (State-space Estimation followed by Hierarchical Clustering) and SE-MC (State-space Estimation followed by Mixture model Clustering), treat the state-space mapping step and the state-space clustering separately. The third algorithm, PE-MC (Parameter Estimation followed by Mixture model Clustering), separates experiment-specific distributional parameter estimation from the joint estimation of other parameters.

Table 4.2 *Simulation Study 1.* A summary of the benchmark algorithms that are compared to MBASIC. Neither SE-MC nor PE-MC perform joint estimation of the model parameters. SE-* algorithms estimate the data-specific model parameters and state spaces as a first step and then cluster the state variables. PE-* algorithms estimate data-specific model parameters and fixes these in joint estimation of the state space and clustering. * Denotes distributional parameters for each experimental replicate.

Algorithm	Is state-space estimation joint with clustering?	Is parameter* estimation joint with clustering?	Clustering model	Include singletons
MBASIC	Joint	Joint	Mixture model	Yes
SE-HC	Separate	Separate	Hierarchical clustering	No
SE-MC	Separate	Separate	Mixture model	Yes
PE-MC	Joint	Separate	Mixture model	Yes
MBASIC0	Joint	Joint	Mixture model	No
SE-MC0	Separate	Separate	Mixture model	No
PE-MC0	Joint	Separate	Mixture model	No

For all the three algorithms, in the first step, observations from each experimental condition $\{Y_{ikl} : 1 \leq i \leq I, 1 \leq l \leq n_k\}$ are fitted according to the following model:

$$(Y_{ikl}|\theta_{ik} = s) \sim f_s(\cdot|\mu_{kls}, \sigma_{kls}, \gamma_{ikls}), \quad P(\theta_{ik} = s) = q_{ks}. \tag{4.42}$$

The standard E-M algorithm can be used for the first step and results in estimates of q_{ks}, μ_{kls}, σ_{kls} as well as the posterior estimates for the state space $P(\theta_{ik} = s|Y)$. In the second step, SE-MC and SE-HC cluster the observational units based on the estimated $P(\theta_{ik} = s|Y)$ from the first step. SE-HC (Algorithm 4.3) uses hierarchical clustering, while SE-MC (Algorithm 4.4) uses MBASIC with degenerate distributions (4.6) for clustering. The second step of PE-MC (Algorithm 4.5) is similar to Algorithm 4.2, except that parameters μ_{kls}, σ_{kls}'s are not updated.

In addition to joint fitting of all model parameters, another important feature of MBASIC is its inclusion of the singleton cluster $\mathscr{C}_0$. To the best of my knowledge, this feature is not included in similar models such as [72] and [73]. I conjecture that in practice, when some units can not be grouped together with other units due to their distinct state-space profiles, including this singleton cluster can enhance model estimation. To test this conjecture, I developed a version of each of the SE-MC, PE-MC, and MBASIC algorithms that ignore the singleton cluster, i.e., forces each unit into a cluster. This is achieved simply by initializing $\zeta = 0$ in the Algorithms 4.2, 4.4, and 4.5. I refer to these algorithms by SE-MC0, PE-MC0, and MBASIC0.

Algorithm 4.3 State-space estimation followed by hierarchical clustering (SE-HC)

Step 1:

for $1 \leq k \leq K$ **do**

Apply the standard E-M algorithm on data $\{Y_{ikl} : 1 \leq i \leq I, 1 \leq l \leq n_k\}$ to estimate posterior probabilities $P(\theta_{ik} = s|Y)$.

end for

Step 2:

Denote vectors $\tilde{\theta}_i = (P(\theta_{ik} = s|Y))_{1\leq k\leq K, 1\leq s\leq S}$. Cluster vectors $\tilde{\theta}_i$ into J clusters using hierarchical clustering algorithm with the Euclidean distance. Estimate w_{jks} as the means within each cluster.

Algorithm 4.4 State-space estimation followed by mixture model clustering (SE-MC)

Step 1:

for $1 \leq k \leq K$ **do**

Apply the standard E-M algorithm on data $\{Y_{ikl} : 1 \leq i \leq I, 1 \leq l \leq n_k\}$ to estimate posterior probabilities $P(\theta_{ik} = s|Y)$.

end for

Step 2:

Denote $\theta^*_{ik} = \arg_s \max P(\theta_{ik} = s|Y)$ for each $1 \leq k \leq K, 1 \leq i \leq I)$. Apply Algorithm 4.2 with $\theta_{ik} \leftarrow \theta^*_{ik}$ and $f_s = I(y = s)$ to obtain estimates for w_{jks}, p_{is}, ζ, and π_j.

Algorithm 4.5 Parameter estimation followed by mixture model clustering (PE-MC)

Step 1:

for $1 \leq k \leq K$ **do**

Apply the standard E-M algorithm on data $\{Y_{ikl} : 1 \leq i \leq I, 1 \leq l \leq n_k\}$ to estimate μ_{kls}, σ_{kls} for each experiment.

end for

Step 2:

Apply Algorithm 4.2 without updating μ_{kls}, σ_{kls} in the Maximization step.

4.4.1.3 Results

I utilized several criteria to compare the performance of MBASIC to the benchmark algorithms in Table 4.2. To estimate how well the state space was characterized for each cluster, I computed the mean-squared error for W (MSE-W) as MSE-W=$\sqrt{\sum_{j,k,s}(\hat{w}_{jks} - w_{jks})^2/(JKS)}$. I also evaluated how well each method recovered the true state variables θ_{ik}'s. This was reflected by the state prediction error (SPE) as the mean squared error between the simulated states θ_{ik}'s and their posterior probabilities: SPE=$\sqrt{\sum_{i,k,s}[1\{\theta_{ik} = s\} - P(\theta_{ik} = s|Y)]^2/(IKS)}$. Finally, to compare the estimated clustering with the simulated true clustering, I computed the Adjusted Rand Index (ARI) [54]. ARI is a measure for the similarity between two different clusterings of the data. Its value ranges between -1 and 1, with 1 indicating perfect match between the two clusterings.

ARI requires the true clusters denoted by $\mathscr{C}_j,\ 0 \leq j \leq J$ and their estimates denoted by $\hat{\mathscr{C}}_j,\ 0 \leq j \leq \hat{J}$. In my simulations, they were computed as:

$$\mathscr{C}_0 = \{1 \leq i \leq I : b_i = 1\};\ \mathscr{C}_j = \{1 \leq i \leq I : b_i = 0,\ z_{ij} = 1\},\ j \leq 1,$$

where $\mathscr{C}_0$ denoted the set of singleton units. $\hat{\mathscr{C}}_j$ was computed from the posterior distributions as follows:

$$\begin{aligned}
\hat{\mathscr{C}}_0 &= \{1 \leq i \leq I : E(b_i|Y) > [1 - E(b_i|Y)] \max_j E(z_{ij}|Y)\}, \\
\hat{\mathscr{C}}_j &= \{1 \leq i \leq I : j = \arg_{j'} \max E(z_{ij'}|Y), E(b_i|Y) \leq [1 - E(b_i|Y)]E(z_{ij}|Y)\}.
\end{aligned}$$

The simulation results under various settings are summarized by the boxplots for each criterion in Figures 4.3, 4.4, and 4.5. Across all different simulation settings, the performance of MBASIC was consistently among the best in all of the ARI, MSE-W, and SPE metrics. This shows that MBASIC could not only recover the clustering structure, but also achieve high accuracy in estimating individual states. SE-HC, SE-MC and SE-MC0 performed the worst in both detecting the clustering structure and estimating the individual states. This suggests that separating state-space inference from joint model fitting can significantly deteriorate model estimation. Different from the SE-* methods, performances of PE-MC and PE-MC0 were much closer to MBASIC. For the

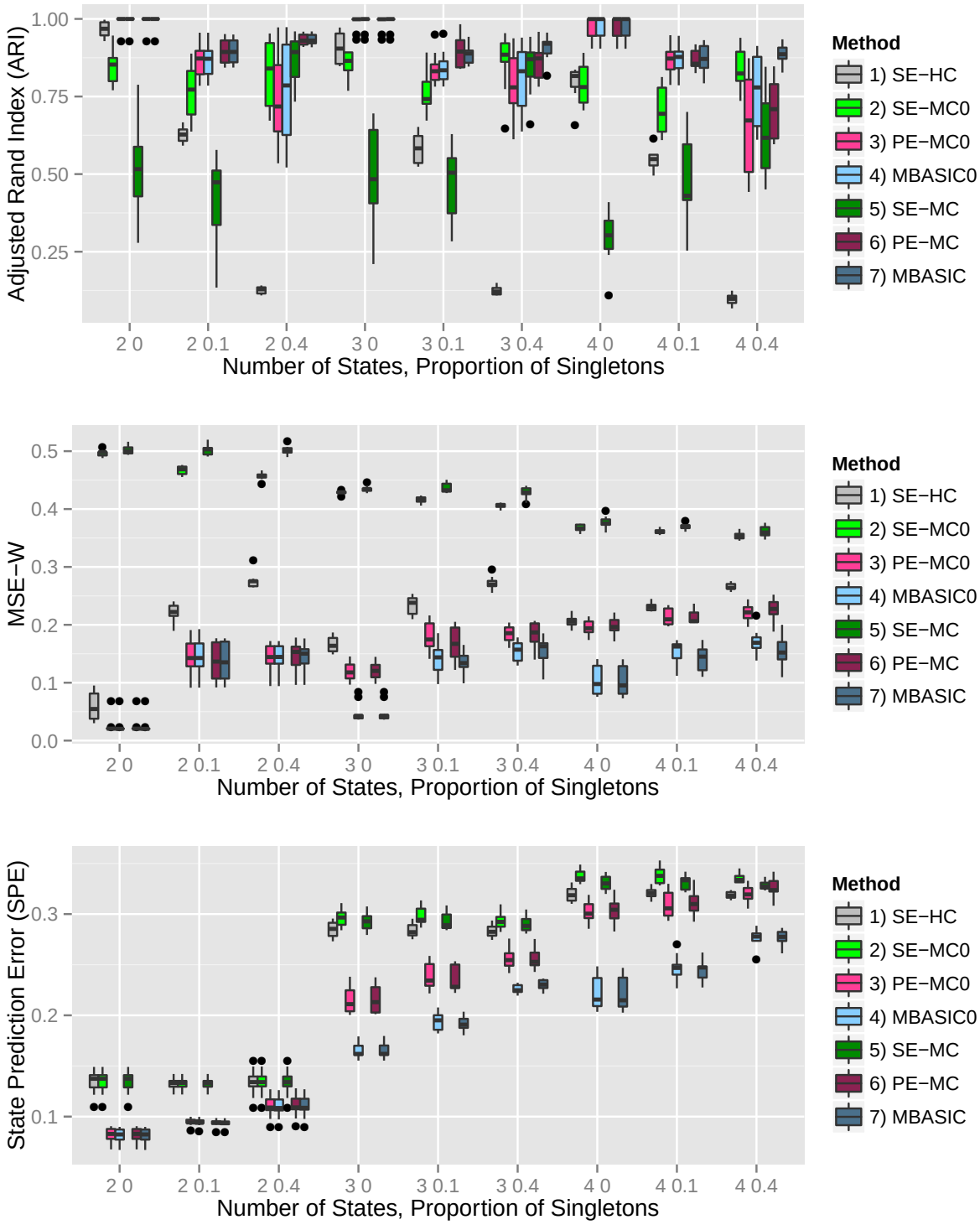


Figure 4.3 *Simulation Study 1, log-normal distribution.* Boxplots for ARI, MSE-W, and SPE across 10 simulated datasets. The number of states is varied at 2, 3, and 4, and the proportion of singletons at 0, 0.1, 0.4. Table 4.2 summarizes the methods compared.

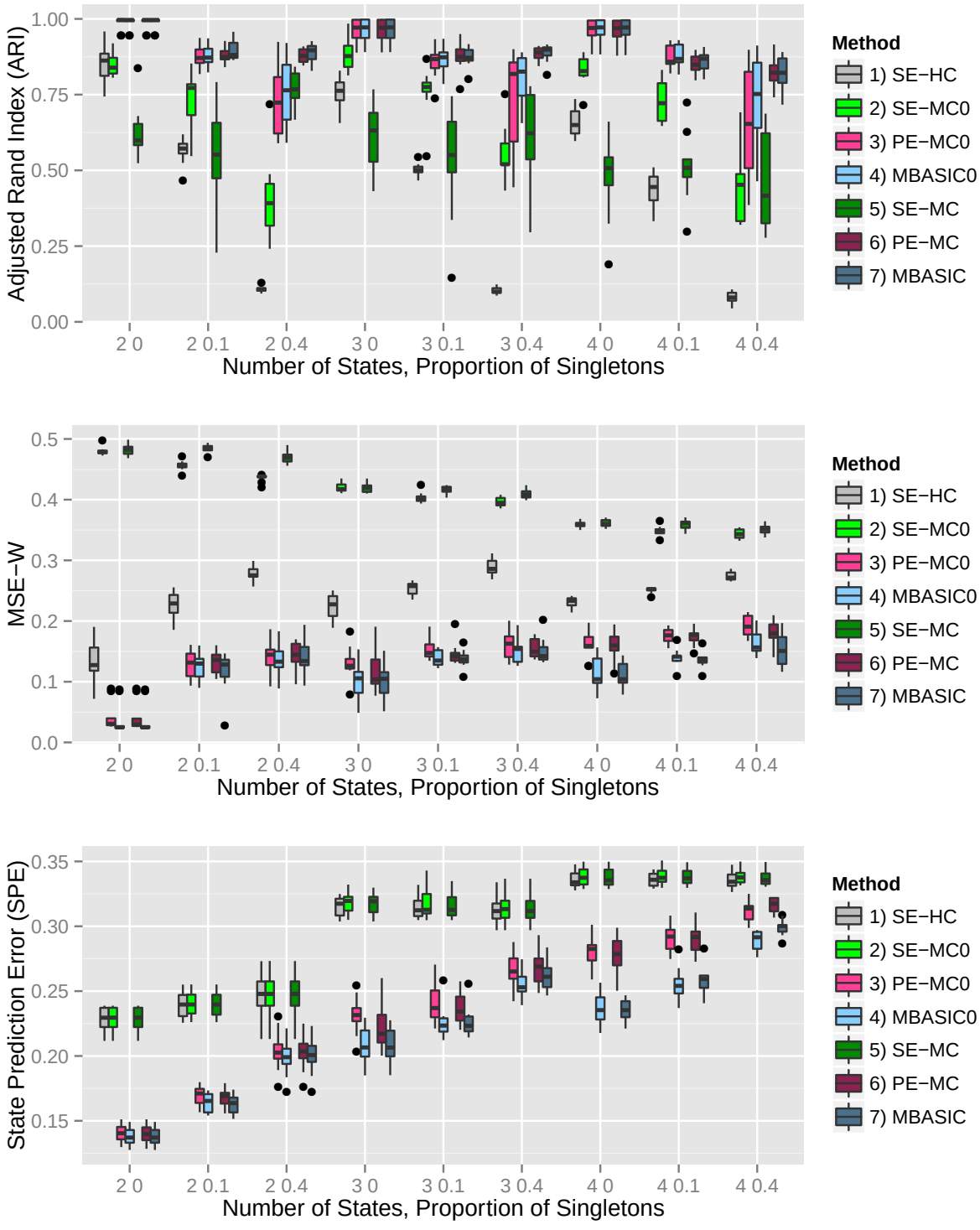


Figure 4.4 *Simulation Study 1, negative binomial distribution.* Boxplots for ARI, MSE-W, and SPE across 10 simulated datasets. The number of states is varied at 2, 3, and 4, and the proportion of singletons at 0, 0.1, 0.4. Table 4.2 summarizes the methods compared.

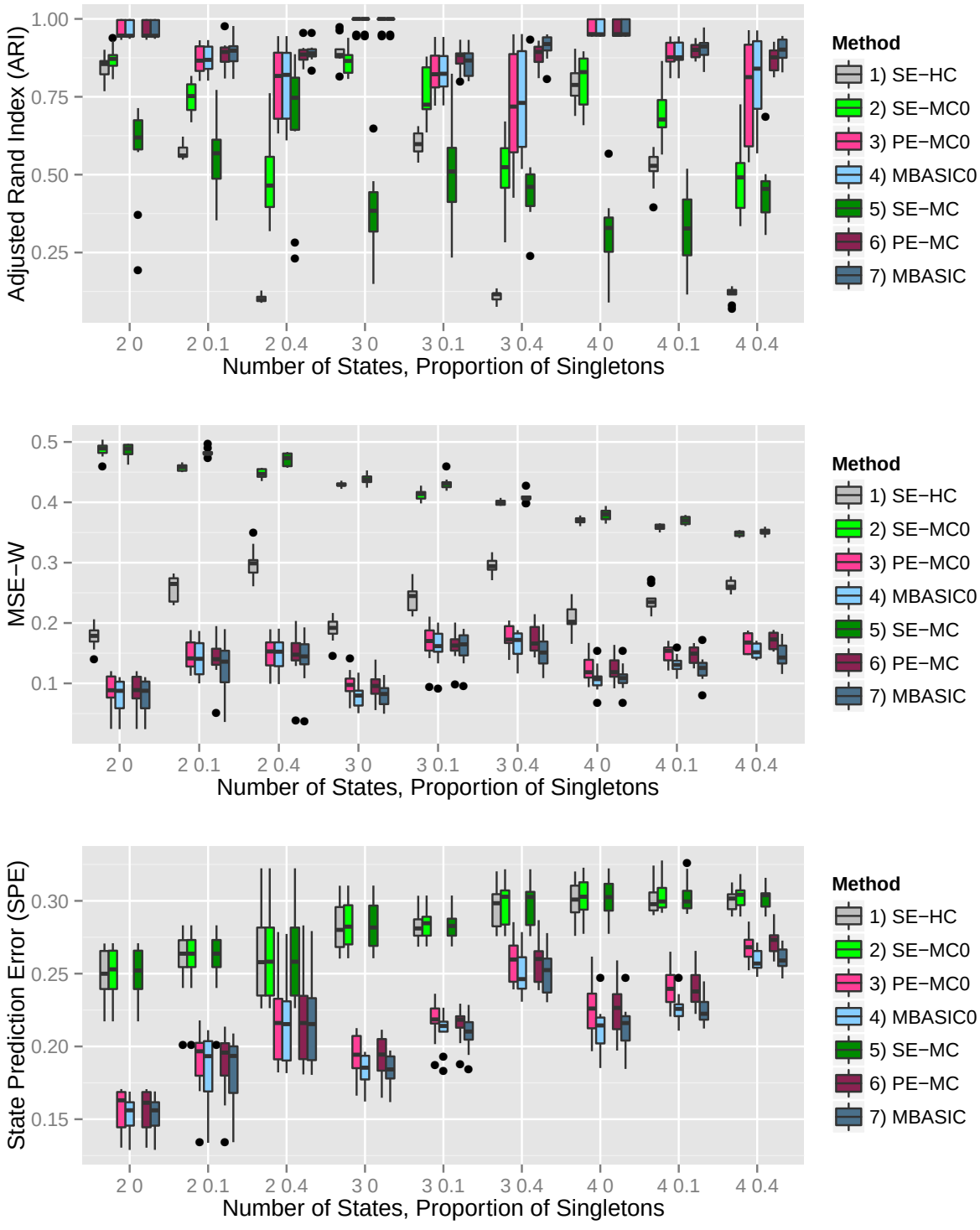


Figure 4.5 *Simulation study 1, binomial distribution.* Boxplots for ARI, MSE-W, and SPE across 10 simulated datasets. The number of states is varied at 2, 3, and 4, and the proportion of singletons at 0, 0.1, 0.4. Table 4.2 summarizes the methods compared.

negative binomial and binomial distributions (Figures 4.4 and 4.5), PE-MC achieved similar ARI levels to MBASIC and slightly larger SPE values. These observations show that by jointly estimating the clusters and the states, data under different conditions could borrow information from each other and thus substantially improve the state-space estimation. Overall, these observations are consistent with the results in [73] and [72].

The simulation results also show the effect of modeling the singleton cluster in various settings. Comparing the performances of MBASIC with MBASIC0 and PE-MC with PE-MC0, I see that modeling the singleton cluster does not have a significant effect when the proportion of singletons is low, i.e., $\zeta = 0$ or 0.1; however, the improvement is highly significant when $\zeta = 0.4$. When $\zeta = 0.4$, including singletons significantly improved the performance with respect to ARI, but did not have an obvious effect on SPE. This has several implications in practice. First, the fact that MBASIC does not under-perform any other methods when $\zeta = 0$ or 0.1 indicates that increasing the model complexity by introducing singletons does not lead to unrobust inference. Because I am always agnostic on the existence of singletons for any real data, keeping them in my model would guard against their adverse influence in inferring the clustering structure. Second, although incorporating the singleton cluster does not improve estimating individual states, some epigenetic studies focus primarily on the association structure between units, as my example in Section 4.5.2. For such studies, the gain in estimating the clustering structure by including the singletons is essential. I notice that modeling the singletons does not necessarily improve estimation for separate model fitting, even when the proportion of singletons is high, e.g., $\zeta = 0.4$, as I compare SE-MC0 with SE-MC for the negative binomial distribution and the binomial distribution (Figures 4.4 and 4.5). This indicates that the state-space estimation step might introduce additional noise to the clustering step, which made it less favorable to infer a complicated clustering structure with singletons.

4.4.2 Simulation Study 2: Model Selection

This second set of simulations aimed to evaluate the use of BIC to select the number of clusters as well as the structural constraints for each cluster. I simulated data sets under two scenarios. For

Table 4.3 *Simulation study 2, Scenario 1, unstructured clusters.* Simulation results for model selection without structural constraints. For each criterion, the mean is computed over 10 simulated data sets, with the standard deviation shown in the parentheses.

Dist.	ζ	J	ARI	MSE-W	SPE
Bin	0.1	20.8 (2.098)	0.94 (0.036)	0.096 (0.018)	0.159 (0.014)
Bin	0.4	20.9 (1.101)	0.914 (0.035)	0.122 (0.034)	0.204 (0.012)
LN	0.1	20.7 (0.823)	0.989 (0.005)	0.044 (0.03)	0.086 (0.006)
LN	0.4	21.3 (1.337)	0.972 (0.007)	0.095 (0.027)	0.107 (0.008)
NB	0.1	21.6 (0.843)	0.947 (0.021)	0.089 (0.028)	0.154 (0.007)
NB	0.4	20.6 (2.271)	0.902 (0.026)	0.112 (0.048)	0.189 (0.007)

Table 4.4 *Simulation study 2, Scenario 2, structured clusters.* Simulation results for model selection with structural constraints. For each criterion, the mean is computed over 10 simulated data sets, with the standard deviation shown in the parentheses.

Dist.	ζ	J_1	J	ARI	MSE-W	SPE
Bin	0.1	10.3 (1.16)	20.7 (1.494)	0.934 (0.022)	0.084 (0.035)	0.162 (0.02)
Bin	0.4	10.3 (1.636)	21 (2.625)	0.897 (0.048)	0.125 (0.03)	0.196 (0.031)
LN	0.1	10.4 (0.516)	20.6 (0.516)	0.984 (0.015)	0.044 (0.032)	0.086 (0.006)
LN	0.4	11.2 (1.619)	22.5 (1.509)	0.968 (0.01)	0.108 (0.037)	0.106 (0.006)
NB	0.1	10.9 (1.197)	21 (1.054)	0.955 (0.019)	0.064 (0.035)	0.155 (0.008)
NB	0.4	11.2 (1.814)	22.2 (1.398)	0.926 (0.014)	0.108 (0.031)	0.184 (0.013)

the first scenario, each data set had $J = 20$ clusters with $K = 30$ experimental conditions, and none of the clusters had structural constraints. For the second scenario, each data set had $J = 20$ clusters over $K = 30$ conditions, but $J_1 = 10$ of the clusters were structurally constrained as follows:

$$w_{j,k,s} = w_{j,k+K/2,s}, \forall j, 1 \leq j \leq J/2; \forall k, 1 \leq k \leq K/2.$$

I refer the two scenarios as the *unstructured scenario* and the *structured scenario*, respectively. I considered log-normal distributions (4.3), negative binomial distributions (4.4) and binomial distributions (4.5) for both cases. I also varied the proportion of singleton units ζ at 0.1 and 0.4. The number of states was fixed at $S = 2$. The remaining parameters were simulated following the same mechanism as in Section 4.4.1.1.

For each simulated data set, I fitted a number of candidate models. For the unstructured scenario, I varied the number of clusters J from 10 to 30. For the structured scenario, I followed the two-phase procedure described in Section 4.3.4. The best model was selected by the minimum BIC value. To assess the performances of these selected models, I computed the ARI, MSE-W and SPE metrics as described in Section 4.4.1[2].

The simulation results are summarized in Tables 4.3 and 4.4. Under each set of parameters, I computed the mean and the standard deviation for each of the criterion as well as the selected value of J and J_1 under 10 simulated data sets. These tables show that the selected values for J and J_1 were very close to the true values. Moreover, MBASIC performed uniformly well with respect to ARI, MSE-W, and SPE under different settings. These results indicate that even if MBASIC may not identify the "true" structure that drives the actual data, the identified structures can still properly represent the state-space association between units.

[2]When the actual J and its estimate $\hat{J}$ are different, MSE-W is redefined as:

$$MSE - W = \left[\frac{\sum_{k,1 \leq j \leq J,s} (w_{jks} - \hat{w}_{kc_1(j)s})^2 + \sum_{k,1 \leq j \leq \hat{J},s} (\hat{w}_{jks} - w_{kc_2(j)s})^2}{KS(J + \hat{J})} \right]^{\frac{1}{2}},$$

where

$$c_1(j) = \arg_{j' \leq \hat{J}} \min \sum_{k,s} (w_{jks} - \hat{w}_{jk's})^2 \ , \ c_2(j) = \arg_{j' \leq J} \min \sum_{k,s} (\hat{w}_{jks} - w_{jk's})^2.$$

4.4.3 Simulation Studies 3-5: Comparison with iASeq and CorMotif

In this section, I compare MBASIC with two recently proposed models for integrative analysis of specific types of genomic data: CorMotif ([73]) and iASeq ([72]). Both models have the similar state-space clustering structure as MBASIC. The main difference from MBASIC is that they each incorporate a more complicated distribution assumption targeting a specific genomic data type. The CorMotif model, inheriting the LIMMA ([63]) framework for integrating gene-expression data, assumes mixture of Gaussian distributions with $S = 2$ states: $s = 1$ for the non-differential state, and $s = 2$ for the differential state. For each experiment condition k, in addition to the n_{k1} replicates, there are n_{k0} control replicates. The CorMotif model has the following state-space mapping structure:

$$\begin{aligned}
\frac{n_k s_k^2}{\sigma_{ik}^2} \sim & \chi^2_{n_k}, \\
\mu_{ik}|\sigma_{ik}^2 \sim & N(0, u_k\sigma_{ik}^2), \\
(Y_{ikl}|\theta_{ik} = 1) \sim & N(\mu_{ik0}, \sigma_{ik}^2),\ l = 1, 2, \cdots, n_{k1}, \\
(Y_{ikl}|\theta_{ik} = 2) \sim & N(\mu_{ik0} + \mu_{ik}, \sigma_{ik}^2),\ l = 1, 2, \cdots, n_{k1}, \\
X_{ikl} \sim & N(\mu_{ik0}, \sigma_{ik}^2),\ l = 1, 2, \cdots, n_{k0}.
\end{aligned}$$

where X_{ikl}'s are the observed data from control experiments, and Y_{ikl} are the observed data from the case experiments. n_k and s_k^2 are hyper parameters specific to each experiment to account for potential heterogeneity among units within the same state, and u_k reflects the strength of differential expression. CorMotif assumes almost the same state-space clustering structure as MBASIC except that it does not include singletons. The iASeq model, targeting at allele-specific binding problems, has the following state-space mapping structure:

$$
\begin{aligned}
Y_{ikl} &\sim Binom(\gamma_{ikl}, p_{ik}),\\
p_{ik}|\theta_{ik} = 2 &\sim Beta(\alpha_k, \beta_k),\\
p_{ik}|\theta_{ik} = 1 &\sim Unif\big(0, \frac{\alpha_k}{\alpha_k + \beta_k}\big),\\
p_{ik}|\theta_{ik} = 3 &\sim Unif\big(\frac{\alpha_k}{\alpha_k + \beta_k}, 1\big).
\end{aligned}
$$

where the α_k, β_k are experiment-specific parameters, and γ_{ikl} is the observed total number of reads between two alleles. The state-space mapping structure for iASeq is almost the same as MBASIC, except that it assumes no singletons, and that one cluster is undifferentiated (i.e. $w_{1k1} = 1,\ \forall 1 \leq k \leq K$).

There are two key differences between CorMotif/iASeq and MBASIC. First, both CorMotif and iASeq address the heterogeneity among the units within the same state, and they introduce additional hyper parameters to model the heterogeneous parameters associated with the distribution of individual units. Compared to MBASIC, where I assume the distributions within the same state are homogeneous, such heterogeneous distributional assumptions are much more realistic. Second, CorMotif and iASeq implement two-stage estimation procedures similar to PE-MC0, which separate parameter estimation from state-space clustering. [73] pointed out that once I have the heterogeneous distributional parameters within each state, joint model fitting for all parameters would require running a Markov Chain Monte-Carlo algorithm rather than the simple E-M algorithm I have developed for MBASIC. Therefore, the computational cost ensued might render its applicability for large real data sets.

In comparison of MBASIC to CorMotif and iASeq, I simulated data according to each of the assumed distributions of CorMotif/iASeq, but fitted MBASIC models using simplified distributions. For data simulated from the iASeq model, I used MBASIC to fit binomial distributions with $S = 3$ states (4.5). For data simulated from the CorMotif model, I first generated two versions of t-statistics as follows. For each unit and experiment, denote $\overline{Y}_{ik} = \sum_{l=1}^{n_{k1}} Y_{ikl}/n_{k1}$, $\overline{X}_{ik} = \sum_{l=1}^{n_{k0}} X_{ikl}/n_{k0}$, $\tilde{s}^2_{ik} = [\sum_{l=1}^{n_{k1}}(Y_{ikl} - \overline{Y}_{ik})^2 + \sum_{l=1}^{n_{k0}}(X_{ikl} - \overline{X}_{ik})^2]/(n_{k1} + n_{k0} - 2)$ and $v_k = 1/n_{k1} + 1/n_{k0}$. I computed the *naive t-statistic* T_{ik} as:

Table 4.5 Summary for the designs of the simulation settings in Simulation Study 4, originally designed by [73].

Simulation Setting	I	J	K
1	10,000	4	4
2	10,000	4	4
3	10,000	5	8
4	10,000	5	20

$$T_{ik} = \frac{\overline{Y}_{ik} - \overline{X}_{ik}}{\sqrt{v_k}\tilde{s}_k}. \tag{4.43}$$

I also computed the *limma t-statistic* $\tilde{T}_{ik}$ by first fitting the data for each condition using LIMMA ([63]) to estimate n_k and s_k^2, then computed:

$$\tilde{T}_{ik} = \frac{\sqrt{n_k + n_{k1} + n_{k0} - 2}(\overline{Y}_{ik} - \overline{X}_{ik})}{\sqrt{v_k[(n_{k1} + n_{k0} - 2)\tilde{s}_k^2 + n_k s_k^2]}}. \tag{4.44}$$

For each set of T_{ik}'s and $\tilde{T}_{ik}$'s, I fitted the MBASIC model with $S = 2$ components of scaled-t distributions:

$$T/\mu_{ks}|\theta_{iks} = 1 \ \sim \ t_{\sigma_{ks}}, \ \ s = 1,\ 2. \tag{4.45}$$

Here, μ_{ks} is the scaling parameter, and σ_{ks} is the degrees of freedom. Because I pooled the replicate level data to generate these t-statistics, the parameters μ and σ no longer depended on l. I refer to the method using $\tilde{T}_{ik}$ as MBASIC-limma, and using T_{ik} as MBASIC-t. Because there is no closed form maximum likelihood solution for t-distributions, I use the moment method to estimate μ_{ks}'s and σ_{ks}'s in the M-step similar to the case of negative binomial distributions.

In Simulation Study 3, I simulated data following the iASeq model. I set $\alpha_k = \beta_k = 2$, and simulated state-space variables the same as in Section 4.4.1 with $I = 4000$. I set $J = 10,\ 20$ and $\zeta = 0,\ 0.1,\ 0.4$. Simulation Studies 4-5 compare MBASIC with CorMotif. In Simulation Study 4, I simulated data in four settings corresponding to Simulations 1-4 of [73] respectively. In these

Table 4.6 *Simulation Studies 3-5.* A summary of the simulation designs, the fitting algorithms compared, and the figure numbers for the results.

Study	J	ζ	True model	Fitting algorithms	Related figures
3	10, 20	0, 0.1, 0.4	iASeq	MBASIC, iASeq	Figure 4.6
4	4, 5	0	CorMotif	MBASIC-limma, MBASIC-t, CorMotif	Figures 4.7, 4.9
5	10, 20	0, 0.1, 0.4	CorMotif	MBASIC-limma, MBASIC-t, CorMotif	Figure 4.8

settings, I had $n_k = 4$, $u_k = 4$, $s_k^2 = 0.02$. Table 4.5 summarizes the settings for the number of clusters, experiment conditions, and units for the state-space variables. I refer my readers to [73] for more details of the state-space design. I note that [73] simulations did not include singletons (i.e., $\zeta = 0$) and furthermore, their settings assumed $w_{jks} \in \{0,\ 1\}$. This means that the state-space variables are completely determined by the clustering structure. In Simulation Study 5, I set n_k and s_k^2 the same as in Simulation Study 4, but varied u_k as $u_k = 8$ for easier distinction between different states. However, I simulated w_{jks} following S-dimensional Dirichlet distributions as in Simulation Study 1 to introduce noises in generating state-space variables. In addition, I simulated data with smaller number of units ($I = 4000$), but more clusters ($J = 10,\ 20$), and varied the proportion of singletons $\zeta = 0,\ 0.1,\ 0.4$. The other details of generating state-space variables were the same as in Section 4.4.1.

Table 4.6 summarizes Simulation Studies. For each set of parameters, I simulated 10 data sets. I computed ARI, MSE-W, and SPE based on both the model with the number of clusters selected by BIC, and the oracle model where the number of clusters is set to its true value. The comparison between MBASIC and iASeq is shown in Figure 4.6. For all the different settings, MBASIC achieved better clustering performance with higher ARI values. However, iASeq performed better in SPE and MSE-W. When $\zeta = 0$, iASeq performed overall better than MBASIC, with similar ARI values as MBASIC but much lower SPE. However, as ζ increased, iASeq's ARI value became significantly smaller than MBASIC, while its SPE value became closer to MBASIC's. In such cases, the benefits of modeling singletons outweigh the loss of using simplified distributional assumptions.

The comparison between MBASIC and CorMotif is shown in Figures 4.7 and 4.8. In Simulation Study 4 (Figure 4.7), because CorMotif models did not allow singletons, I also excluded

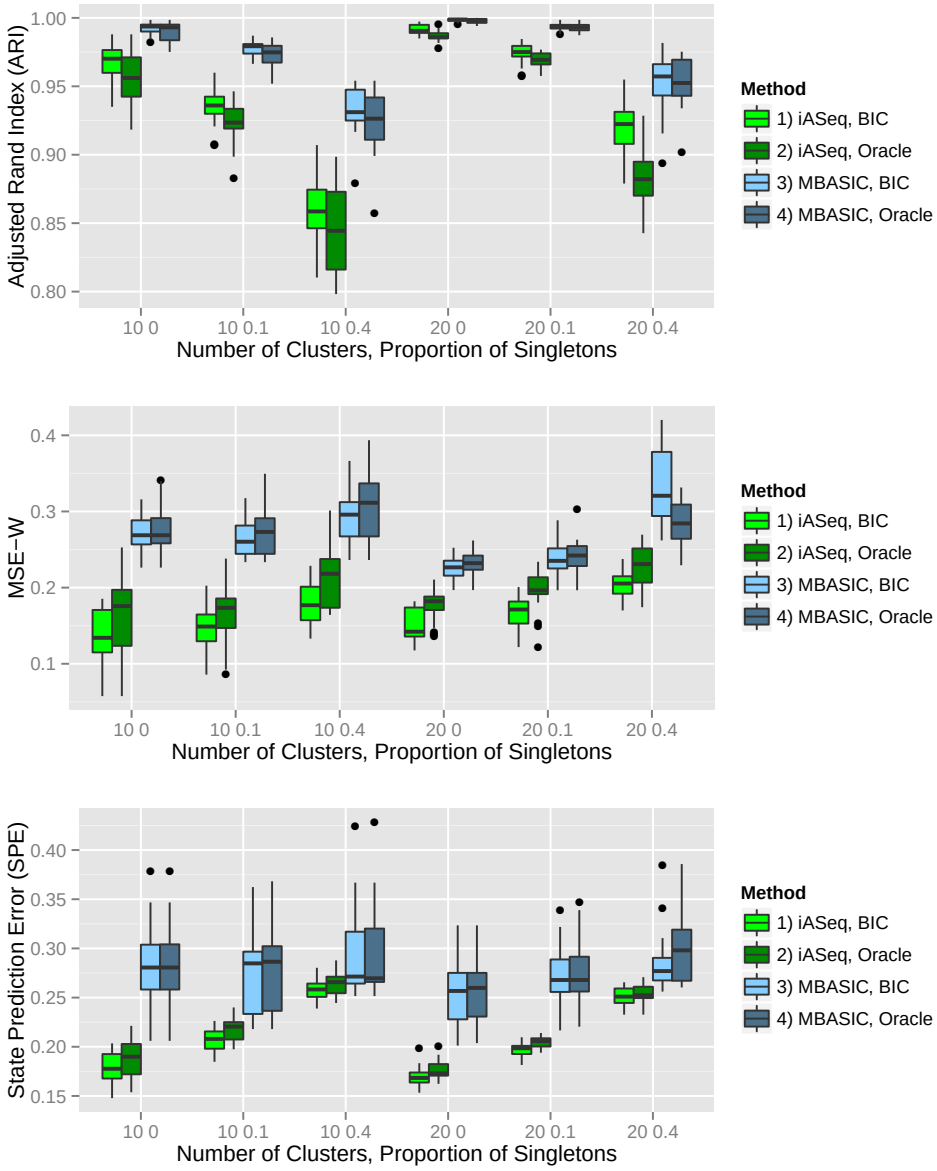


Figure 4.6 *Simulation Study 3, comparison between MBASIC and iASeq.* I varied the number of clusters at 10, 20 and the proportion of singletons at 0, 0.1 and 0.4. Results are summarized over 10 simulations under each setting.

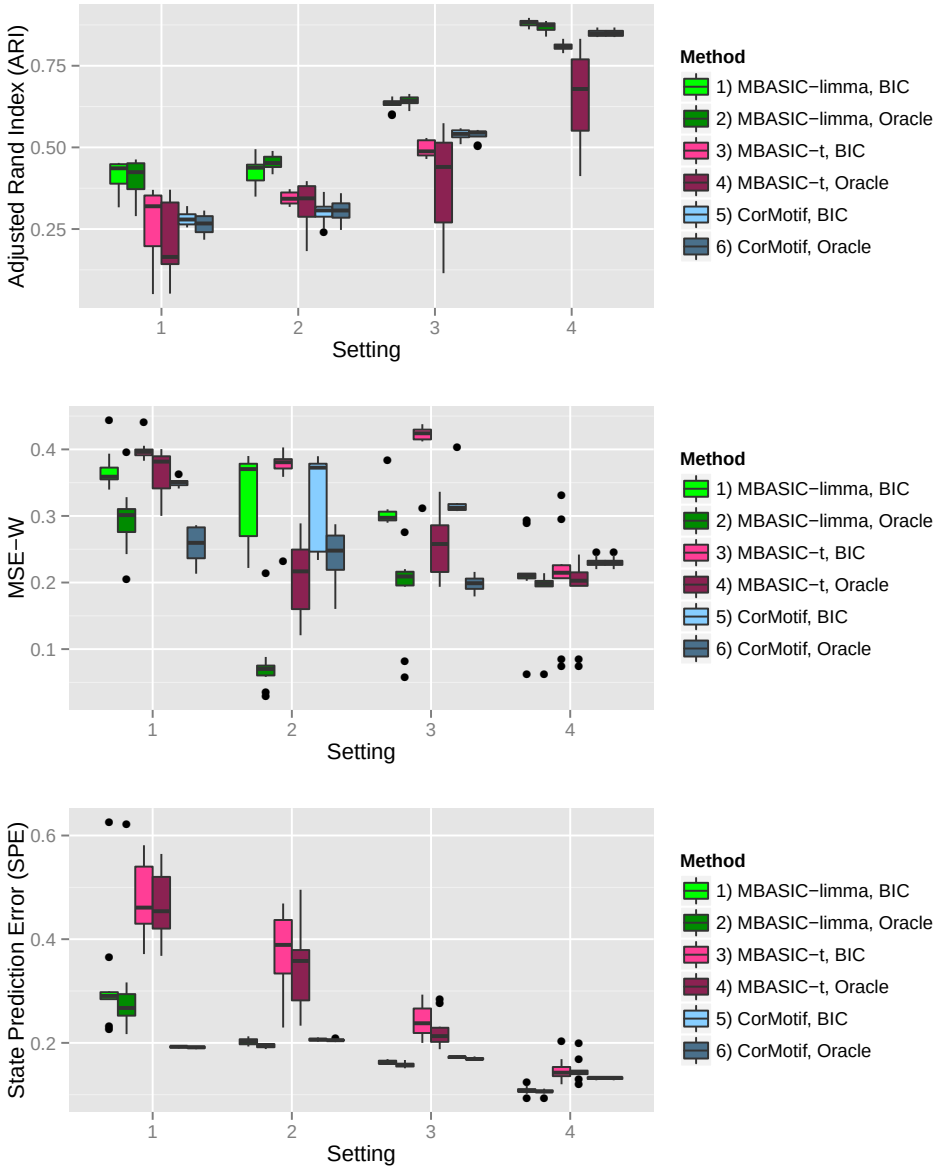


Figure 4.7 *Simulation Study 4, comparison between MBASIC and CorMotif.* I simulated data under four settings as in Table 4.5. Results are summarized over 10 simulations under each setting.

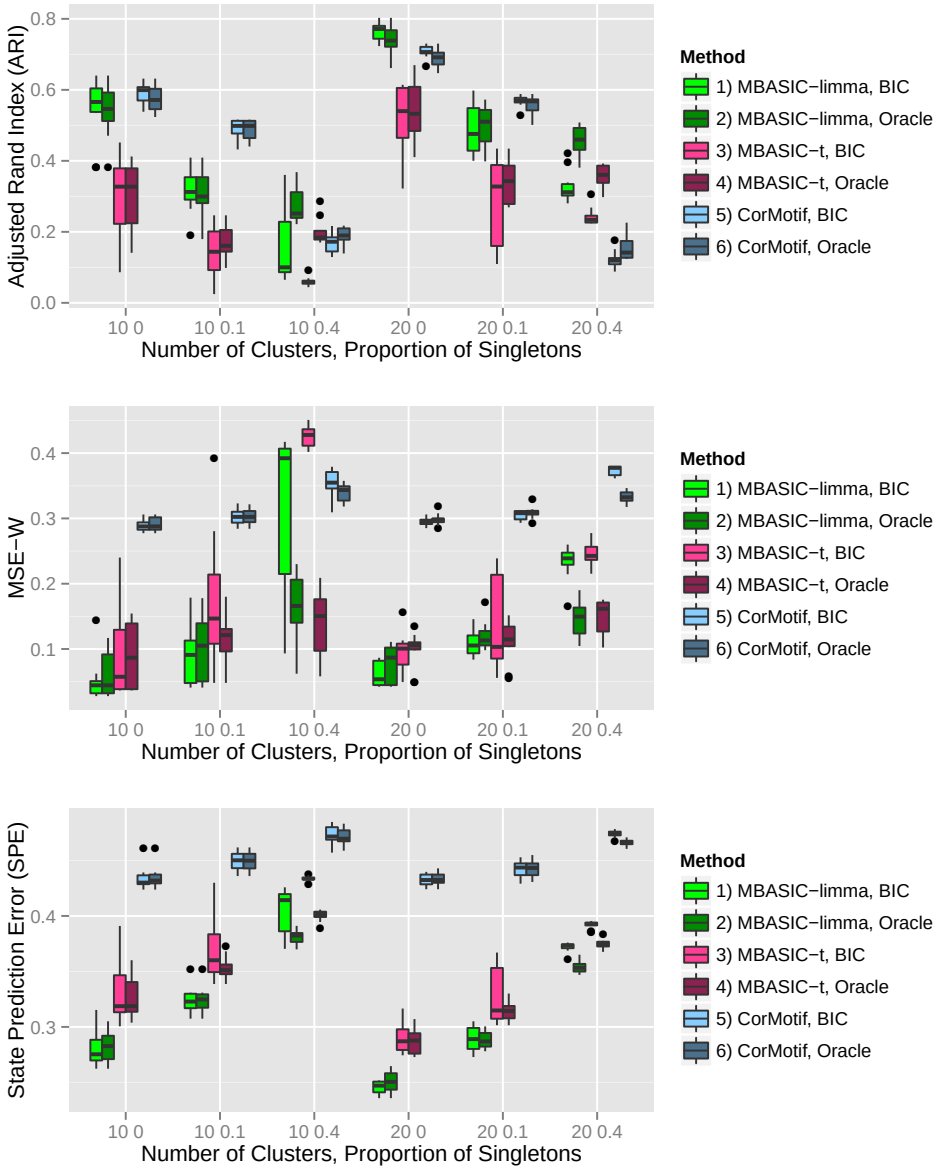


Figure 4.8 *Simulation Study 5, comparison between MBASIC and CorMotif.* I varied the number of clusters at 10, 20 and the proportion of singletons at 0, 0.1 and 0.4. Results are summarized over 10 simulations under each setting.

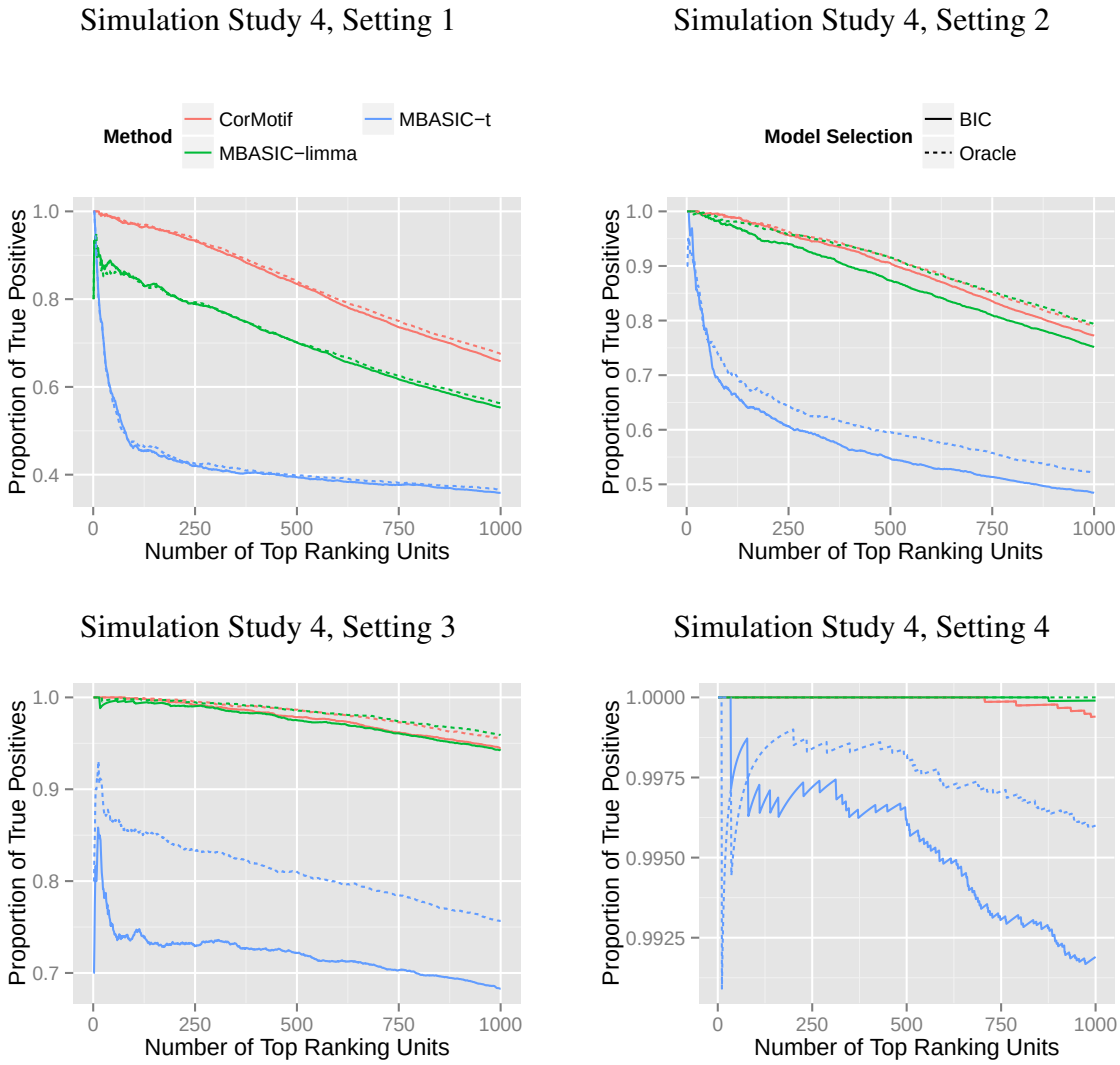


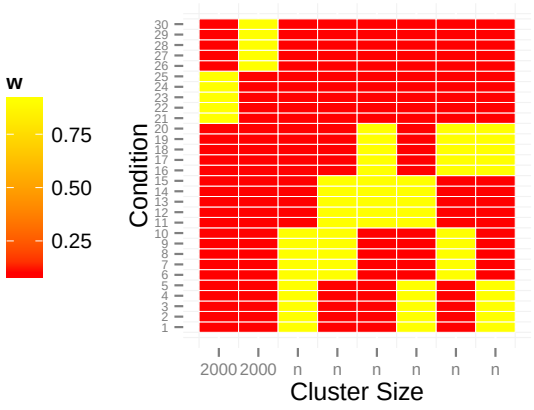
Figure 4.9 *Simulation Study 4, comparison between MBASIC and CorMotif.* The average true positive rate for the 10 simulations among the top 1000 significant unit-experiment pairs for each of the four simulation settings in Table 4.5. For each simulation, the "true positive" consists of $(i,\ k)$'s with $\theta_{ik} = 2$, and the significance is based on the posterior probability $P(\theta_{ik} \neq 2|Y)$.

the singleton cluster in fitting MBASIC models. MBASIC-limma performed the best except in the first setting, where CorMotif achieved the best SPE. For each of the four settings, I show the average true positive rate in detecting states with $\theta_{ik} = 2$ among the top 1000 significant units as in Figure 4.9. Except for Setting 1, MBASIC-limma performed equally well as CorMotif. I note that Setting 1 has the fewest clusters $J = 4$ and the fewest experimental conditions $K = 4$, while the other settings have more complicated state-space structures. Performance of MBASIC-t was the worst in all the four settings. This suggests that neglecting the heterogeneity in these cases can significantly increase estimation error. Although MBASIC model alone does not address the heterogeneity issue, fitting MBASIC models after a data pre-processing step that incorporates the heterogeneity structure, such as computing $\tilde{T}_{ik}$ in MBASIC-limma, can significantly improve model inference. In Simulation Study 5 where I had stronger signals in separating distribution components but noisy state-space clusters, CorMotif resulted in the largest SPE values in all settings (Figure 4.8). Although its performance in ARI was comparable with MBASIC-limma when $\zeta \leq 0.1$, it deteriorated with increasing proportion of singletons, i.e., $\zeta = 0.4$. Simulation Studies 4 and 5 collectively suggest that MBASIC's performance is competitive with CorMotif in settings where I have less noise in clustering structure, small numbers of clusters, and some level of singletons despite the fact that the distributional assumptions of MBASIC might be mis-specified. This indicates that for real data sets where I am agnostic about the true data generating structure, MBASIC might be a more general and robust approach.

4.4.4 Simulation Study 6: Weak Clusters

[73] pointed out that in state-space clustering without accommodating singletons, when the size of one cluster is small, it could be merged with other clusters with distinct state-space profiles. In other words, the estimated cluster profiles could be parsimonious representation of the true underlying cluster structure. Such a phenomenon may alter the interpretation of the W matrix, because each column may represent the average state-space pattern of several small clusters that lack the data support. It is therefore important to investigate whether such a phenomenon still exists for MBASIC where I include a singleton clusters.

(a) Simulations 1-3



(b) Simulations 4-6

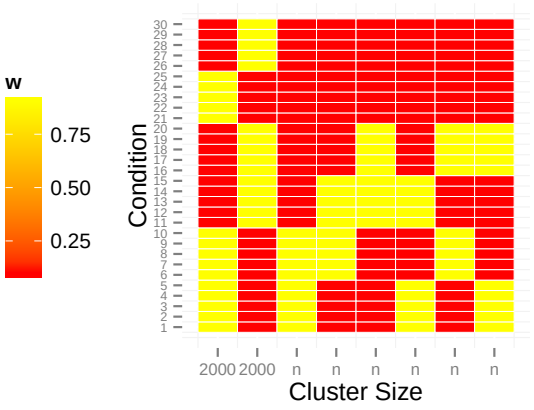


Figure 4.10 *Simulation Study 6.* Two settings of the true cluster patterns, represented by the matrix $(w_{jk2})^T_{1\leq j\leq 8,\ 1\leq k\leq 30}$.

I conducted six simulations in Simulation Study 6 to investigate this issue. I simulated data according to the log-normal distribution with $S = 2$ states, $J = 8$ clusters, and $K = 30$ conditions. The number of replicates within each condition, as well as the distribution parameters within each state is the same as in Section 4.4.1.1. I set the sizes of the first two clusters as 2000, and varied the size of each of the other six clusters n_{small} as 20 or 100. To vary the level of state-space similarity between the two big clusters and the small clusters, I had two settings for the state-space pattern, shown in Figure 4.10. For Simulations 1-3, the conditions in which the small clusters have state $s = 2$ are distinct from the two big clusters, while for Simulations 4-6, the patterns between the small and large clusters are more similar. To control these cluster patterns, I set $w_{jks} \in \{0.1,\ 0.9\}$. Finally, I included $n_{\text{singleton}} = 0$ or 2000 singletons in each simulated data set. The states for the singleton units were generated the same as in Section 4.4.1.1.

In each simulation, I fitted the data using MBASIC and MBASIC0 with BIC to select the number of clusters. MBASIC0 differs from MBASIC only by the exclusion of the singleton feature. Therefore, comparing these two methods allow us to assess how fitting a singleton cluster may affect the small cluster merging problem. I then compare the confusion matrices between the fitted and the true clusters in Tables 4.7-4.9. I also display the state-space patterns in the fitted models in Figures 4.11 and 4.12. In Simulations 1 and 4, with $n_{\text{small}} = 20$ units in each of the small clusters, MBASIC classified them as singletons, while MBASIC0 merged them to form a spurious cluster. The state-space pattern estimated by MBASIC represented the two real big clusters. When the data included singletons, as in Simulations 2 and 5, MBASIC0 formed more spurious clusters, while MBASIC continued to allocate the small clusters as singletons. When I had $n_{\text{small}} = 100$ units in each of the small clusters, both methods identified these small cluster patterns in Simulation 6 (Figure 4.12), but formed spurious clusters in Simulation 3 (Figure 4.11). I compare the ARI, MSE-W and SPE between MBASIC and MBASIC0 in Table 4.10. Performances of these two methods are close when I have no singletons, but differentiate otherwise. From these simulations, I conclude that fitting a singleton cluster can substantially avoid merging weak clusters. The state-space patterns estimated by the W matrix are more likely to reflect true underlying clusters rather than the average of several small clusters. I acknowledge that how well modeling the singletons can

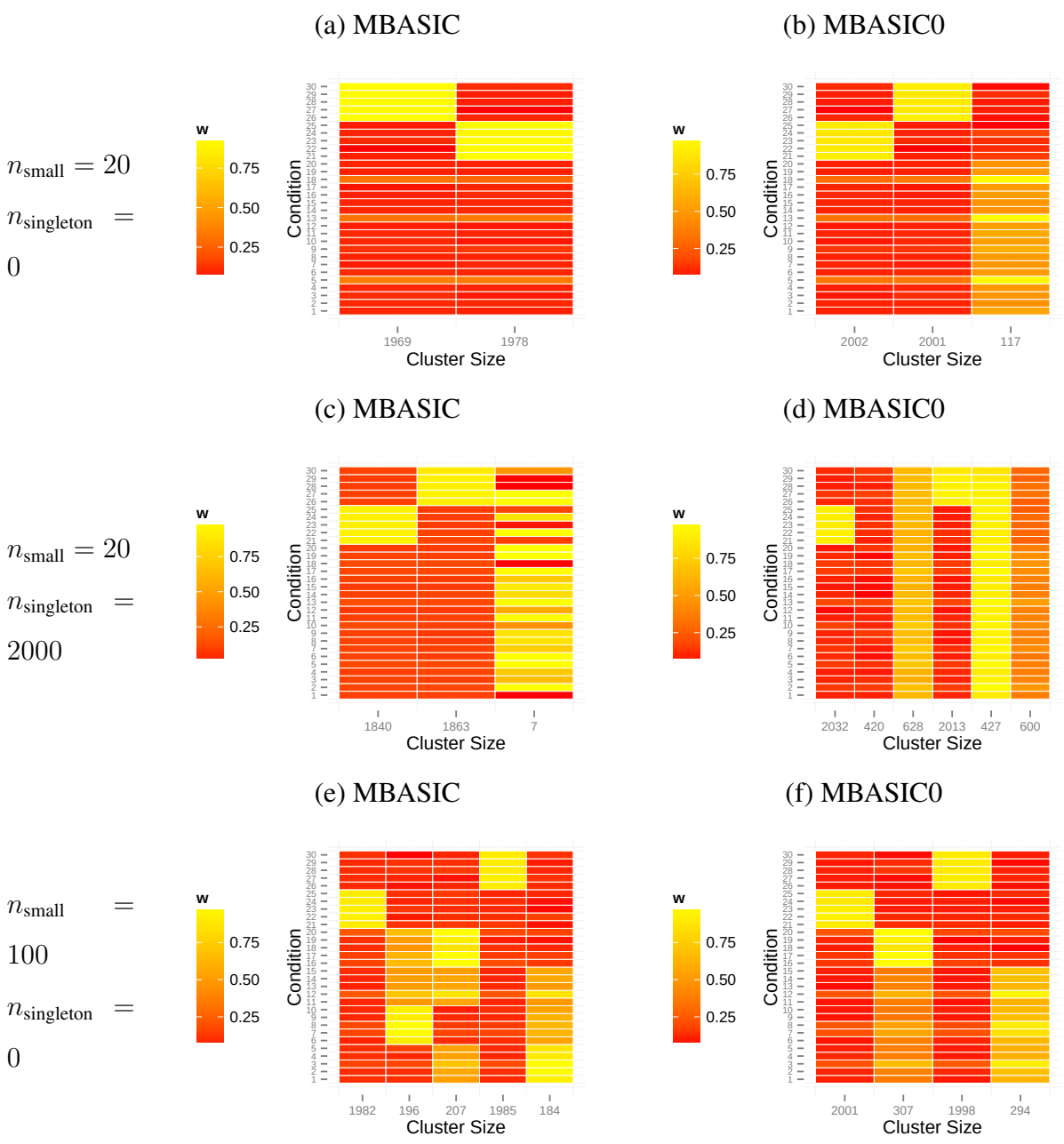


Figure 4.11 *Simulation Study 6, Simulations 1-3.* Estimated cluster patterns by (a, c, e) MBASIC and (b, d, f) MBASIC0. The true clustering pattern is shown in Figure 4.10(a).

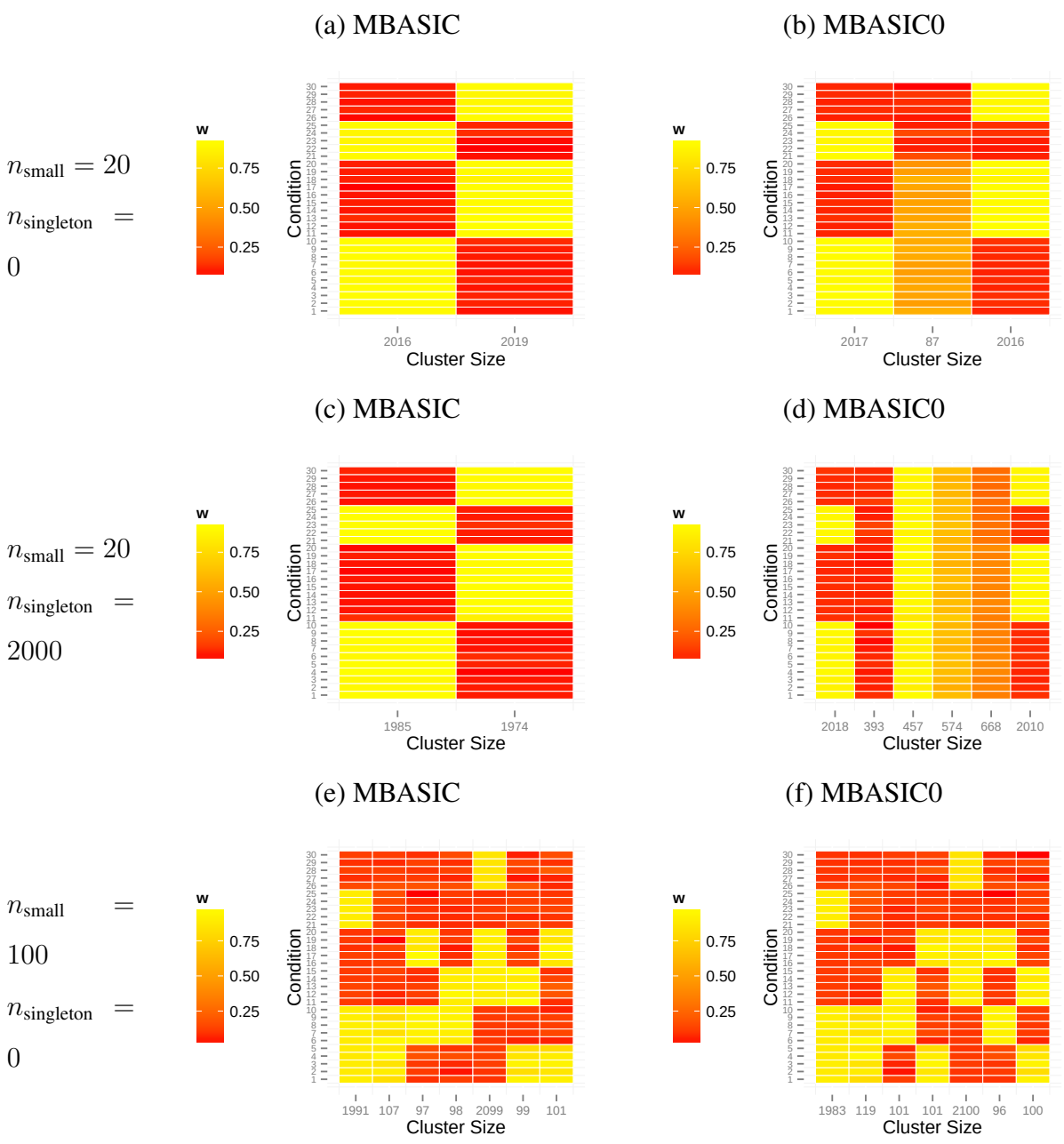


Figure 4.12 *Simulation Study 6, Simulations 4-6.* Estimated cluster patterns by (a, c, e) MBASIC and (b, d, f) MBASIC0. The true clustering pattern is shown in Figure 4.10(b).

Table 4.7 *Simulation Study 6, Simulations 1 and 4.* Confusion matrix between the true clusters and the estimated clusters. The true cluster pattern is shown in Figure 4.10.

Simulation 1, $n_{\text{small}} = 20$, $n_{\text{singleton}} = 0$

	MBASIC		
True	0	1	2
1	23	2	1975
2	30	1967	3
3	20	0	0
4	20	0	0
5	20	0	0
6	20	0	0
7	20	0	0
8	20	0	0

	MBASIC0		
True	1	2	3
1	1998	2	0
2	4	1995	1
3	0	1	19
4	0	1	19
5	0	0	20
6	0	1	19
7	0	0	20
8	0	1	19

Simulation 4, $n_{\text{small}} = 20$, $n_{\text{singleton}} = 0$

	MBASIC		
True	0	1	2
1	5	1995	0
2	1	0	1999
3	1	19	0
4	19	1	0
5	1	0	19
6	20	0	0
7	18	1	1
8	20	0	0

	MBASIC0		
True	1	2	3
1	1999	1	0
2	0	0	2000
3	17	3	0
4	0	20	0
5	0	5	15
6	0	20	0
7	0	19	1
8	1	19	0

Table 4.8 *Simulation Study 6, Simulations 2 and 5.* Confusion matrix between the true clusters and the estimated clusters. The true cluster pattern is shown in Figure 4.10.

Simulation 2, $n_{\text{small}} = 20$, $n_{\text{singleton}} = 2000$

	MBASIC			
True	0	1	2	3
0	1957	19	17	7
1	180	1818	2	0
2	153	3	1844	0
3	20	0	0	0
4	20	0	0	0
5	20	0	0	0
6	20	0	0	0
7	20	0	0	0
8	20	0	0	0

	MBASIC0					
True	1	2	3	4	5	6
0	67	372	620	59	427	455
1	1957	25	0	4	0	14
2	8	23	3	1950	0	16
3	0	0	1	0	0	19
4	0	0	0	0	0	20
5	0	0	1	0	0	19
6	0	0	1	0	0	19
7	0	0	1	0	0	19
8	0	0	1	0	0	19

Simulation 5, $n_{\text{small}} = 20$, $n_{\text{singleton}} = 2000$

	MBASIC		
True	0	1	2
0	1986	7	7
1	31	1969	0
2	45	0	1955
3	11	9	0
4	20	0	0
5	8	0	12
6	20	0	0
7	20	0	0
8	20	0	0

	MBASIC0					
True	1	2	3	4	5	6
0	16	393	457	561	560	13
1	1990	0	0	5	5	0
2	0	0	0	5	12	1983
3	12	0	0	1	7	0
4	0	0	0	1	19	0
5	0	0	0	0	6	14
6	0	0	0	0	20	0
7	0	0	0	0	20	0
8	0	0	0	1	19	0

Table 4.9 *Simulation Study 6, Simulations 3 and 6.* Confusion matrix between the true clusters and the estimated clusters. The true cluster pattern is shown in Figure 4.10.

Simulation 3, $n_{\text{small}} = 100$, $n_{\text{singleton}} = 0$

	MBASIC					
True	0	1	2	3	4	5
1	21	1974	0	0	5	0
2	12	7	0	1	1980	0
3	1	1	9	0	0	89
4	5	0	89	0	0	6
5	6	0	2	92	0	0
6	1	0	0	13	0	86
7	0	0	96	4	0	0
8	0	0	0	97	0	3

	MBASIC0			
True	1	2	3	4
1	1993	0	7	0
2	7	1	1991	1
3	1	0	0	99
4	0	2	0	98
5	0	99	0	1
6	0	6	0	94
7	0	100	0	0
8	0	99	0	1

Simulation 6, $n_{\text{small}} = 100$, $n_{\text{singleton}} = 0$

	MBASIC									MBASIC0						
True	0	1	2	3	4	5	6	7	True	1	2	3	4	5	6	7
1	3	1983	14	0	0	0	0	0	1	1976	24	0	0	0	0	0
2	1	0	0	0	0	1999	0	0	2	0	0	0	0	2000	0	0
3	0	8	92	0	0	0	0	0	3	7	93	0	0	0	0	0
4	1	0	0	1	98	0	0	0	4	0	0	99	0	0	1	0
5	1	0	0	0	0	99	0	0	5	0	0	0	0	99	0	1
6	1	0	0	0	0	0	99	0	6	0	0	1	0	0	0	99
7	1	0	1	96	0	1	0	1	7	0	2	1	1	1	95	0
8	0	0	0	0	0	0	0	100	8	0	0	0	100	0	0	0

Table 4.10 *Simulation Study 6.* ARI, MSE-W, and SPE in all simulations.

			MBASIC			MBASIC0		
Simulation	n_{small}	$n_{\text{singleton}}$	ARI	MSE-W	SPE	ARI	MSE-W	SPE
1	20	0	0.967	0.433	0.185	0.991	0.29	0.189
2	20	2000	0.79	0.442	0.192	0.727	0.34	0.193
3	100	0	0.969	0.203	0.187	0.975	0.233	0.193
4	20	0	0.977	0.384	0.169	0.983	0.260	0.167
5	20	2000	0.926	0.384	0.185	0.773	0.333	0.184
6	100	0	0.949	0.087	0.172	0.947	0.087	0.171

avoid merging weak clusters requires further investigation in more dynamic settings as I vary the similarity among clusters, the difference among the states, as well as other variables that influence cluster structures such as J, K, S. I leave such potential investigations as future research.

4.5 Applications of MBASIC to Genome Research Problems

4.5.1 Transcription Factor Enrichment Network

Regulation of gene expression relies heavily on the context-specific combinatorial activities of TFs. Gene clustering analysis based on TF occupancy data, i.e., ChIP-seq, aims to identify combinatorial patterns of TF occupancy and group genes based on such patterns. The ENCODE consortium ([67]) has generated TF ChIP-seq datasets for over 100 TFs across multiple cell types, and has motivated several integrative studies for learning regulation patterns ([15], [71]). In this study, I applied MBASIC to the analysis of such data. Specifically, I focused on the TF enrichment patterns at the promoter regions, i.e., -5000 bps and +1000 bps the transcription start site, of the 10290 genes that had significant expression, as measured by RNA-seq, in either the Gm12878 or the K562 cells. The input data to MBASIC were the mapped numbers of reads at these promoter regions from the uniformly processed ChIP-seq data by [15]. I chose the cell types Gm12878 and K562 because they had the largest numbers of TF ChIP-seq experiments. The final dataset utilized included ChIP-seq data for $I = 10290$ observational units over 30 TFs corresponding to $K = 60$ experimental conditions (cell type $\times$ TF) with a total of 166 replicate experiments.

I fitted MBASIC with $S = 2$ states and used log normal distributions as in Equation (4.3). $s = 1$ corresponded to the unenriched state, and I let $\gamma_{ikl1} = \log(1 + x_{ik})$, where x_{ik} is the count from the matching control experiment at unit i. $s = 2$ corresponded to the enrichment state, and I let $\gamma_{ikl2} = 1$ for all loci.

I followed the two-phase procedure using BIC from Section 4.3.4 to select both the number of clusters and the structure of each cluster. In Phase 1, I selected the number of clusters as 24. In Phase 2, I considered two types of structural constraints for each cluster, referred to by *TF-homogeneity* and *cell type-homogeneity* and defined as $w_{jk_1s} = w_{jk_2s}$ if k_1 and k_2 corresponded to the same TF or cell type. I found that imposing cell type-homogeneity to any cluster would cause

(a) (b)

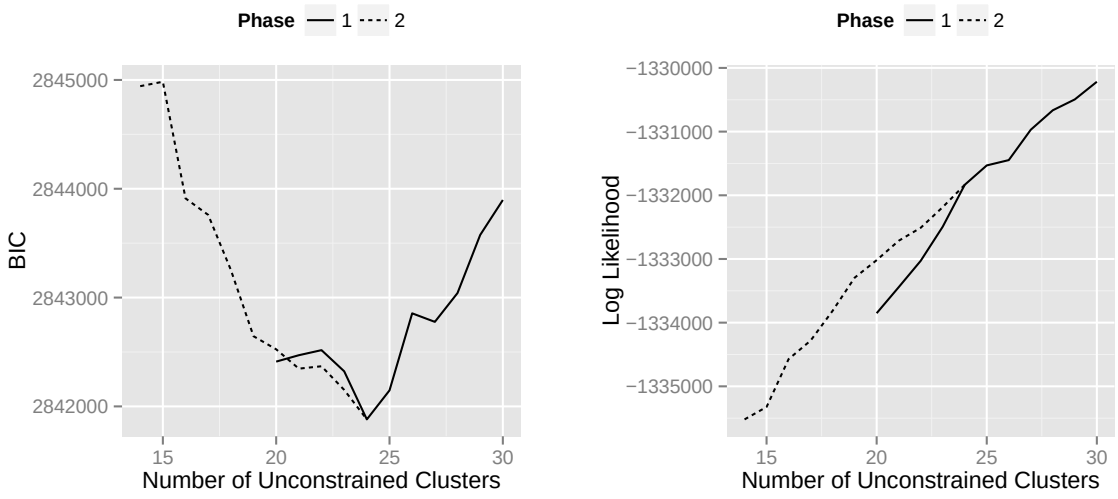


Figure 4.13 (a) BIC and (b) log-likelihood values for models with different structures. All the clusters are unstructured in the Phase 1 models and the x-axis denotes the total number of clusters. The total number of clusters is 24 for Phase 2 models and the x-axis denotes the number of unconstrained clusters. The remaining clusters have TF-homogeneity.

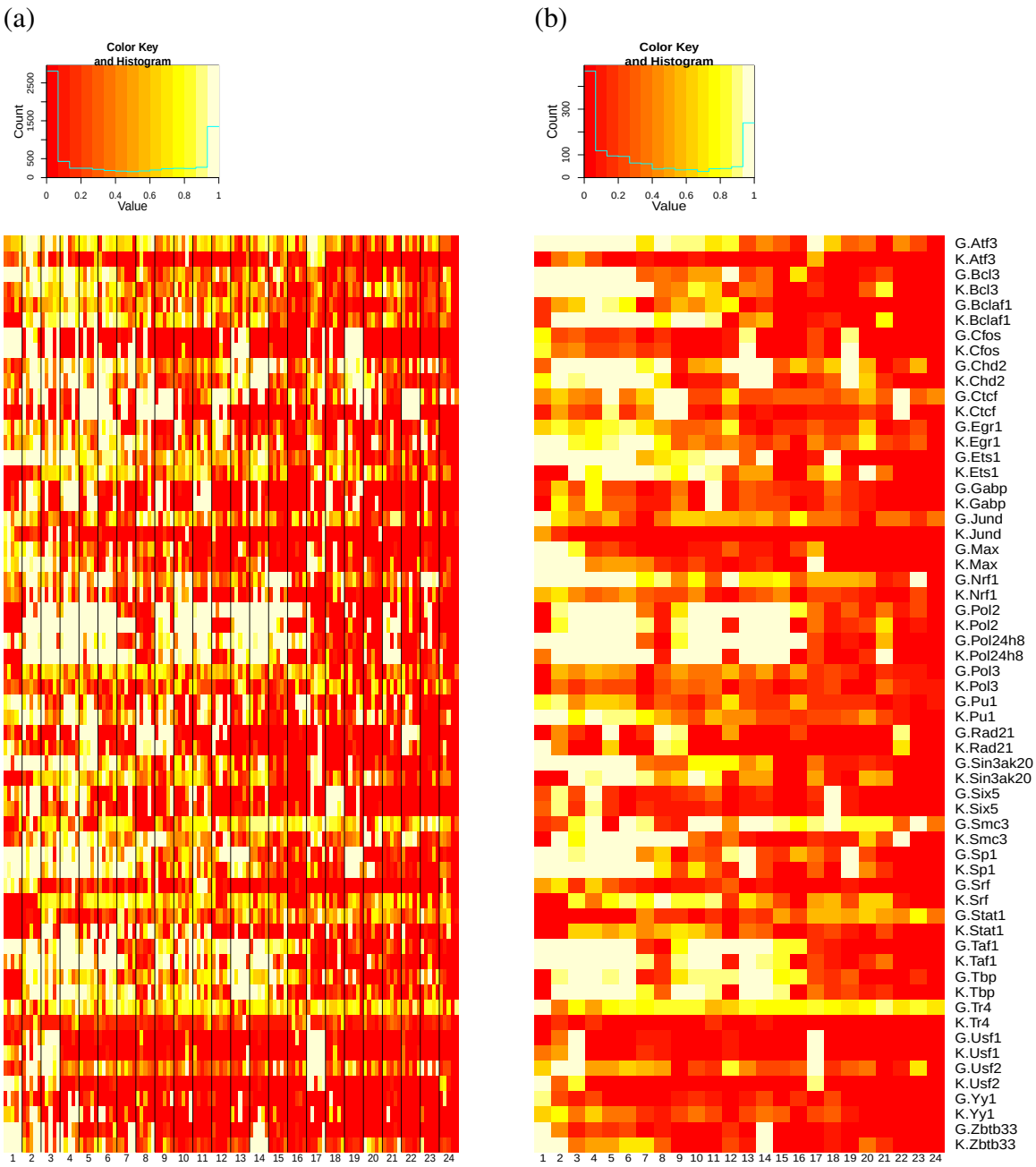


Figure 4.14 (a) Normalized data for each cell-TF combination at five sub-sampled loci within each cluster. (b) Estimated enrichment probability at each cell-TF combination for each cluster.

that cluster to be degenerate (i.e., no unit was assigned to that cluster). Therefore, I chose the final model among those with TF-homogeneity structures. The BIC and log likelihood values for different models fitted in both phases are shown in Figure 4.13. The final model had 24 unconstrained clusters, consisting of $1-\zeta = 89.8\%$ of the 10290 loci. The ranges of the estimated distribution parameters among replicates within the same cell type-TF combination is shown in Figures 4.15 and 4.16. I notice that these parameters can be substantially different among replicated experiments. This provides further support for my replicate specific parametrization.

To compare the normalized data and the predicted enrichment probability for each cluster, I computed the normalized signals [3] and compared them to the estimated cluster parameters. Figure 4.14(a) depicts such normalized signals from five randomly selected loci within each predicted cluster, as well as the predicted enrichment probabilities at the corresponding condition and cluster (w_{jk2}'s). I observe that the estimated enrichment probabilities at the cluster level capture the commonality among loci within each cluster. In addition, each loci cluster exhibits distinct combinatorial patterns of activity across all cell type-TF combination. The cell type-TF combination enriched within each cluster is listed in the Table B.1.

My clustering results are consistent with the existing literature on the TF enrichment networks. For example, cooperating TFs tend to be enriched at the same loci. This pattern can be observed in Figure 4.14 between Bcl3 and Bclaf1. Pol2 and Pol24h8 represent Pol2 experiments with different antibodies. As expected, I observe enrichment at the same loci for these two different version of Pol2 experiments (Figure 4.14(a)). Moreover, pairs of TFs that have similar binding motifs have similar enrichment probabilities over the clusters. For example, [71] discovered the UA1 motif as common to both Chd2 and Ets1 and the USF motif for Max, Usf1, and Usf2. Interactions between Taf1 and Tbp have also been studied by [1]. Similar enrichment probabilities of these TFs across clusters can be observed in Figure 4.14(a). In addition to these observations that are consistent with the literature, my results illustrate how the genome-wide TF association patterns can be attributed

[3] The normalized signal for unit i and condition k is:

$$\tilde{\theta}_{ik} = \frac{\prod_{l=1}^{n_k} f_s(y_{ikl}|\hat{\mu}_{kl2}, \hat{\sigma}_{kl2}, \gamma_{ikl1})}{\prod_{l=1}^{n_k} f_s(y_{ikl}|\hat{\mu}_{kl2}, \hat{\sigma}_{kl2}) + \prod_{l=1}^{n_k} f_s(y_{ikl}|\hat{\mu}_{kl1}, \hat{\sigma}_{kl1}, \gamma_{ikl2})}$$

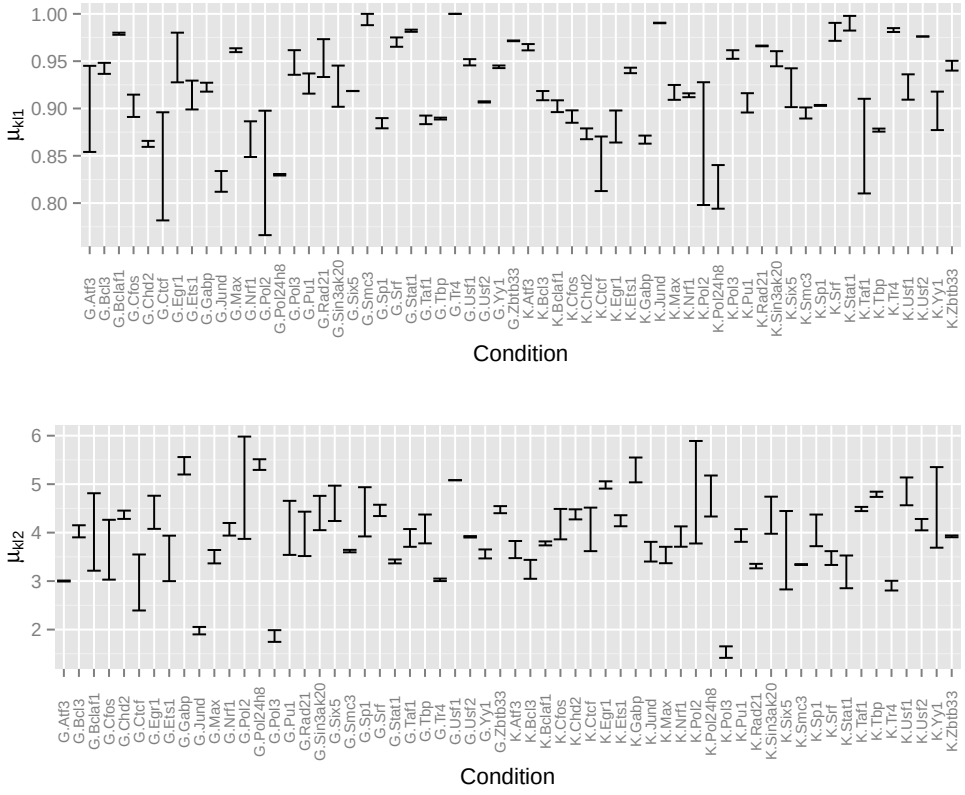


Figure 4.15 The range of the estimated μ_{kls} among the different replicates under the same experimental condition for the transcription factor enrichment network data in Section 4.5.1.

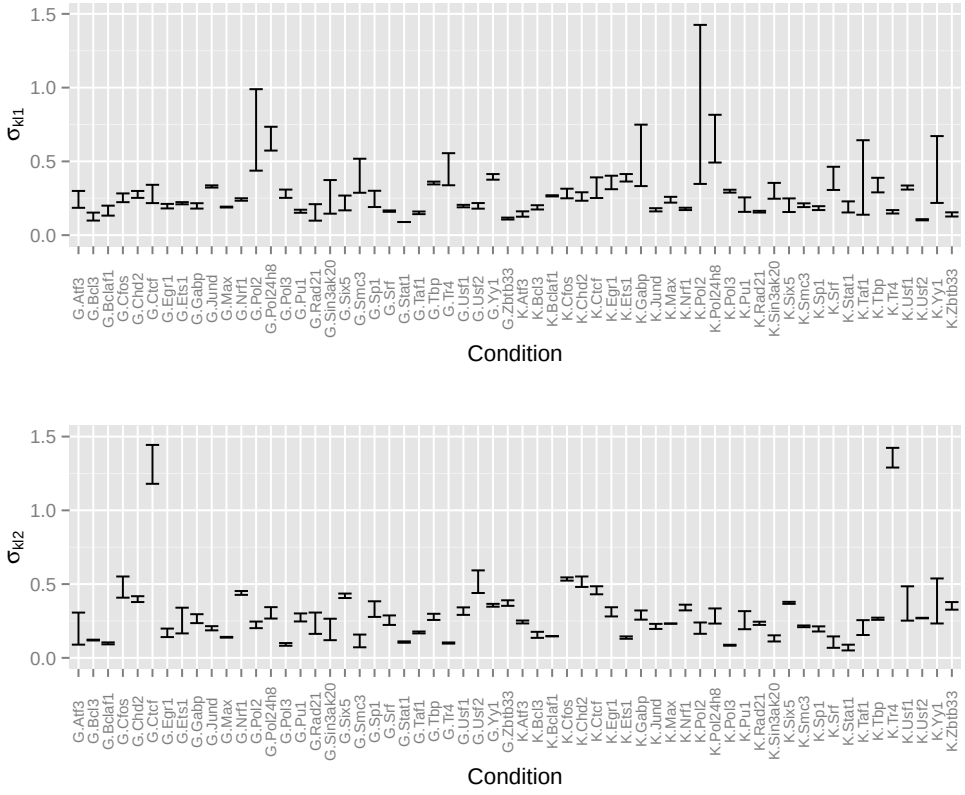


Figure 4.16 The range of the estimated σ_{kls} among the different replicates under the same experimental condition for the transcription factor enrichment network data in Section 4.5.1.

to specific clusters. I explored the loci clusters with distinct patterns between cell types (e.g., Pol2 in Cluster 12, Figure 4.19), TFs from the same families (e.g., Bcl3 v.s. Bclaf1 in Cluster 3, Figure 4.17), and TFs with similar genome-wide enrichment (e.g., Max v.s. Usf1 in Cluster 2, Figure 4.18) using raw data.

I further evaluated each cluster of genes for their KEGG pathway enrichment ([66]), and identified 8 KEGG pathways that are significantly enriched in individual clusters (Table 4.11). Three of my clusters (Clusters 7, 9, and 19) have more than half of their genes in one single pathway. Since KEGG pathways curate the known knowledge of molecular interaction systems, these clusters may be driven by unknown biological processes that warrant further investigation.

Table 4.11 Significantly enriched KEGG pathways across the 24 clusters.

KEGG.name	# Genes Overlapped	Z Score	Cluster	Cluster Size
Protein processing in endoplasmic reticulum	156	5.652	7	391
Fatty acid elongation in mitochondria	7	7.518	8	133
B cell receptor signaling pathway	74	6.016	9	146
Lysine biosynthesis	3	6.53	9	146
D-Glutamine and D-glutamate metabolism	3	5.548	12	184
Vitamin B6 metabolism	4	5.28	14	156
Non-homologous end-joining	12	7.539	17	213
Lysosome	116	5.402	19	187

MBASIC infers the clustering structure based on its own estimates of the state-space profiles. The ENCODE consortium provides the estimated enrichment regions (i.e., *peaks*) for each experiment in this study. Then, a natural question is whether MBASIC reveals more information compared to clustering of genes based on ENCODE-estimated binary enrichment profiles of TFs. To address this, I created a binary vector for each gene by overlapping its promoter with the ENCODE peaks. Then, I applied the state-of-the-art MClust model ([12]) to cluster the 10290 promoter regions based on these peak profiles. MClust selected 90 clusters based on BIC. Figure 4.20

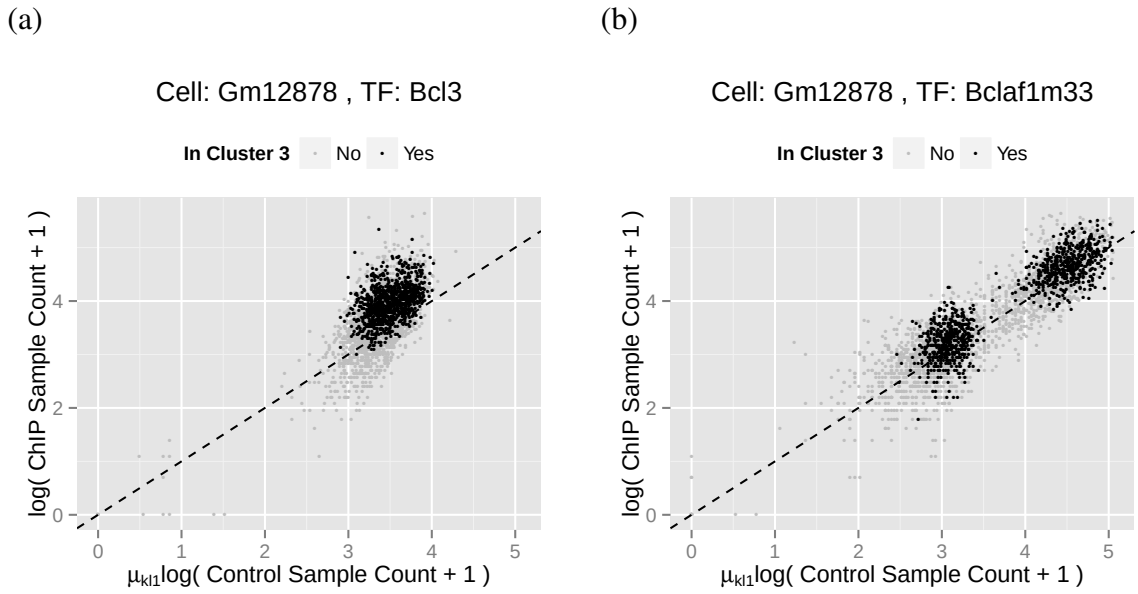


Figure 4.17 Plots of the transformed ChIP sample read counts against the transformed control sample read counts for all units in the Gm12878 cell for (a) Bcl3 and (b) Bclaf1. Data from unenriched units are expected to locate around the 45 degree dashed line.

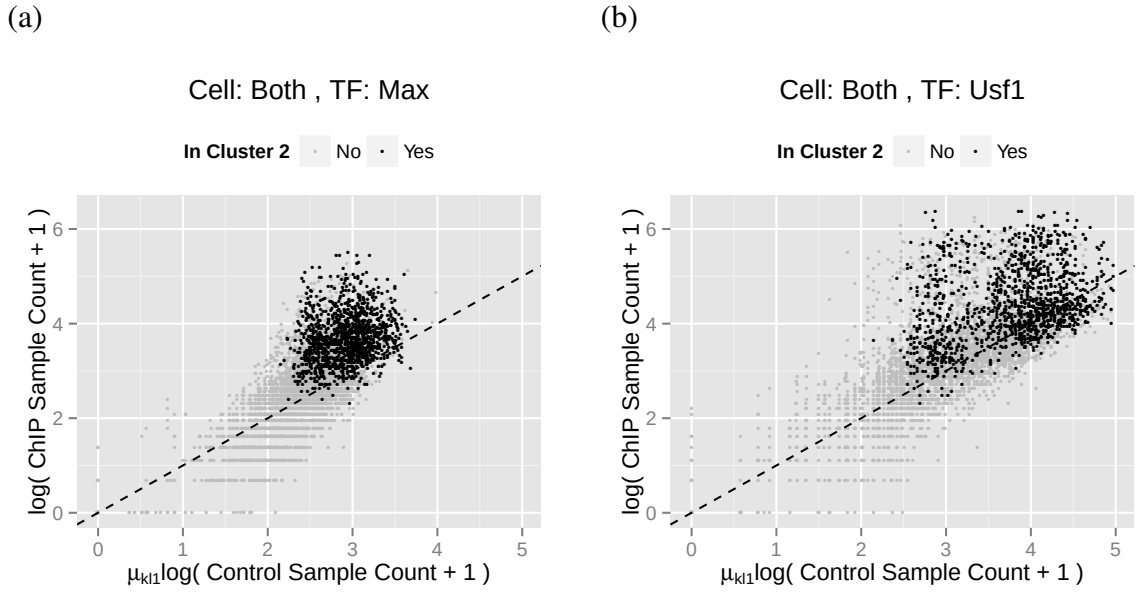


Figure 4.18 Plots of the transformed ChIP sample read counts against the transformed control sample read counts for all units in both Gm12878 and K562 cells for (a) Max and (b) Usf1. Data from unenriched units are expected to locate around the 45 degree dashed line.

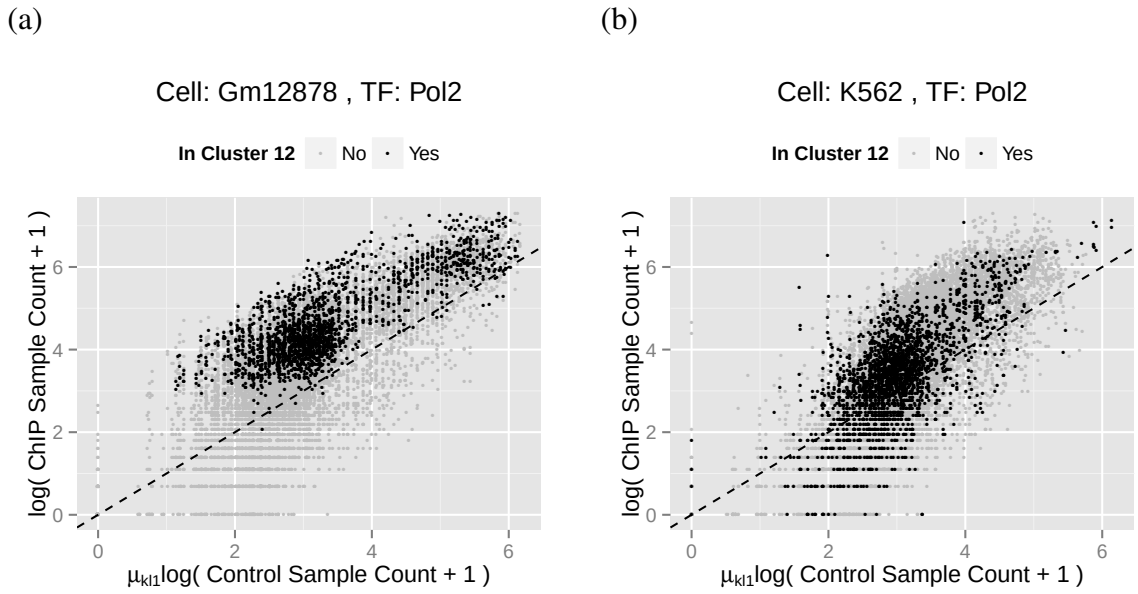


Figure 4.19 (a, b) Plots of the transformed Pol2 ChIP sample read counts against the transformed control sample read counts for all units in (a) Gm12878 and (b) K562 cells. Data from unenriched units are expected around the 45 degree dashed line.

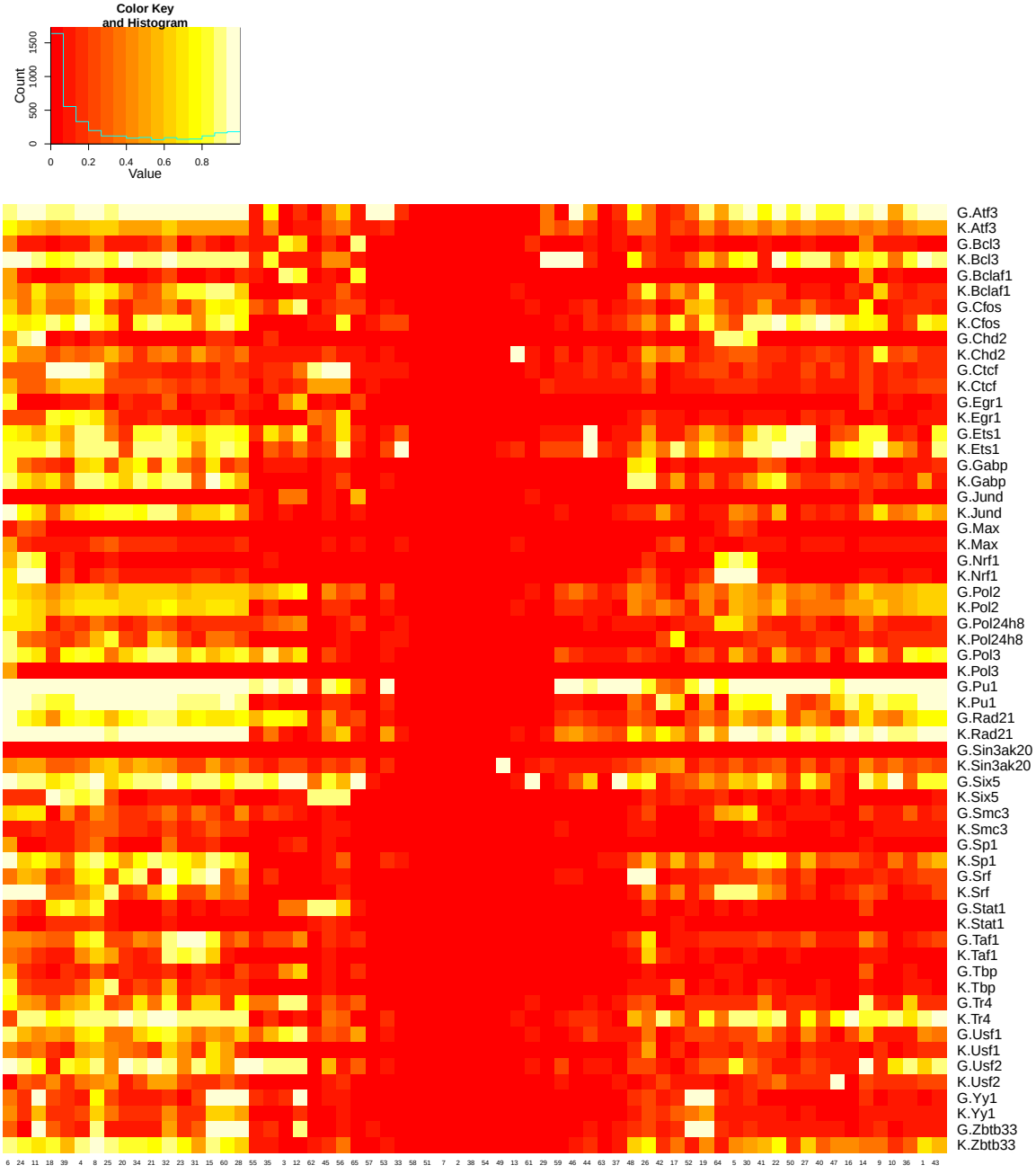


Figure 4.20 Estimated enrichment probability for each of the 90 clusters identified by MClust.

displays cluster-level estimated enrichment probabilities of TFs across the conditions considered. Compared to Figure 4.14, I can see that many of the MClust clusters have very similar enrichment profiles. For example, Clusters 51, 7, 8, 32, 54 contained almost no enrichment for any TFs, but are classified as distinct clusters. The association between units across these clusters are thus non-trivial to interpret. In addition, I found that for some conditions, the enrichment states predicted by MBASIC are quite different than those from the ENCODE peak profiles (e.g., Figure 4.21). This is because the ENCODE peaks are identified by whole genome-wide analysis and may not reflect the differences between the ChIP and control samples at the local promoter regions. MBASIC attains larger raw data fidelity by directly modeling the counts at each unit rather than inheriting results from existing analyses.

4.5.2 Genome-wide Identification of +9.5-like Composite Elements

[27] and [14] described the requirement of the intronic +9.5 site, an Ebox-GATA composite element located at chr6: 88143884-88157023 in the mouse genome (genome version mm9), to establish the hematopoietic stem/progenitor cell (HSC) compartment in the fetal liver and for hematopoietic stem cell genesis in the aorta-gonad-mesonephros (AGM), respectively. Furthermore, [27] and [22] showed that heterozygous +9.5 mutations cause a human immunodeficiency associated with myelodysplastic syndrome (MDS) and acute myeloid leukemia (AML). Because the +9.5 site is the only known *cis*-element deletion of which depletes fetal liver HSCs and is lethal at E13-14 of embryogenesis, identifying additional loci that have similar functionality is extremely important for establishing mechanisms that enable GATA factor-bound regions with nonredundant activity and have the potential to reveal novel targets for therapeutic modulation of hematopoiesis. In this application, I identified 4803 genomic regions with the Ebox-GATA motif (CATCTG-N[7-9]-AGATAA where N[7-9] denotes a variable size spacer of 7 and 9 nucleotides) in the human genome (genome version hg19). I considered a 150 bps window anchored at each of the 4803 composite elements as the observational unit. To analyze the TF occupancy activities at these units and identify a group of composite elements with occupancy profiles similar to that of the +9.5

(a) (b)

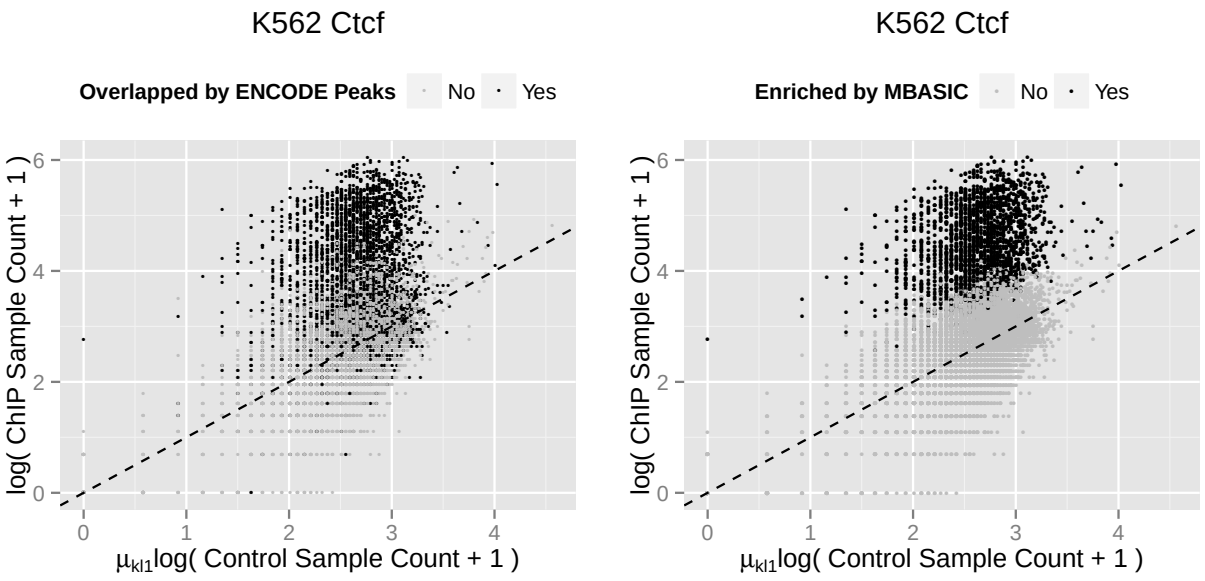


Figure 4.21 (a, b) Transformed ChIP versus control sample read counts from a Gm12878-Ctcf dataset. Enrichment states are annotated by (a) ENCODE peak profiles and (b) MBASIC estimation. In MBASIC, an observational unit is estimated to be enriched if its enrichment probability satisfies $P(\theta_{ik} = 2|Y) > 0.5$.

composite element, I downloaded all ChIP-seq data for the Huvec and K562 cells from [15]. In total, the data set contained 224 replicates spanning $K = 84$ experimental conditions and 77 TFs.

I used negative binomial distributions with $S = 2$ states, where $s = 1$ denoted the unenriched (unoccupied) state, in the MBASIC framework. I chose $\gamma_{ikl1} = 1 + x_{ik}$, where x_{ik} is the count for unit i from the matching control experiment for condition k, to incorporate data from the accompanying control experiments of the ChIP samples. For $s = 2$, I utilized the following mixture distribution to account for the heavy tails observed in the raw data:

$$Y_{ikl} - 3|\theta_{ik} = 2 \overset{ind.}{\sim} \nu_{ikl} NB(\mu_{kl2}, \sigma_{kl2}) + (1 - \nu_{ikl}) NB(\mu_{kl3}, \sigma_{kl3}),$$

$$\nu_{ik} \overset{\text{i.i.d.}}{\sim} Bernoulli(v_{kl}).$$

Here, the constant 3 represents the minimum count threshold for enrichment estimation. The use of mixture distributions to capture heavy tailed count data was previously considered by [85]. I note that an alternative approach to capture heavy tailed counts would be to fit a model using $S = 3$ states, with $s = 2, 3$ representing two distinct enrichment components. Such an approach would differ from the proposed approach in a subtle yet important way. In this alternative approach, allocation of each unit to different enrichment components would affect the clustering estimation, while in my approach, clustering is only determined by the enrichment status of the individual unit regardless of which enrichment component it follows. The E-M algorithm for this setting requires a slight modification as discussed in Section 4.5.1.

Following the two phase model selection procedure using BIC, I selected the model with 3 clusters, 2 of which were cell type-homogeneous. The ranges of the estimated distribution parameters among replicates within the same condition are displayed in Figures 4.23 and 4.24. The three clusters (denoted by C1, C2, and C3) included 332, 837, 157 composite elements, respectively, and the remaining 3477 composite elements were identified as singletons. A heatmap for the enrichment probability of each unit under each cell type-TF combination across the three clusters is shown in Figure 4.22. The +9.5 element is a member of cluster C3 which consists of a total of 157 +9.5-like composite elements. A detailed genomic annotation of these elements are provided in

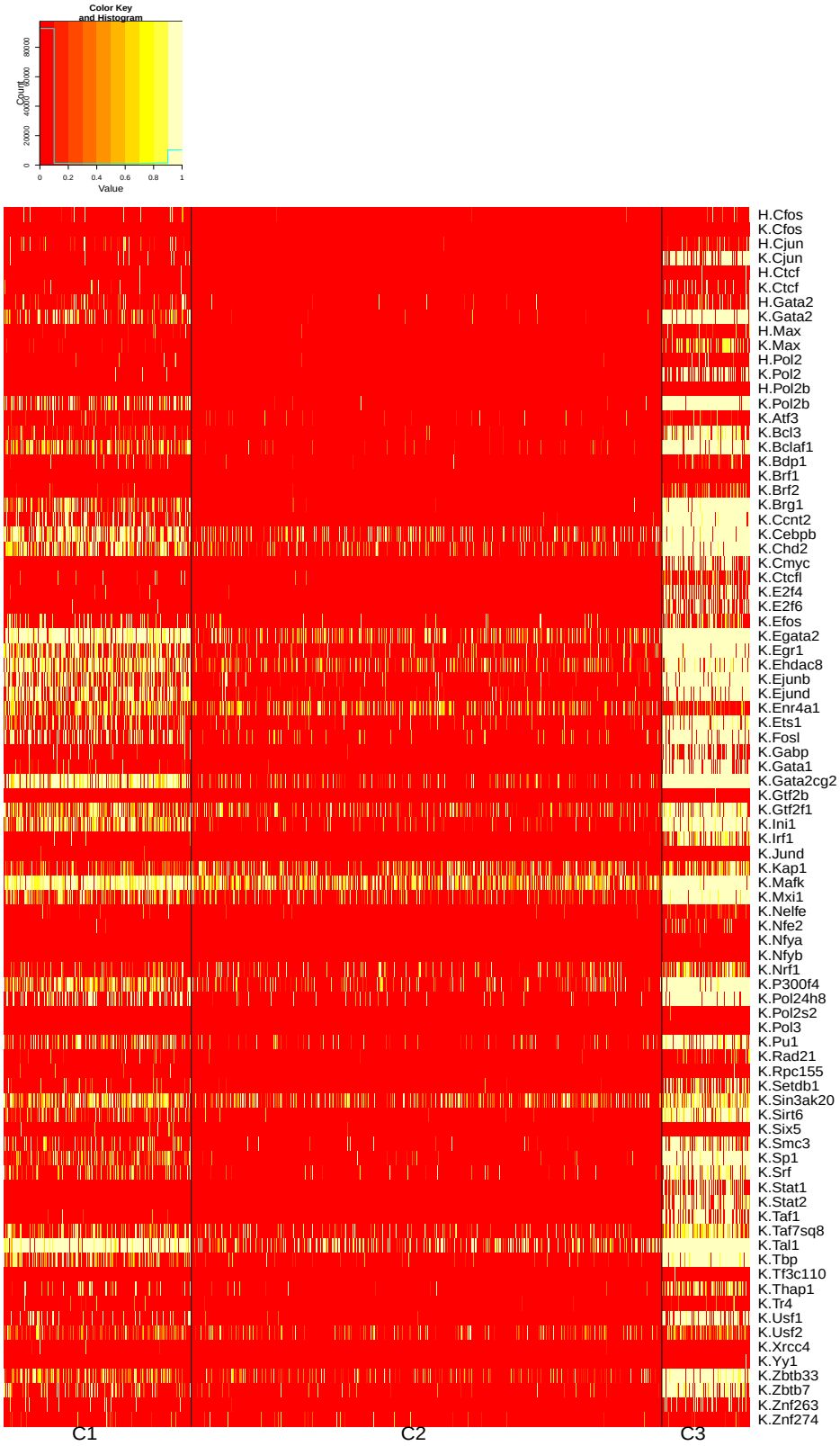


Figure 4.22 Posterior enrichment probability (i.e,. $P(\theta_{ik} = 2|Y)$) for all units in the three clusters. The right most column of the C3 cluster corresponds to the +9.5 element.

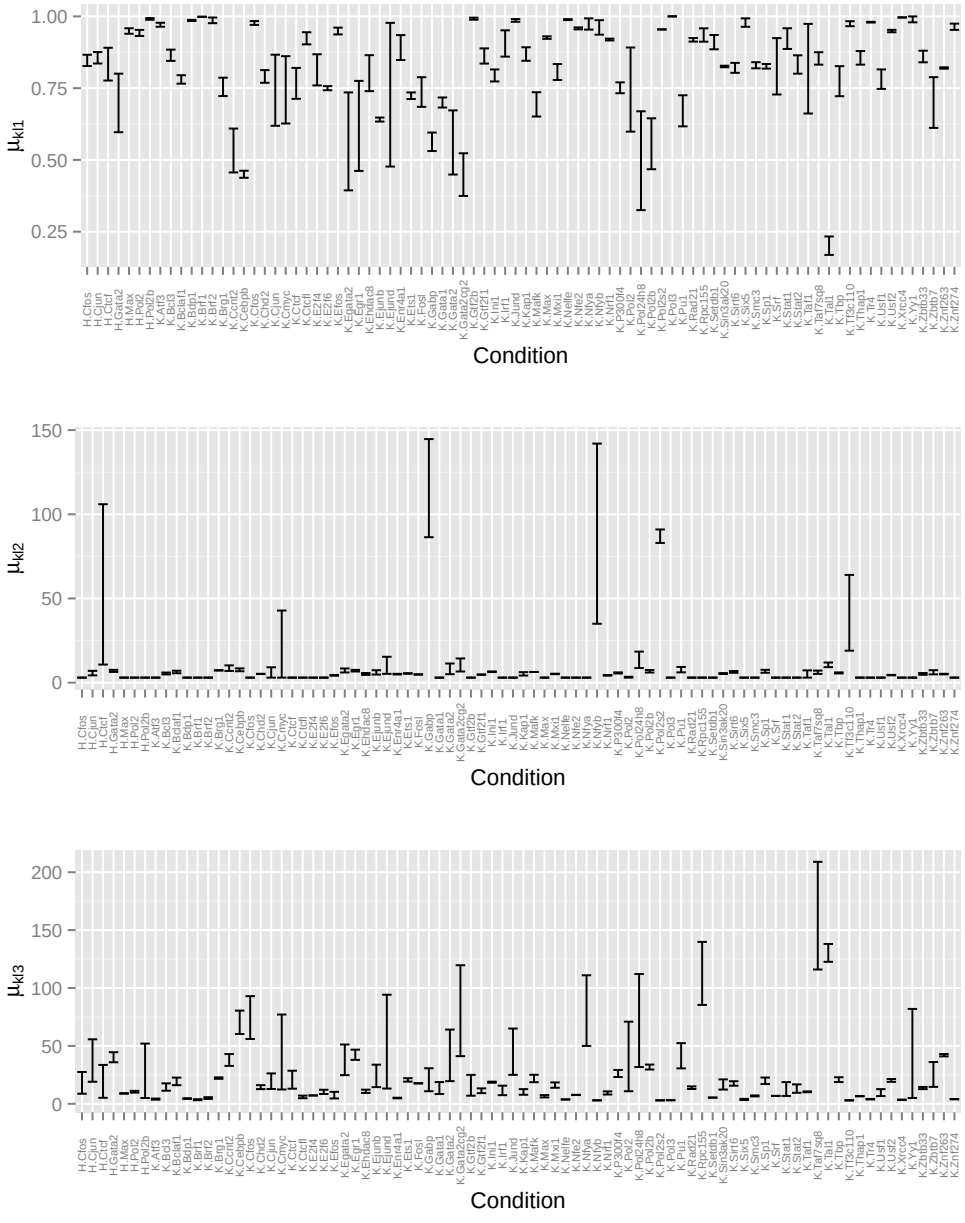


Figure 4.23 The range of the estimated parameters μ_{kls} among the different replicates under the same experimental condition for the +9.5 composite element data in Section 4.5.2.

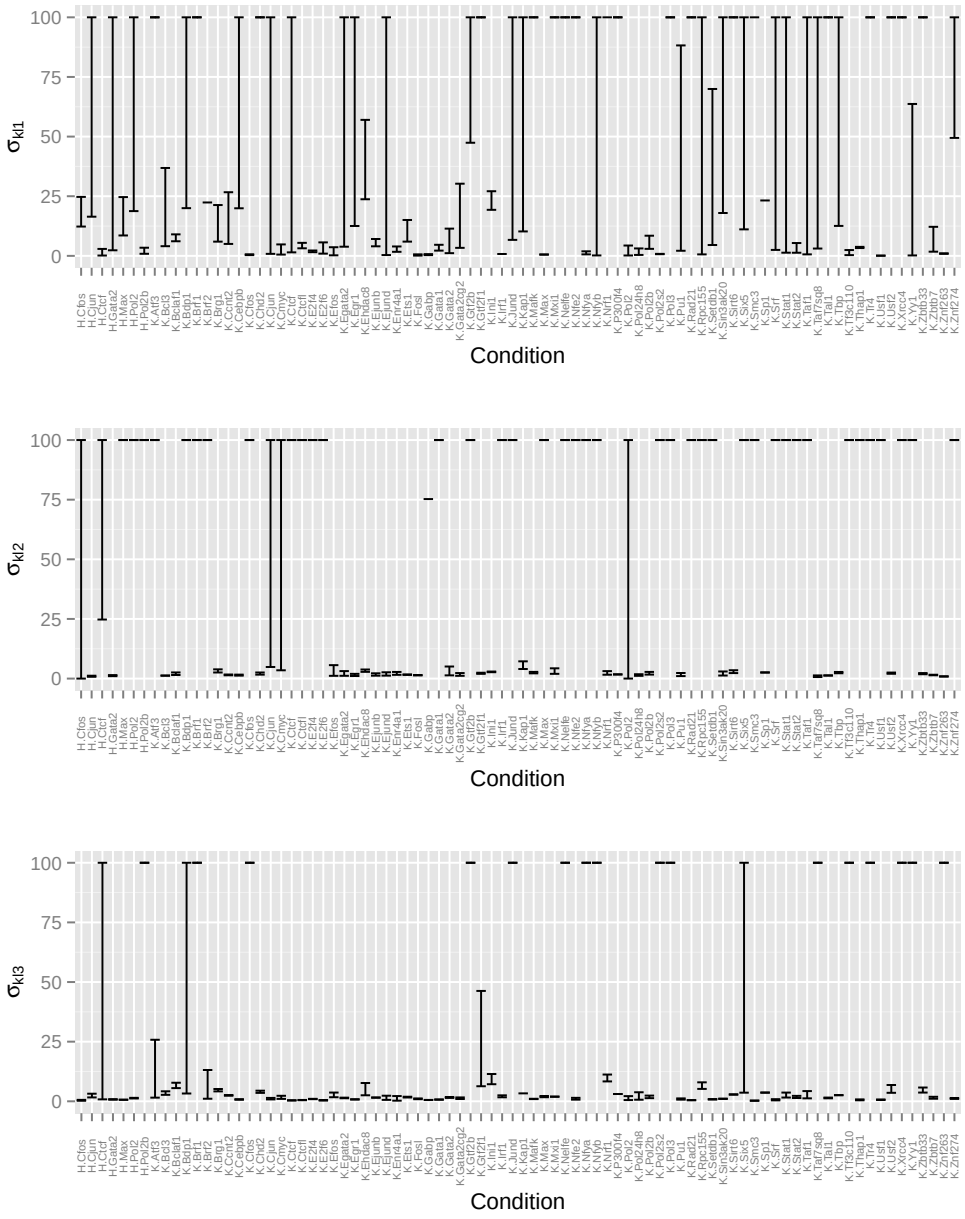


Figure 4.24 The range of the estimated parameters σ_{kls} among the different replicates under the same experimental condition for the +9.5 composite element data in Section 4.5.2. For some replicate, the data for a state can be under-dispersed, resulting in a negative value for the estimated size parameter in the negative binomial distribution. In that case, I set $\sigma_{kls} = 100$. Although my model do not fully capture the under-dispersion patterns, the large variations in σ_{kls} for some conditions suggest that assuming replicate-specific distributions is quite necessary.

Table B.2. Notably, 46% of the C3 elements reside in intronic regions and 42% of these are within first intron. Only 15% of the cluster are located up to 10Kb upstream of transcription start sites.

A detailed analysis of Figure 4.22 reveals that cluster C3 is driven by several transcription factors with known associations to GATA2. First, we note that a large fraction of the C3 loci are bound by BRG1. The chromatin remodeler BRG1 is involved in GATA1-mediated chromatin looping ([31, 32]) and co-localizes with GATA1 at some chromatin sites [23]. BRG1 has broad functions in many cell types; however, conditional knockouts of BRG1 reveal its importance in specific cell and tissue contexts [21]. Another factor that clearly stands out as having a GATA2-like profile in cluster C3 is ETS1. Prior work identified the propensity of occupied GATA motifs to reside near Ets motifs [40, 11] has highlighted GATA2-ETS co-localization.

I next performed an alternative naive analysis by utilizing the list of peaks provided by the ENCODE project. As in the case of the Transcription Factor Enrichment Network example of Section 4.5.1, these peaks, provided by the ENCODE consortium, were identified by analyzing each dataset individually with ENCODE's uniform ChIP-seq processing pipeline. Figure 4.26 displays the ENCODE peak profiles for my cell type-TF conditions. For each of the 4803 composite elements, I constructed a *peak profile*, which is a binary vector indicating whether the element overlaps with the ENCODE peaks for each cell type-TF combination. I then computed the peak profile based similarity between the +9.5 site and each the of the composite elements using the `R` function `dist.binary` with the "Jaccard index" option. For comparison, I computed *pseudo-binary similarities* between each element and the +9.5 site using the MBASIC estimated enrichment probabilities across all conditions[4]. I then ranked the composite elements based on both ENCODE and MBASIC estimated similarities. Figure 4.25 provides a comparison of the two lists as a function of top ranking composite elements. Overall, I observe that the rankings based on MBASIC estimation are consistent with the rankings based on the ENCODE peak profiles.

Although the rankings of the composite elements with respect to their +9.5 similarity using both the ENCODE peak profiles and MBASIC estimation were quite similar, the two approaches

[4]The pseudo-binary similarity between two units i_1 and i_2 is calculated as $s(i_1, i_2) = \frac{\sum_k P\{\theta_{i_1 k}=1|Y\}P\{\theta_{i_2 k}=1|Y\}}{\sum_k P\{\theta_{i_1 k}=1|Y\}+P\{\theta_{i_2 k}=1|Y\}-P\{\theta_{i_1 k}=1|Y\}P\{\theta_{i_2 k}=1|Y\}}$.

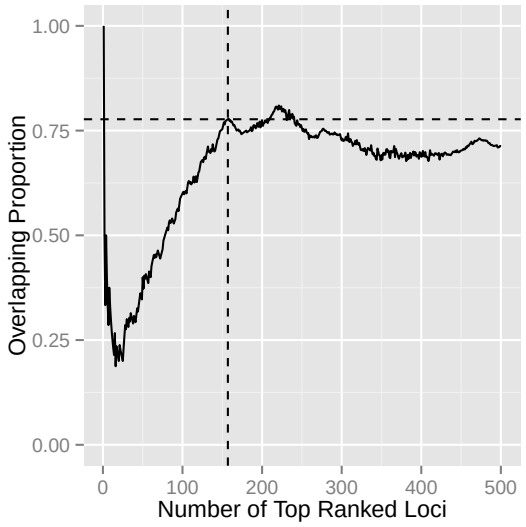


Figure 4.25 Proportion of overlap between the top ranked +9.5-like composite elements identified by MBASIC and ENCODE peak profiles. The overlap proportion is calculated by considering the same number of top ranked units (x-axis) in both the ENCODE-based and MBASIC-based similarities to the +9.5 site. The dashed lines mark that 78% of the C3 units are ranked in the top 157 based on the ENCODE peak profiles.

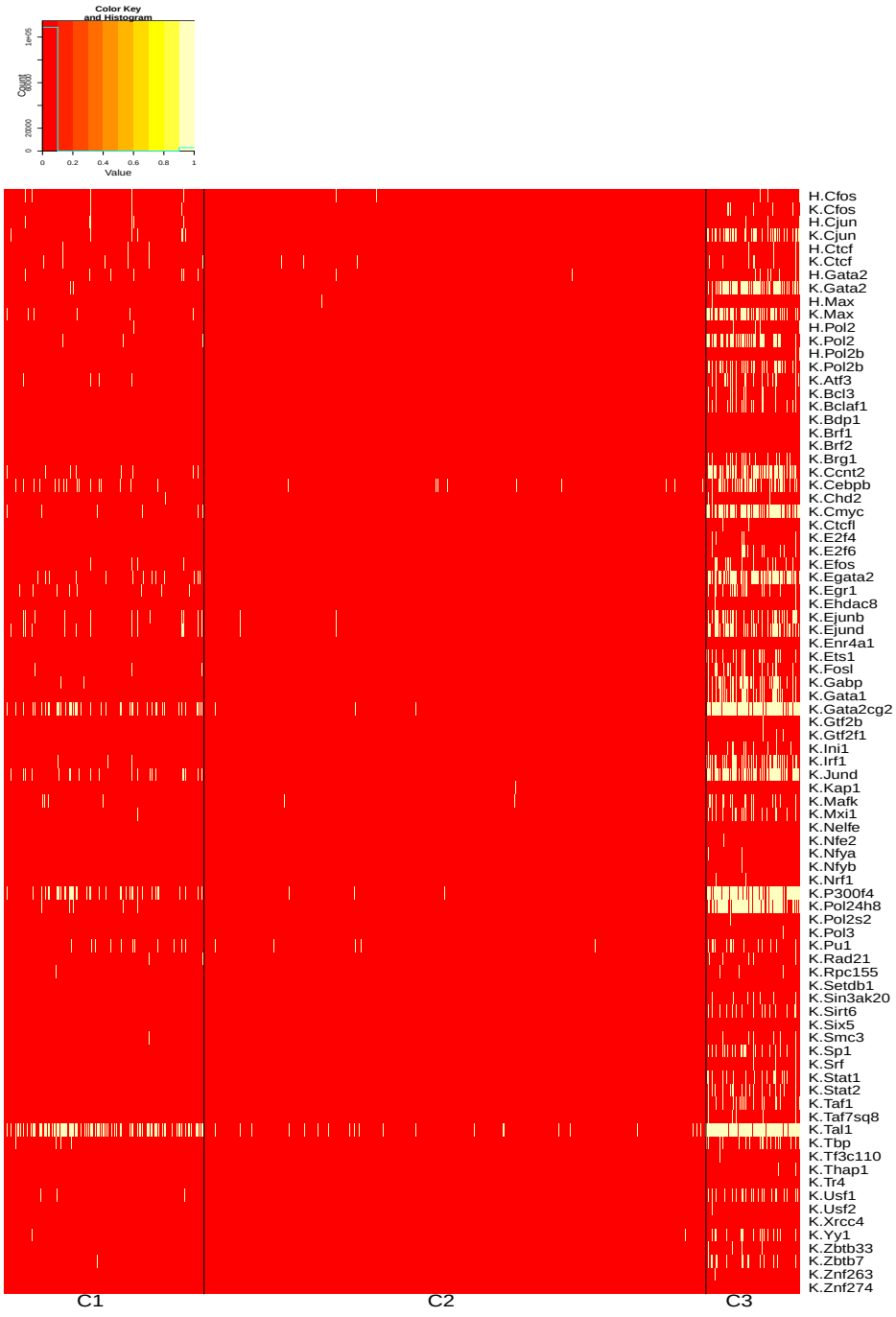


Figure 4.26 Enrichment states provided by the ENCODE peak profiles. Note that only a small percentage of the composite elements which harbor the canonical GATA binding site are identified as enriched for GATA family transcription factors. This suggests that the ENCODE peak profiles can be conservative.

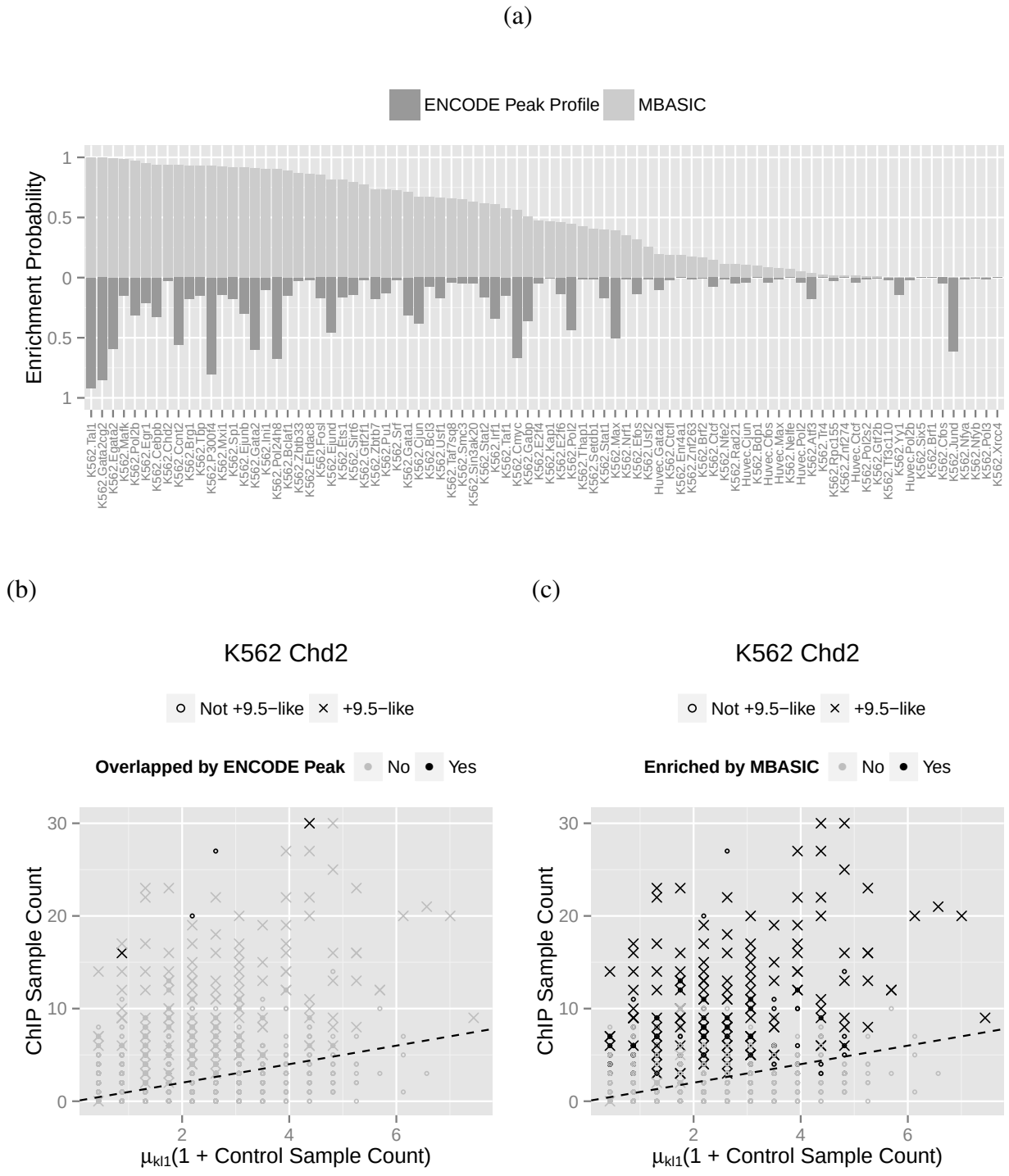


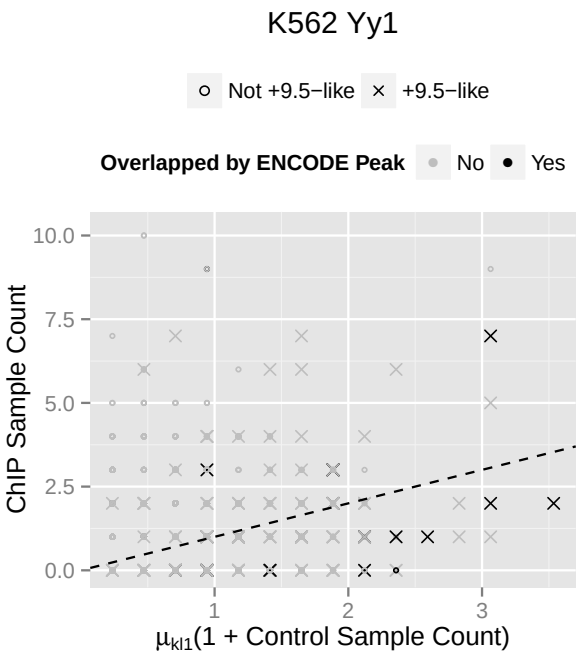
Figure 4.27 (a) Top half: Enrichment probabilities for the C3 units across all experimental conditions estimated by MBASIC. Bottom half: Proportion of C3 units that are overlapped by the ENCODE peaks for each condition. (b, c) ChIP sample read counts against normalized control sample read counts for one replicate with K562-Chd2. Enrichment status are annotated by (a) the ENCODE peak profiles and (c) MBASIC prediction.

resulted in different enrichment estimation at the individual TF-cell combination level. Figure 4.27(a) compares the estimated cluster-level enrichment probabilities of each cell type-TF combination for cluster C3 against their average ENCODE peak profiles and highlights the difference between the two procedures. To further investigate these differences, I plotted the raw data for individual replicates and compared the composite elements that were estimated to be enriched by the two methods. An example using data from K562-Chd2 is displayed in Figures 4.27(b) and (c). Although many elements have significantly higher counts in the ChIP sample compared to the control sample, they are not identified as occupied by Chd2 in K562 according to ENCODE peak annotation. Another example using a replicate from K562-Yy1 is shown in Figure 4.28, where several elements with zero ChIP count are overlapped by ENCODE peaks. These results indicate that MBASIC provides a grouping of the Ebox-GATA composite elements that is more consistent with the raw data compared to grouping based on ENCODE peak annotation.

4.6 Conclusions and Discussion

Clustering analysis based on an underlying state-space is a common problem for many genomic and epigenomic studies where multiple data sets over many observational units are integrated. In this chapter, I developed a unified statistical framework, called MBASIC, for addressing these class of problems. MBASIC simultaneously projects the observations onto a hidden state-space and infers clustered units in this space. The hierarchical structure of MBASIC enables the information of the state-space clusters to be fed back into the projection of the raw data, thus reinforces the accuracy of predicting the state-space states of individual units. The MBASIC framework offers flexibility in a number of aspects of experimental design, such as different numbers of replicates under individual experimental conditions and missing values. Additionally, it is applicable to many parametric distributions. My computational studies highlighted good operating characteristics of MBASIC and the two genomic applications illustrated how large numbers of ChIP-seq datasets can be integrated for addressing specific problems. In both of the applications, MBASIC algorithm converged within 20 minutes for a fixed model on a 64 bit machine with Intel Xeon 3.0GHz processor and 64GB of RAM. For model selection, I utilized `R` package `snow` to

(a)



(b)

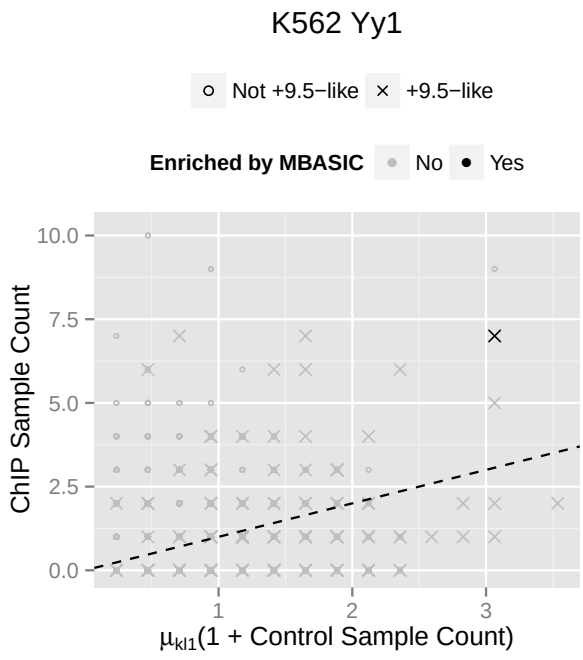


Figure 4.28 (a, b) ChIP sample read counts against control sample read counts for one replicate with K562-Yy1. Enrichment status are annotated by (a) the ENCODE peak profiles and (b) MBASIC prediction.

implement the 2-phase procedure with parallel fitting of different candidate models using a 8-core 64 bit, 64GB RAM machine with 8 Intel Xeon 3.0GHz processors. These runs were completed under 2 hours. The computational efficiency of my model depends on the simple, closed-form updates in my E-M algorithm. Such a mathematical form is due, at least in part, to my modeling assumption that the rows of my state-space matrix is clustered. I have argued that this assumption, as compared to the PCA-type model structures, offers easier interpretation and is well suited for many genomic applications. A preliminary version of the `R` package `MBASIC` is available at https://github.com/chandlerzuo/mbasic.

Chapter 5

A MAD-Bayes Algorithm for State-space Inference and Clustering[1]

5.1 Introduction

Integrative analysis for genomic and epigenomic data sets that reveal genome regulatory mechanism under a variety of biological conditions has become a popular topic in recent years. For example, [15] and [71] each studied human regulatory network by analyzing over one hundred transcription factor (TF) occupancy data sets. [78] identified clustered TF binding patterns in cancer cells by investigating 565 different TF data sets. [79] examined conservation pattern between human and mouse genome from over 1000 data sets encompassing TF binding, gene expression and DNase hypersensitivity experiments. In such studies, genomic loci that exhibit signals, i.e. enriched loci, are identified for individual data sets, and loci with the same combinatorial enrichment patterns across the experiments are clustered.

Such applications have motivated a number of new statistical models for systematic data integration. Earlier models, such as [2], [26] and [80], require separate analysis for enrichment detection and loci clustering. Later, models that jointly detect enrichment and clusters were proposed, i.e. [72] and [73]. These models, targeting at specific data types, enable information sharing among different data sets to increase statistical power. In Chapter 4, I proposed the Matrix Based Analysis for State-space Inference and Clustering (MBASIC) framework, a hierarchical framework generally applicable to different data types.

[1] The manuscript for this chapter is to be submitted. Method in this chapter is implemented in the `R` package `MBASIC` and is freely available at `http://github.com/chandlerzuo/mbasic`.

Among the three joint fitting algorithms, all of them assumed mixture data distributions and implemented Expectation-Maximization algorithms ([10]) to estimate parameters. A drawback of these algorithms is that its convergence rate is slow. One data set in [84] which involved 10290 loci and 166 data sets took over 20 min to converge. With the increasing number of data sets and the scale of each data set in such analysis, such computational performances can only be more challenging. From the application point of view, it is very important to develop new algorithms with greater scalability and efficiency.

To that end, I propose a MAD-Bayes algorithm for the MBASIC framework ([84]) in this chapter. The MAD-Bayes (i.e. Maximum a posteriori-based Asymptotic Derivation from Bayes) framework, was proposed by [6]. Recall that in estimating mixture Gaussian models, the small-variance asymptotic of the standard Expectation-Maximization algorithm leads to the K-means algorithm. [6] generalized this small-variance asymptotic view and proposed the MAD-Bayes framework that could derive a family of K-means-like algorithms for many Bayesian nonparametric models. To adopt this approach in the MBASIC framework, I first reparametrized its hierarchical model in a Bayesian framework, then derived the small-variance asymptotic algorithm for posterior maximization. Such an algorithm potentially inherits the simplicity and scalability of the K-means algorithm.

Despite the benefits of the MAD-Bayes framework, it has two under-appreciated drawbacks. First, a K-means-like algorithm for discrete parameters only converges to a local optimum. In a hierarchical model as MBASIC, this local optima problem is only exacerbated. [6] proposed random restarts and careful initialization as practical remedies. For the MAD-Bayes construction of the MBASIC algorithm, my numerical experiments show that the local optima can achieve comparable estimation with the E-M algorithm. Second, and more challenging, is that it leads to important tuning parameters in the objective function that determines the model structure. [6] evaluated several MAD-Bayes algorithms based on fixed tuning parameter values, and kept their choice for practical data sets as an unsolved problem. My MAD-Bayes algorithm contains two tuning parameters, one dictating the number of clusters and the other one dictating the relative importance between state-space inference and state-space clustering. I proposed a heuristic method

based on the conjugacy between the tuning parameters and the number of clusters that gives a range of candidate tuning parameter values. I further use the Silhouette score ([57]) based on the normalized data as the guide to select the best tuning parameter values. In addition to developing a MAD-Bayes formulation for MBASIC, I also developed a robust procedure for selecting tuning parameters, therefore directly applicable for real data analyses. My simulation studies demonstrate that this Silhouette score can reliably guide us to select models that restores the simulated "truth".

5.2 The Bayesian MBASIC Model

I first introduce my Bayesian MBASIC model that imposes a Bayesian structure on the MBASIC model of Chapter 4. I assume that data are taken from I observational units indexed from $i = 1, \cdots, I$, under K different experimental conditions. In a ChIP-seq experiment, for example, I can view the units as pre-specified loci and each condition as a particular pair of transcription factor and cell type. I assume that under the k-th experimental condition, there are n_k experimental replicates. The observed value for the i-th unit under condition k for the l-th replicate is denoted by Y_{ikl}, for $1 \leq i \leq I$, $1 \leq k \leq K$ and $1 \leq l \leq n_k$.

The Bayesian MBASIC model assumes a latent state is associated with the i-th unit and the k-th condition. θ_{iks} is the indicator for the state to be s, where s takes values in a discrete state-space $\mathscr{S} = \{1, \cdots, S\}$. For instance, in the ChIP-seq experiment, I may take $\mathscr{S} = \{1, 2\}$, where $\theta_{ik1} = 1$ or $\theta_{ik2} = 1$ means that the i-th unit is un-enriched or enriched under condition k respectively.

The Bayesian MBASIC model consists of two parts. The first part, *State-space Mapping*, assumes the following distribution of Y_{ikl} conditional on θ_{ik}:

$$(Y_{ikl}|\theta_{iks} = 1) \overset{\text{i.i.d.}}{\sim} f_s(\cdot|\mu_{kls}, \sigma_{kls}, \gamma_{ikls}),$$

where f_s is a density function with parameters μ_{kls}, σ_{kls}, and γ_{ikls} denotes the variable for the background information. The original MBASIC model can take f_s as many different distribution classes, and allows that different distributions can be taken for different states. In the Bayesian MBASIC model, I restrict myself to the log-normal distribution:

$$(\log(Y_{ikl}+1)|\theta_{iks}=1) \overset{\text{i.i.d.}}{\sim} N(\mu_{kls}\gamma_{ikls}, \sigma^2_{kls}). \tag{5.1}$$

I take conjugate priors $\mu_{kls} \sim N(\xi, \tau^2)$ and $\sigma^{-2}_{kls} \sim Gamma(\omega, \nu)$. The log-normal distribution leads to straight forward derivation for the Bayes and MAD-Bayes algorithms, as I show in Section 5.3. I discuss the extension to other distributions in the Discussion section.

The second part of the Bayesian MBASIC model is *State-space Clustering*. I assume that experimental units can be clustered into J groups, i.e. $\{1,\ 2, \cdots,\ I\} = \mathscr{C}_1 \cup \cdots \cup \mathscr{C}_J$. States for the units within the same cluster follow the product multinomial distribution:

$$(\theta_{iks})_{1\le s\le S}|i \in \mathscr{C}_j \overset{\text{i.i.d.}}{\sim} \text{MultiNomial}(1, (w_{jks})_{1\le s\le S}),\quad \sum_{s=1}^{S} w_{jks} = 1. \tag{5.2}$$

I take the non-informative prior $(w_{jks})_{1\le s\le S} \sim Dir(1,\ 1, \cdots,\ 1)$. Unlike the MBASIC model, which assumes that J is fixed, I assume a Chinese Restaurant Process as a prior for the cluster number J. Let α be a hyper parameter of the model. The first unit forms $\mathscr{C}_1$ in the beginning. I recursively assign the cluster for each unit. Suppose I have assigned units $1,\ \cdots,\ i-1$ to J' clusters. The i-th unit is then assigned to $\mathscr{C}_{j'},\ j' \le J'$ with probability proportional to the size of $\mathscr{C}_{j'}$. It can also form a new cluster $\mathscr{C}_{J'+1}$ with probability proportional to α. The prior density for a partition with J clusters is thus ([6]):

$$f(\mathscr{C}_1,\ \cdots,\ \mathscr{C}_J) = \alpha^{J-1}\frac{\Gamma(\alpha+1)}{\Gamma(\alpha+I)}\prod_{j=1}^{J}(|\mathscr{C}_j|-1)!. \tag{5.3}$$

Introduce $z_{ij} = 1\{i \in \mathscr{C}_j\}$. The posterior probability of the model is written as:

$$\begin{aligned} P(\theta, z, \mu, \sigma, w, J|y) \propto & \prod_{i=1}^{I}\prod_{j=1}^{J}\prod_{k=1}^{K}\prod_{s=1}^{S} w_{jks}^{\theta_{iks}z_{ij}} \cdot \alpha^{J-1}\prod_{j=1}^{J}\left(\sum_{i=1}^{I} z_{ij}-1\right)! \\ & \cdot \prod_{k=1}^{K}\prod_{l=1}^{n_k}\prod_{s=1}^{S}\frac{1}{\tau}e^{-\frac{(\mu_{kls}-\xi)^2}{2\tau^2}} \cdot \prod_{k=1}^{K}\prod_{l=1}^{n_k}\prod_{s=1}^{S}\frac{1}{\sigma_{kls}^{2(\omega+1)}}e^{-\frac{1}{\nu\sigma^2_{kls}}} \\ & \cdot \prod_{i=1}^{I}\prod_{k=1}^{K}\prod_{s=1}^{S}\left[\prod_{l=1}^{n_k}\frac{1}{\sigma_{kls}}e^{-\frac{[\log(y_{ikl}+1)-\mu_{kls}\gamma_{ikls}]^2}{2\sigma^2_{kls}}}\right]^{\theta_{iks}}. \end{aligned} \tag{5.4}$$

A Gibbs sampling algorithm can be derived in a straightforward way from the posterior density. In each step, I update the quantities as the following:

$$
\begin{aligned}
(w_{jks})_{1\le s\le S} &\sim Dir\left((1+\sum_{i=1}^{I}\theta_{iks}z_{ij})_{1\le s\le S}\right),\\
(\mu_{kls}|\cdot) &\sim N\left(\frac{\xi\sigma_{kls}^2+\tau^2\sum_{i=1}^{I}\gamma_{ikls}\log(y_{ikl}+1)\theta_{iks}}{\sigma_{kls}^2+\tau^2\sum_{i=1}^{I}\theta_{iks}\gamma_{ikls}^2}, \frac{\sigma_{kls}^2\tau^2}{\sigma_{kls}^2+\tau^2\sum_{i=1}^{I}\theta_{iks}\gamma_{ikls}^2}\right),\\
\left(\frac{1}{\sigma_{kls}^2}|\cdot\right) &\sim Gamma\left(\omega+\frac{\sum_{i=1}^{I}\theta_{iks}}{2}, \left[\frac{1}{\nu}+\frac{(\log(y_{ikl}+1)-\gamma_{ikls}\mu_{kls})^2}{2}\right]^{-1}\right),\\
P(z_{ij}=1|\cdot) &\propto \frac{\sum_{i'\neq i}z_{i'j}}{I-1+\alpha}\prod_{k=1}^{K}\prod_{s=1}^{S}w_{jks}^{\theta_{iks}},\\
P(i \text{ forms a new cluster}|\cdot) &\propto \frac{\alpha}{I-1+\alpha}\frac{1}{S^K}.
\end{aligned}
$$

5.3 The MAD-Bayes Algorithm

Although my model results in a simple Gibbs sampling algorithm, its computational efficiency is much restricted in practice. I have applied the Gibbs sampling to a ChIP-seq data set on 10,290 units and 166 replicates in [84]. The algorithm took over 20 hours to mix. This further motivates the *MAD-Bayes* construction for MBASIC.

Recall that the K-means algorithm can be derived from a small variance asymptotic view of maximizing the posterior density for a Bayesian model. In an earlier work, [34] used this idea to develop an algorithm named "DP-means" for the Chinese Restaurant Processes. [6] extended this idea to develop algorithms for the Dirichlet Processes. In addition, the authors proposed that this idea can be generally applied to different Bayesian models, and named such algorithms as "MAD-Bayes" (***M***aximum a posteriori-based **A**symptotic **D**erivation of **Bayes**).

I consider the following small variance asymptotic assumptions for the Bayesian MBASIC model:

Assumption 5.1. *All data have equal variance:* $\sigma_{kls}^2=\sigma^2\to 0$.

Assumption 5.2. $w_{jks}\in\{1-(S-1)e^{-\lambda_w/\sigma^2/}, e^{-\lambda_w/\sigma^2}\}$.

Assumption 5.3. $\alpha = e^{-\lambda_w \lambda_r / 2\sigma^2}$.

Recall that w_{jks} denotes the probability for the j-th cluster to have state s in the k-th condition. Both λ_w and λ_r in my assumptions are some chosen constants such that my MAD-Bayes algorithm will have non-trivial cluster assignments. Notice that as $\sigma^2 \to 0$, $e^{-\lambda_w/\sigma^2} \to 0$, $\alpha \to 0$.

Proposition 5.4. *With Assumptions 5.1, 5.2 and 5.3, we have*

$$\begin{aligned} -2\sigma^2 \log(\theta, z, \mu, \sigma, w, J|y) = & \sum_{i=1}^{I} \sum_{k=1}^{K} \sum_{l=1}^{n_k} \sum_{s=1}^{S} \theta_{iks} [\log(y_{ikl} + 1) - \mu_{kls}\gamma_{ikls}]^2 \\ & + \lambda_w \left\{ \sum_{i=1}^{I} \sum_{j=1}^{J} z_{ij} \left[\sum_{k=1}^{K} \sum_{s=1}^{S} (\theta_{iks} - w_{jks})^2 \right] + \lambda_r (J-1) \right\} + Const + o(1). \end{aligned} \tag{5.5}$$

See Section 5.3.3 for the proof. In Eqn. (5.5), the first term originates from the densities of Gaussian distributions (Eqn. (5.1)), the second term from the Multinomial distributions (Eqn. (5.2)), and the third term is a penalty for the number of clusters. From Eqn. (5.5), I see that maximizing the posterior density is asymptotically equivalent to the following minimization problem:

$$\begin{aligned} \min_{\mu, z, \theta, w, J} & \sum_{i=1}^{I} \sum_{k=1}^{K} \sum_{l=1}^{n_k} \sum_{s=1}^{S} \theta_{iks} [\log(y_{ikl} + 1) - \mu_{kls}\gamma_{ikls}]^2 \\ & + \lambda_w \left\{ \sum_{i=1}^{I} \sum_{j=1}^{J} z_{ij} \left[\sum_{k=1}^{K} \sum_{s=1}^{S} (\theta_{iks} - w_{jks})^2 \right] + \lambda_r (J-1) \right\} \end{aligned} \tag{5.6}$$

A local minimization algorithm with objective function (5.5) can be derived as Algorithm 5.1. Similarly to the K-Means algorithm, it sequentially assigns new values for the variables in each iteration. Notice that each step of this algorithm does not increase the objective function, and the updates for w_{jks}'s and μ_{kls}'s minimize the objective function for a fixed configuration of θ_{iks}'s and z_{ij}'s. Moreover, there are finite number of combinations for θ_{iks}'s and z_{ij}'s such that no cluster is empty and all clusters are distinct from one another. With such observations, one can conclude that:

Proposition 5.5. *Algorithm 5.1 converges within finite number of iterations to a local minimum of the objective function Eqn. (5.6).*

Algorithm 5.1 The MAD-Bayes algorithm for the Bayesian MBASIC model.

repeat

Step 1. Update the log-normal parameter μ_{kls}'s. For $1 \leq k \leq K$, $1 \leq l \leq n_k$ and $1 \leq s \leq S$,

$$\mu_{kls} \leftarrow \frac{\sum_{i=1}^{I} \theta_{iks} \log(y_{ikl} + 1)}{\sum_{i=1}^{I} \theta_{iks} \gamma_{ikls}}.$$

Step 2. Update the Multinomial parameter w_{jks}'s. For each $1 \leq j \leq J$, $1 \leq k \leq K$ and $1 \leq s \leq S$,

$$w_{jks} \leftarrow \frac{\sum_{i=1}^{I} z_{ij} \theta_{iks}}{\sum_{i=1}^{I} z_{ij}}.$$

Step 3. Update the cluster labels z_{ij}'s. For each $1 \leq i \leq I$, relabel the distinct clusters for all the rest $I - 1$ units as $1, 2, \cdots, J$. Compute

$$t_j = \sum_{k=1}^{K} \sum_{s=1}^{S} (\theta_{iks} - w_{jks})^2, \ 1 \leq j \leq J.$$

If $\min_{1 \leq j \leq J} t_j < \lambda_r$, assign the unit i to the cluster j with minimum t_j. Otherwise, generate a cluster $J + 1$ with a single unit i.

Step 4. Assign the states θ_{iks}'s. For each $1 \leq i \leq I$, $1 \leq k \leq K$, let

$$s_0 \leftarrow \arg\min_{s} \sum_{l=1}^{n_k} (\log(y_{ikl} + 1) - \mu_{kls} \gamma_{ikls})^2 + \lambda_w \sum_{j=1}^{J} z_{ij} \left[(1 - w_{jks})^2 + \sum_{s' \neq s} w_{jks}^2 \right],$$

and let $\theta_{iks_0} = 1$, $\theta_{iks} = 0$ for $s \neq s_0$.

until Convergence.

5.3.1 Model Initialization

It is known that the K-means algorithm can be sensitive to the choice of cluster initialization, and with its increasing complexity in the parameter space, this problem can even be more serious for the MAD-Bayes algorithm ([6]). Random initialization alone may address this issue, but it has no control of the starting point in the parameter space, and in some cases this would lead to excessive number of iterations. Therefore, a guided initialization strategy is often needed.

For Algorithm 5.1, I first initialize θ_{iks}'s and μ_{kls}'s together as the minimizer of the following objective:

$$\min_{\mu,\theta} \sum_{i=1}^{I} \sum_{k=1}^{K} \sum_{l=1}^{n_k} \sum_{s=1}^{S} \theta_{iks} [\log(y_{ikl} + 1) - \mu_{kls}\gamma_{ikls}]^2. \tag{5.7}$$

Notice that (5.7) is a degenerate form of (5.6) with $\lambda_w = 0$. Therefore, the algorithm to minimize (5.7) is Algorithm 5.2, which degenerates from Algorithm 5.1.

Algorithm 5.2 Initializing θ_{iks}'s and μ_{kls}'s.

---

repeat

Step 1. Update the log-normal parameter μ_{kls}'s. For $1 \le k \le K$, $1 \le l \le n_k$ and $1 \le s \le S$,

$$\mu_{kls} \leftarrow \frac{\sum_{i=1}^{I} \theta_{iks} \log(y_{ikl} + 1)}{\sum_{i=1}^{I} \theta_{iks}\gamma_{ikls}}.$$

Step 2. Assign the states θ_{iks}'s. For each $1 \le i \le I$, $1 \le k \le K$, let

$$s_0 \leftarrow \arg\min_{s} \sum_{l=1}^{n_k} (\log(y_{ikl} + 1) - \mu_{kls}\gamma_{ikls})^2,$$

and let $\theta_{iks_0} = 1$, $\theta_{iks} = 0$ for $s \neq s_0$.

until Convergence.

I then initialize z_{ij}'s and w_{jks}'s based on the fixed values of the initialized θ_{iks}'s. I propose three initializing methods: K-means, K-means++ and adaptive K-means++. Both the K-means and the K-means++ initialization depends on a pre-determined value of J, initialization of which is

discussed in Section 5.3.2. For now, assume the value of initial J is given. For the K-means initialization, I run the K-means algorithm on the I state-space vectors $(\theta_{iks})_{1\leq k\leq K, 1\leq s\leq S}$ for $1 \leq i \leq I$ until convergence. The center of each cluster is used as the initial values of w_{jks}'s. For the K-means++ initialization, I adopt the procedure in [4] which was introduced as a guided initialization for the K-means algorithm with guaranteed global optimality. The adaptive K-means++ initialization is similar to the initialization procedure in [6]. It uses a K-means++ style algorithm to find a local minimizer for the following objective function:

$$\min_{z,w,J} \sum_{i=1}^{I}\sum_{j=1}^{J} z_{ij}\left[\sum_{k=1}^{K}\sum_{s=1}^{S}(\theta_{iks}-w_{jks})^2\right] + \lambda_r(J-1). \tag{5.8}$$

Algorithm 5.3 Adaptive K-means++ Initialization.

Let $J = 1$, $z_{i1} = 1$, $w_{1ks} = \sum_{i=1}^{I}\theta_{iks}/I$.

repeat

1. Let $d_i = \sum_{j=1}^{J} z_{ij}\sum_{k=1}^{K}\sum_{s=1}^{S}(\theta_{iks}-w_{jks})^2$.
2. Sample i_0 from $\{1,\ 2, \cdots,\ I\}$ with probability proportional to d_i.
3. For each i, if $\sum_{j=1}^{J} z_{ij}\sum_{k=1}^{K}\sum_{s=1}^{S}(\theta_{iks}-w_{jks})^2 < \sum_{k=1}^{K}\sum_{s=1}^{S}(\theta_{iks}-\theta_{i_0ks})^2$, then set $z_{i,J+1} = 1$, $z_{ij} = 0$ for $j \leq J$.
4. Update each w_{jks} as
$$w_{jks} \leftarrow \frac{\sum_{i=1}^{I} z_{ij}\sum_{k=1}^{K}\sum_{s=1}^{S}\theta_{iks}}{\sum_{i=1}^{I} z_{ij}}.$$
5. $J \leftarrow J + 1$.

until The value of Eqn. (5.8) does not decrease.

5.3.2 Selecting the Tuning Parameters

Although selecting the tuning parameters is a cretical step for real data, this issue was not addressed in [6]. Even for the models with one tuning parameters, the authors of [6] acknowledged the difficulty in choosing their appropriate values in practice. Given the hierarchical structure of

my model, the objective function Eqn. (5.6) has two tuning parameters λ_w and λ_r, making this problem more challenging for my case. Notice that since my goal is to develop a time-efficient algorithm, cross-validation is an un-affordable method.

I propose my tuning parameter selection method through heuristic studies. For λ_w, recall one of my assumptions in Proposition 5.4 that $e^{-\frac{\lambda_w}{\sigma^2}} \to 0$ as $\sigma^2 \to 0$. This asymptotic degeneracy should not be expected of real data, but we should expect small $e^{-\frac{\lambda_w}{\sigma^2}}$ values since it represents the probability of not having a particular state. I propose choosing λ_w as $2\hat{\sigma}^2$ in practice, where $\hat{\sigma}^2$ is the initial estimate for σ^2 as a byproduct of Algorithm 5.2:

$$\hat{\sigma}^2 = \min_{\mu,\theta} \sum_{i=1}^{I} \sum_{k=1}^{K} \sum_{l=1}^{n_k} \sum_{s=1}^{S} \theta_{iks} [\log(y_{ikl} + 1) - \mu_{kls}\gamma_{ikls}]^2.$$

I will show in Section 5.4 that varying λ_w in the order of $\hat{\sigma}^2$ does not affect model inference.

Once λ_w is fixed, λ_r determines the number of clusters as shown by Eqn. (5.6). I first decide a range of λ_r that yield numbers of clusters of interest. My intuition comes from the following conjugacy between λ_r and J. Suppose the global minimum of Eqn. (5.6) is attained with J clusters. Then, fixing the θ_{iks}'s, z_{ij}'s, λ_w, λ_r and J jointly minimize:

$$\sum_{i=1}^{I} \sum_{j=1}^{J} z_{ij} \left[\sum_{k=1}^{K} \sum_{s=1}^{S} (\theta_{iks} - w_{jks})^2 + \lambda_r (J - 1) \right]. \tag{5.9}$$

Let

$$L(J) = \min_{z,w} \sum_{i=1}^{I} \sum_{j=1}^{J} z_{ij} \left[\sum_{k=1}^{K} \sum_{s=1}^{S} (\theta_{iks} - w_{jks})^2 \right],$$

I should have:

$$L(J) + \lambda_r (J - 1) \leq L(J') + \lambda_r (J' - 1), \ \forall J' \neq J.$$

Therefore, the λ_r value that can yield J clusters in the global solution must satisfy:

$$\sup_{J'>J} \frac{L(J) - L(J')}{J - J'} \leq \lambda_r \leq \inf_{J'>J} \frac{L(J') - L(J)}{J' - J}. \tag{5.10}$$

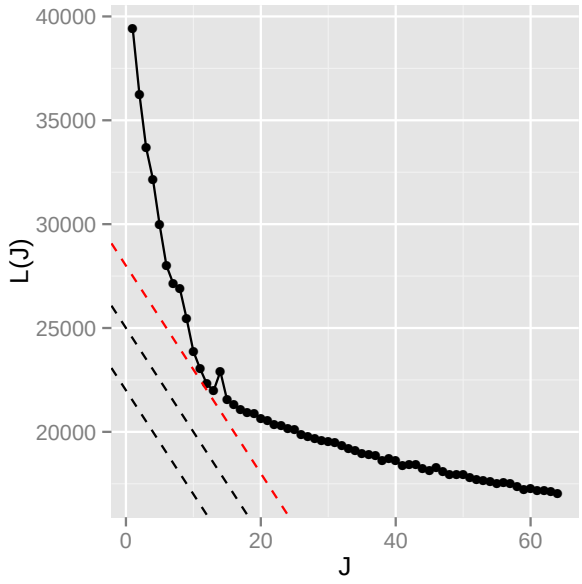


Figure 5.1 A graphic interpretation of Eqn. (5.10).

In practice, I can get surrogate values of $L(J)$ from the initialization of the clusters. For the K-means initialization, I take $L(J)$ as the total within group variance with J clusters. For the K-means++ initialization, I take $L(J)$ as the total squared distance from each unit to its cluster centroid. For the Adaptive K-means++ initialization, I apply Algorithm 5.3 except that I repeat the iterations until I get J clusters, then take $L(J)$ as the total within group variance.

A graphic interpretation for the conjugacy between λ_r and J is shown in Figure 5.1. I use the K-means initialization to compute surrogate values for $L(J)$ for various values of $J \geq 1$. If λ_r satisfies Eqn. (5.10), a line with slope $-\lambda_r$ crossing $(J, L(J))$ on the graph should be tangent to the trace of all $L(J)$ values. Notice that using the surrogate $L(J)$ values, the curve connecting the $L(J)$ values can be non-convex, so the solution for λ_r for some J not hold. Nevertheless, as my goal here is only to get the number of clusters in a reasonable range rather than a specific value, I could get a convex approximation for the trace of $L(J)$ so that the solution for Eqn. (5.10) exists for each J. A simpler approach is to put the $L(J)$ values in a decreasing value and require the following condition for λ_r instead of Eqn. (5.10):

$$L(J) - L(J+1) \leq \lambda_r \leq L(J-1) - L(J). \tag{5.11}$$

Algorithm 5.4 applies this idea to pick m candidate λ_r values. Each J corresponds to a λ_r of value $[L(J-1) - L(J+1)]/2$ that satisfies Eqn. (5.11). It tries to identify the range of λ_r that leads up to $\sqrt{I}$ number of clusters. Notice that Step 4 in Algorithm 5.4 also determines the initial number of clusters for the K-means and K-means++ initialization.

Algorithm 5.4 Choose m candidate λ_r values.

1. Compute surrogate values of $L(j)$ for $1 \leq j \leq \lfloor \sqrt{I} \rfloor := J_{max}$. Order $L(j)$'s decreasingly as $L_1 \geq L_2 \geq \cdots \geq L_{J_{max}}$.
2. Let $\lambda'_j = (L_{j-1} - L_{j+1})/2$ for $2 \leq j \leq J_{max} - 1$.
3. Choose the $k/(m+2)$-th quantiles in the set $\{\lambda'_j\}$ as the candidate λ_r values.
4. For the K-means and K-means++ initialization, given a selected λ_r, choose the initial number of clusters as $J \leftarrow \arg\min_j |\lambda'_j - \lambda_r|$.

Finally, to select the best λ_r value, I propose using the Silhouette score ([57]) based on the normalized data. The Silhouette score has been demonstrated as one of the most useful criteria for determining the number of clusters in clustering analysis (e.g. [75], [41]). Based on the estimated parameters, I compute $\sigma^2_{kls} = \sum_{i=1}^{I} (\log(y_{ikl}+1) - \mu_{kls})^2/I$, and let $f_{iks} = \prod_{l=1}^{n_k} \phi[(\log(y_{ikl}+1) - \mu_{kls})/\sigma_{kls}]$, where ϕ is the density function for the standard normal distribution. I further compute

$$\tilde{\theta}_{iks} = \frac{f_{iks}}{\sum_{s=1}^{S} f_{iks}}.$$

The Silhouette score based on the I vectors $(\tilde{\theta}_{iks})_{1\leq k\leq K,\ 1\leq s\leq S}$ and their cluster labels z_{ij}. My normalization for $\tilde{\theta}_{iks}$'s involves all my parameter estimates, so its Silhouette score summarizes their joint goodness of fit for the observed data.

5.3.3 Proof of Proposition 5.4

I finish this section by the proof of Proposition 5.4.

Proof. With $\sigma_{kls}^2 = \sigma^2$, I can rewrite the joint posterior density Eqn. (5.4) as:

$$\begin{aligned}\log P(\theta, z, \mu, \sigma, w, J|y) = &-\frac{1}{2\sigma^2}\sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{l=1}^{n_k}\sum_{s=1}^{S}\theta_{iks}[\log(y_{ikls}+1) - \mu_{kls}\gamma_{ikls}]^2 \\ &+ \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{s=1}^{S}\theta_{iks}z_{ij}\log w_{jks} + \log\alpha(J-1) \\ &+ \frac{NS}{\nu\sigma^2} - [(\omega+1)NS + NI]\log\sigma^2 + O(1),\end{aligned} \tag{5.12}$$

where $N = \sum_{k=1}^{K} n_k$ is the total number of replicates, and $O(1)$ collects terms unrelated to σ^2. Therefore, as $\sigma^2 \to 0$,

$$\begin{aligned}-2\sigma^2\log P(\theta, z, \mu, \sigma, w, J|y) = &\sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{l=1}^{n_k}\sum_{s=1}^{S}\theta_{iks}[\log(y_{ikls}+1) - \mu_{kls}\gamma_{ikls}]^2 \\ -2\sigma^2&\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{s=1}^{S}\theta_{iks}z_{ij}\log w_{jks} - 2\sigma^2\log\alpha(J-1) + 2NS + o(1)\end{aligned} \tag{5.13}$$

For the term including w_{jks}, notice that by assumption, for each j, k, $w_{jks} = e^{-\frac{\lambda_w}{\sigma^2}}$ except for one s. Thus, when $\theta_{iks'} = 1$ for $s' = \arg\max_s w_{jks}$,

$$-2\sigma^2\sum_{s}\theta_{iks}\log w_{jks} = -2\sigma^2\log[1-(S-1)e^{-\frac{\lambda_w}{\sigma^2}}] = o(1),$$

$$\sum_{s}(\theta_{iks} - w_{jks})^2 = S(S-1)e^{-\frac{2\lambda_w}{\sigma^2}} = o(1);$$

otherwise,

$$-2\sigma^2\sum_{s}\theta_{iks}\log w_{jks} = 2\lambda_w,$$

$$\sum_{s}(\theta_{iks} - w_{jks})^2 = S(S-1)e^{-\frac{2\lambda_w}{\sigma^2}} - 2Se^{\frac{\lambda_w}{\sigma^2}} + 2 = 2 + o(1).$$

In both cases, I have

$$-2\sigma^2 \sum_s \theta_{iks} \log w_{jks} = \lambda_w \sum_s (\theta_{iks} - w_{jks})^2 + o(1). \tag{5.14}$$

Substitute (5.14) in (5.12), and also notice that $-2\sigma^2 \log(\alpha) = \lambda_w \lambda_r$ by my assumption, I prove Eqn. (5.5).

□

5.4 A Simulation Study

I present a simulation study to prove the utility of my algorithms. Notice that although my MAD-Bayes algorithm is derived from the Bayesian MBASIC model, my algorithm is essentially non-parametric and does not depend on specific distributions. In this section, I rely on the MBASIC model ([84]) to generate synthetic data sets of various structures. As in Chapter 4, I have already shown that the MBASIC model can be required to a wide range of genomic and epigenomic data sets, I hope that their model can let us investigate my algorithm's performance in realistic settings.

The MBASIC model differs from the Bayesian MBASIC model (Section 5.2) in two ways. First, it assumes that some units, called "singletons", can fall out of the J clusters. Given a latent variable b_i that represents the indicator of singleton, the distribution of $(\theta_{iks})_{1\leq k\leq K,\ 1\leq s\leq S}$ follows:

$$(\theta_{iks})_{1\leq s\leq S} | b_i = 0,\ z_{ij} = 1 \overset{\text{i.i.d.}}{\sim} \text{Multinom}(1, (w_{jks})_{1\leq s\leq S});$$
$$(\theta_{iks})_{1\leq s\leq S} | b_i = 1 \overset{\text{i.i.d.}}{\sim} \text{Multinom}(1, (p_{is})_{1\leq s\leq S}).$$

where p_{is}'s are the idiosyncratic state probabilities for each unit satisfying $\sum_{s=1}^{S} p_{is} = 1$. Second, MBASIC does not have the Chinese Restaurant Process prior for z_{ij}'s and J. It assumes the following prior for the cluster and singleton indicators:

$$b_i \overset{\text{i.i.d.}}{\sim} Bin(\zeta),\ (z_{ij})_{1\leq j\leq J} \overset{\text{i.i.d.}}{\sim} \text{Multinom}(1, (\zeta_j)_{1\leq j\leq J}).$$

In my simulations, I set $K = 20$, and $\gamma_{ikls} = 1,\ \forall k,\ l$. I simulated $(\pi_j)_{1\leq j\leq J} \sim \text{Dir}(1, 1, \cdots, 1)$. I let $(w_{jks})_{1\leq s\leq S}$ and $(p_{is})_{1\leq s\leq S}$ each follow an S-dimensional dirichlet distribution $\text{Dir}(0.1,\ 0.1, \cdots,\ 0.1)$ so that for each S-dimensional vector the probability mass tended to concentrate in one state. I set

the log-normal distribution parameters as $\mu_{kls} \overset{\text{i.i.d.}}{\sim} N(2s,\ 0.05^2)$ and $\sigma_{kls} = 0.5$. I varied ζ =0 or 0.4 so that singletons could be included in the data to let us evaluate my algorithm's robustness against violations to its assumptions.

My MAD-Bayes algorithm also relies on the assumption that $\sigma_{kls} = \sigma$ for all $k,\ l,\ s$. To account for the heterogeneity between the data under different conditions in my simulation, I always normalized my data such that the average of $\{\log(Y_{ikl} + 1) : 1 \leq i \leq I\}$ is 1 for each replicate before running the algorithm.

5.4.1 An Empirical Approach of Choosing Tuning Parameters

In Section 5.3.2, I have stated that the solution of the algorithm is not sensitive to the choice of λ_w, and that the Silhouette score based on the normalized data can be used to assess the goodness of fit. I now demonstrate these statements using simulations. I simulated data sets with $I = 4000$, $J = 10$, and applied the MAD-Bayes algorithm with different values of λ_w and λ_r. I varied λ_w from 1 to 10 folds of the initial σ^2 estimate $\hat{\sigma}^2$, and varied λ_r between 5 to 200. For each pair of choices for λ_w and λ_r, I ran the MAD-Bayes algorithm with 10 random reinitializations, and picked the best result with the minimum objective function. I then computed the Adjusted Rand Index (ARI) as well as the Silhouette score based on the clustering results.

For each of the three initialization methods, I compared the variation of ARI against choices in λ_w and λ_r. I show these results with $\zeta = 0.4$ and $S = 2$ in Figures 5.2 and 5.3. As shown by the heatmaps in Figure 5.2, with all three initialization methods, ARI is much robust against variation in λ_w, but sensitive to the choice of λ_r. This suggests that as long as I have a method to select the appropriate λ_r given λ_w, my algorithm should give reasonable result. The comparison between ARI and the Silhouette score is shown in Figure 5.3. For all the three different initialization methods, I observe high correlation between these two quantities. I observe similar patterns under settings with $\zeta = 0$ or $S = 4$ (Figures C.1-C.6). In conclusions, these observations justifies that setting $\lambda_w = 2\hat{\sigma}^2$ while selecting λ_r based on the Silhouette score is a useful approach.

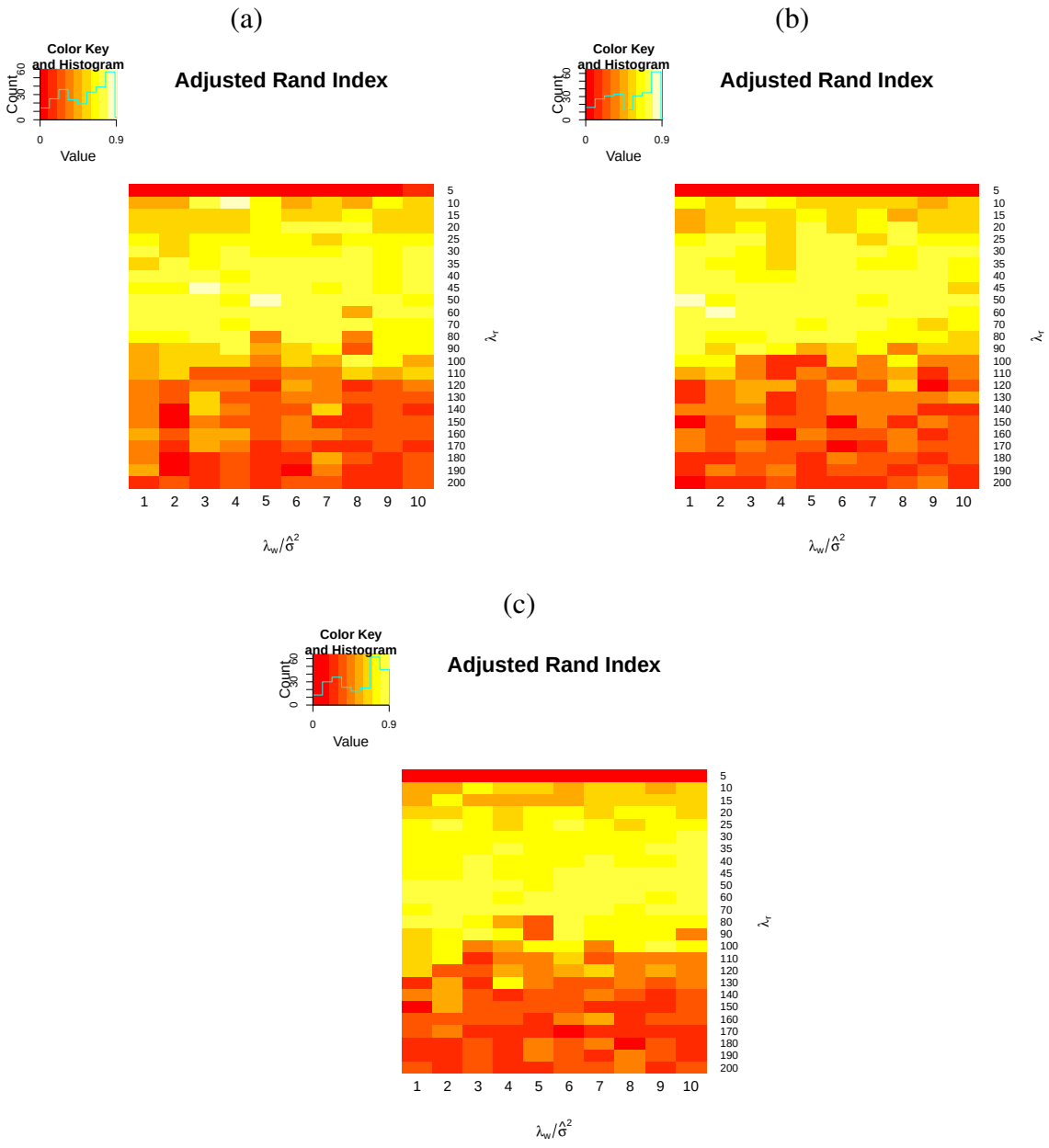


Figure 5.2 Performance of ARI for different choices of λ_w and λ_r based on the (a) K-means (b) K-means++ and (c) Adaptive K-means++ initialization. Data simulated with $\zeta = 0.4$ and $S = 2$.

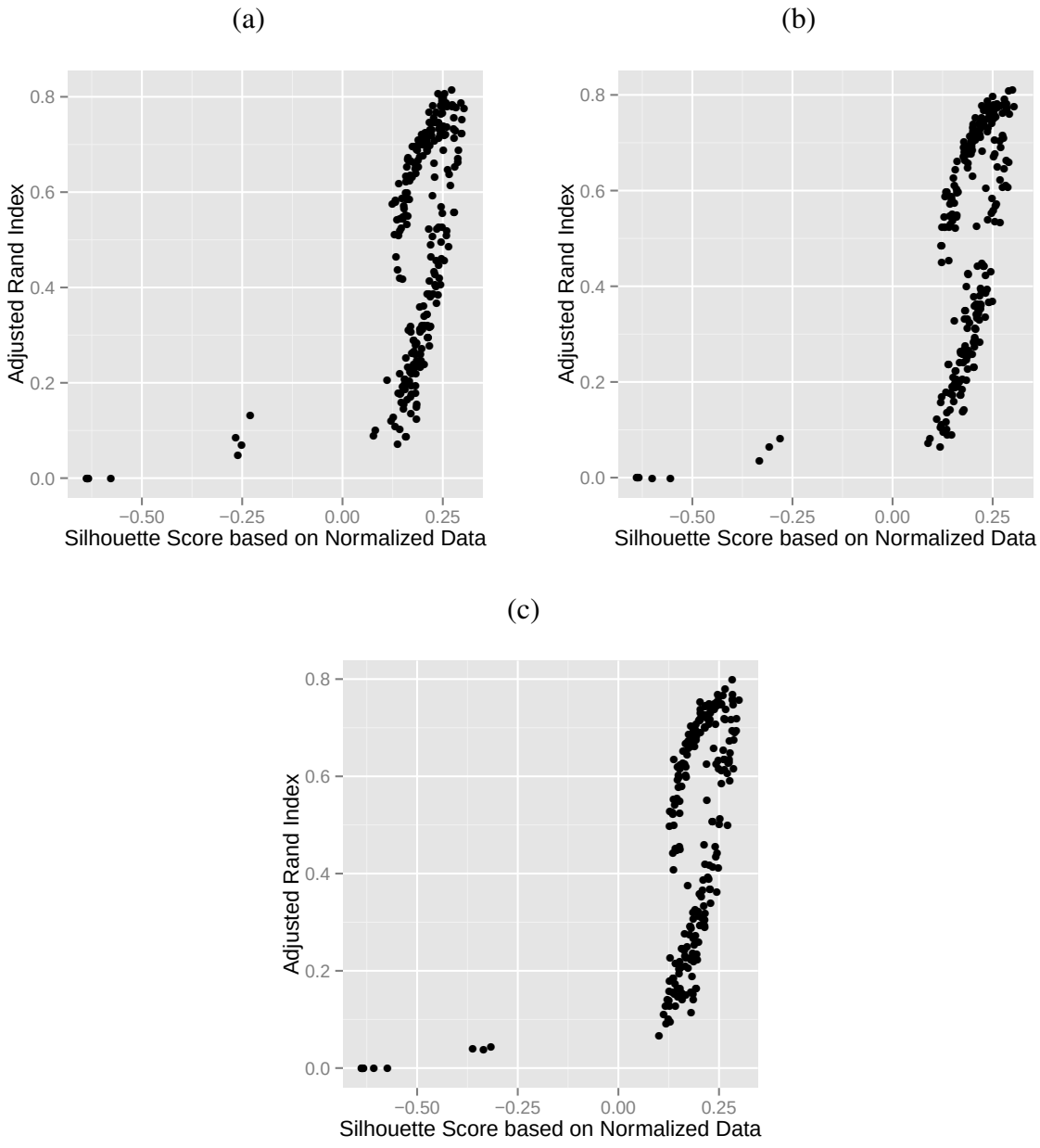


Figure 5.3 Relationship between ARI and the Silhouette score based on the normalized data for different choices of λ_w and λ_r based on the (a) K-means (b) K-means++ and (c) Adaptive K-means++ initialization. Data simulated with $\zeta = 0.4$ and $S = 2$.

5.4.2 Comparing MAD-Bayes to MBASIC

In this section, I evaluate my algorithm in more diverse settings and focus on its estimation performance. Besides varying $S = 2,\ 4$ and $\zeta = 0,\ 0.4$, I also varied $(I,\ J) = (4000,\ 10)$ or $(10000,\ 20)$ to investigate the algorithm's performance with data of different scales. Besides the log-normal distribution, I also simulated data with the negative binomial distribution, I let $\mu_{kls} \overset{\text{i.i.d.}}{\sim} N(2 \cdot 5^{s-1},\ 0.5^2)$, $\rho_{kls} = 5$, and $Y_{ikl}|\theta_{iks} = 1$ follow the negative binomial distribution with mean μ_{kls} and variance $\mu_{kls}(1 + \mu_{kls}/\rho_{kls})$. The combinations of (I, J), ζ, S and distributions in total gave us 16 different simulation settings. I simulated 10 data sets under each setting. For each data set, I chose 50 λ_r values based on Algorithm 5.4, and ran the algorithm once for each one.

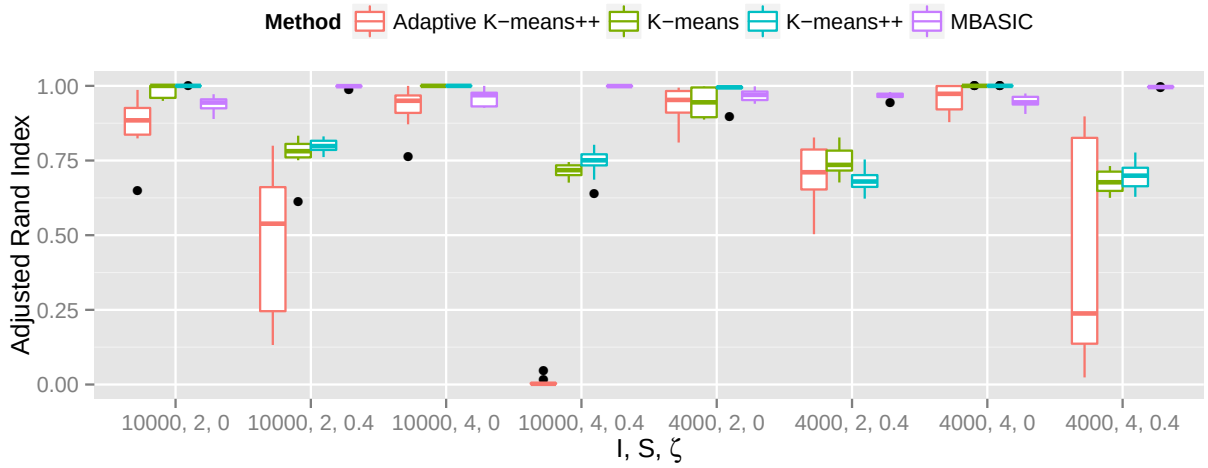
For each simulated data set, I compared the performance between the MAD-Bayes algorithm with three different initialization methods and MBASIC's generic E-M algorithm. Besides ARI, I also computed the State Prediction Error (SPE) for each method as follows:

$$\text{SPE} = \sqrt{\frac{\sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{s=1}^{S}(\theta_{iks} - \hat{\theta}_{iks})^2}{IKS}}.$$

While ARI measures how each method captures the clustering structure, SPE measures the accuracy in restoring the states of each unit. I also evaluated the execution time for each method. I determined that the MAD-Bayes algorithm converged if there was no more updates in parameters. For MBASIC's E-M algorithm, I determined that it converged if the relative increment in the log likelihood function was less than 10^{-10}, or the maximum relative update among its parameters was less than 10^{-5}. For model selection, I ran the MAD-Bayes algorithm with 50 λ_r values according to Algorithm 5.4 with each initialization method. I fitted MBASIC models with numbers of clusters between $J - 9$ to $J + 10$ and chose the best model according to the Bayesian Information Criteria. For each algorithm, I paralleled the fits for different candidate models using 20 CPUs.

I show by boxplots the comparison between the MAD-Bayes algorithm and MBASIC's E-M algorithm in Figures 5.4, 5.5 and 5.6. Its performance in ARI was among the best under all settings (Figures 5.4). For the log-normal distribution, the E-M algorithm performed the best in SPE (Figure 5.5(a)). For the negative binomial distribution, the E-M algorithm performed better

(a)



(b)

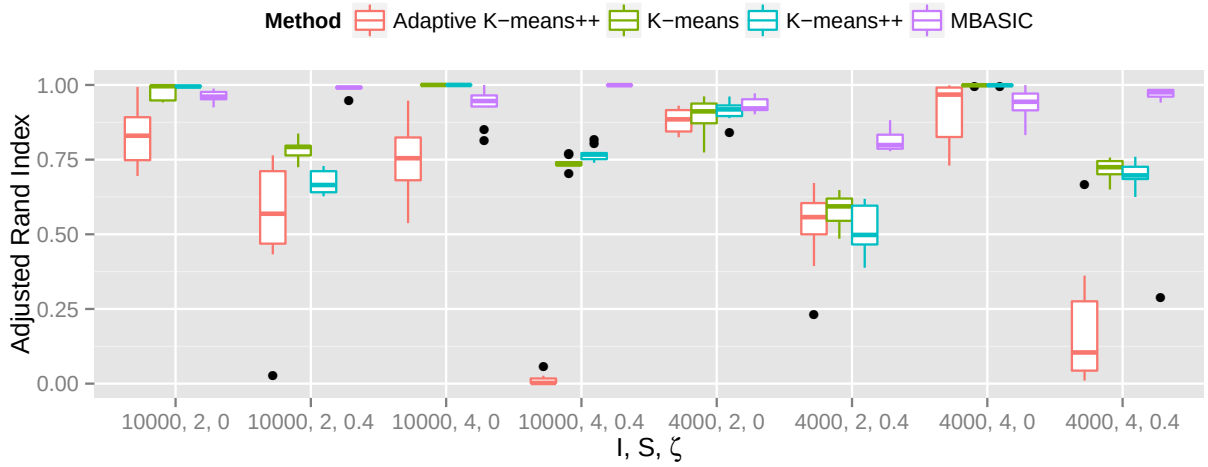
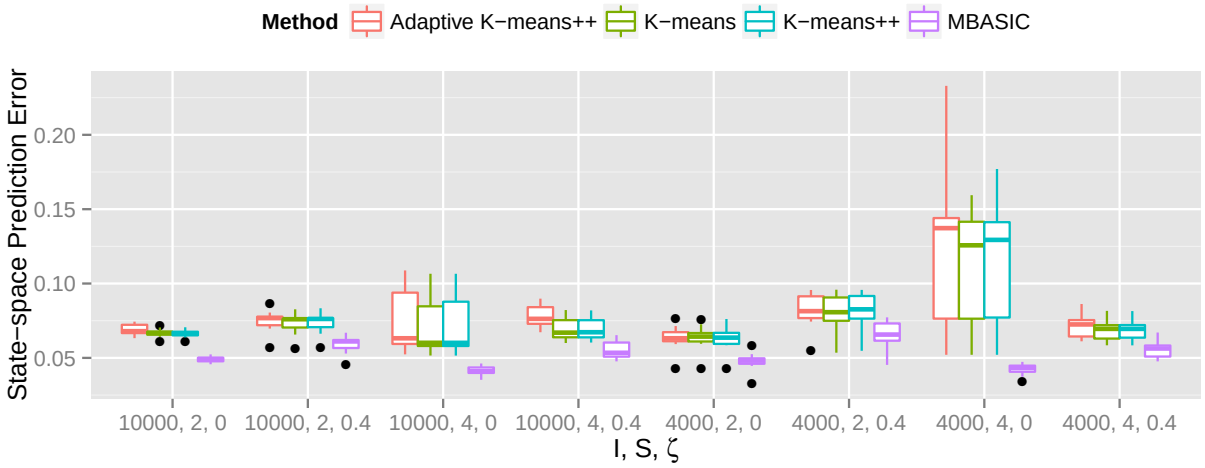


Figure 5.4 Comparison in ARI for the MAD-Bayes algorithm with three different initialization methods and MBASIC's E-M algorithm with (a) log-normal and (b) negative binomial distribution.

(a)



(b)

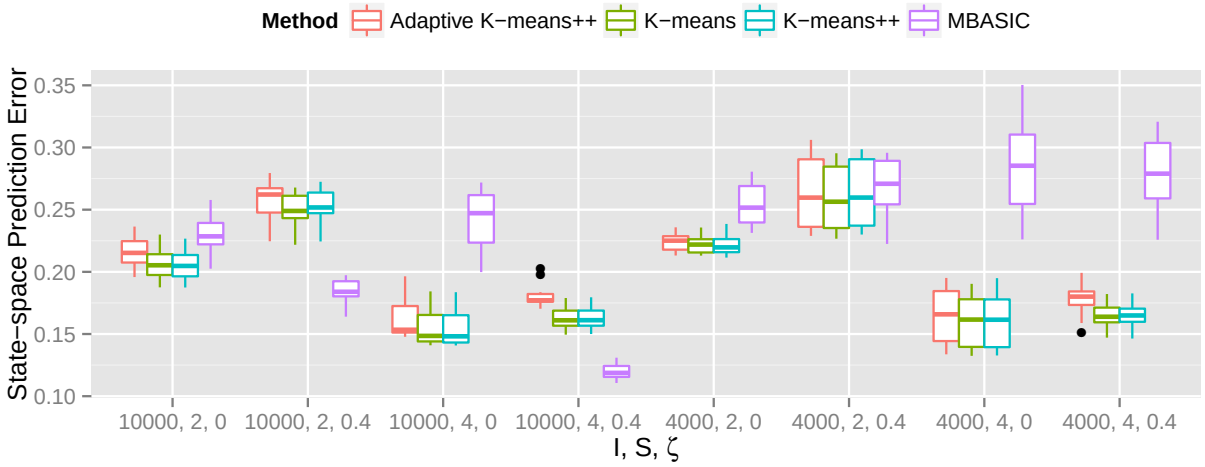
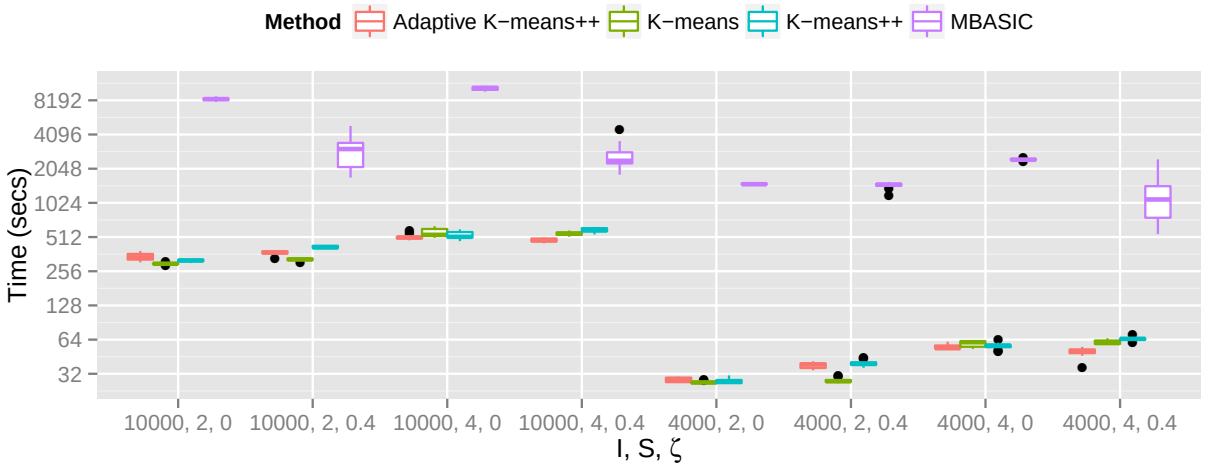


Figure 5.5 Comparison in SPE-W for the MAD-Bayes algorithm with three different initialization methods and MBASIC's E-M algorithm with (a) log-normal and (b) negative binomial distribution.

(a)



(b)

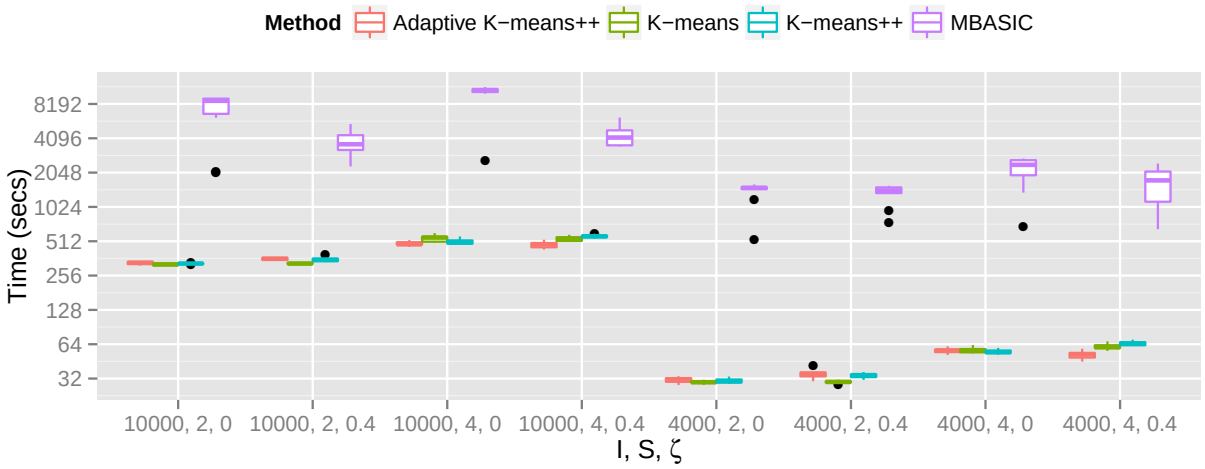


Figure 5.6 Comparison in execution time for the MAD-Bayes algorithm with three different initialization methods and MBASIC's E-M algorithm with (a) log-normal and (b) negative binomial distribution.

than MAD-Bayes under some settings, but worse in the others. Notably, when $\zeta = 0.4$, the E-M algorithm achieved higher ARI than MAD-Bayes. This was expected since MBASIC explicitely modeled the existence of singletons. With the K-means or the K-means++ initialization, MAD-Bayes achieved reasonable performance in ARI and SPE in general, even with mis-specified model assumptions. Specifically, MAD-Bayes performed consistently between the log-normal (Figure 5.4(a) and 5.5(a)) and the negative binomial distributions (Figure 5.4(b) and 5.5(b)). The biggest difference between the two algorithms is the execution time. In all settings, there are approximately 5 to 10 fold decrease in time for MAD-Bayes as compared to the E-M algorithm (Figures 5.6). Such an order of magnitude in time difference shows the potential advantage of applying MAD-Bayes in real settings. When I am agnostic about the "true" model structure of a real data set or only exploratory results are needed, applying MAD-Bayes to get fast model output can be a much preferred approach.

Across the three initialization methods, I notice that performances of the adaptive K-means++ initialization showed the least stability. Even with the log-normal distribution, it achieved poor ARI when $\zeta = 0.4$ and $(I, J) =$ (4000, 4), (10000, 2) or (10000, 4) (Figure 5.4(a)). I found that model fits with different λ_r values led to either too many or too few clusters. This problem remained as I tried more candidate λ_r values using Algorithm 5.4. This suggests that the adaptive K-means++ initialization may be so aggressive in jointly determining the number of clusters and the cluster centers that it exacerbate the local optima problem. This phenomenon was not observed for the K-means and K-means++ initialization. Overall, initialization using K-means or K-means++ seems more reliable than the adaptive K-means++ proposed by [6].

5.5 Conclusions and Discussion

I have developed an efficient algorithm for state-space inference and clustering in this chapter. I adopt the MAD-Bayes framework to derive a hard assignment objective function for the Bayesian MBASIC model. Compared to the generic E-M algorithm for the MBASIC model, my algorithm has shown great advantage in time efficiency.

A major contribution of this paper is that my proposed data-driven methods to select tuning parameters. [6] only showed that for properly chosen tuning parameters, the MAD-Bayes algorithm can restore the clustering structures. However, it did not propose a method for tuning parameter selection for practical analysis. I have demonstrated that the model fitting can be very sensitive to the tuning parameters; therefore, an automated procedure to determine their proper values is extremely important. Given the similarity between their objective function and Eqn. (5.9), I believe that Algorithm 5.4 is potentially applicable to their model as well.

The derivation of my algorithm depends on the log-normal distribution assumption. I acknowledge that for other distributions, the extension of such simple derivation is not straightforward. From a non-parametric point of view, my algorithms can still be applied as a exploratory analysis, similar to how K-means algorithms are applied without Gaussian assumptions. My simulation study shows that its performance in restoring the clustering structure and the state prediction can be robust against different data distributions and singletons.

Chapter 6

Conclusions

In this thesis, I present several novel computational tools for large-scale genomic and epigenomic inference.

In Chapter 2, I present the CSSP (**C**hIP-**S**eq **S**tatistical **P**ower) framework to calculate the power for ChIP-Seq experiments. I build an empirical Bayesian model for the process of accumulating genome-wide ChIP-Seq reads and estimate the power of calling enriched sites as a function of the sequencing depths. This framework also considers the fold change and minimum intensity thresholds to filter spurious enriched regions, thereby can be used to estimate the required sequencing depth for ChIP-Seq experiments using pilot data. This framework also enables the analysis of lab and batch effects, as well as the input depth of the matching control experiments.

In Chapter 3, I describe the atSNP (**a**ffinity **t**esting for regulatory **SNP**s) software, a `R` package for identifying rSNPs *in silico* from millions of SNP-motif pairs. Regulatory SNPs (rSNPs) are such SNPs that affect gene regulation by changing transcription factor (TF) binding affinities to genomic sequences. atSNP implements an importance sampling algorithm coupled with a first-order Markov model for the background nucleotide sequences to test the significance of affinity scores and SNP-driven changes in these scores. Application of atSNP with $>$20K SNPs indicates that atSNP is the only available tool for such a large-scale task.

In Chapter 4, I discuss a unified statistical framework, called MBASIC (**M**atrix **B**ased **A**nalysis for **S**tate-space **I**nference and **C**lustering), to address the problem of clustering analysis based on an underlying state-space. MBASIC simultaneously projects the observations onto a hidden state-space and infers clustered units in this space. The MBASIC framework offers flexibility in a number of aspects of experimental design, such as different numbers of replicates under individual

experimental conditions and missing values. Additionally, it is applicable to many parametric distributions. I have also developed an Expectation and Maximization algorithm with closed form E-step updates, thus enables the application of MBASIC to large scale integrative analysis for genomic and epigenomic data.

In Chapter 5, I establish a MAD-Bayes algorithm to perform simultaneous state-space inference and clustering. This algorithm is derived from an small variance asymptotic view of the Bayesian MBASIC model. Compared to MBASIC's E-M algorithm, my MAD-Bayes algorithm shows orders-of-magnitude improvement in computational time without sacrificing much estimation accuracy. I have also developed empirical procedures for selecting tuning parameters related to model selection, which is potentially applicable to other MAD-Bayes algorithms.

Methods in this thesis are implemented as R packages listed in Table 6.1. While I have conducted a number of numerical studies using both synthetic and real data to prove the usage of the tools in this thesis, applying them to newly generated data sets in order to derive insights in the gene regulatory systems should be a future research direction. For example, the atSNP software in Chapter 2 can now be used to fast screening millions of SNP-motif pairs to predict putative regulatory SNPs; such discoveries can be thus connected with the already available GWAS data to explore gene pathology. Methods in Chapter 4 and 5 can be applied to jointly analyze data collected on a set of loci from different experiment types, such as gene expression, transcription factor binding and allele-specific binding. The unified analytical approaches can improve the current practice of separate analyzing different data sets using different tools by not only enhancing computational efficiency but also retaining data fidelity.

Another research direction is to incorporate more complex model structures to warrant realistic data structures. For example, the CSSP framework in Chapter 2 assumes that data from different loci are independently distributed; the MBASIC framework in Chapter 4 further assumes that they have homogeneous distributions. Incorporating correlated and/or heterogeneous distribution assumptions would challenge the computational efficiency emphasized across this thesis, but would be worthwhile as the evolving data generation technology continues to drive the need for more

Table 6.1 A list of software for the methods in this thesis.

Package	Method Implemented	Availability
`CSSP`	CSSP (Chapter 2)	`http://bioconductor.org/packages/release/bioc/html/CSSP.html`
`atSNP`	atSNP (Chapter 3)	`http://github.com/chandlerzuo/atsnp`
`MBASIC`	MBASIC, MAD-Bayes(Chapter 4, 5)	`http://github.com/chandlerzuo/mbasic`

granular and fundamental understanding of the genome systems. Such extensions may also become feasible with the ever-improving power of the computer systems.

Bibliography

[1] Madhanagopal Anandapadamanaban, Cecilia Andresen, Sara Helander, Yoshifumi Ohyama, Marina I. Siponen, Patrik Lundstrm, Tetsuro Kokubo, Mitsuhiko Ikura, Martin Moche, and Maria Sunnerhagen. High-resolution structure of TBP with TAF1 reveals anchoring patterns in transcriptional regulation. *Nature Structural & Molecular Biology*, 20:1008–1014, 2013.

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

[3] Malin C. Andersen, Pär G. Engström, Stuart Lithwick, David Arenillas, Per Eriksson, Boris Lenhard, Wyeth W. Wasserman, and Jacob Odeberg. In silico detection of sequence variations modifying transcriptional regulation. *PLoS Computational Biology*, 4(1):e5, 2008.

[4] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

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

[6] Tamara Broderick, Brian Kulis, and Michael I. Jordan. Mad-bayes: Map-based asymptotic derivations from bayes. *Proceedings of the 30th International Conference on Machine Learning*, 28:226234, 2013.

[7] Hock Peng Chan, Nancy Ruonan Zhang, and Louis H.Y. Chen. Importance sampling of word patterns in DNA and protein sequences. *Journal of Computational Biology*, 17(12):16971709, 2010.

[8] Yiwen Chen, Nicolas Negre, Qunhua Li, Joanna O. Mieczkowska, Matthew Slattery, Tao Liu, Yong Zhang, Tae-Kyung Kim, Housheng H. He, Jennifer Zieba, Yijun Ruan, Peter J. Bickel, Richard M. Myers, Barbara J. Wold, Kevin P. White, Jason D. Lieb, and X. Shirley Liu. Systematic evaluation of factors influencing ChIP-seq fidelity. *Nature Methods*, advance online publication, 2012.

[9] Chao Cheng, Koon-Kiu Yan, Woochang Hwang, Jiang Qian, Nitin Bhardwaj, Joel Rozowsky, Zhi John Lu, Wei Niu, Pedro Alves, Masaomi Kato, Michael Snyder, and Mark Gerstein. Construction and analysis of an integrated regulatory network derived from high-throughput sequencing data. *PLoS Computational Biology*, 7, 2011.

[10] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

[11] Louis C. Doré, Timothy M. Chlon, Christopher D. Brown, Kevin P. White, and John D. Crispino. Chromatin occupancy analysis reveals genome-wide GATA factor switching during hematopoiesis. *Blood*, 119(16):3724–3733, 2012.

[12] Chris Fraley and Adrian E. Raftery. Model-based clustering, discriminant analysis and density estimation. *Journal of the American Statistical Association*, 97:611–631, 2002.

[13] Debra L. Fulton, Saravanan Sundararajan, Gwenael Badis, Timothy R. Hughes, Wyeth W. Wasserman, Jared C. Roach, and Rob Sladek. Tfcat: the curated catalog of mouse and human transcription factors. *Genome Biology*, 10(3):R29, 2009.

[14] Xin Gao, Kirby D. Johnson, Yuan-I Chang, Meghan E. Boyer, Colin N. Dewey, Jing Zhang, and Emery H. Bresnick. Gata2 cis-element is required for hematopoietic stem cell generation in the mammalian embryo. *Journal of Experimental Medicine*, 210(13):2833–42, 2013.

[15] Mark B. Gerstein, Anshul Kundaje, Manoj Hariharan, Stephen G. Landt, Koon-Kiu Yan, Chao Cheng, Xinmeng Jasmine Mu, Ekta Khurana, Joel Rozowsky, Roger Alexander, Renqiang Min, Pedro Alves, Alexej Abyzov, Nick Addleman, Nitin Bhardwaj, Alan P. Boyle, Philip Cayting, Alexandra Charos, David Z. Chen, Yong Cheng, Declan Clarke, Catharine Eastman, Ghia Euskirchen, Seth Frietze, Yao Fu, Jason Gertz, Fabian Grubert, Arif Harmanci, Preti Jain, Maya Kasowski, Phil Lacroute, Jing Leng, Jin Lian, Hannah Monahan, Henriette O/'Geen, Zhengqing Ouyang, E. Christopher Partridge, Dorrelyn Patacsil, Florencia Pauli, Debasish Raha, Lucia Ramirez, Timothy E. Reddy, Brian Reed, Minyi Shi, Teri Slifer, Jing Wang, Linfeng Wu, Xinqiong Yang, Kevin Y. Yip, Gili Zilberman-Schapira, Serafim Batzoglou, Arend Sidow, Peggy J. Farnham, Richard M. Myers, Sherman M. Weissman, and Michael Snyder. Architecture of the human regulatory network derived from ENCODE data. *Nature*, 489:91–100, 2012.

[16] Mark B. Gerstein, Zhi John Lu, Eric L. Van Nostrand, Chao Cheng, Bradley I. Arshinoff, Tao Liu, Kevin Y. Yip, Rebecca Robilotto, Andreas Rechtsteiner, Kohta Ikegami, et al. Integrative analysis of the Caenorhabditis elegans genome by the modENCODE project. *Science*, 330(6012):1775–1787, 2010.

[17] Charles E. Grant, Timothy L. Bailey, and William Stafford Nobel. FIMO: Scanning for occurrences of a given motif. *Bioinformatics*, 7:1017, 2011.

[18] Obi L. Griffith, Stephen B. Montgomery, Bridget Bernier, Bryan Chu, Katayoon Kasaian, Stein Aerts, Shaun Mahony, Monica C. Sleumer, Mikhail Bilenky, Maximilian Haeussler, et al. Oreganno: an open-access community-driven resource for regulatory annotation. *Nucleic acids research*, 36(suppl 1):D107–D113, 2008.

[19] Jennifer Harrow, Adam Frankish, Jose M. Gonzalez, Electra Tapanari, Mark Diekhans, Felix Kokocinski, Bronwen L. Aken, Daniel Barrell, Amonida Zadissa, Stephen Searle, If Barnes, Alexandra Bignell, Veronika Boychenko, Toby Hunt, Mike Kay, Gaurab Mukherjee, Jeena Rajan, Gloria Despacio-Reyes, Gary Saunders, Charles Steward, Rachel Harte, Michael Lin, Cdric Howald, Andrea Tanzer, Thomas Derrien, Jacqueline Chrast, Nathalie Walters, Suganthi Balasubramanian, Baikang Pei, Michael Tress, Jose Manuel Rodriguez, Iakes Ezkurdia, Jeltje van Baren, Michael Brent, David Haussler, Manolis Kellis, Alfonso Valencia, Alexandre Reymond, Mark Gerstein, Roderic Guig, , and Tim J. Hubbard. GENCODE: The reference human genome annotation for The ENCODE Project. *Genome Research*, 22:1760–1774, 2012.

[20] Joshua Ho, Eric Bishop, Peter Karchenko, Nicolas Negre, Kevin White, and Peter Park. ChIP-chip versus ChIP-seq: Lessons for experimental design and data analysis. *BMC Genomics*, 12(1):134, 2011.

[21] Darcy W. Holley, Beezly S. Groh, Glenn Wozniak, Dallas R. Donohoe, Wei Sun, Virginia Godfrey, and Scott J. Bultman. The BRG1 Chromatin Remodeler Regulates Widespread Changes in Gene Expression and Cell Proliferation During B Cell Activation. *Journal of Cellular Physiology*, 229(1):44–52, 2014.

[22] Amy P. Hsu, Kirby D. Johnson, E Liana Falcone, Rajendran Sanalkumar, Lauren Sanchez, Dennis D. Hickstein, Jennifer Cuellar-Rodriguez, Jacob E. Lemieux, Christa S. Zerbe, Emery H. Bresnick, et al. Gata2 haploinsufficiency caused by mutations in a conserved intronic element leads to monomac syndrome. *Blood*, 121(19):3830–3837, 2013.

[23] Gangqing Hu, Dustin E. Schones, Kairong Cui, River Ybarra, Daniel Northrup, Qingsong Tang, Luca Gattinoni, Nicholas P. Restifo, Suming Huang, and Keji Zhao. Regulation of nucleosome landscape and transcription factor targeting at tissue-specific enhancers by BRG1. *Genome Research*, 21(10):1650–1658, 2011.

[24] James X. Hu, Hongyu Zhao, and Harrison H. Zhou. False discovery rate control with groups. *Journal of the American Statistical Association*, 105(491):1215–1227, 2010.

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

[26] Hongkai Ji, Xia Li, Qian-fei Wang, and Yang Ning. Differential principle component analysis of ChIP-seq. *Proceedings of the National Academy of Sciences*, 110:6789–6794, 2013.

[27] Kirby D. Johnson, Amy P. Hsu, Myung-Jeom Ryu, Jinyong Wang, Xin Gao, Meghan E. Boyer, Yangang Liu, Youngsook Lee, Katherine R. Calvo, Sündüz Keleş, et al. Cis-element mutation in a GATA-2-dependent immunodeficiency syndrome governs hematopoiesis and vascular integrity. *The Journal of Clinical Investigation*, 122(10):3692, 2012.

[28] Maya Kasowski, Fabian Grubert, Christopher Heffelfinger, Manoj Hariharan, Akwasi Asabere, Sebastian M. Waszak, Lukas Habegger, Joel Rozowsky, Minyi Shi, Alexander E. Urban, et al. Variation in transcription factor binding among humans. *Science*, 328(5975):232–235, 2010.

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

[30] Pouya Kheradpour and Manolis Kellis. Systematic discovery and characterization of regulatory motifs in ENCODE TF binding experiments. *Nucleic acids research*, 42(5):2976–2987, 2014.

[31] Shin-Il Kim, Emery H. Bresnick, and Scott J. Bultman. BRG1 directly regulates nucleosome structure and chromatin looping of the α globin locus to activate transcription. *Nucleic Acids Research*, 37(18):6019–6027, 2009.

[32] Shin-Il Kim, Scott J. Bultman, Christine M. Kiefer, Ann Dean, and Emery H. Bresnick. BRG1 requirement for long-range interaction of a locus control region with a downstream promoter. *Proceedings of the National Academy of Sciences*, 106(7):2259–2264, 2009.

[33] Pei Fen Kuan, Dongjun Chung, Guangjin Pan, James A. Thomson, Ron Stewart, and Sündüz Keleş. A statistical framework for the analysis of ChIP-seq data. *Journal of the American Statistical Association*, 106(495):891–903, 2011.

[34] Brian Kulis and Michael I. Jordan. Revisiting k-means: New algorithms via bayesian nonparametrics. *Proceedings of the 29 th International Conference on Machine Learning*, pages 513–520, 2012.

[35] Galih Kunarso, Na-Yu Chia, Justin Jeyakani, Catalina Hwang, Xinyi Lu, Yun-Shen Chan, Huck-Hui Ng, and Guillaume Bourque. Transposable elements have rewired the core regulatory network of human embryonic stem cells. *Nature Genetics*, 42(7):631–634, 2010.

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

[37] Seokho Lee, Jianhua Huang, and Jianhua Hu. Sparse logistic principal components analysis for binary data. *The Annals of Applied Statistics*, 4:1579–1601, 2010.

[38] Kun Liang and Sündüz Keleş. Normalization of ChIP-seq data with control. *BMC Bioinformatics*, 13(1):199, 2012.

[39] Kun Liang and Sündüz Keleş. Detecting differential binding of transcription factors with ChIP-seq. *Bioinformatics*, 28:121–122, 2012.

[40] Amelia K. Linnemann, Henriette OGeen, Sündüz Keleş, Peggy J. Farnham, and Emery H. Bresnick. Genetic framework for gata factor function in vascular biology. *Proceedings of the National Academy of Sciences*, 108(33):13641–13646, 2011.

[41] Lovisa Lovmar, Annika Ahlford, Mats Jonsson, and Ann-Christine Syvänen. Silhouette scores for assessment of SNP genotype clusters. *BMC Genomics*, 6(1):35, 2005.

[42] Geoff Macintyre, James Bailey, Izhak Haviv, and Adam Kowalczyk. is-rSNP: a novel technique for in silico regulatory SNP detection. *Bioinformatics*, 26(18):524–530, 2010.

[43] Anthony Mathelier, Xiaobei Zhao, Allen W. Zhang, François Parcy, Rebecca Worsley-Hunt, David J. Arenillas, Sorana Buchman, Chih-yu Chen, Alice Chou, Hans Ienasescu, Jonathan Lim, Casper Shyr, Ge Tan, Michelle Zhou, Boris Lenhard, Albin Sandelin, and Wyeth W. Wasserman. JASPAR 2014: an extensively expanded and updated open-access database of transcription factor binding profiles. *Nucleic Acids Research*, 42(D1):D142–D147, 2014.

[44] Anthony Mathelier, Xiaobei Zhao, Allen W. Zhang, François Parcy, Rebecca Worsley-Hunt, David J. Arenillas, Sorana Buchman, Chih-yu Chen, Alice Chou, Hans Ienasescu, et al. Jaspar 2014: an extensively expanded and updated open-access database of transcription factor binding profiles. *Nucleic acids research*, page gkt997, 2013.

[45] Matthew T. Maurano, Richard Humbert, Eric Rynes, Robert E. Thurman, Eric Haugen, Hao Wang, Alex P. Reynolds, Richard Sandstrom, Hongzhu Qu, Jennifer Brody, et al. Systematic localization of common disease-associated variation in regulatory DNA. *Science*, 337(6099):1190–1195, 2012.

[46] Ryan McDaniell, Bum-Kyu Lee, Lingyun Song, Zheng Liu, Alan P. Boyle, Michael R. Erdos, Laura J. Scott, Mario A. Morken, Katerina S. Kucera, Anna Battenhouse, et al. Heritable individual-specific and allele-specific chromatin signatures in humans. *Science*, 328(5975):235–239, 2010.

[47] Kevin S. Myers, Huihuang Yan, Irene M. Ong, Dongjun Chung, Kun Liang, Frances Tran, Sündüz Keleş, Robert Landick, and Patricia J. Kiley. Genome-scale analysis of escherichia coli fnr reveals the complexity of bacterial regulon structure. *PLoS Genetics*, 9(6):e1003565, 2013.

[48] Richard M. Myers, John Stamatoyannopoulos, Michael Snyder, Ian Dunham, Ross C. Hardison, Bradley E. Bernstein, Thomas R. Gingeras, W James Kent, Ewan Birney, Barbara Wold, and et al. A Users Guide to the Encyclopedia of DNA Elements (ENCODE). *PLoS Biology*, 9(4):21, 2011.

[49] Shane Neph1, Andrew B. Stergachis, Alex Reynolds, Richard Sandstrom, Elhanan Borenstein, and John A. Stamatoyannopoulos. Circuitry and dynamics of human transcription factor regulatory networks. *Cell*, 150:12741286, 2012.

[50] David A. Nix, Samir J. Courdy, and Kenneth M. Boucher. Empirical methods for controlling false positives and estimating confidence in ChIP-seq peaks. *BMC Bioinformatics*, 9(1):523, 2008.

[51] H. Pages. BSgenome: Infrastructure for Biostrings-based genome data packages., 2014. http://www.bioconductor.org/packages/release/bioc/html/BSgenome.html.

[52] Athma A. Pai, Jonathan K. Pritchard, and Yoav Gilad. The genetic and mechanistic basis for variation in gene regulation. *PLoS Genetics*, 11(1):e1004857, 2015.

[53] William C. Parr and William R. Schucany. Minimum distance and robust estimation. *Journal of the American Statistical Association*, 75(371):616–624, 1980.

[54] William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[55] Naim U. Rashid, Paul G. Giresi, Joseph G. Ibrahim, Wei Sun, and Jason D. Lieb. ZINBA integrates local covariates with DNA-seq data to identify broad and narrow regions of enrichment, even within amplified genomic regions. *Genome Biology*, 12(7):R67, 2011.

[56] Alberto Riva. Large-scale computational identification of regulatory SNPs with rSNP-MAPPER. *BMC Genomics*, 13(suppl 4):S7, 2012.

[57] Peter J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[58] Sushmita Roy, Jason Ernst, Peter V. Kharchenko, Pouya Kheradpour, Nicolas Negre, Matthew L. Eaton, Jane M. Landolin, Christopher A. Bristow, Lijia Ma, Michael F. Lin, and with modENCODE Consortium et al. Identification of Functional Elements and Regulatory Circuits by Drosophila modENCODE. *Science*, 330(6012):1787–1797, 2010.

[59] Sushmita Roy, Ilan Wapinski, Jenna Pfiffner, Courtney French, Amanda Socha, Jay Konieczka, Naomi Habib, Manolis Kellis, Dawn Thompson, and Aviv Regev. Arboretum: reconstruction and analysis of the evolutionary history of condition-specific transcriptional modules. *Genome Research*, 23(6):1039–1050, 2013.

[60] Joel Rozowsky, Ghia Euskirchen, Raymond K. Auerbach, Zhengdong D. Zhang, Theodore Gibson, Robert Bjornson, Nicholas Carriero, Michael Snyder, and Mark B. Gerstein. Peakseq enables systematic scoring of ChIP-seq experiments relative to controls. *Nature Biotechnology*, 27(1):66–75, 2009.

[61] Dominic Schmidt, Michael D. Wilson, Benoit Ballester, Petra C. Schwalie, Gordon D. Brown, Aileen Marshall, Claudia Kutter, Stephen Watt, Celia P. Martinez-Jimenez, Sarah Mackay, et al. Five-vertebrate ChIP-seq reveals the evolutionary dynamics of transcription factor binding. *Science*, 328(5981):1036–1040, 2010.

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

[63] Gordon K. Smyth. Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, 3(1):1–25, 2004.

[64] John D. Storey. The positive false discovery rate: a bayesian interpretation and the q-value. *The Annals of Statistics*, 31(6):2013–2035, 2003.

[65] Gary D. Stormo, Thomas D. Schneider, Larry Gold, and Andrzej Ehrenfeucht. Use of the perceptronalgorithm to distinguish translational initiation sites in e. coli. *Nucleic Acids Research*, 10(9):2997–3011, 1982.

[66] Aravind Subramanian, Pablo Tamayo, Vamsi K. Mootha, Sayan Mukherjee, Benjamin L. Ebert, Michael A. Gillette, Amanda Paulovich, Scott L. Pomeroy, Todd R. Golub, Eric S. Lander, and Jill P. Mesirov. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 102(43):15545–15550, 2005.

[67] The ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489:57–74, 2012.

[68] Gert Thijs, Magali Lescot, Kathleen Marchal, Stephane Rombauts, Bart De Moor, Pierre Rouzé, and Yves Moreau. A higher-order background model improves the detection of promoter regulatory elements by Gibbs sampling. *Bioinformatics*, 17(12):1113–1122, 2001.

[69] Morgane Thomas-Chollier, Andrew Hufton, Matthias Heinig, Sean O'Keeffe, Nassim El Masri, Helge G. Roider, Thomas Manke, and Martin Vingron. Transcription factor binding predictions using trap for the analysis of ChIP-seq data and regulatory SNPs. *Nature Protocols*, 6(12):1860–1869, 2011.

[70] Peter Waltman, Thadeous Kacmarczyk, Ashley R. Bate, Daniel B. Kearns, David J. Reiss, Patrick Eichenberger, and Richard Bonneau. Multi-species integrative biclustering. *Genome Biology*, 11(9):R96, 2010.

[71] Jie Wang, Jiali Zhuang, Sowmya Iyer, XinYing Lin, Troy W. Whitfield, Melissa C. Greven, Brian G. Pierce, Xianjun Dong, Anshul Kundaje, Yong Cheng, Oliver J. Rando, Ewan Birney, Richard M. Myers, William S. Noble, Michael Snyder, and Zhiping Weng. Sequence features and chromatin structure around the genomic regions bound by 119 human transcription factors. *Genome Research*, 22:1798–1812, 2012.

[72] Yingying Wei, Xia Li, Qian fei Wang, and Hongkai Ji. iASeq: integrative analysis of allele-specificity of protein-DNA interactions in multiple ChIP-seq datasets. *BMC Genomics*, 13, 2012.

[73] Yingying Wei, Toyoaki Tenzen, and Hongkai Ji. Joint analysis of differential gene expression in multiple studies using correlation motifs. *Biostatistics*, 16:31–46, 2015.

[74] Elizabeth G. Wilbanks and Marc T. Facciotti. Evaluation of algorithm performance in ChIP-seq peak detection. *PloS One*, 5(7):e11471, 2010.

[75] Gary D. Wu, Jun Chen, Christian Hoffmann, Kyle Bittinger, Ying-Yu Chen, Sue A. Keilbaugh, Meenakshi Bewtra, Dan Knights, William A. Walters, Rob Knight, et al. Linking long-term dietary patterns with gut microbial enterotypes. *Science*, 334(6052):105–108, 2011.

[76] Weisheng Wu, Yong Cheng, Cheryl A. Keller, Jason Ernst, Swathi Ashok Kumar, Tejaswini Mishra, Christapher Morrissey, Christine M. Dorman, Kuan-Bei Chen, Daniela Drautz, et al. Dynamics of the epigenetic landscape during erythroid differentiation after gata1 restoration. *Genome Research*, 21(10):1659–1671, 2011.

[77] Han Xu, Lusy Handoko, Xueliang Wei, Chaopeng Ye, Jianpeng Sheng, Chia-Lin Wei, Feng Lin, and Wing-Kin Sung. A signal–noise model for significance analysis of ChIP-seq with negative control. *Bioinformatics*, 26(9):1199–1204, 2010.

[78] Jian Yan, Martin Enge, Thomas Whitington, Kashyap Dave, Jianping Liu, Inderpreet Sur, Bernhard Schmierer, Arttu Jolma, Teemu Kivioja, Minna Taipale, et al. Transcription factor binding in human cells occurs in dense clusters formed around cohesin anchor sites. *Cell*, 154(4):801–813, 2013.

[79] Feng Yue, Yong Cheng, Alessandra Breschi, Jeff Vierstra, Weisheng Wu, Tyrone Ryba, Richard Sandstrom, Zhihai Ma, Carrie Davis, Benjamin D. Pope, et al. A comparative encyclopedia of DNA elements in the mouse genome. *Nature*, 515(7527):355–364, 2014.

[80] Xin Zeng, Rajendran Sanalkumar, Emery H. Bresnick, Hongda Li, Qiang Chang, and Sündüz Keleş. jMOSAiCS: joint analysis of multiple ChIP-seq datasets. *Genome Biology*, 14(4):R38, 2013.

[81] Chaolin Zhang, Zhenyu Xuan, Stefanie Otto, John R. Hover, Sean R. McCorkle, Gail Mandel, and Michael Q. Zhang. A clustering property of highly-degenerate transcription factor binding sites in the mammalian genome. *Nucleic Acids Research*, 34(8):2238–2246, 2006.

[82] Xuekui Zhang, Gordon Robertson, Martin Krzywinski, Kaida Ning, Arnaud Droit, Steven Jones, and Raphael Gottardo. Pics: Probabilistic inference for ChIP-seq. *Biometrics*, 67(1):151–163, 2011.

[83] Yong Zhang, Tao Liu, Clifford A. Meyer, Jérôme Eeckhoute, David S. Johnson, Bradley E. Bernstein, Chad Nusbaum, Richard M. Myers, Myles Brown, Wei Li, et al. Model-based analysis of ChIP-seq (macs). *Genome Biology*, 9(9):R137, 2008.

[84] Chandler Zuo, Kyle J. Hwitt, Emery H. Bresnick, and Sündüz Keleş. A hierarchical framework for state-space inference and clustering. *arXiv preprint arXiv:1505.04883*, 2015.

[85] Chandler Zuo and Sündüz Keleş. A statistical framework for power calculations in ChIP-seq experiments. *Bioinformatics*, 2013.

[86] Chandler Zuo, Sunyoung Kim, and Sündüs Keleş. atSNP: transcription factor binding affinity testing for regulatory SNP detection. *Bioinformatics*, 2015. To appear.

Appendix A: Supplementary Figures for Chapter 3

A.1 Sequence logo plots for commonly identified SNP-PWM pairs in Table 3.2

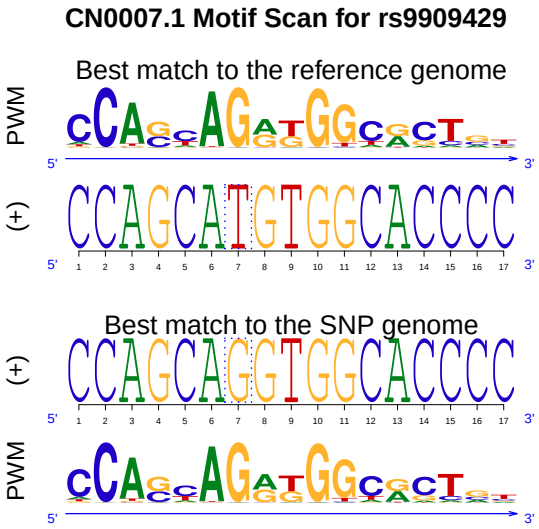


Figure A.1 Sequence logo plot for CN0007.1-rs9909429.

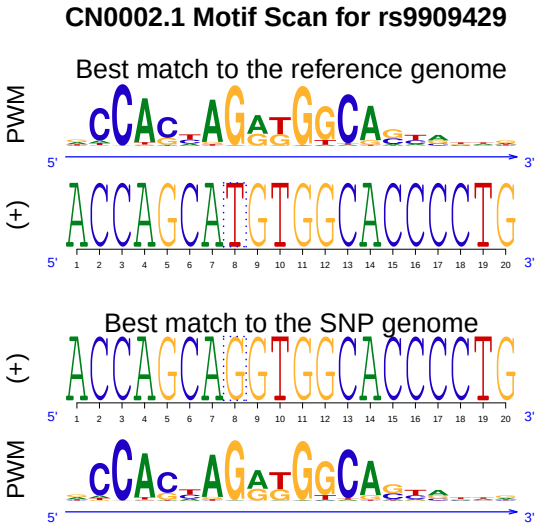


Figure A.2 Sequence logo plot for CN0002.1-rs9909429.

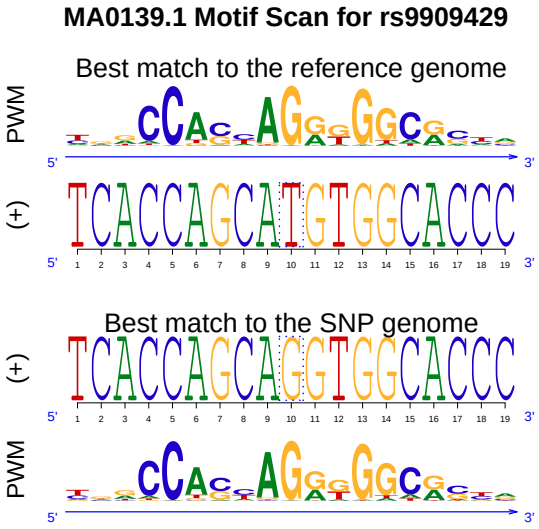


Figure A.3 Sequence logo plot for MA0139.1-rs9909429.

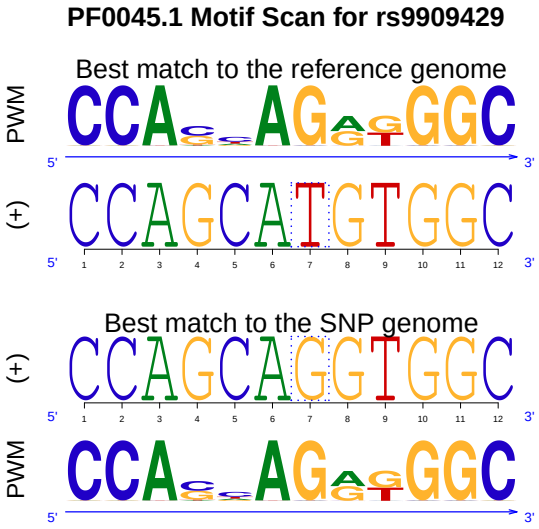


Figure A.4 Sequence logo plot for PF0045.1-rs9909429.

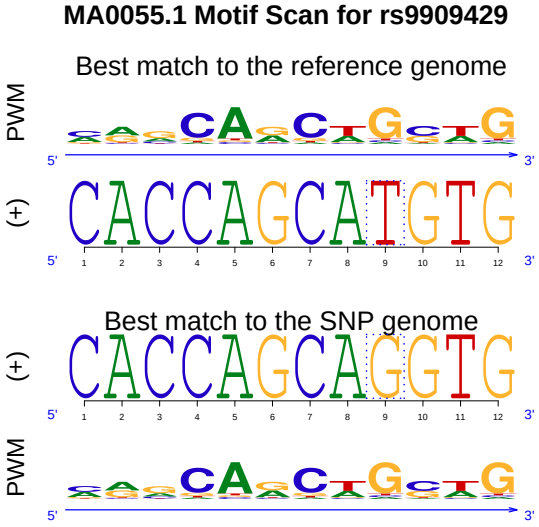


Figure A.5 Sequence logo plot for MA0055.1-rs9909429.

A.2 Sequence logo plots for SNP-PWM pairs identified only by atSNP in Table 3.2

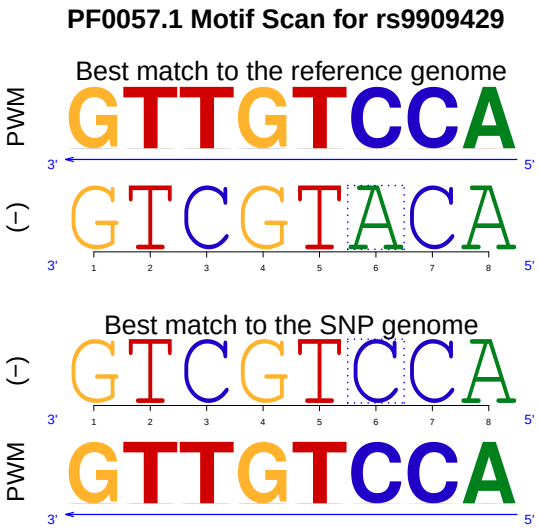


Figure A.6 Sequence logo plot for PF0057.1-rs9909429.

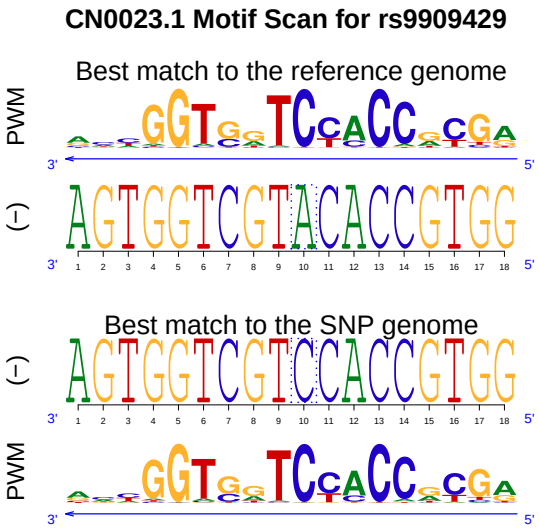


Figure A.7 Sequence logo plot for CN0023.1-rs9909429.

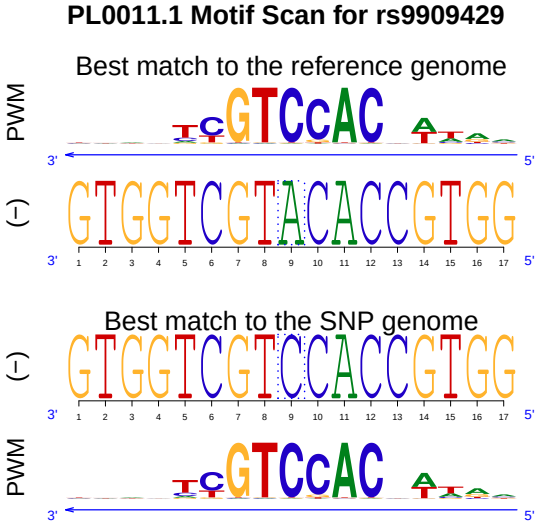


Figure A.8 Sequence logo plot for PL0011.1-rs9909429.

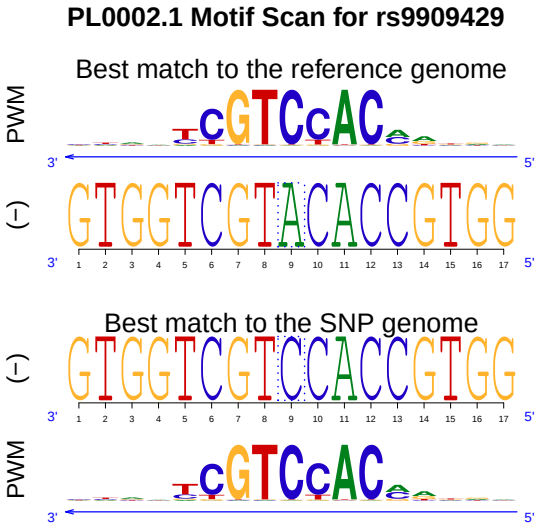


Figure A.9 Sequence logo plot for PL0002.1-rs9909429.

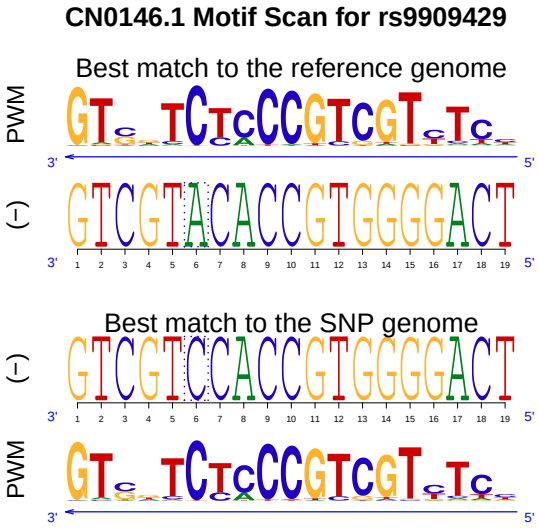


Figure A.10 Sequence logo plot for CN0146.1-rs9909429.

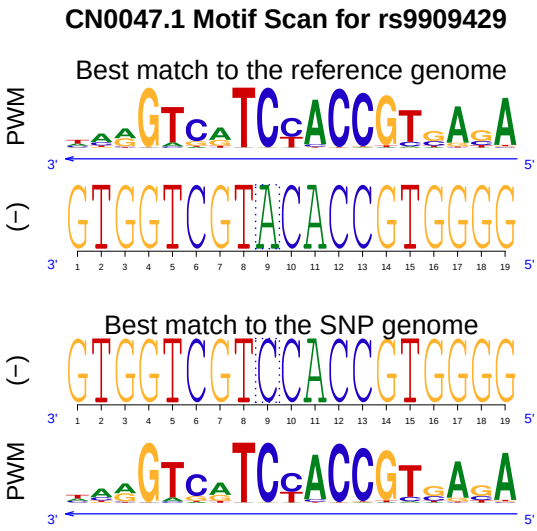


Figure A.11 Sequence logo plot for CN0047.1-rs9909429.

A.3 Sequence logo plots for SNP-PWM pairs identified only by is-rSNP in Table 3.2

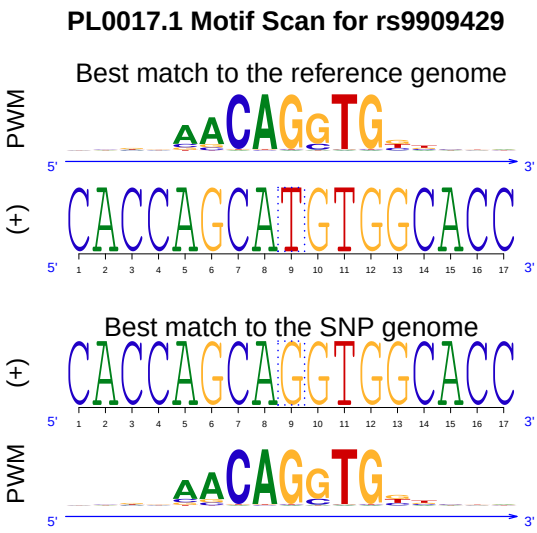


Figure A.12 Sequence logo plot for PL0017.1-rs9909429.

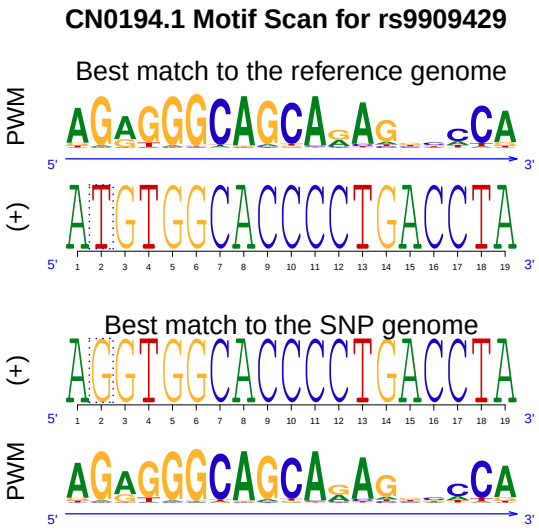


Figure A.13 Sequence logo plot for CN0194.1-rs9909429.

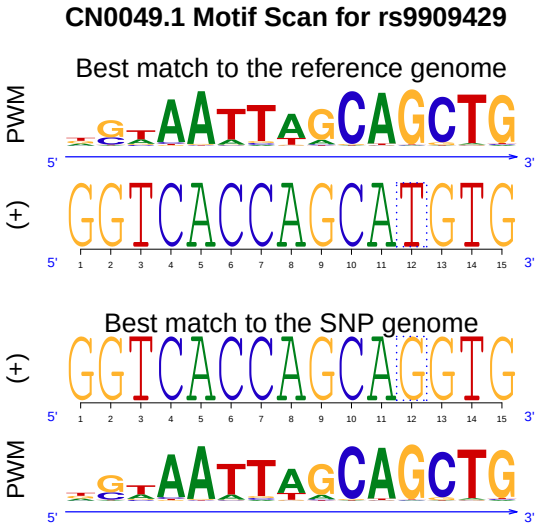


Figure A.14 Sequence logo plot for CN0049.1-rs9909429.

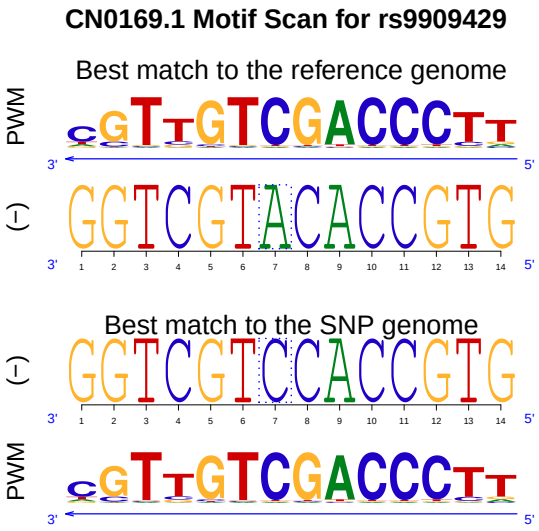


Figure A.15 Sequence logo plot for CN0169.1-rs9909429.

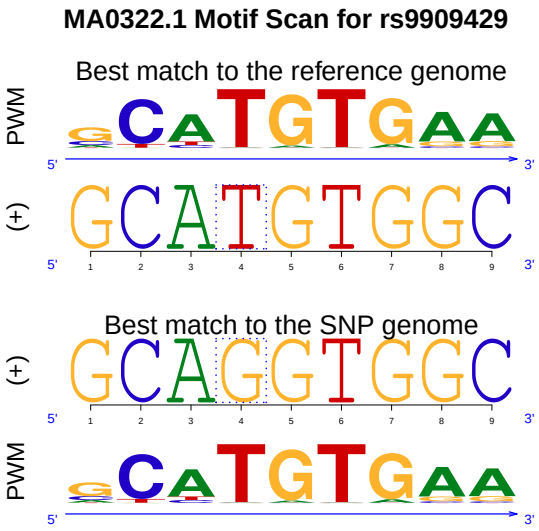


Figure A.16 Sequence logo plot for MA0322.1-rs9909429.

A.3.1 Sequence logo plots for SNP-PWM pairs in Table 3.3

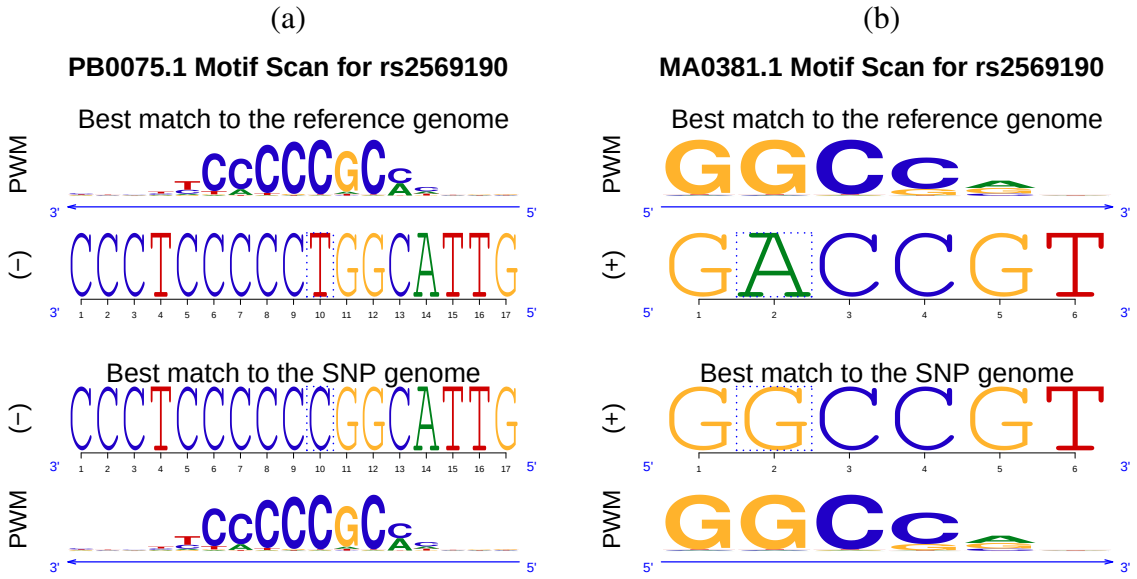


Figure A.17 Sequence logo plot for rs2569190 with the motif of the top regulatory effect in (a) the reported TF family; and (b) the whole JASPAR library.

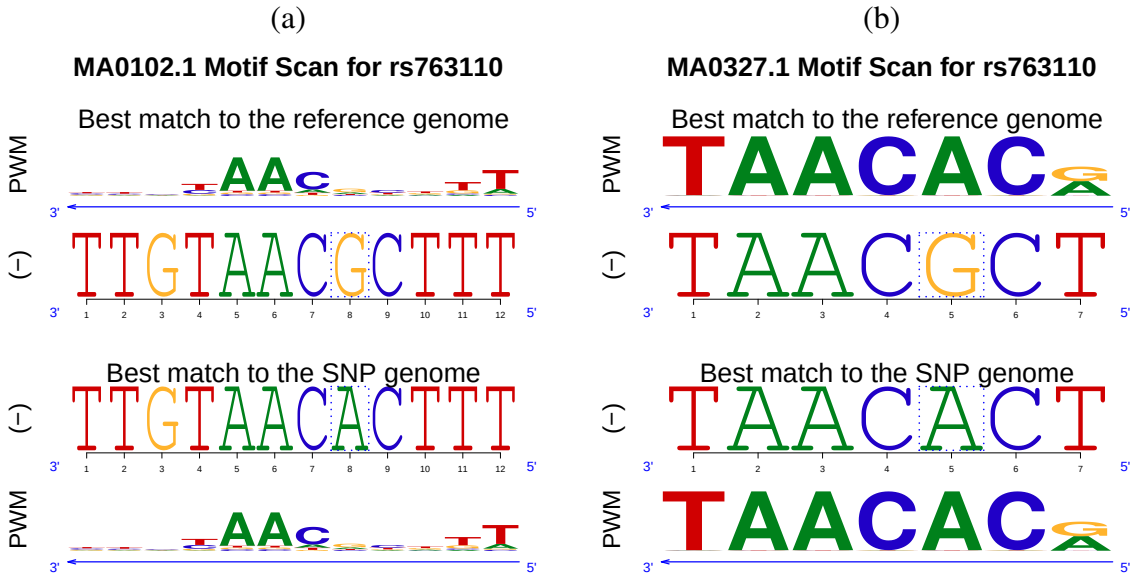


Figure A.18 Sequence logo plot for rs763110 with the motif of the top regulatory effect in (a) the reported TF family; and (b) the whole JASPAR library.

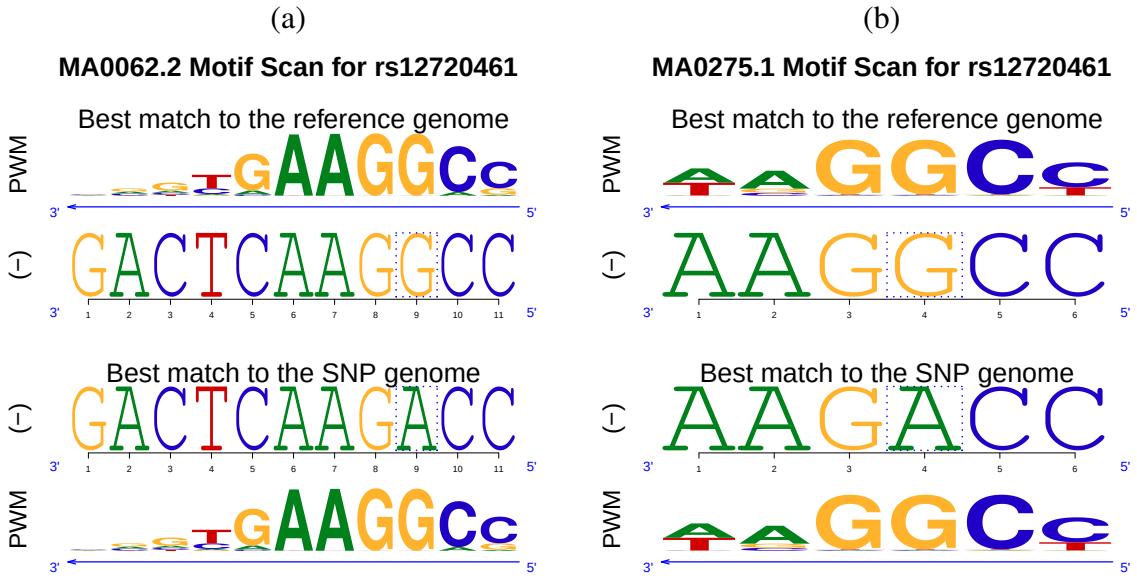


Figure A.19 Sequence logo plot for rs12720461 with the motif of the top regulatory effect in (a) the reported TF family; and (b) the whole JASPAR library.

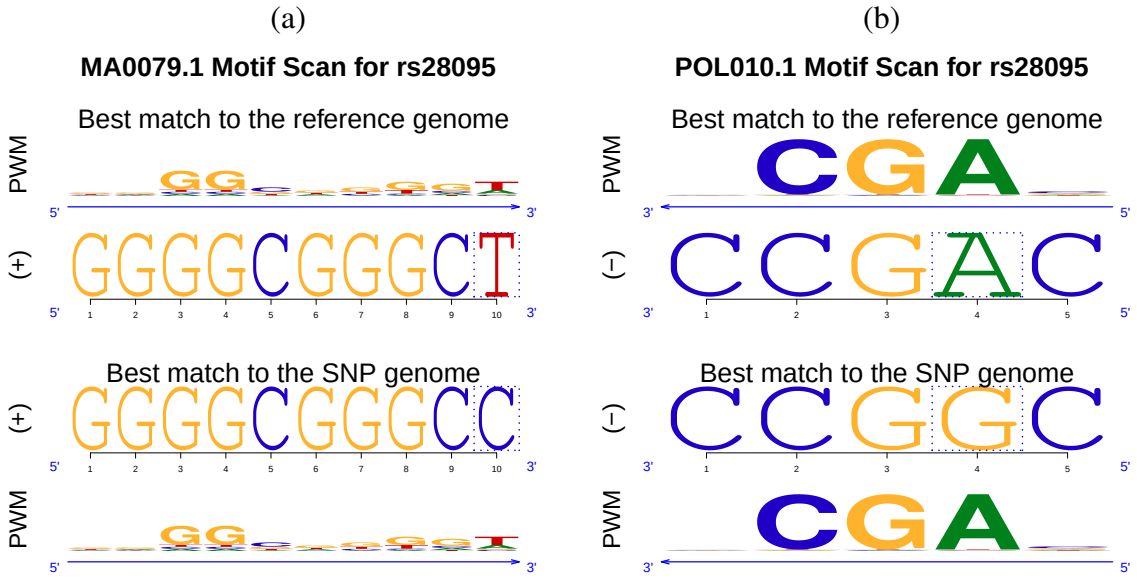


Figure A.20 Sequence logo plot for rs28095 with the motif of the top regulatory effect in (a) the reported TF family; and (b) the whole JASPAR library.

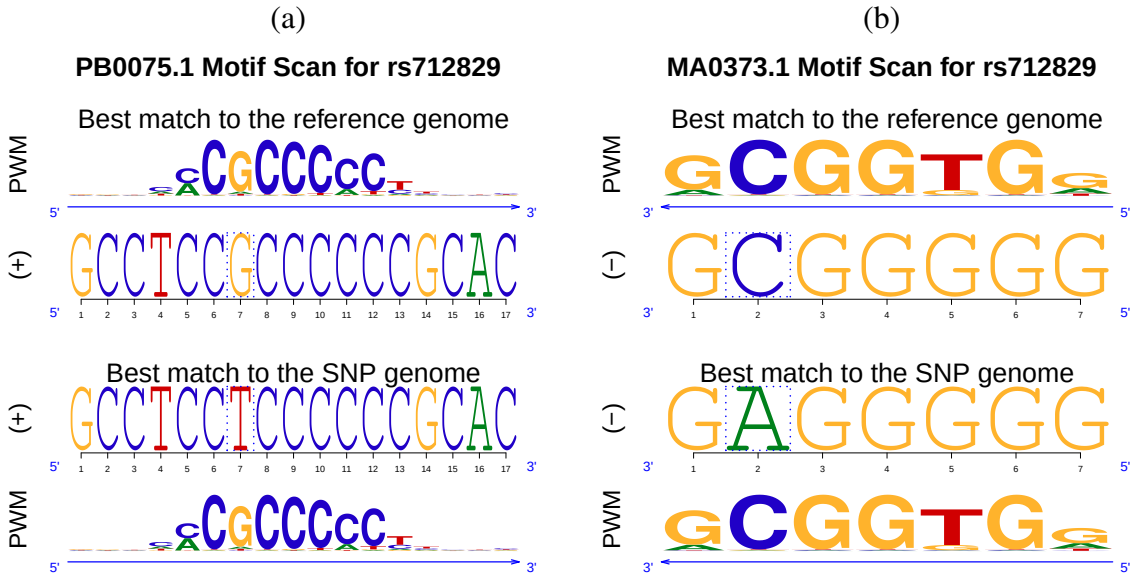


Figure A.21 Sequence logo plot for rs712829 with the motif of the top regulatory effect in (a) the reported TF family; and (b) the whole JASPAR library.

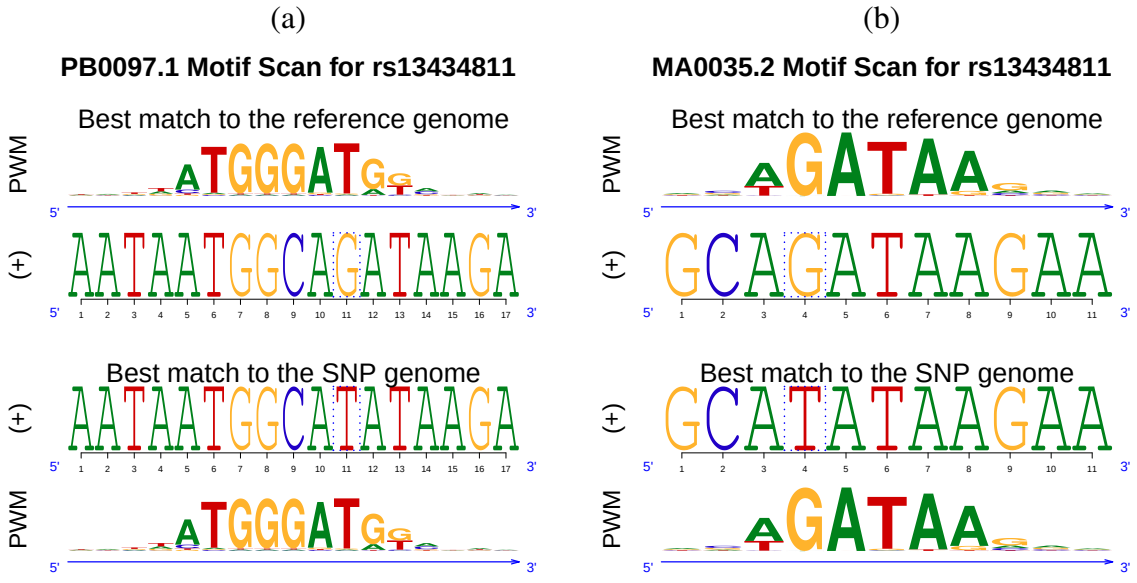


Figure A.22 Sequence logo plot for rs13434811 with the motif of the top regulatory effect in (a) the reported TF family; and (b) the whole JASPAR library.

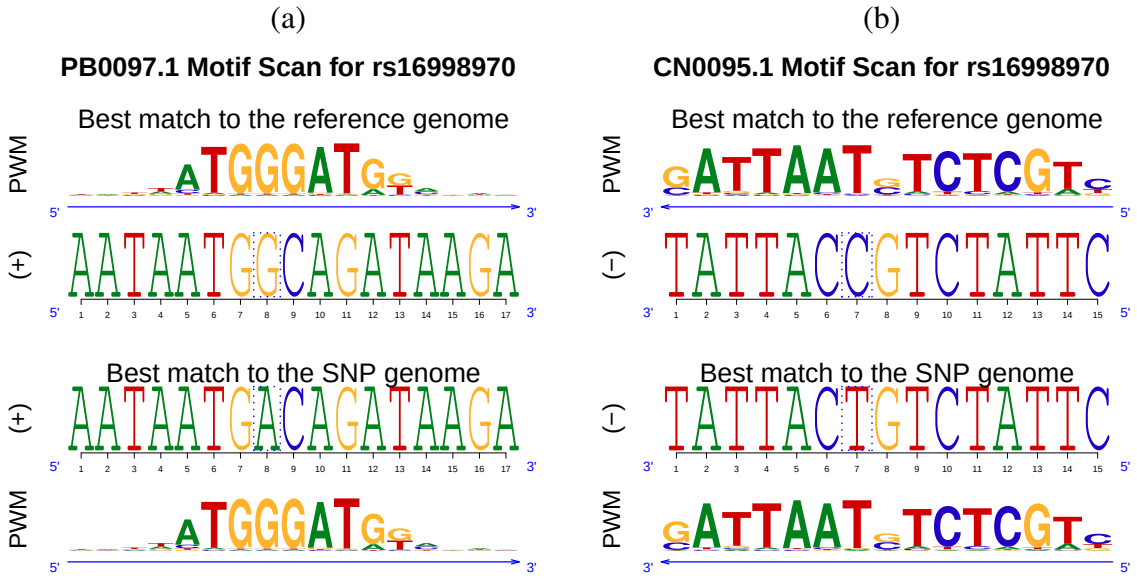


Figure A.23 Sequence logo plot for rs16998970 with the motif of the top regulatory effect in (a) the reported TF family; and (b) the whole JASPAR library.

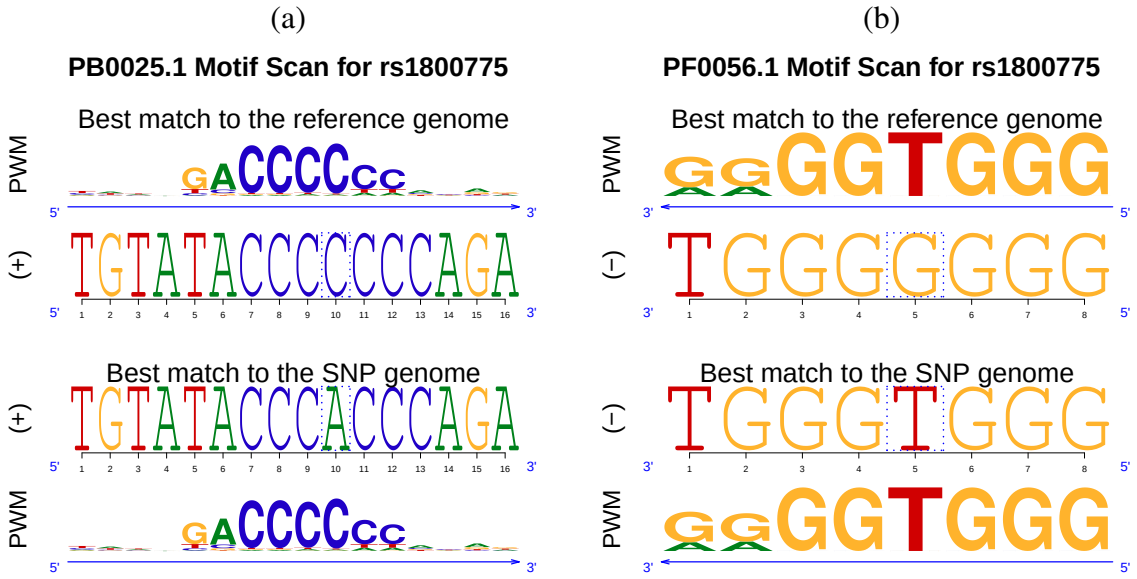


Figure A.24 Sequence logo plot for rs1800775 with the motif of the top regulatory effect in (a) the reported TF family; and (b) the whole JASPAR library.

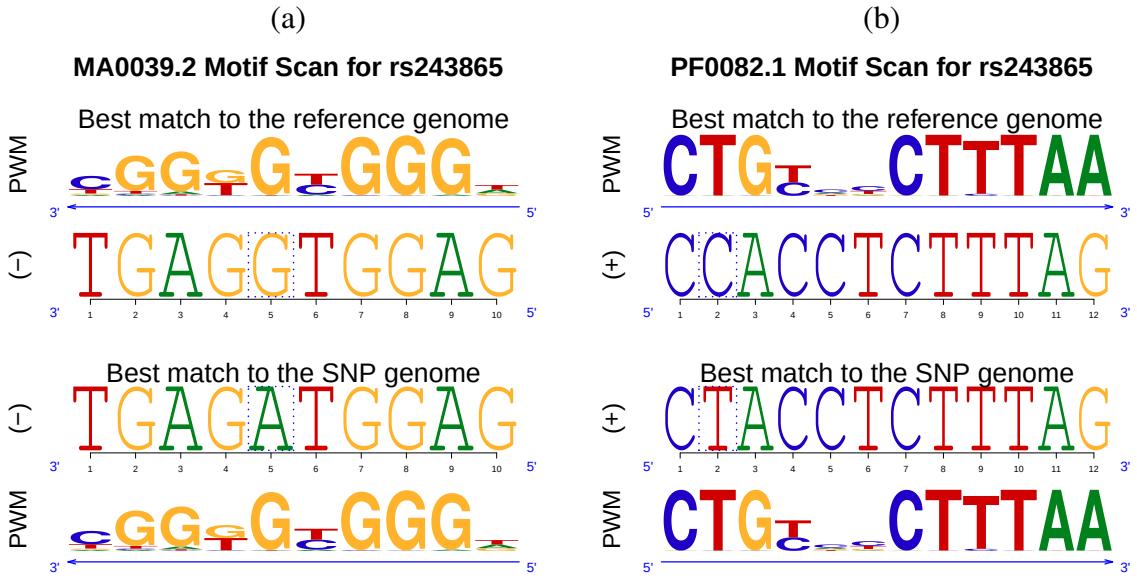


Figure A.25 Sequence logo plot for rs243865 with the motif of the top regulatory effect in (a) the reported TF family; and (b) the whole JASPAR library.

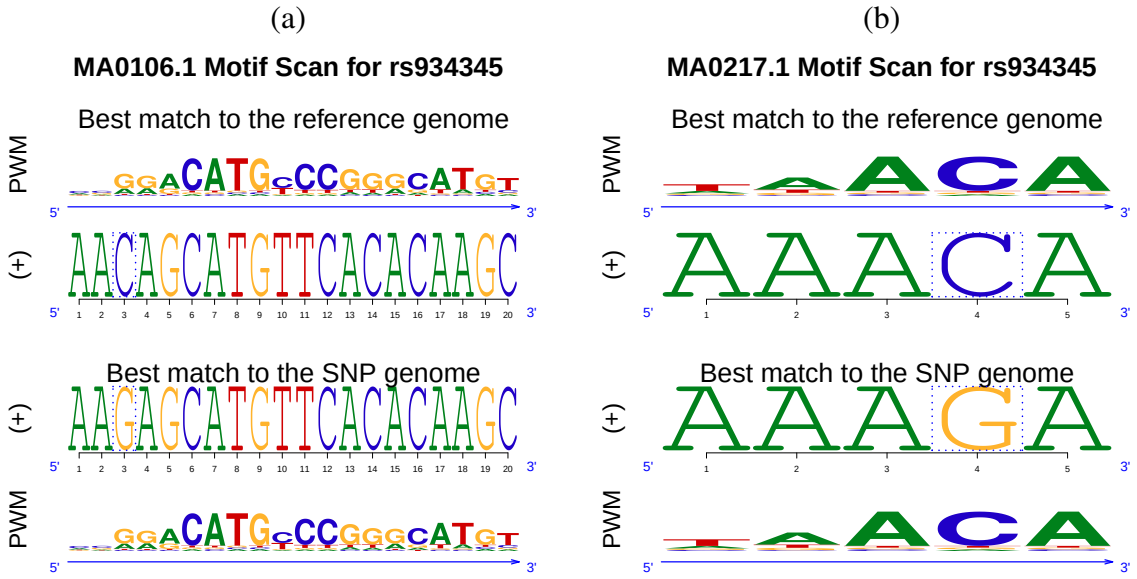


Figure A.26 Sequence logo plot for rs934345 with the motif of the top regulatory effect in (a) the reported TF family; and (b) the whole JASPAR library.

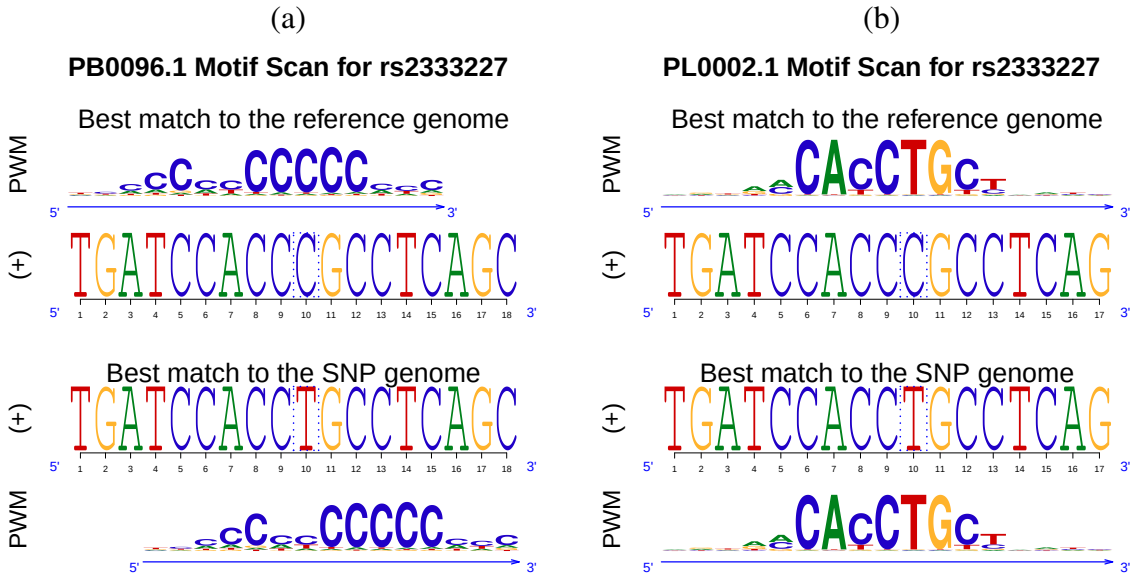


Figure A.27 Sequence logo plot for rs2333227 with the motif of the top regulatory effect in (a) the reported TF family; and (b) the whole JASPAR library.

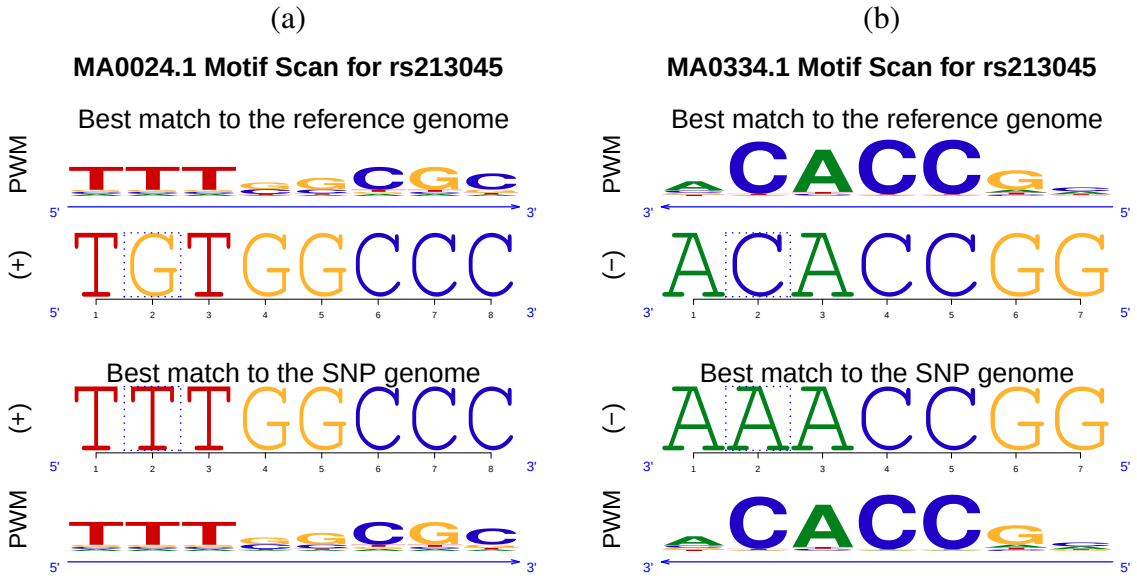


Figure A.28 Sequence logo plot for rs213045 with the motif of the top regulatory effect in (a) the reported TF family; and (b) the whole JASPAR library.

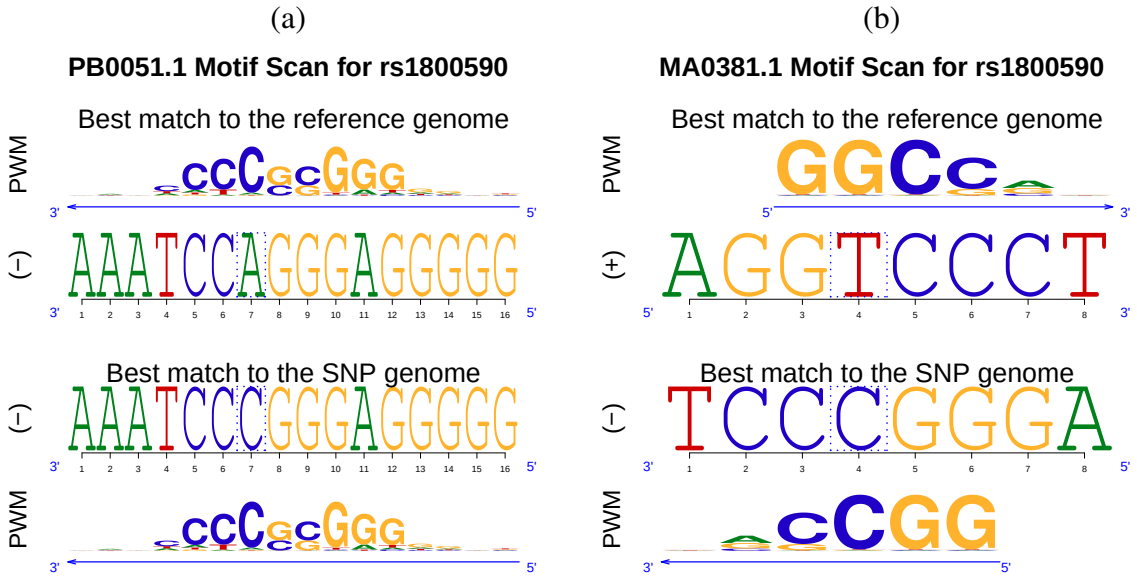


Figure A.29 Sequence logo plot for rs1800590 with the motif of the top regulatory effect in (a) the reported TF family; and (b) the whole JASPAR library.

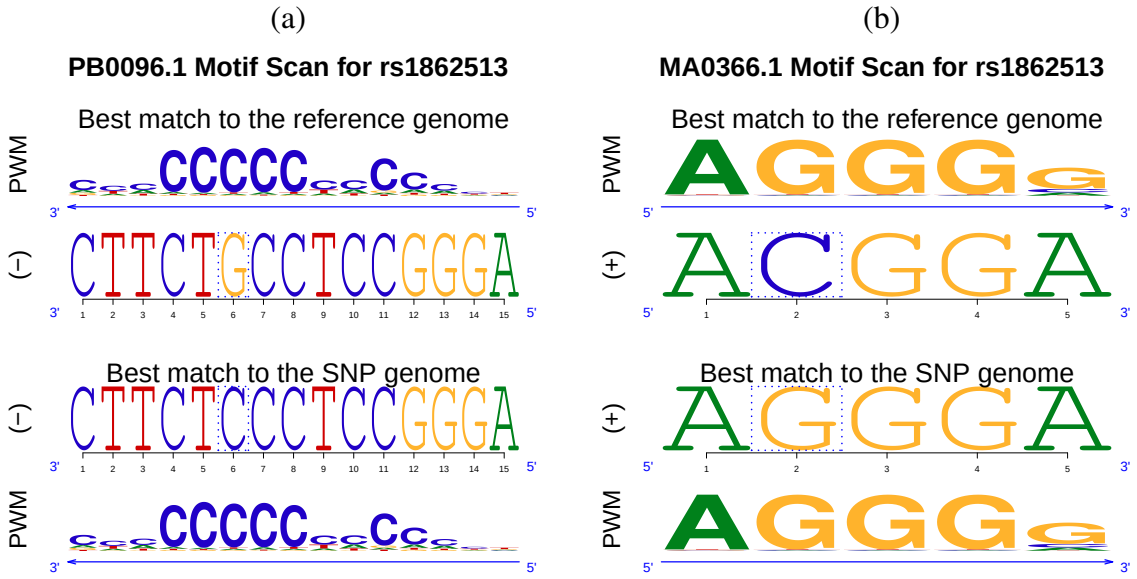


Figure A.30 Sequence logo plot for rs1862513 with the motif of the top regulatory effect in (a) the reported TF family; and (b) the whole JASPAR library.

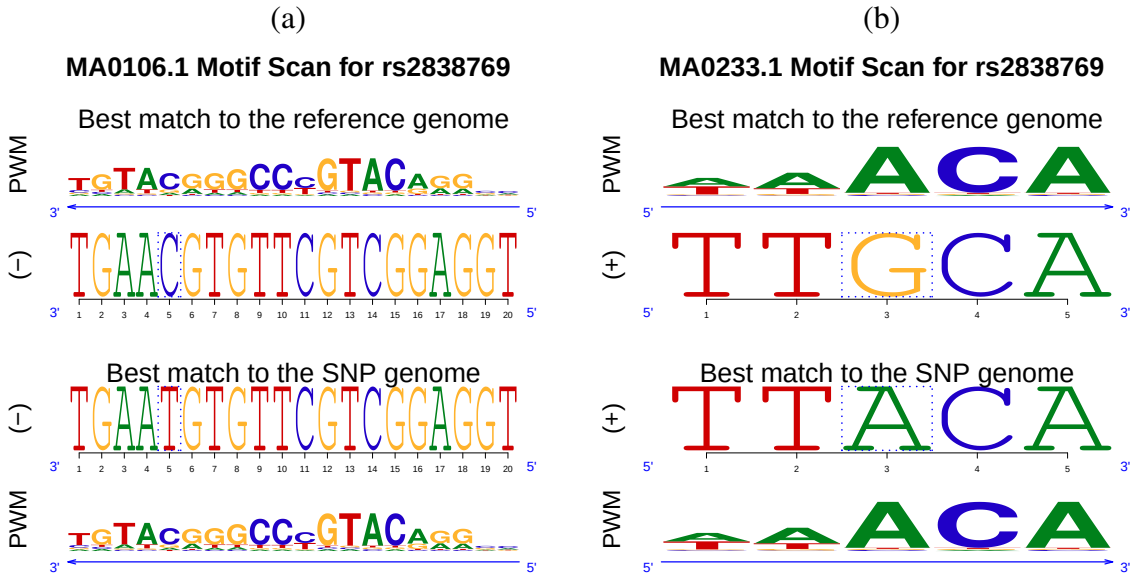


Figure A.31 Sequence logo plot for rs2838769 with the motif of the top regulatory effect in (a) the reported TF family; and (b) the whole JASPAR library.

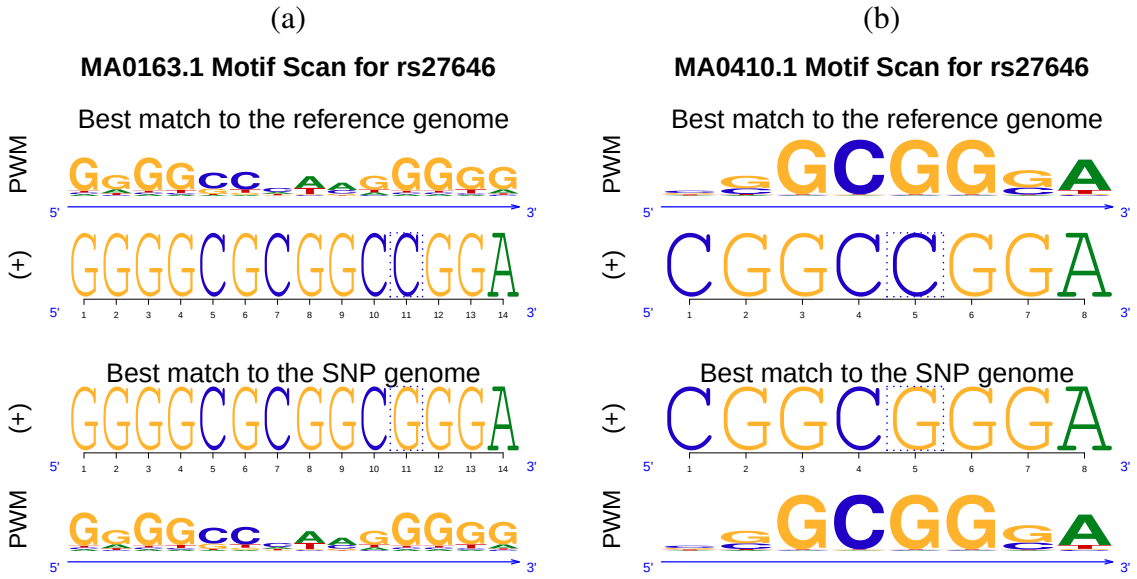


Figure A.32 Sequence logo plot for rs27646 with the motif of the top regulatory effect in (a) the reported TF family; and (b) the whole JASPAR library.

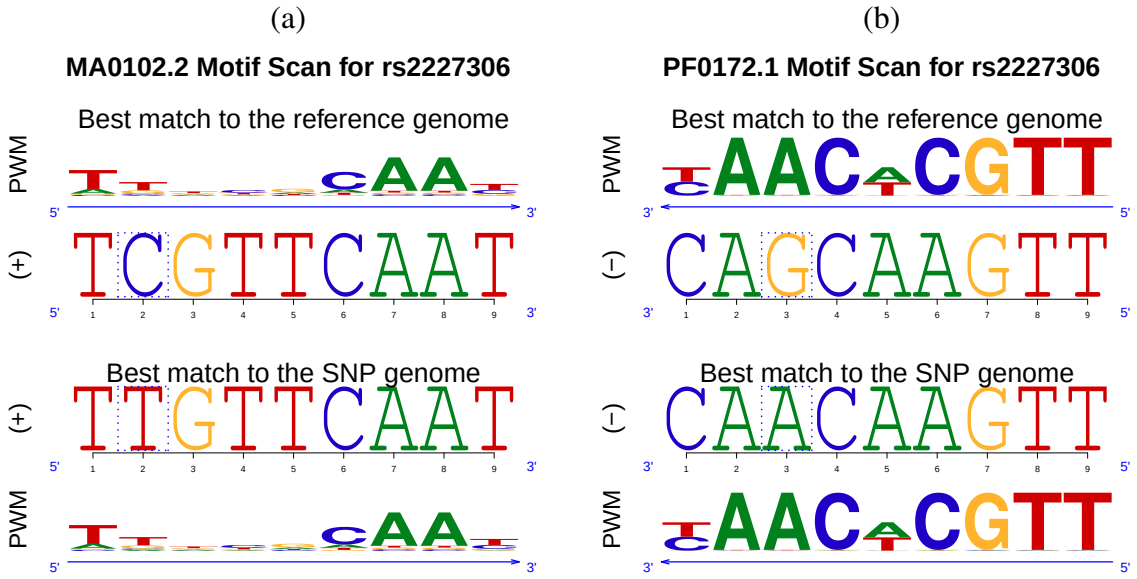


Figure A.33 Sequence logo plot for rs2227306 with the motif of the top regulatory effect in (a) the reported TF family; and (b) the whole JASPAR library.

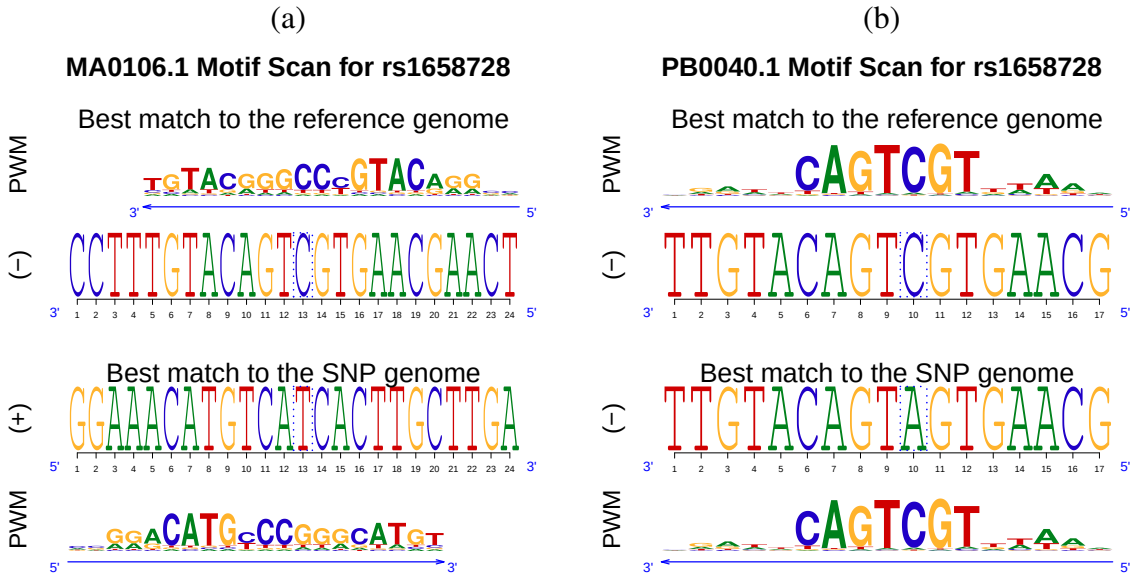


Figure A.34 Sequence logo plot for rs1658728 with the motif of the top regulatory effect in (a) the reported TF family; and (b) the whole JASPAR library.

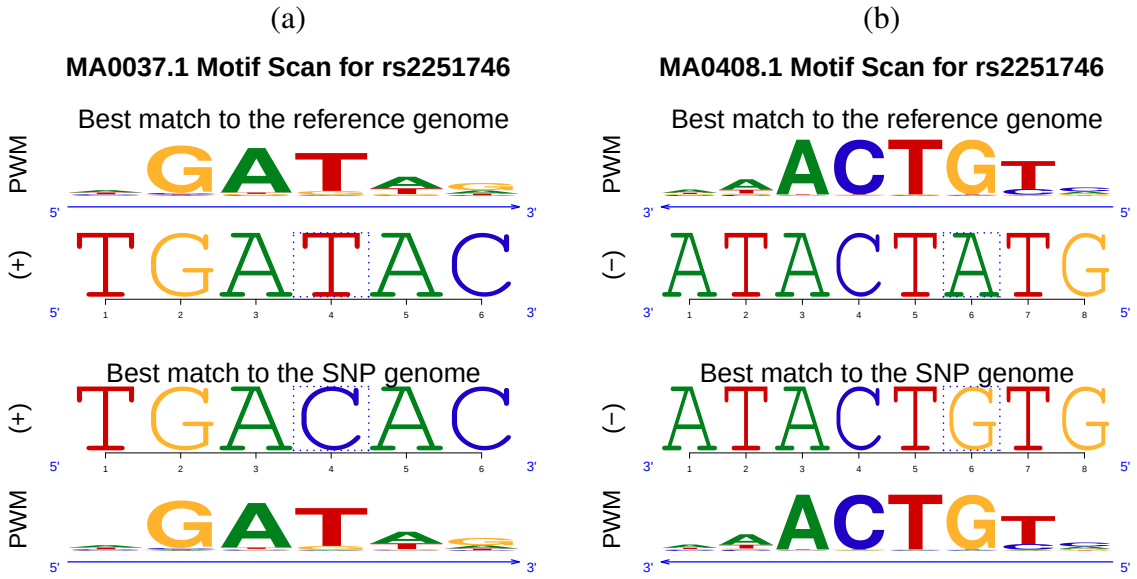


Figure A.35 Sequence logo plot for rs2251746 with the motif of the top regulatory effect in (a) the reported TF family; and (b) the whole JASPAR library.

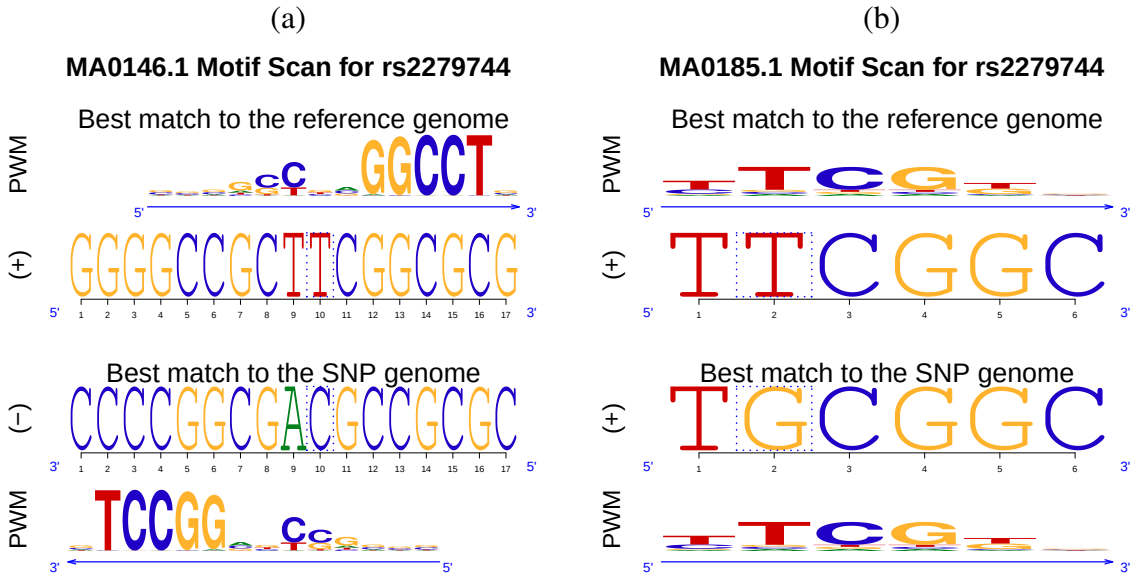


Figure A.36 Sequence logo plot for rs2279744 with the motif of the top regulatory effect in (a) the reported TF family; and (b) the whole JASPAR library.

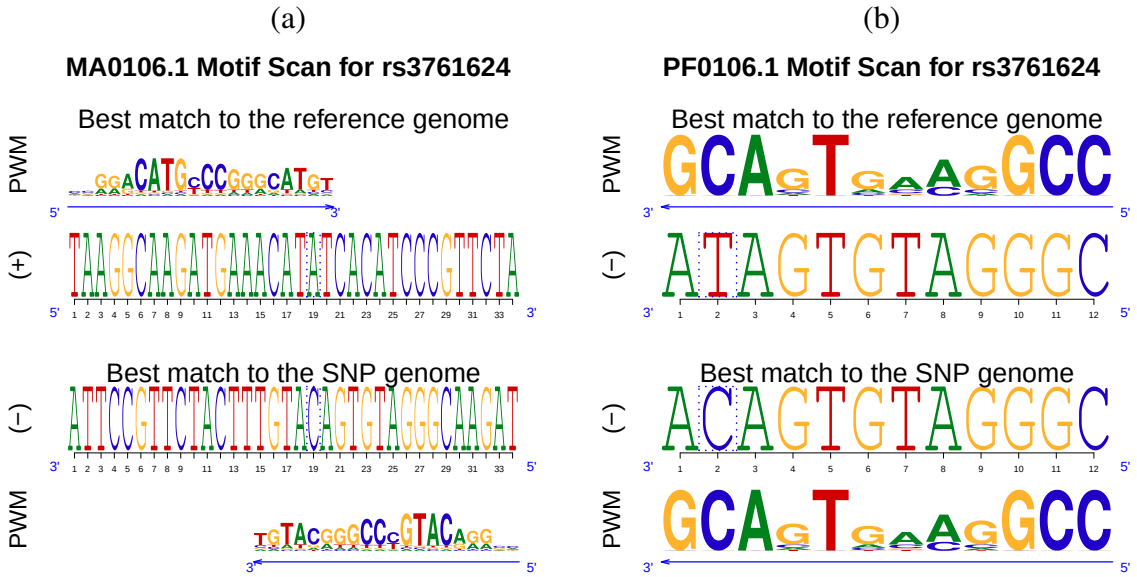


Figure A.37 Sequence logo plot for rs3761624 with the motif of the top regulatory effect in (a) the reported TF family; and (b) the whole JASPAR library.

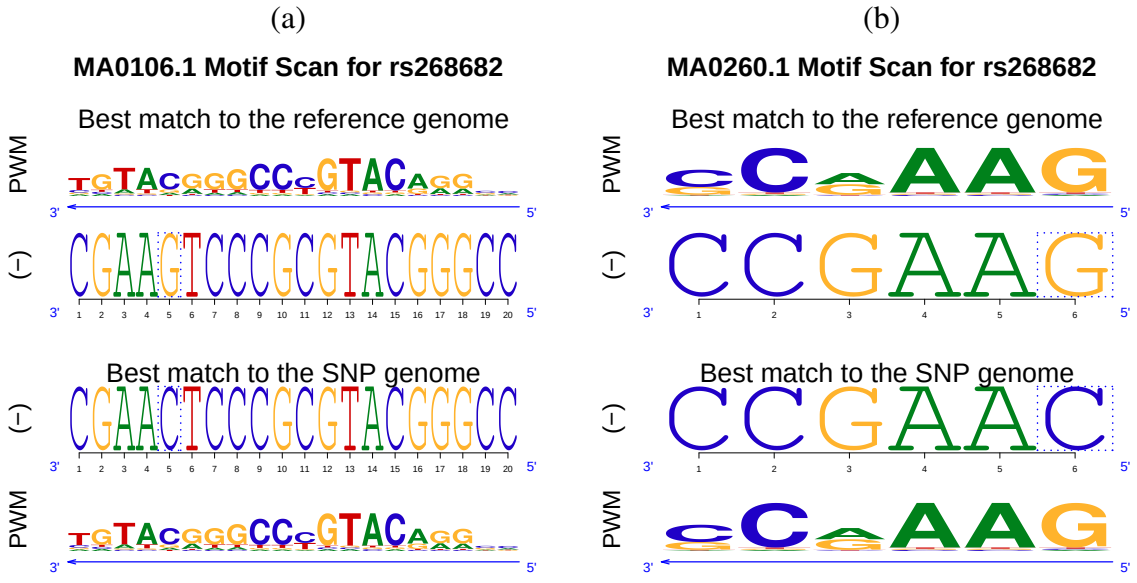


Figure A.38 Sequence logo plot for rs268682 with the motif of the top regulatory effect in (a) the reported TF family; and (b) the whole JASPAR library.

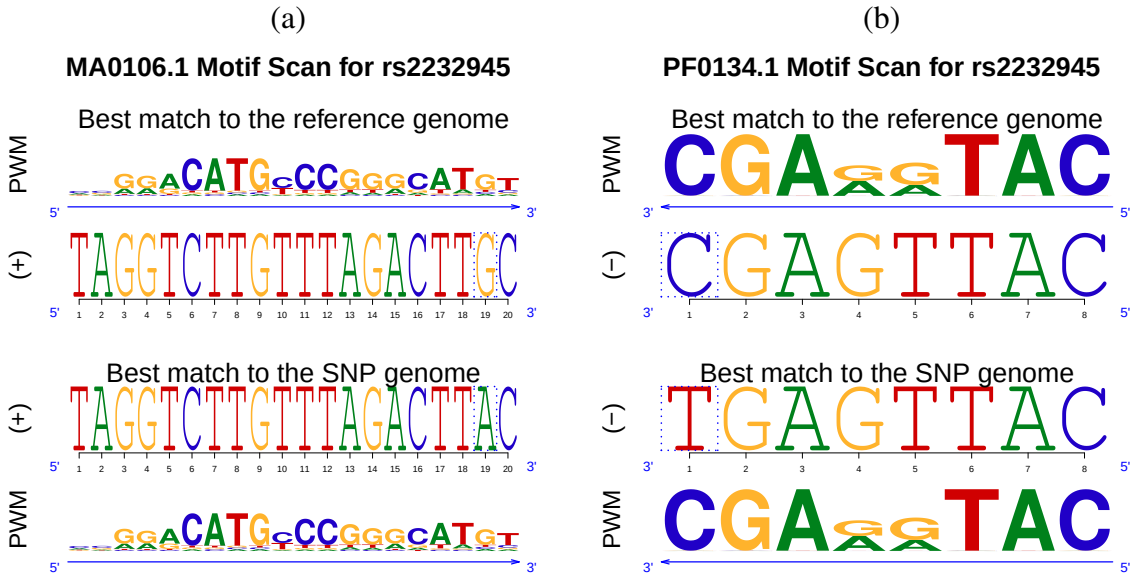


Figure A.39 Sequence logo plot for rs2232945 with the motif of the top regulatory effect in (a) the reported TF family; and (b) the whole JASPAR library.

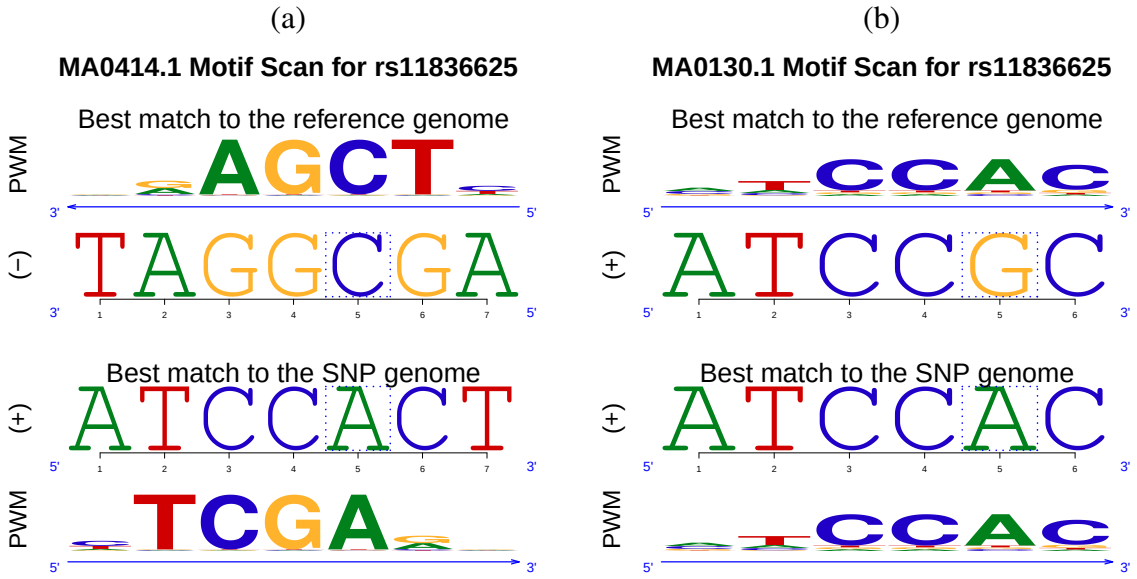


Figure A.40 Sequence logo plot for rs11836625 with the motif of the top regulatory effect in (a) the reported TF family; and (b) the whole JASPAR library.

Appendix B: Supplementary Table for Chapter 4

Table B.1: Enriched cell type-TF combination for each cluster in the TF enrichment network analysis of Section 4.5.1. TFs with estimated enrichment probability $> 95\%$ are listed for each cluster.

Cluster	# of Loci	Common TF	Gm12878 Specific	K562 Specific
1	34	Bcl3, Max, Sp1, Taf1, Zbtb33	Atf3, Ets1, Jund, Nrf1, Pol24h8, Sin3ak20, Tr4	Egr1, Pu1, Rad21, Usf2
2	317	Bcl3, Chd2, Max, Pol24h8, Sp1, Taf1	Atf3, Ets1, Nrf1, Sin3ak20, Usf2	Bclaf1, Egr1, Pu1, Smc3, Tbp, Zbtb33
3	490	Ets1, Pol2, Pol24h8, Sin3ak20, Taf1, Tbp, Usf1	Atf3, Chd2, Nrf1, Usf2	Bcl3, Bclaf1, Max, Sp1
4	555	Bcl3, Bclaf1, Chd2, Pol2, Pol24h8, Sp1, Taf1, Tbp	Atf3, Ets1, Nrf1, Sin3ak20, Six5, Smc3	Pu1
5	428	Bcl3, Chd2, Ets1, Pol24h8, Sp1, Taf1	Atf3, Ctcf, Nrf1, Rad21, Sin3ak20	Bclaf1, Pol2, Smc3, Tbp
6	729	Bcl3, Chd2, Ets1, Pol2, Pol24h8, Sin3ak20, Smc3, Sp1, Taf1	Atf3, Nrf1	Bclaf1, Egr1, Tbp

To continue on the next page.

Cluster	# of Loci	Common TF	Gm12878 Specific	K562 Specific
7	391			Bcl3, Bclaf1, Chd2, Egr1, Ets1, Smc3, Sp1
8	133	Ctcf	Atf3, Chd2, Rad21	Smc3, Srf
9	146	Ctcf		Bclaf1, Pol24h8, Smc3, Tbp
10	469	Pol2, Pol24h8, Taf1	Smc3	Bclaf1, Ets1, Sin3ak20, Tbp
11	440	Gabp, Pol24h8, Taf1	Ets1, Smc3	Pol2
12	184		Bcl3, Chd2, Ets1, Nrf1, Pol24h8, Taf1	
13	277	Chd2, Pol2, Pol24h8, Sp1, Taf1, Tbp	Smc3	Cfos
14	156	Chd2, Pol2, Pol24h8, Tbp, Zbtb33	Ets1, Smc3, Taf1	
15	412	Pol2, Pol24h8		
16	327		Pol24h8	
17	213	Usf1	Atf3, Usf2	Max
18	241	Six5	Ets1, Smc3	
19	187	Chd2, Sp1		Cfos
20	222			Ets1

To continue on the next page.

Cluster	# of Loci	Common TF	Gm12878 Specific	K562 Specific
21	385			Pol24h8
22	343	Ctcf		Smc3
23	449		Nrf1, Smc3	
24	1674			

Table B.2: Annotations for +9.5 Element-like loci in 5p1 (2Kb upstream of transcription start site (TSS)), 5p2 (2Kb to 10Kb upstream of TSS) and intronic regions.

Ref ID	Gene	Chr	Strand	Gene Start	Gene End	Region	Distance	Peak Start	Peak End	+9.5 Similarity
NM_001145662	GATA2	chr3	-	128198264	128206764	intron	4601	128202079	128202248	0.964
NM_001145661	GATA2	chr3	-	128198264	128207373	intron	5210	128202079	128202248	0.964
NM_032638	GATA2	chr3	-	128198264	128212030	intron	9867	128202079	128202248	0.964
NM_005225	E2F1	chr20	-	32263292	32274210	5p2	-3886	32278012	32278182	0.774
NM_001166	BIRC2	chr11	+	102217965	102249394	5p2	-5004	102212877	102213046	0.753
NM_203343	EPB41	chr1	+	29213602	29446558	intron	39968	29253487	29253655	0.74
NM_203342	EPB41	chr1	+	29213602	29446558	intron	39968	29253487	29253655	0.74
NM_001166007	EPB41	chr1	+	29213602	29446558	intron	39968	29253487	29253655	0.74
NM_004437	EPB41	chr1	+	29213602	29446558	intron	39968	29253487	29253655	0.74
NM_001166005	EPB41	chr1	+	29213602	29446558	intron	39968	29253487	29253655	0.74
NM_001166006	EPB41	chr1	+	29241087	29391731	intron	12483	29253487	29253655	0.74
NM_173485	TSHZ2	chr20	+	51588876	52103965	intron	203292	51792084	51792253	0.735
NM_007077	AP4S1	chr14	+	31494682	31555007	intron	13234	31507832	31508001	0.733
NM_001128126	AP4S1	chr14	+	31494682	31562634	intron	13234	31507832	31508001	0.733
NM_001430	EPAS1	chr2	+	46524540	46613842	intron	42757	46567214	46567382	0.728
NM_018119	POLR3E	chr16	+	22308740	22345341	intron	833	22309489	22309658	0.719
NM_181442	ADNP	chr20	-	49506882	49547527	intron	27423	49520020	49520189	0.718
NM_015339	ADNP	chr20	-	49506882	49547527	intron	27423	49520020	49520189	0.718
NM_020359	PLSCR2	chr3	-	146151081	146213722	5p1	-921	146214559	146214728	0.718

Continued on the next page.

Ref ID	Gene	Chr	Strand	Gene Start	Gene End	Region	Distance	Peak Start	Peak End	+9.5 Similarity
NM_006257	PRKCQ	chr10	-	6469104	6622238	intron	106706	6515449	6515617	0.713
NM_018309	TBC1D23	chr3	+	99979685	100044078	intron	28727	100008329	100008497	0.711
NM_020382	SETD8	chr12	+	123868703	123893898	intron	4170	123872789	123872958	0.71
NM_015385	SORBS1	chr10	-	97071530	97321171	intron	29191	97291896	97292065	0.709
NM_024991	SORBS1	chr10	-	97071530	97321171	intron	29191	97291896	97292065	0.709
NM_000440	PDE6A	chr5	-	149237519	149324356	intron	4584	149319688	149319857	0.699
NM_012091	ADAT1	chr16	-	75632997	75657154	intron	2033	75655038	75655206	0.693
NM_005033	EXOSC9	chr4	+	122722471	122738175	5p2	-6644	122715743	122715912	0.691
NM_001034194	EXOSC9	chr4	+	122722471	122738175	5p2	-6644	122715743	122715912	0.691
NM_004099	STOM	chr9	-	124101353	124132545	intron	388	124132073	124132243	0.689
NM_198194	STOM	chr9	-	124101356	124132545	intron	388	124132073	124132243	0.689
NM_014395	DAPP1	chr4	+	100737980	100791344	intron	25687	100763583	100763752	0.682
NM_181078	IL21R	chr16	+	27413722	27462115	intron	28911	27442549	27442718	0.681
NM_181079	IL21R	chr16	+	27414422	27462115	intron	28211	27442549	27442718	0.681
NM_021798	IL21R	chr16	+	27438578	27462115	intron	4055	27442549	27442718	0.681
NM_002492	NDUFB5	chr3	+	179322574	179342287	intron	10827	179333318	179333486	0.68
NM_021831	AGBL5	chr2	+	27274490	27293489	intron	11452	27285858	27286027	0.678
NM_020132	AGPAT3	chr21	+	45285115	45407474	5p2	-4281	45280751	45280919	0.677
NM_007356	LAMB4	chr7	-	107663995	107770801	intron	50003	107720714	107720883	0.677
NM_001010985	MYBPHL	chr1	-	109834986	109849663	5p1	-613	109850192	109850361	0.67
NM_006253	PRKAB1	chr12	+	120105760	120119428	5p2	-2184	120103492	120103661	0.668
NM_015226	CLEC16A	chr16	+	11038344	11276044	intron	27935	11066196	11066364	0.667
NM_020448	NIPAL3	chr1	+	24742244	24799472	intron	22892	24765053	24765221	0.663

Continued on the next page.

Ref ID	Gene	Chr	Strand	Gene Start	Gene End	Region	Distance	Peak Start	Peak End	+9.5 Similarity
NM_015560	OPA1	chr3	+	193310932	193415599	intron	67680	193378528	193378697	0.659
NM_130832	OPA1	chr3	+	193310932	193415599	intron	67680	193378528	193378697	0.659
NM_130831	OPA1	chr3	+	193310932	193415599	intron	67680	193378528	193378697	0.659
NM_130834	OPA1	chr3	+	193310932	193415599	intron	67680	193378528	193378697	0.659
NM_130837	OPA1	chr3	+	193310932	193415599	intron	67680	193378528	193378697	0.659
NM_130836	OPA1	chr3	+	193310932	193415599	intron	67680	193378528	193378697	0.659
NM_130835	OPA1	chr3	+	193310932	193415599	intron	67680	193378528	193378697	0.659
NM_130833	OPA1	chr3	+	193310932	193415599	intron	67680	193378528	193378697	0.659
NM_001004342	TRIM67	chr1	+	231298673	231357314	5p2	-2591	231295998	231296167	0.659
NM_020201	NT5M	chr17	+	17206679	17250975	intron	1325	17207920	17208089	0.658
NM_173054	RELN	chr7	-	103112232	103629963	intron	331913	103297966	103298135	0.657
NM_005045	RELN	chr7	-	103112232	103629963	intron	331913	103297966	103298135	0.657
NM_014206	C11orf10	chr11	-	61556602	61560085	5p2	-8290	61568291	61568461	0.657
NR_030342	MIR611	chr11	-	61559967	61560033	5p2	-8342	61568291	61568461	0.657
NM_173685	NSMCE2	chr8	+	126104082	126379367	intron	242235	126346233	126346402	0.655
NM_001127511	APC	chr5	+	112043217	112181935	5p2	-3243	112039890	112040060	0.653
NM_021926	ALX4	chr11	-	44282277	44331716	intron	39937	44291695	44291864	0.652
NM_016213	TRIP4	chr15	+	64680019	64747500	intron	42584	64722519	64722688	0.649
NM_007217	PDCD10	chr3	-	167401696	167452594	intron	10656	167441855	167442023	0.649
NM_145860	PDCD10	chr3	-	167401696	167452630	intron	10692	167441855	167442023	0.649
NM_145859	PDCD10	chr3	-	167401696	167452651	intron	10713	167441855	167442023	0.649
NM_203318	MYO18A	chr17	-	27400527	27507407	intron	17382	27489941	27490111	0.645
NM_078471	MYO18A	chr17	-	27400527	27507407	intron	17382	27489941	27490111	0.645

Continued on the next page.

Ref ID	Gene	Chr	Strand	Gene Start	Gene End	Region	Distance	Peak Start	Peak End	+9.5 Similarity
NM_001626	AKT2	chr19	-	40736224	40791265	intron	12426	40778755	40778924	0.643
NM_004767	GPR37L1	chr1	+	202092028	202098633	5p2	-4769	202087175	202087344	0.642
NM_015531	C2CD3	chr11	-	73745479	73882064	intron	84354	73797626	73797796	0.637
NM_002738	PRKCB	chr16	+	23847299	24231930	intron	166844	24014060	24014228	0.634
NM_212535	PRKCB	chr16	+	23847299	24231930	intron	166844	24014060	24014228	0.634
NM_004571	PKNOX1	chr21	+	44394642	44453688	intron	15907	44410465	44410634	0.632
NR_026749	SKINTL	chr1	-	48567386	48648100	intron	4923	48643093	48643262	0.629
NM_005560	LAMA5	chr20	-	60884122	60942368	intron	9836	60932449	60932617	0.626
NM_001080826	SGK223	chr8	-	8175258	8239257	intron	9672	8229501	8229670	0.622
NM_130465	TSPAN17	chr5	+	176074387	176086058	intron	1462	176075765	176075934	0.621
NM_012171	TSPAN17	chr5	+	176074387	176086058	intron	1462	176075765	176075934	0.621
NM_001006616	TSPAN17	chr5	+	176074387	176086058	intron	1462	176075765	176075934	0.621
NM_013326	C18orf8	chr18	+	21083461	21111742	5p2	-4363	21079015	21079183	0.616
NM_138371	FAM113B	chr12	+	47610051	47630441	5p2	-8921	47601047	47601215	0.616
NM_182498	ZNF428	chr19	-	44111376	44124014	intron	3381	44120549	44120718	0.615
NM_025179	PLXNA2	chr1	-	208195589	208417665	intron	207170	208210411	208210580	0.614
NM_020133	AGPAT4	chr6	-	161551056	161695107	intron	12103	161682920	161683089	0.611
NM_013427	ARHGAP6	chrX	-	11155662	11683821	intron	252809	11430929	11431097	0.609
NM_006125	ARHGAP6	chrX	-	11161516	11683821	intron	252809	11430929	11431097	0.609
NM_001669	ARSD	chrX	-	2822011	2847392	intron	6868	2840441	2840609	0.607
NM_009589	ARSD	chrX	-	2831654	2847392	intron	6868	2840441	2840609	0.607
NM_032359	C3orf26	chr3	+	99536677	99897476	intron	243457	99780050	99780220	0.605
NM_182909	FILIP1L	chr3	-	99551988	99833349	intron	53215	99780050	99780220	0.605

Continued on the next page.

Ref ID	Gene	Chr	Strand	Gene Start	Gene End	Region	Distance	Peak Start	Peak End	+9.5 Similarity
NM_001042459	FILIP1L	chr3	-	99566772	99833349	intron	53215	99780050	99780220	0.605
NM_194298	SLC16A9	chr10	-	61410521	61469649	5p2	-3726	61473291	61473460	0.603
NM_007356	LAMB4	chr7	-	107663995	107770801	intron	39404	107731314	107731482	0.594
NM_203456	PPIE	chr1	+	40204529	40229585	intron	18825	40223270	40223439	0.594
NM_152726	EFHA1	chr13	-	22066839	22178307	intron	81061	22097162	22097331	0.592
NM_001025107	ADAR	chr1	-	154554535	154600437	5p1	-1420	154601773	154601942	0.592
NM_001130966	TBXAS1	chr7	+	139478046	139720123	intron	75241	139553203	139553372	0.591
NM_001166254	TBXAS1	chr7	+	139478046	139720123	intron	75241	139553203	139553372	0.591
NM_030984	TBXAS1	chr7	+	139528951	139720123	intron	24336	139553203	139553372	0.591
NM_001166253	TBXAS1	chr7	+	139528951	139720123	intron	24336	139553203	139553372	0.591
NM_001061	TBXAS1	chr7	+	139528951	139720123	intron	24336	139553203	139553372	0.591
NR_029394	TBXAS1	chr7	+	139528951	139720123	intron	24336	139553203	139553372	0.591
NM_173542	PLBD2	chr12	+	113796370	113827458	intron	20911	113817197	113817366	0.591
NM_001159727	PLBD2	chr12	+	113796370	113827458	intron	20911	113817197	113817366	0.591
NM_138356	SHF	chr15	-	45459413	45493373	intron	31722	45461567	45461736	0.589
NM_021908	ST7	chr7	+	116593380	116863955	intron	92727	116686023	116686192	0.588
NM_018412	ST7	chr7	+	116593380	116870073	intron	92727	116686023	116686192	0.588
NM_017681	NUP62CL	chrX	-	106366657	106449670	intron	53243	106396343	106396512	0.587
NM_020845	PITPNM2	chr12	-	123468026	123594975	intron	73786	123521105	123521274	0.587
NM_001135054	SIGIRR	chr11	-	405715	414999	5p2	-7383	422299	422467	0.586
NM_021805	SIGIRR	chr11	-	405715	417397	5p2	-4985	422299	422467	0.586
NM_001135053	SIGIRR	chr11	-	405715	417397	5p2	-4985	422299	422467	0.586
NM_001012302	ANO9	chr11	-	417929	442011	intron	19629	422299	422467	0.586

Continued on the next page.

Ref ID	Gene	Chr	Strand	Gene Start	Gene End	Region	Distance	Peak Start	Peak End	+9.5 Similarity
NM_001098816	ODZ4	chr11	-	78364328	79151695	intron	773881	78377730	78377899	0.582
NM_178865	SERINC2	chr1	+	31885962	31907524	5p2	-2738	31883140	31883309	0.581
NM_004481	GALNT2	chr1	+	230202955	230417875	5p1	-672	230202200	230202368	0.579
NM_032427	MAML2	chr11	-	95711439	96076344	intron	19672	96056588	96056758	0.577
NM_021961	TEAD1	chr11	+	12695968	12966298	intron	202724	12898608	12898778	0.577
NM_016436	PHF20	chr20	+	34359922	34538288	intron	130764	34490602	34490771	0.573
NM_003128	SPTBN1	chr2	+	54683453	54898582	intron	117920	54801290	54801458	0.573
NM_178313	SPTBN1	chr2	+	54785530	54889444	intron	15843	54801290	54801458	0.573
NM_001037165	FOXK1	chr7	+	4721929	4811074	intron	30769	4752614	4752783	0.573
NM_005802	TOPORS	chr9	-	32540542	32552601	5p1	-1681	32554199	32554367	0.573
NM_182739	NDUFB6	chr9	-	32553522	32573182	intron	18900	32554199	32554367	0.573
NM_002493	NDUFB6	chr9	-	32553522	32573182	intron	18900	32554199	32554367	0.573
NM_004466	GPC5	chr13	+	92050934	93519485	intron	7890	92058740	92058909	0.569
NM_001145169	GPR113	chr2	-	26531040	26541970	intron	2083	26539803	26539972	0.563
NM_153835	GPR113	chr2	-	26531040	26569685	intron	29798	26539803	26539972	0.563
NM_001145168	GPR113	chr2	-	26532812	26541917	intron	2030	26539803	26539972	0.563
NM_000593	TAP1	chr6	-	32812986	32821748	5p2	-3584	32825248	32825417	0.549
NM_002800	PSMB9	chr6	+	32821937	32827626	intron	3395	32825248	32825417	0.549
NM_148954	PSMB9	chr6	+	32821937	32827626	intron	3395	32825248	32825417	0.549
NM_033104	STON2	chr14	-	81736910	81864927	intron	94218	81770625	81770794	0.546
NM_001001894	TTC3	chr21	+	38445570	38575406	intron	12717	38458203	38458372	0.544
NM_003316	TTC3	chr21	+	38455246	38575406	intron	3041	38458203	38458372	0.544
NM_000147	FUCA1	chr1	-	24171573	24194859	5p2	-4297	24199072	24199241	0.544

Continued on the next page.

Ref ID	Gene	Chr	Strand	Gene Start	Gene End	Region	Distance	Peak Start	Peak End	+9.5 Similarity
NM_016063	HDDC2	chr6	-	125596495	125623282	5p1	-844	125624043	125624211	0.544
NM_000404	GLB1	chr3	-	33038099	33138694	intron	89213	33049397	33049566	0.541
NM_001135602	GLB1	chr3	-	33038099	33138694	intron	89213	33049397	33049566	0.541
NM_001079811	GLB1	chr3	-	33038099	33138314	intron	88833	33049397	33049566	0.541
NM_017803	DUS2L	chr16	+	68057203	68113183	intron	17562	68074682	68074850	0.54
NM_001101417	ISPD	chr7	-	16127151	16460947	intron	160970	16299893	16300063	0.532
NM_001101426	ISPD	chr7	-	16127151	16460947	intron	160970	16299893	16300063	0.532
NM_002736	PRKAR2B	chr7	+	106685177	106802255	intron	31362	106716455	106716624	0.532
NR_024448	LOC91316	chr22	-	23980676	24059610	intron	32642	24026884	24027054	0.529
NM_153615	RGL4	chr22	+	24033047	24041358	5p2	-6079	24026884	24027054	0.529
NM_033631	LUZP1	chr1	-	23410515	23495351	intron	53595	23441673	23441841	0.526
NM_001142546	LUZP1	chr1	-	23410515	23495351	intron	53595	23441673	23441841	0.526
NM_001134492	HS2ST1	chr1	+	87380334	87564124	intron	77077	87457327	87457496	0.52
NM_012262	HS2ST1	chr1	+	87380334	87575680	intron	77077	87457327	87457496	0.52
NM_001085481	MAP1LC3B2	chr12	+	116997185	117014425	intron	2387	116999488	116999657	0.499
NM_079834	SCAMP4	chr19	+	1905372	1926011	intron	2224	1907513	1907681	0.487
NM_138422	ADAT3	chr19	+	1905416	1913443	intron	2180	1907513	1907681	0.487
NM_032932	RAB11FIP4	chr17	+	29718641	29865232	intron	33750	29752307	29752476	0.48
NM_033129	SCRT2	chr20	-	642240	656823	5p2	-8983	665723	665891	0.47
NM_012079	DGAT1	chr8	-	145538246	145550567	intron	1560	145548924	145549092	0.443

Appendix C: Supplementary Figures for Chapter 5

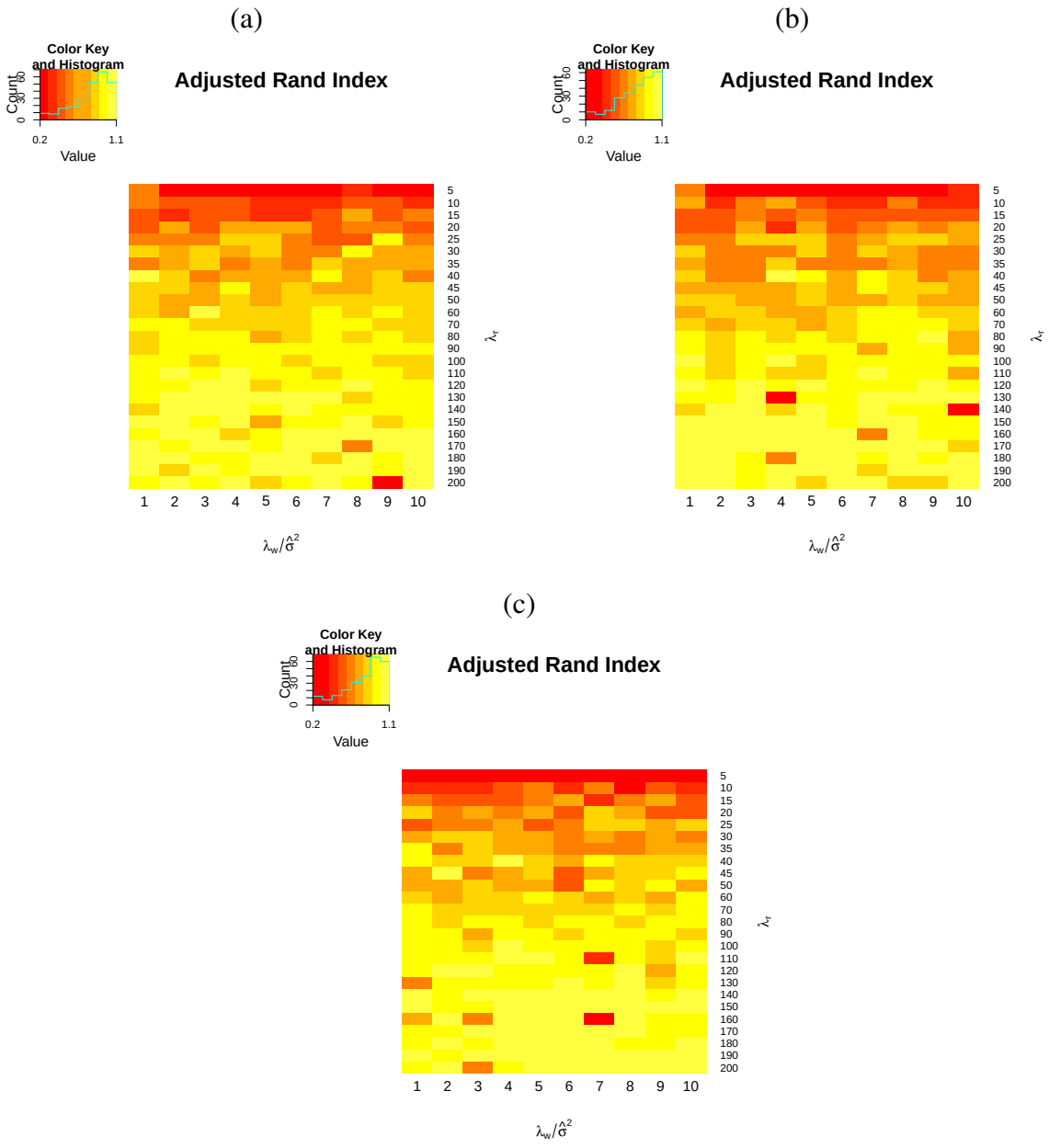


Figure C.1 Performance of ARI for different choices of λ_w and λ_r based on the (a) K-means (b) K-means++ and (c) Adaptive K-means++ initialization. Data simulated with $\zeta = 0$ and $S = 2$.

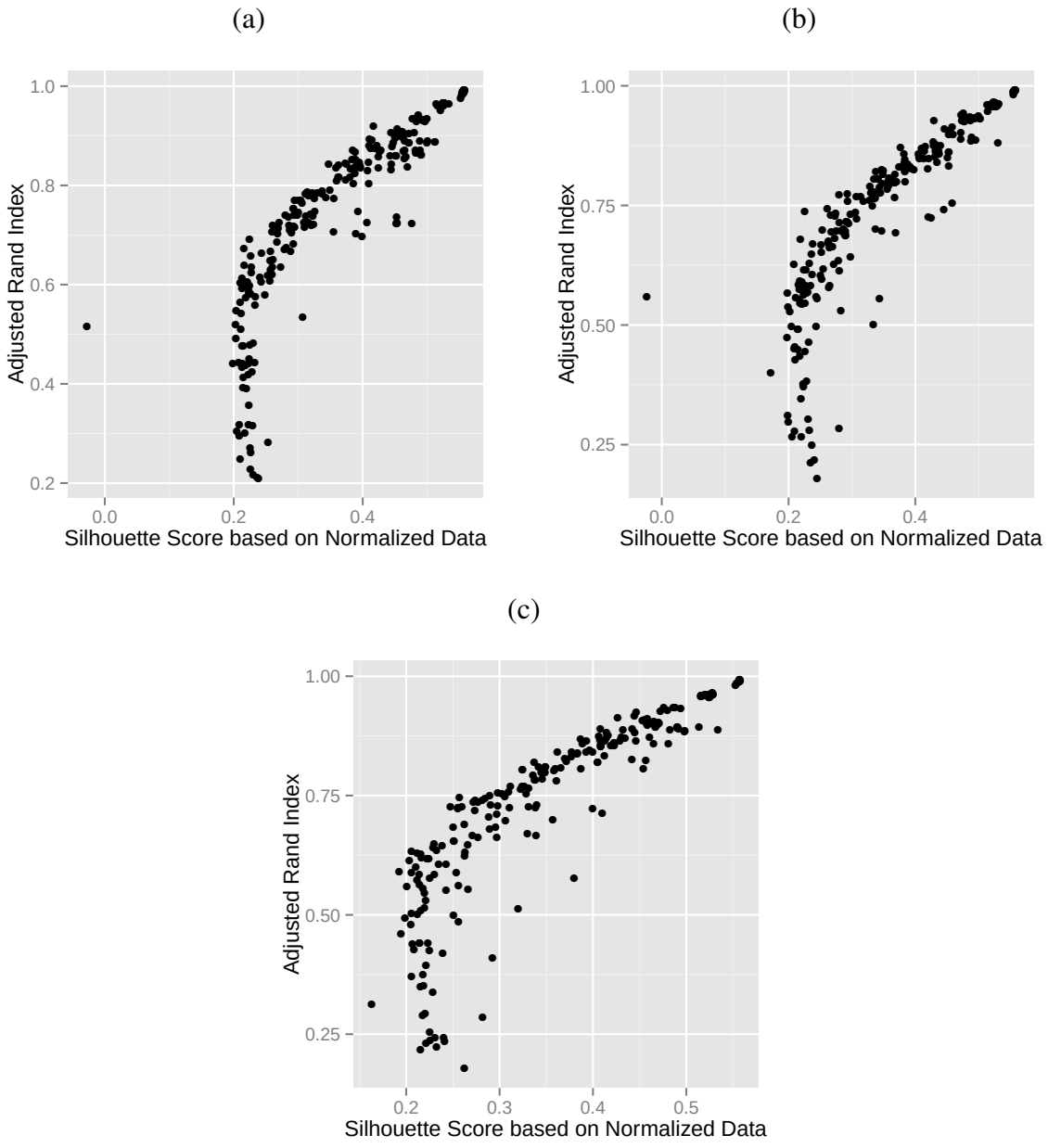


Figure C.2 Relationship between ARI and the Silhouette score based on the normalized data for different choices of λ_w and λ_r based on the (a) K-means (b) K-means++ and (c) Adaptive K-means++ initialization. Data simulated with $\zeta = 0$ and $S = 2$.

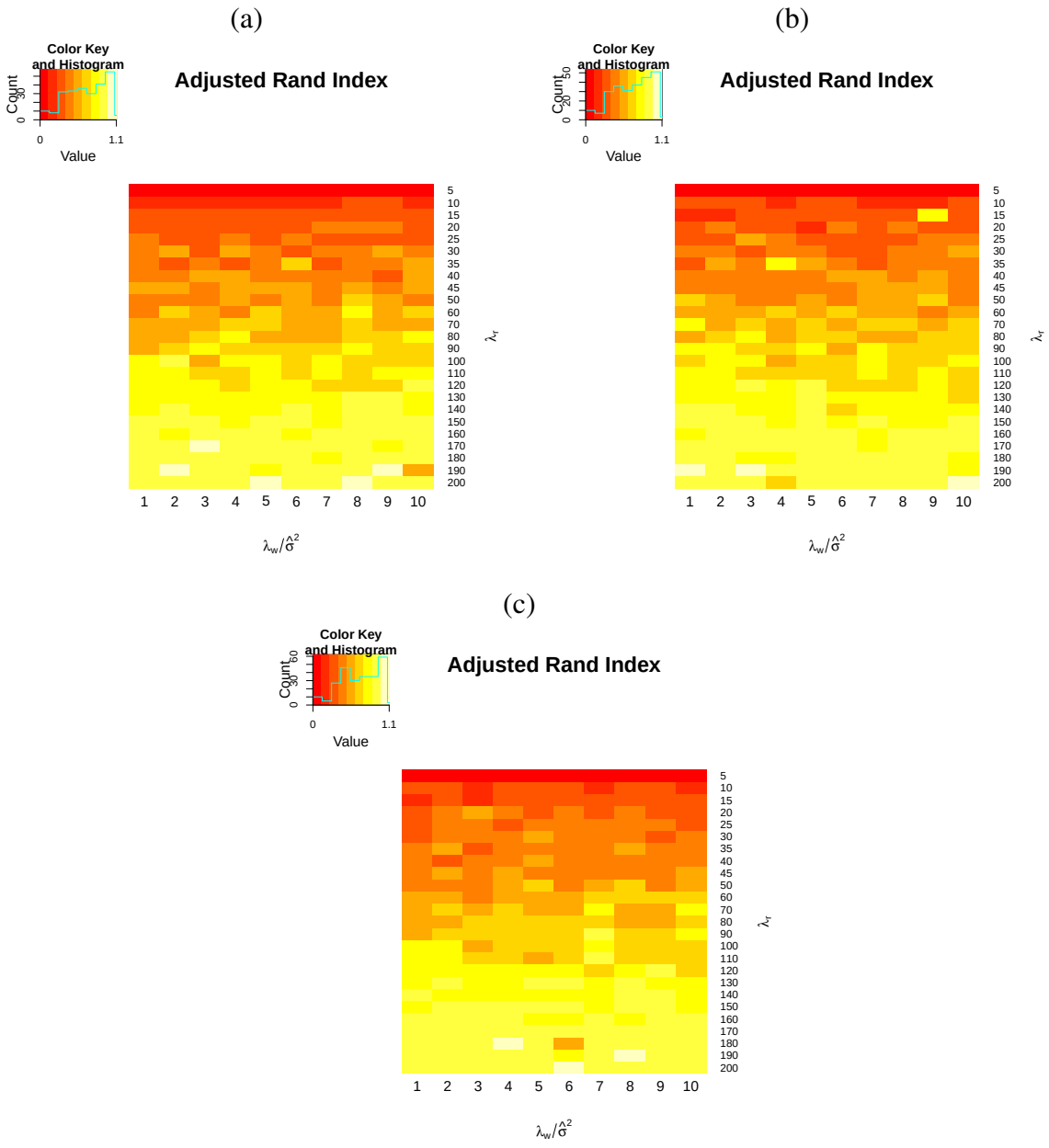


Figure C.3 Performance of ARI for different choices of λ_w and λ_r based on the (a) K-means (b) K-means++ and (c) Adaptive K-means++ initialization. Data simulated with $\zeta = 0$ and $S = 4$.

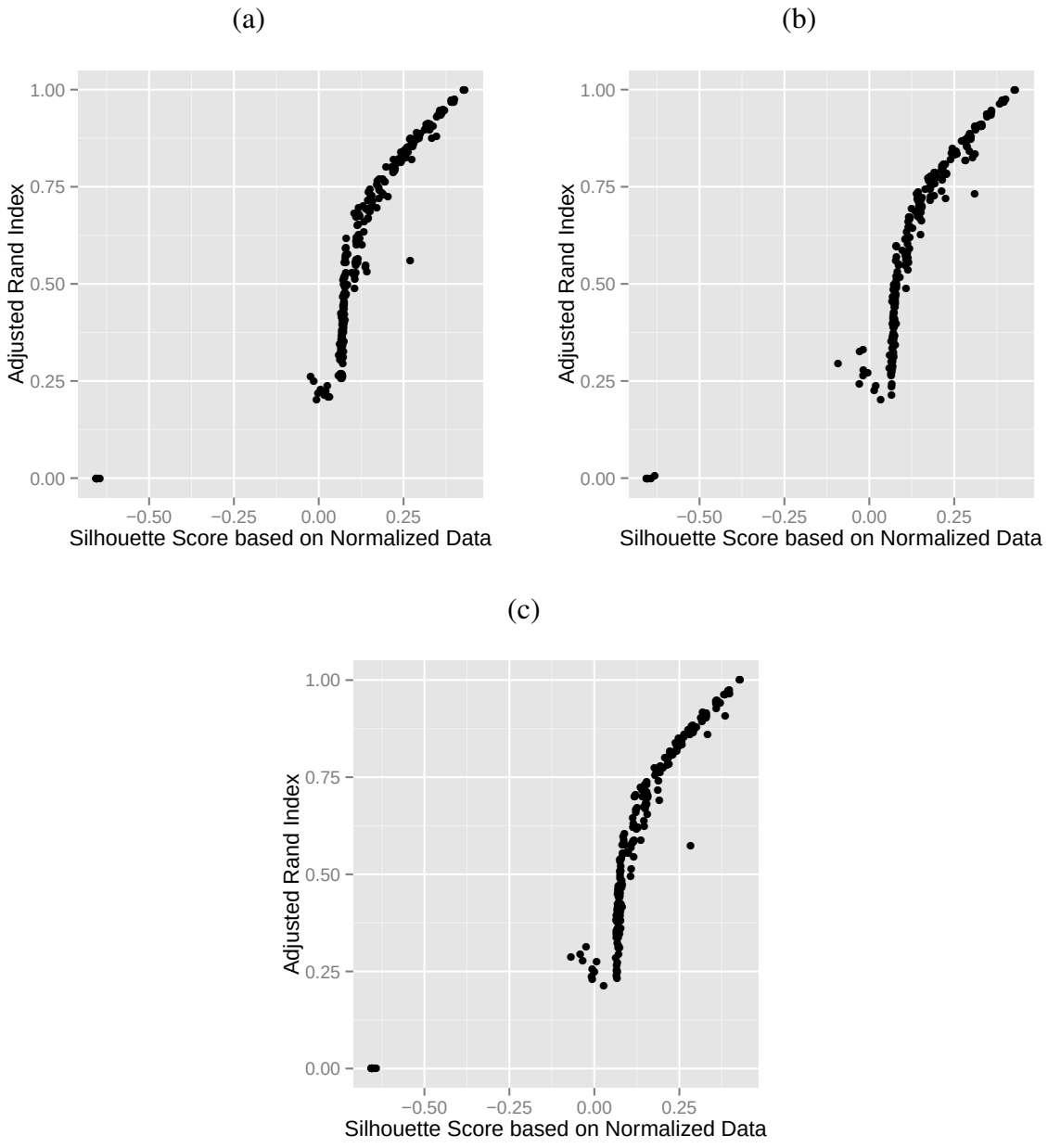


Figure C.4 Relationship between ARI and the Silhouette score based on the normalized data for different choices of λ_w and λ_r based on the (a) K-means (b) K-means++ and (c) Adaptive K-means++ initialization. Data simulated with $\zeta = 0$ and $S = 4$.

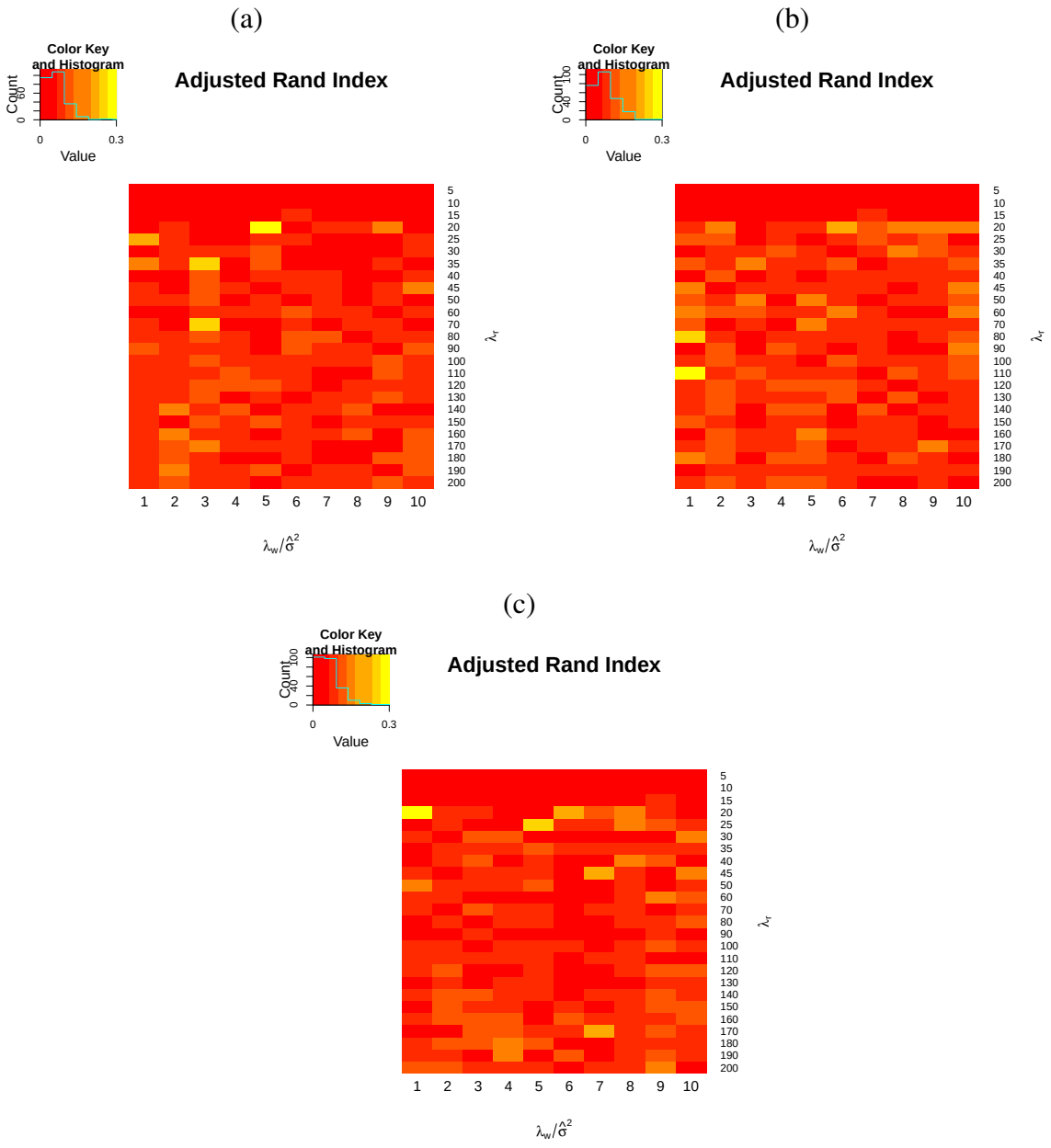


Figure C.5 Performance of ARI for different choices of λ_w and λ_r based on the (a) K-means (b) K-means++ and (c) Adaptive K-means++ initialization. Data simulated with $\zeta = 0.4$ and $S = 4$.

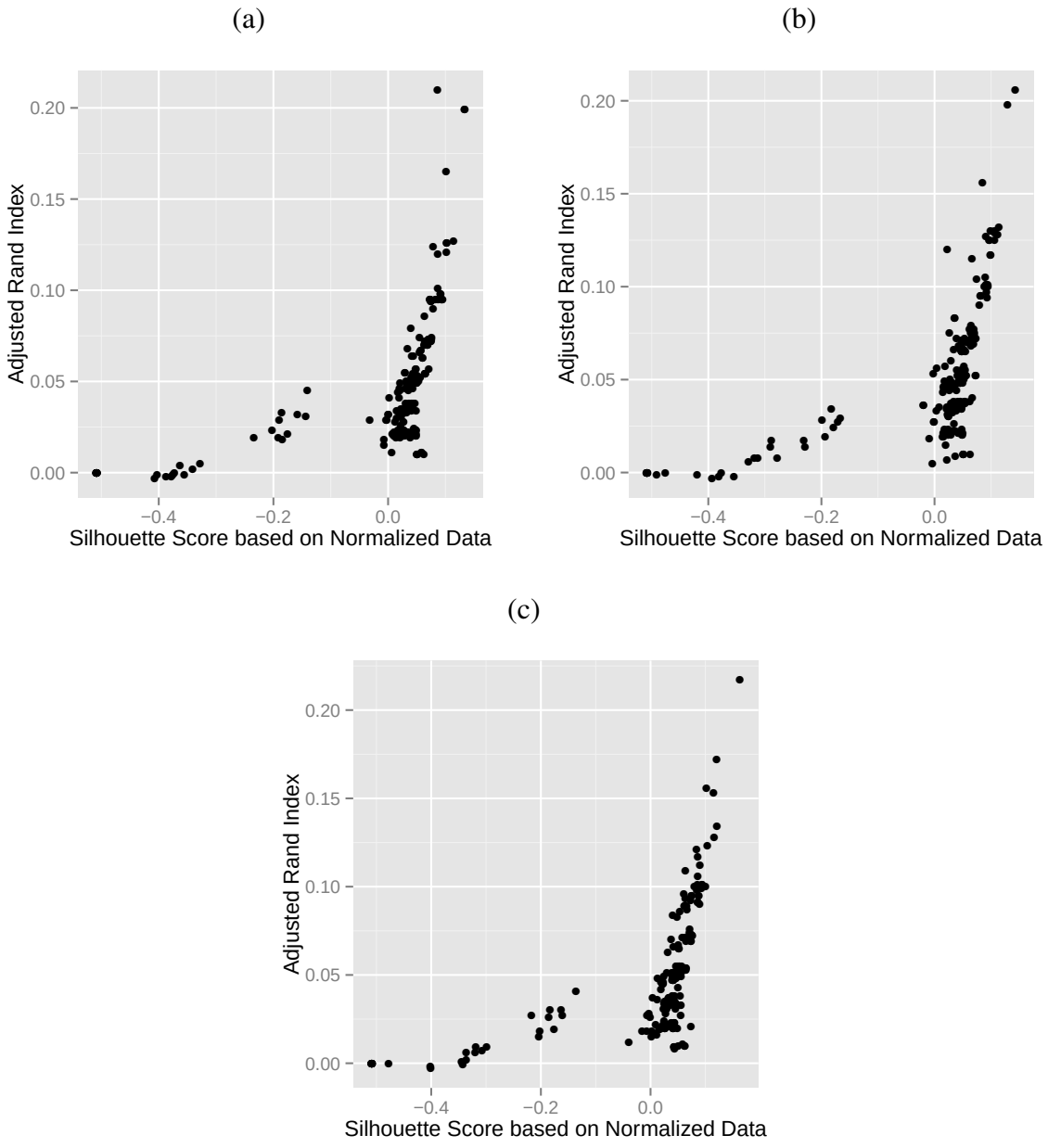


Figure C.6 Relationship between ARI and the Silhouette score based on the normalized data for different choices of λ_w and λ_r based on the (a) K-means (b) K-means++ and (c) Adaptive K-means++ initialization. Data simulated with $\zeta = 0.4$ and $S = 4$.