UNIVERSITY OF ADELAIDE

Measuring genome wide changes in chromatin state using ChIP-seq

Author:

Catisha Leigh COBURN

Supervisor:

Prof. David ADELSON and Assoc. Prof. Gary GLONEK

A thesis submitted for the degree of Master of Philosophy

in the

School of Biological Sciences

Faculty of Sciences

April 11, 2019

THE UNIVERSITY *of* ADELAIDE

Declaration of Authorship

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree. I give permission for the digital version of my thesis to be made available on the web, via the Universitys digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time. I acknowledge the support I have received for my research through the provision of an Australian Government Research Training Program Scholarship.

Signed:

_____

Date:

_____

Acknowledgements

I would like to thank my supervisors, Professor David Adelson and Associate Professor Gary Glonek, for their help and guidance throughout my Masters.

I would also like to thank my family; my mum, Robyn, for always supporting me and my dad, Malcolm, who encouraged me to pursue a career in science.

I'm also grateful for my cat, Loki, who provided many therapeutic cuddles.

Finally, I would like to thank Kevin, my partner, for his endless patience and support. I could not have done it without you.

Contents

Declaration of Authorship **iii**

Acknowledgements **v**

Abstract **1**

1 Introduction **3**

1.1 Introduction to the problem 3

1.1.1 Epigenetics and chromatin 3

1.1.2 Observing epigenetic changes 6

1.1.3 Limitations and difficulties of analysis 10

1.1.4 Peak finding software 11

1.1.5 Analysis of differential regions 11

1.2 Further Analysis 14

1.3 Latent Class Analysis 15

1.4 Research Aims 19

2 Analysis of multiple ChIP-seq programs **21**

2.1 Introduction 21

2.2 Programs and Dataset 23

2.2.1 Dataset 23

2.2.2 Calling Programs 23

2.3 Latent Class Analysis of ChIP-seq Peak Calling Programs 27
2.3.1 The Simple LCA Model 27
2.3.2 LCA with a random effect 31
Two Class LCRE with constant loading 32
One Class LCRE with non-constant loading 35
Two class LCRE with non-constant loading 39
2.3.3 LCA with a random effect: without enRich 45
2.4 Conclusions 49

3 Simulation Study **53**
3.1 Introduction 53
3.2 Methods 54
3.2.1 Simulating ChIP-seq data 54
3.2.2 Generation of Test Data 55
3.2.3 Model Fit of Simulation Data 57
3.2.4 Method Assesment 58
3.3 Results 60
3.3.1 Comparing the Simple LCA and LCRE models 60
Correlation to MGMM 60
RMSE 64
BIC 68
Sum of Scores 70
Summary of the measures 73
Conclusions 74
3.3.2 Should the BIC be used to select the best model? 76
Correlation to the MGMM 77
RMSE 80

Sum of Scores 80
Conclusion 84
3.3.3 Investigating preference of One Class vs Two Class models using the BIC 85
Analysis of two class data 85
Analysis of one class data 89
Conclusion 91
3.4 Conclusions 94

4 Changing Threshold Method **97**
4.1 Introduction 97
4.2 Method 98
4.3 Results 100
4.3.1 Correlation to MGMM 100
4.3.2 RMSE 104
4.3.3 Binding Accuracy using p_0 106
4.3.4 Sum of Scores 109
4.4 Conclusions 111

5 Applying new LCA method to data **113**
5.1 Introduction 113
5.2 Methods 114
5.3 Results 117
5.4 Conclusions 123

6 Conclusions and Future Directions **127**

A Software **135**

A.1 ChIP-seq Peak Identification Software 135
A.2 R Software and Scripts 136
A.2.1 Chapter 2 136
A.2.2 Chapter 3 137
A.2.3 Chapter 4 139
A.2.4 Chapter 5 140
A.3 Other Software 140

B Chapter 3 Full Results **143**
B.1 Comparing the Simple LCA and LCRE models Results 143
B.1.1 Average Correlation to MGMM and Standard Deviation of Correlation to MGMM for three models 143
B.1.2 RMSE for three models 160
B.1.3 BIC for each of the three models 162
B.1.4 Sum of Scores with matching replicate results 163
B.2 Investigating preference of One Class vs Two Class models using the BIC 165
B.2.1 Frequencies of one class preference for data with two clusters 165
B.2.2 Frequencies of one class preference for data with one clusters 167

C Chapter 4 Full Results **169**
C.1 Average correlation and standard deviation for threshold method results 169
C.2 RMSE results 174
C.3 Binding Accuracy 175

Bibliography **177**

Abstract

As research into epigenetics grows, it is clear that modifications to DNA through histones and other proteins can change behaviour within the cell, and is an important aspect of cellular function. One of the methods to observe these modifications is chromatin immunoprecipitation sequencing (ChIP-seq), which specifically targets protein-bound DNA to determine its location along the genome. The outcome of this technique are sequences of DNA, which indicate regions of DNA that may be bound by the protein. A drawback of this technique is that noise within the data can hide the true location of these proteins, and thus ChIP-seq peak calling software is needed to identify putative binding sites, which can then be associated with genes.

There are a number of these programs available, but they tend to have a low level of agreement. This is because they use a wide variety of peak identification models that rely on different assumptions about the data. Ideally, the results from a number of tools could be combined to identify a combined, robust set of associated genes. One candidate technique is Latent Class Analysis (LCA).

The aim of this thesis is to apply LCA to ChIP-seq data, and use it to identify a reliable set of bound genes.

Three different LCA models were considered; a simple model, as well as models with additional random effects. These random effects had either constant loading among the programs, or non-constant loading. In Chapter 1, I applied these models to ChIP-seq data to observe the initial results.

Next, in Chapter 2, I performed a series of simulations with varying parameters,

and analysed them with the three models, to clarify and extend upon the results from Chapter 1. In this case, the underlying truth was known, so I could measure the performance of each model. These measurements included the correlation to a Multivariate Gaussian Mixture Model (MGMM) results, which was fitted to the underlying data, and the root mean squared error to the MGMM results.

An additional measurement was the BIC. Aside from comparing the models for accuracy, I also assessed the use of BIC for both determining the correct number of classes to use, and as a method of determining the best model using the simulations.

Finally, in Chapter 4, I developed and tested using simulations a new method of using the LCA models to acquire a more accurate set of putative binding genes. This was analysed using the MGMM, as well as by comparing the proportion of binding genes with the known expected number. I then applied this new method to the original data in Chapter 5.

Based on initial results in Chapter 1, the LCA model without random effects generated a reasonable set of binding genes. This was further confirmed using the results of the simulations in Chapter 2, which indicated that the posterior probabilities are more accurate using this model. In addition, the BIC was not found to accurately determine the best number of classes. When assessing the use of the BIC to choose a model, it was found that it did not necessarily find the best performing model, and, based on the simulations, selecting the LCA is better. Finally, assessments of the new method indicated that it performed well compared to using a single model.

In conclusion, the approach that incorporates changing thresholds with the LCA was shown to be the most effective at producing a combined robust set of genes.

Chapter 1

Introduction

1.1 Introduction to the problem

1.1.1 Epigenetics and chromatin

Epigenetics is the study of heritable changes to DNA that are not due to changes in the DNA sequence (Berger et al., 2009). These changes include the modification of histones; large proteins that the DNA wraps around to form complex structures called nucleosomes, as well as the addition of methyl (-CH_3) groups to bases in a process called methylation.

Modifications to DNA methylation and histones are key to cell differentiation within the body. These modifications of the DNA affect transcription, and lead to the differential expression of proteins. Methyl groups have been found to inhibit protein binding, preventing key transcriptional proteins such as RNA polymerase from acting on regions of DNA. This leads to regulation of transcription (Jeong et al., 2016).

Nucleosomes are the principal component of chromatin. The packaging of the DNA into tight clusters is organised around these chemical spools (see Figure 1.2). The inclusion of nucleosomes affects the accessibility of the DNA to proteins and

hence allows the cell to control gene expression.

Histones can be modified in a number of ways to regulate genetic activity. These modifications, including histone methylation (addition of methyl groups) and histone acetylation (addition of an acetyl group -CH_3CO), lead to changes in the composition and positioning of the nucleosome (see Figure 1.1). For instance, different histone variants influence the stability of the nucleosome and lead to changes in the structure of chromatin. Positioning of the nucleosome is affected by a large number of enzymes. These enzymes can be used by the cell to tag nucleosomes for removal and insertion into different areas of the genome. Other enzymes may lead to the shifting of nucleosomes a relatively short distance along the DNA, allowing for more dynamic control of transcription. While the effect of particular histone modifications has been characterised, the exact mechanisms causing these effects remains poorly understood.

Nucleosomes are also the basis for more complex packaging of the DNA; nucloesomes can be packaged close together leading to the formation of the metaphase chromosome (see Figure 1.2). Generally, DNA is more loosely packaged to allow for protein access (Venkatesh and Workman, 2015).

Different cell types within an organism have different patterns of chromatin modification (Stueve et al., 2016). Regions which are accessed often will have open or loose chromatin, where histones are more sparsely located along the DNA, and where proteins are more likely to bind. Conversely, regions accessed rarely will be found in a closed or tight chromatin state, with a greater number of histones and a lower chance of bound proteins (Even-Faitelson et al., 2016). Similarly, methylation is also more common in areas of low access, and vice versa for regions of high access (Jeong et al., 2016). Through these mechanisms, the cell has fine control of gene expression at the level of the DNA. Since these epigenetic changes are also reversible,

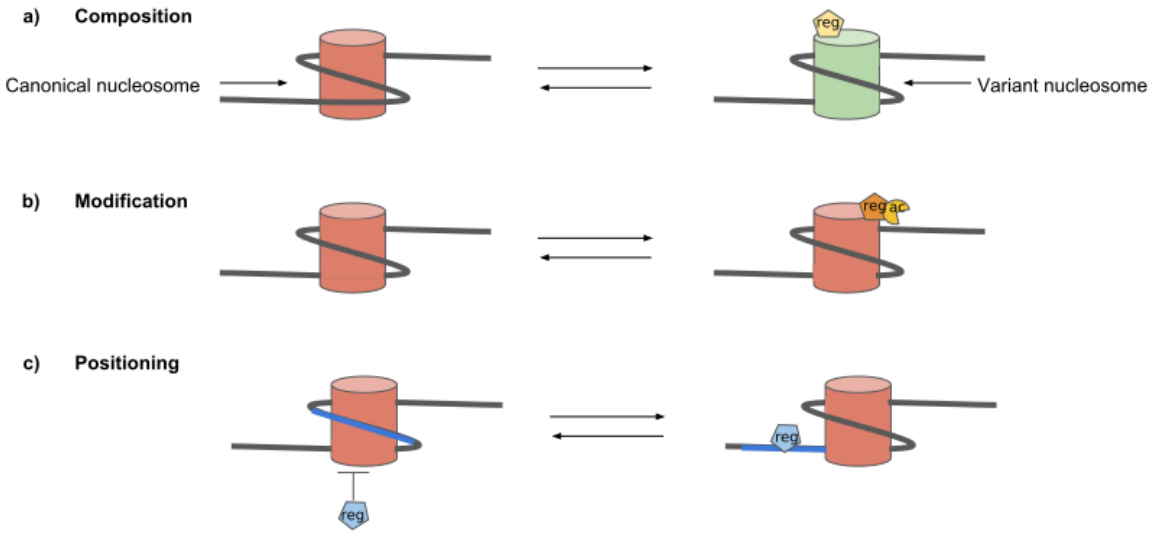


FIGURE 1.1: Histone modifications that lead to changes in transcription. a) Different histone variants can be incorporated into the nucleosome, influencing the stability of the nucleosome and structure of the chromatin. b) Addition of chemical groups to histones may affect the expression of genes on the associated DNA. c) Enzymes can affect the positioning of the nucleosome, leading to the shifting of the protein complex along the DNA, or its entire removal (not shown).

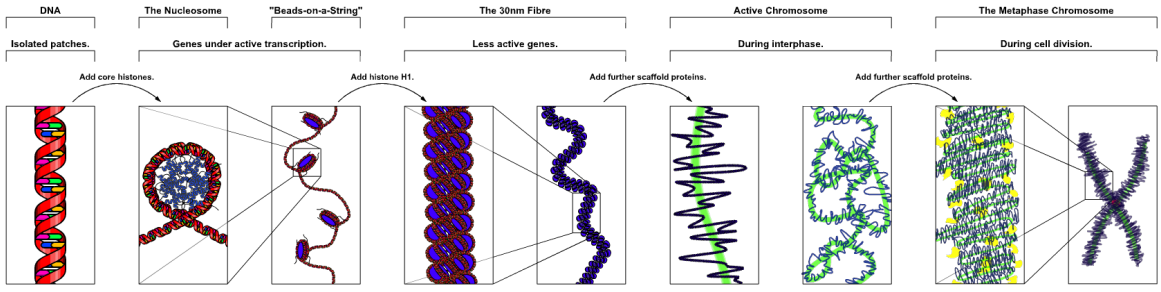


FIGURE 1.2: Structure of chromatin within the nucleus of the cell adapted from en.wikipedia

the cell is also able to change its gene expression in response to environmental cues (Venkatesh and Workman, 2015).

The effects of epigenetics are often observed in disease. Epigenetics can cause disease by the deregulation of epigenetic modification pathways, leading to changes in gene expression, or through inhibiting access of RNA Polymerase to functional genes (Jeong et al., 2016; Portela and Esteller, 2010). In cancer, it is common for there to be significant changes in chromatin that lead to increased expression of genes promoting cell proliferation, cell growth and survival and decreased expression in genes that downregulate these pathways and initiate apoptosis. Some inherited diseases, such as Prader-Willi syndrome, occur because the only functional copy of a particular gene is methylated, leaving only the mutated gene available for transcription (Portela and Esteller, 2010).

The types of condition needed to effect changes to the chromatin state are still being investigated. It is of interest to determine not only the conditions necessary for change, but also the dynamics of such change within the genome. This is achieved by performing time course analyses, where multiple samples are obtained over time from the same experiment. There are well established methods of finding locations of closed chromatin, however the statistical analysis of the resulting data is complex. This literature review will examine different methods for analysing this type of data, in particular current tools available for implementation.

1.1.2 Observing epigenetic changes

It is necessary to determine when epigenetic changes are occurring, and where in the genome, in order to investigate this phenomenon. However, sequencing the genome will not normally give information about chromatin or epigenetics, as bound proteins are removed from the DNA during the sequencing preparation process, and

methylated DNA cannot be distinguished from ordinary DNA during normal sequencing (Ku et al., 2011).

CpG methylation is relatively simple to measure using bisulphite sequencing. Bisulphite is used to treat samples of DNA, which causes the conversion of unmethylated cytosine to uracil, but leaves methylated cytosine nucleotides unchanged. After sequencing, the treated sample is compared with an untreated sample. The remaining cytosines indicate the original location of the methylated cytosines (Jeong et al., 2016).

In contrast, histone modification is difficult to measure directly. In recent years several new procedures have been developed to determine locations of tightly bound chromatin (see Figure 1.3). These techniques are presented in the following section.

The procedures are performed on samples of cells with qualities of interest. While all of the techniques below are used to determine the state of chromatin, chromatin immunoprecipitation sequencing (ChIP-seq) specifically targets protein-bound DNA, while both the assay for transposon accessible chromatin sequencing (ATAC-seq) and formaldehyde assisted isolation of regulatory elements (FAIRE-seq) identify regions of open chromatin (Johnson et al., 2007; Buenrostro et al., 2013; Giresi et al., 2007). While this seems to imply that ChIP-seq will return different regions of the genome to ATAC-seq and FAIRE-seq, this is largely dependent on the protein chosen. Many proteins, for example RNA polymerase, are associated with open regions of chromatin (Phillips and Shaw, 2008). The output of the techniques are fragments of DNA that can be sequenced.

ChIP-Seq

ChIP-seq is one of the older methods developed to determine locations of tightly bound chromatin, or any DNA binding protein. Proteins are covalently bound to the DNA using formaldehyde, and the DNA is then extracted and sonicated to break it

into small fragments. Next immunoprecipitation occurs; specific antibodies attach to the protein of interest, allowing these DNA-protein complexes to be extracted from the sample. The DNA-protein binding is reversed, and the sample is purified to isolate the DNA (Johnson et al., 2007).

ATAC-seq

ATAC-seq uses a transposase known as Tn5. When a cell is treated with this transposase, it simultaneously fragments exposed areas of the genome and adds a sequencing tag in a process called tagmentation. The transposase will tend to only affect areas where chromatin is not tightly bound, since these areas allow for binding to occur more often. Using these tags, genome fragments are then isolated and amplified for sequencing (Buenrostro et al., 2013).

FAIRE-seq

FAIRE-seq uses formaldehyde to first bind all proteins covalently to DNA, as in ChIP-seq. The sample is sonicated to break the DNA into small fragments, and a phenol-chloroform extraction is performed. This creates two phases within the sample. The DNA fragments bound to nucleosomes will preferentially sit in one of these phases, allowing it to be extracted from the rest. The remaining DNA, which corresponds to open chromatin, can be sequenced (Giresi et al., 2007).

Each technique has its advantages and disadvantages in terms of both experimental procedure and analysis. ChIP-seq is the most widely used of the three techniques, and the most flexible in that it can be used to find non-epigenetic proteins as well as specific histone marks (Johnson et al., 2007). The most significant cause of bias in ChIP-seq is the use of antibodies, as different antibodies bind to proteins of interest with different strengths. This leads to changes in the relative strengths of sample

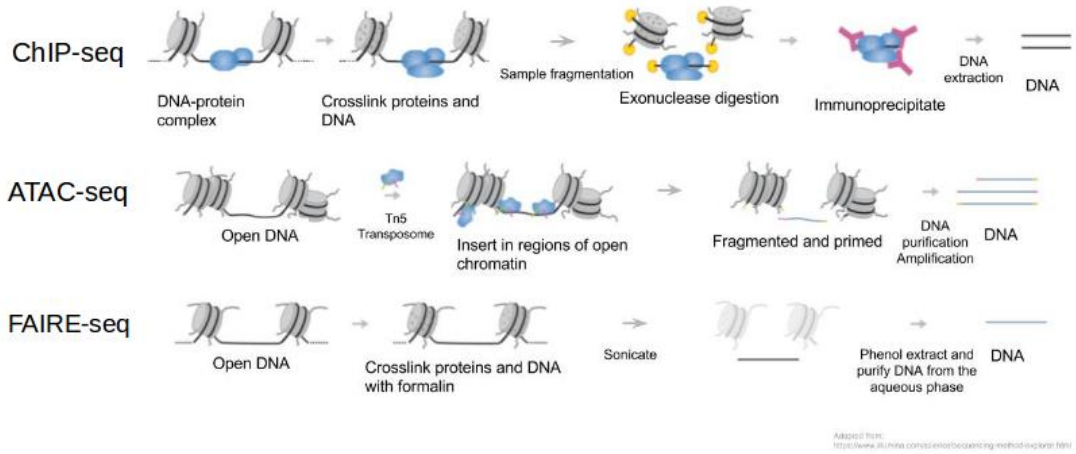


FIGURE 1.3: ChIP-seq, FAIRE-seq and ATAC-seq methods, adapted from the Illumina Sequencing Method explorer

peaks and these differences in data quality reduce accuracy in analysis (Meyer and Liu, 2014). It is recommended that ChIP-seq experiments are accompanied by a control, preferably a spike-in control, where quantities of known readily identifiable nucleic acids are added to the sample before immunoprecipitation. While ATAC-seq and FAIRE-seq require less experimental calibration and do not have the difficulties associated with immunoprecipitation, these technologies are still relatively new and the associated biases are not fully understood (Meyer and Liu, 2014).

All of these techniques require some method of reading large numbers of fragments. This became possible with the advent of next generation sequencing, which allows for high throughput of data (Schuster, 2008). The most common method of sequencing is performed by Illumina, and the end result is short (50-200 bp) single or paired end reads. Next, the samples are aligned to a reference genome, allowing for analysis.

Since most techniques of analysis are created with ChIP-seq in mind, this technique will be the main focus for the rest of this review. However, tools and techniques discussed here are also relevant to ATAC-seq and FAIRE-seq.

1.1.3 Limitations and difficulties of analysis

One of the challenges of ChIP-seq and similar data is that the analysis is more difficult than in related techniques. Two methods that ChIP-seq is often compared to, and which have similar analyses, are bisulphite sequencing (as previously described) and RNA-seq. This is a technique that collects and analyses RNA fragments to determine the expression levels of genes in the cell at the time of collection (Wang, Gerstein, and Snyder, 2009). This technique is well-characterised and many tools are available for analysis

The analysis of ChIP-seq data is more difficult for a number of reasons. Unlike in RNA-seq, the space in which there could be potential changes to chromatin are not limited to within genes, but instead cover the entire genome. Additionally, there is no limit on the amount of signal that could be obtained from this type of data, unlike CpG methylation, which is constrained to a finite interval of 0 to 100% methylated. Furthermore, depending on the particular method employed, one can expect to see considerable noise between samples of data, especially when those samples were not obtained during the same experiment or from the same laboratory. This complication means that normalisation of the data is a key step within analysis. Finally, the length and shape of enriched regions will differ significantly, depending on the target protein. This variation is observed because proteins vary in size, and different lengths of DNA will be enriched in the sample (Steinhauser et al., 2016). In the case of epigenetic modifications, these regions are usually larger (Shen et al., 2013; Xu et al., 2014).

1.1.4 Peak finding software

After alignment, the next step is to identify locations of bound chromatin by differentiating peaks in the data from the noise (Zhang et al., 2008). There are a number of tools readily available to do this; collectively known as peak-finding software.

The method of determining enriched peaks differs based on the tool used. Tools currently available for this purpose include BELT, SISSRs, QuEST, PeakSeq and MACS. BELT uses a percentile rank of the bins to determine significant enrichment levels, and SISSRs calculates binding sites based on the number of forward and reverse reads between windows (Lan et al., 2010; Narlikar and Jothi, 2012). Both QuEST and PeakSeq use a control to compare enrichment in windows and specify regions that differ significantly as potential peaks (Valouev et al., 2008; Rozowsky et al., 2009). MACS models the data as a Poisson distribution and then finds candidate peaks using the p-value for significant enrichment (Zhang et al., 2008).

To rank peaks, most tools estimate a False Discovery Rate (FDR) for the data (Zhang et al., 2008; Lan et al., 2010; Narlikar and Jothi, 2012; Rozowsky et al., 2009), while others rank the peaks using different scoring methods. For example, QuEST uses kernel density estimation derived scores (Valouev et al., 2008). The result of all tools is a list of peaks, denoting locations of bound protein within the genome.

1.1.5 Analysis of differential regions

One of the key applications of ChIP-seq and similar technology is determining the effect of different cell types or treatments on chromatin. By comparing two different samples that differ only by a condition of interest, it is possible to determine its effect on the chromatin state. In particular, the changes in the location of closed chromatin, and thus potential changes in gene expression are of interest (Shen et al., 2013). A simple visualisation of this is given in Figure 1.4. Here we can see that in

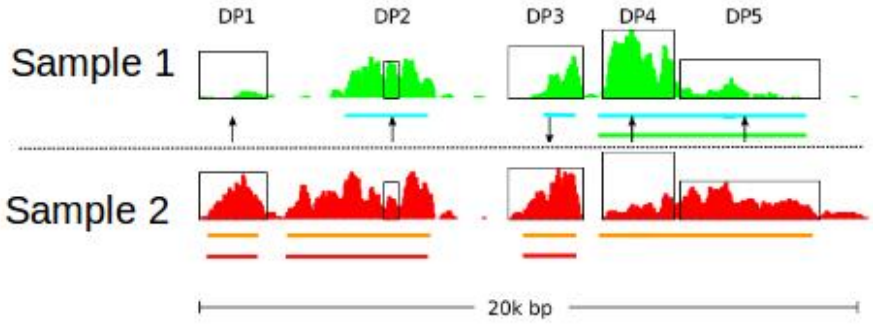


FIGURE 1.4: Comparing two samples for locations of differential binding. Putative regions are emphasised by the boxes. Adapted from (Allhoff et al., 2014).

our two samples, there is an apparent difference in binding at one location, but not in the other. To identify regions across the entire genome we require complex statistical tools. The effect of certain mutations, potential nuclear drug targets, variations between cell types and temporal changes to chromatin state are some themes that could be explored (Chahwan, Wontakal, and Roa, 2011).

Due to the overwhelmingly large number of reads generated by ChIP-seq and similar technology, computers are required to identify regions that appear to vary between conditions. ChIP-seq is a recent technological advance and more established techniques have developed well understood methods of identifying differential regions within the genome. There are many tools that identify differences in DNA methylation using the results of bisulphite sequencing, including IMA and QDMR (Zhang et al., 2011; Wang et al., 2012).

Two popular computational tools used for determining differential RNA expression are edgeR and DESeq, and these are often used in the analysis of ChIP-seq data as well (Anders and Huber, 2010; Robinson, McCarthy, and Smyth, 2010). One disadvantage of using these tools specialised for RNA-seq is that they were originally designed for finding peaks over the relatively short regions that form individual genes. In contrast, when ChIP-seq and similar technology is used for determining

locations of closed chromatin, the peaks cover much larger regions of the genome. If this difference is not taken into account it may lead to erroneous results (Shen et al., 2013).

There are a number of tools that identify differential regions of ChIP-seq, FAIRE-seq and ATAC-seq data (Steinhauser et al., 2016). While most are initially designed for ChIP-seq, most researchers will use the same tools for FAIRE-seq and ATAC-seq, under the assumption that the data is similarly distributed. There are a variety of methods available, and depending on the experiment, the most appropriate tool to use will vary. Most tools will then perform three main steps of analysis; peak calling, normalisation and statistical testing (Steinhauser et al., 2016). These steps may be done discretely, or in one complete process.

Statistical tests allow the two samples to be compared in order to determine whether any given peak could be considered common to both or unique. There are a number of methods that can be used to compare sets of peaks. While each tool has its own implementation of a method, there are a number of common approaches. This may include using a fold change threshold, applying model based analysis approach, using a non-parametric test or utilising Hidden Markov Models (HMM).

In a fold change threshold method, bins between the two samples are considered equal if the read numbers do not exceed a pre-specified fold-change threshold. This is used by HOMER and MACS2 (Heinz et al., 2010; Zhang et al., 2008).

When using a model based analysis approach, equivalent bins between the samples are compared and a p-value is calculated based on either the Poisson or the negative binomial (NB) distribution, with a null hypothesis that the true expression between the two bins is the same based on the number of counts observed (Zang et al., 2009). This type of approach is used by SICER (Zang et al., 2009; Xu et al.,

2014).

Programs that use a non parametric approach include QChiPat, created by Liu et al., (2013), which uses Wilcoxon rank tests to distinguish differentially enriched regions. Finally, some tools, such as ODIN and RSEG, determine differentially enriched regions with a Hidden Markov Model (HMM) (Allhoff et al., 2014; Song and Smith, 2011). This approach differs significantly from the more typical methods described above, particularly in that the entire analysis occurs in one step, rather than in discrete steps for peak calling, normalisation and testing.

1.2 Further Analysis

The vast array of tools available to analyse ChIP-seq data give researchers the ability to identify the effect of these tools on the same set of data. As has been identified in a number of other studies (Thomas et al., 2016; Steinhauser et al., 2016), ChIP-seq programs tend to have a low level of agreement. Since every program relies on a number of assumptions regarding their model for ChIP-seq and similar data, it is reasonable to expect that all tools generate false positive as well as false negative results (Cantarel et al., 2014). Due to the nature of ChIP-seq data, a "gold-standard" does not exist, because while it is possible to individually validate some sites via other molecular biology techniques, this is not feasible to undertake for an entire genome.

Currently, the main technique used by many ChIP-seq tools in order to justify their approach is to create simulated data for assessment by their tool as well as other established tools (Zhang et al., 2008; Ranciati, Viroli, and Wit, 2015; Allhoff et al., 2014). This allows them to compare the results and indicate their tool is the best for ChIP-seq data for the particular use case. However, this technique can be

problematic; when the data is simulated using a model based on the assumptions on which the ChIP-seq tool was designed, this will bias the tool for which the paper was written.

An approach taken in other areas of bioinformatics that also lack a gold standard for judging programs is to create a combined data set of interest by using the results of multiple tools (Cantarel et al., 2014; Elsik et al., 2007; Chen et al., 2007). Some approaches tend to be *ad hoc* and rely on finding a "majority vote" set or other intersection based methods. Others are more complex and use statistical classsification techniques such as Latent Class Analysis (LCA). This has been used successfully to generate a set of genome variants, create a consensus gene set based on gene models and to infer orthologous genes from different genomes (Cantarel et al., 2014; Elsik et al., 2007; Chen et al., 2007).

Such an approach in ChIP-seq would allow the different strengths of the programs to be combined to give a more reliable putative peak set to be investigated further. Currently, complex statistical classification techniques to combine ChIP-seq programs have not been applied to ChIP-seq data in the literature.

1.3 Latent Class Analysis

LCA can be applied to the problem of combining multiple tools in a statistically robust manner. LCA is a popular technique in psychology and social sciences, originally used for finding latent groups or classes based on a number of variables (Linda M. Collins, 2010). In these applications, a group of people are asked questions relevant to a variable of interest (for example, prevalence of alcohol and drug use in teenagers). These questions are designed to be categorical, so the participants may select one of a number of options.

The questions are called response variables and the options response categories. Suppose that there are $t = 1, ...T$ observed variables and observed variable t has R_t response categories. The responses are placed in a contingency table, which is formed by cross tabulating the T variables and has $W = \prod_{t=1}^{T} R_t$ cells. Thus each row represents a response pattern $\mathbf{y} = (y_1, y_2, ..., y_t)$, each of which is associated with a probability $P(\mathbf{Y} = \mathbf{y})$.

For a model with $c = 1, ..., C$ latent classes, each class has a probability of membership, which is called the prevalence of latent class c and is represented by γ_c. The probability of a response r_t given membership in class c is called the item-response probability and is denoted by $\rho_{t,r_t|c}$. Thus the probability of observing response y conditional on membership in latent class c is given by:

$$P(\mathbf{Y} = \mathbf{r}|L = c) = \prod_{t=1}^{T} \prod_{r_t=1}^{R_j} \rho_{t,r_t|c}^{I(y_t=r_t)} \tag{1.1}$$

where L is the latent variable and $I(y_t = r_t)$ is an indicator function that equals 1 when $y_t = r_t$ and 0 otherwise. Finally the probability of observing response y regardless of class c is given by:

$$P(\mathbf{Y} = \mathbf{r}) = \sum_{c=1}^{C} \gamma_c \prod_{t=1}^{T} \prod_{r_t=1}^{R_t} \rho_{t,r_t|c}^{I(y_t=r_t)} \tag{1.2}$$

Using Bayes Theorem, we can find the posterior probability of a class c given the response pattern y:

$$\begin{aligned} P(L = c|Y = y) &= \frac{P(y = y|L = c)P(L = c)}{P(Y = y)} \\ &= \frac{\gamma_c \prod_{t=1}^{T} \prod_{r_t=1}^{R_t} \rho_{t,r_t|c}^{I(y_t=r_t)}}{\sum_{c=1}^{C} \gamma_c \prod_{t=1}^{T} \prod_{r_t=1}^{R_t} \rho_{t,r_t|c}^{I(y_t=r_t)}} \end{aligned}$$

Returning to the example of a group of people asked questions relating to a variable of interest, classes will consist of similar sets of responses to the questions. We can then identify the most likely class for each person based on their response. For given C, the Estimation-Maximisation (EM) algorithm can be used to estimate the parameters of the LCA model.

There are a number of programs available to estimate the number of classes automatically based on the input of the contingency table or equivalent data. Multiple LCAs are performed, with differing numbers of classes, and the results are compared using BIC or similar to determine the best number of classes for the data (Qu, Tan, and Kutner, 1996).

LCA can be applied to ChIP-seq data to combine the responses of different callers. For the application to the problem of identifying a binding and non-binding class using the results from multiple programs, we can consider genes as the responders and the programs as our variables.

The response items would be binary, either binding or non-binding, and based on the results of a program. Furthermore, the maximum expected class size would be two, where the genes are separated into a binding or non-binding group overall. Based on these changes, the posterior probability can be simplified to:

$$P(L = c|Y = y) = \frac{\gamma_c \prod_{t=1}^{T} \pi_{t|c}^{y_t} (1 - \pi_{t|c})^{(1-y_t)}}{\sum_{c=1}^{2} \gamma_c \prod_{t=1}^{T} \pi_{t|c}^{y_t} (1 - \pi_{t|c})^{(1-y_t)}} \tag{1.3}$$

Where $\pi_{t|c}$ is the probability that y_t is 1 given that it is in class c (Beath and Heller, 2009).

One fundamental assumption of the LCA model is that there is local independence, such that the observed variables are independent (Linda M. Collins, 2010). However, this may not always be true for the application. An example from medicine

could be the assessment of patients for a particular disease using different tests. If two or more of the tests rely on similar underlying information, such as a blood sample, this could lead to a dependence between the two tests, and invalidate the assumption of LCA. Similarly for the application of ChIP-seq, if two of the programs rely on similar assumptions or the same model within the data, this could cause similarities between the results that isn't reliant on the true binding or not-binding status (Qu, Tan, and Kutner, 1996).

For binary response, a simple model for this dependence is to assume an unobserved continuous random variable $\lambda_i \sim N(0,1)$ for gene i, which is incorporated into the above equation through $\pi_{ti|c}$:

$$\pi_{ti|c} = \phi^{-1}(a_{t|c} + b_{t|c}\lambda_i) \tag{1.4}$$

where $a_{t|c}$ determines the item response probability for a value of 0 for the random effect and $b_{t|c}$ scales the random effect and is usually known as the loading or discriminant (Qu, Tan, and Kutner, 1996). This loading can be the constant or non-constant for each program. Thus the marginal probability, found by summing over the classes and integrating over λ becomes:

$$P(Y = y) = \sum_{c=1}^{C} \gamma_c \int_\lambda \phi(\lambda) \prod_{t=1}^{T} \pi_{t|c}^{y_t}(1 - \pi_{t|c})^{(1-y_t)} \tag{1.5}$$

and hence the posterior probability is:

$$P(L = c|Y = y) = \frac{\gamma_c \prod_{t=1}^{T} \pi_{t|c}^{y_t}(1 - \pi_{t|c})^{(1-y_t)}}{\sum_{c=1}^{2} \gamma_c \int_\lambda \phi(\lambda) \prod_{t=1}^{T} \pi_{t|c}^{y_t}(1 - \pi_{t|c})^{(1-y_t)}} \tag{1.6}$$

For calculating the parameters, the integration makes it necessary to use an approximation, for example the Gauss-Hermite quadrature. An algorithm has been implemented in R for this complex LCA, in the package randomLCA (Beath, 2008).

1.4 Research Aims

The overall aim of this thesis is to determine suitable methods of combining the results of multiple programs in a statistical rigorous manner. These goals can be broken down into the following aims:

- Determine the suitability of Latent Class Analysis for combining the results of multiple ChIP-seq programs
- Evaluate the performance of this method for a range of data using simulations
- Use the results from the simulations to make improvements upon the original analysis, if possible

Chapter 2

Analysis of multiple ChIP-seq programs

2.1 Introduction

As ChIP-seq experiments have become more popular, there has been a rise in the number of programs available to identify putative ChIP-seq peaks (Zhang et al., 2008; Heinz et al., 2010; Allhoff et al., 2014; Bao et al., 2014; Harmanci, Rozowsky, and Gerstein, 2014; Xing et al., 2012). These tools assume various different read distribution models in order to determine when peaks are likely to represent true binding events rather than noise. These differing assumptions mean that the tools do not necessarily agree and, often, the levels of agreement are suprisingly low (Steinhauser et al., 2016). It is unclear which program has the most accurate model. Since so many peaks are identified by any given program, it is difficult and prohibitively expensive to use other molecular methods to corroborate these peaks. Furthermore, depending on the target of the ChIP-seq experiment, we may observe different read distributions. For example, some histone marks, such as H3K36me3, have peaks that are low in read number but wide in range, while others, such as H3K4me3 are high in read number but narrow in range (Even-Faitelson et al., 2016). Thus there is

no gold standard for ChIP-seq peak identification.

As mentioned in Chapter 1, one way to gain a more reliable set of peaks is to use multiple programs. One approach is to find the intersection of associated genes found between programs. However, this method does not take into account programs with a high level of disagreement, which will remove genuine binding genes from the gene set. In contrast, if the programs are too similar in their assumptions, this may have the opposite problem, leading to the inclusion of non-genuine binding genes.

Latent Class Analysis (LCA) is a statistical method that can be used to provide a more principled approach to combining results from multiple callers (Cantarel et al., 2014). It is described in greater detail in Chapter 1. An LCA model that uses random effects may be more appropriate when there is correlation between the programs, as this correlation violates assumptions in the simple LCA model (Beath and Heller, 2009). For example, certain programs that make similar assumptions will create these correlations. Therefore, three models will be considered; a simple LCA model (the LCA model), a LCA model with random effects and constant loading (LCRE with constant loading) and a LCA model with random effects and non-constant loading (LCRE with non-constant loading).

In this chapter, I examined these three different models for categorising the genes found by multiple programs. Initially I described the data as well as the programs that were used. I then applied a simple LCA model to the data, as well as an LCA that will include a random element with constant loading and with non-constant loading, and compared the results.

2.2 Programs and Dataset

2.2.1 Dataset

I used the Encyclopedia of DNA Elements (ENCODE) portal to identify a set of high quality samples (Consortium, 2012). In addition, the portal provided filtered alignments, so this saved alignment and processing time. The ChIP-seq target for the samples was H3K36me3 for *Homo sapiens* neutrophil cells (Experiment ENCSR373WCB). The resulting reads were mapped to assembly GRCh38 using the tool BWA (Li and Durbin, 2009). Two anisogenic replicates were available, to improve the quality of the peak identification. In addition, control samples (two anisogenic replicates) from the same laboratory were also obtained from ENCODE for control of noise (Experiment ENCSR557RDB). Details on the processing pipeline for the samples are available at www.encodeproject.org (Consortium, 2012). See Appendix A.3 for further information.

H3K36me3 is a well-defined histone modification - the trimethylation of lysine 36 of histone H3. This has been found to be tightly associated with active transcription (III and Reinberg, 2009). The goal of ENCODE, is to build a comprehensive list of functional elements in the genome, and this experiment is part of that investigation (Consortium, 2012). In particular, these samples are meant to indicate the standard locations of this mark in this particular cell type.

2.2.2 Calling Programs

A number of programs were applied to the H3K36me3 dataset. The programs were selected to represent a variety of read distribution models. While many of the programs are differential peak callers, here they are used in a single peak calling capacity. This means for some, only the peak calling step was used, while in others

the different peak calling method was used with the control as the second data set. Table 2.1 summarises these programs briefly.

Tool	Peak Calling	Normalisation	Differential peak calling method	Reference
MACS2	Sliding window approach, identifies peaks using Poisson distribution	Library size normalisation	Fold change threshold: Log_{10} likelihood ratio cutoff.	Zhang et al., 2008
HOMER	Window based approach, identifies peaks using Poisson distribution	Library size normalisation	Fold change threshold	Heinz et al., 2010
THOR			Hidden Markov model with three states. Models with mixture of Poison distributions. Takes into account replicates.	Allhoff et al., 2014
enRich			Markov random field model. Models with a zero inflated negative binomial. Now archived.	Bao et al., 2014
MUSIC	Uses multiscale decomposition to identify significantly enriched regions at 7 different scales, then merges these to gain a final set of enriched regions.	Both control and ChIP reads are filtered for duplicates and then uses control data to normalise before peak calling.		Harmanci, Rozowsky, and Gerstein, 2014
BCP	Designed for broad enrichment. Uses a stochastic Bayesian Change-Point method to calculate posterior means and categorise the genome. Only uses one replicate.	Uses a control to filter false candidates.		Xing et al., 2012

TABLE 2.1: A summary of tools used for identification of binding genes with LCA. Methods are separated into peak calling, normalisation and differential peak calling method.

The data were analysed using the recommended settings for each tool, given the type of protein and experiment, resulting in a series of ranges across the genome for each program. These ranges correspond to putative binding sites. The remaining analysis was performed with R (see Appendix A).

The ranges from each tool were used to annotate the genome in order to find associated genes. This accounted for variation in peak length and generated a more comparable data set for each tool. The outputs from each tool were compared to the locations of genes, and genes that were identified as having a closely associated putative peak ($\pm$ 200bp around gene range) were retained. This was performed

with the package biomaRt using the Ensembl gene dataset for *H. sapiens* (Zerbino et al., 2018). Genes were identified based on their Entrezgene IDs. These genes were then used for the remainder of the analysis.

An assessment of the intersections of the genes found by each of the tools was performed using UpSetR (Lex et al., 2014) and is given in Figure 2.1. The largest intersection contained all peak-calling programs except for enRich, and the top 5 peaks all included MACS2. The intersection of all programs was also large, with slightly less than 1000 genes being common across all 5 programs. Additionally, there were 11,590 genes not found by any program (not included in the Figure 2.1). Notably, MACS2, enRich and BCP were the only programs that called peaks that were associated with unique genes. While there are a considerable number of genes commonly found by all the programs, for MACS2, HOMER, and BCP this constituted at most 61% of the genes found associated with those programs.

The number of genes found to be bound by H3K36me3 is shown in Table 2.2. enRich not only found the fewest genes in general, but also had the fewest genes in common with the other programs. Conversely, MUSIC both had the most genes and had the most in common with other programs. MACS2, MUSIC and BCP also found a number of genes independently of the other tools, despite HOMER finding more genes than BCP.

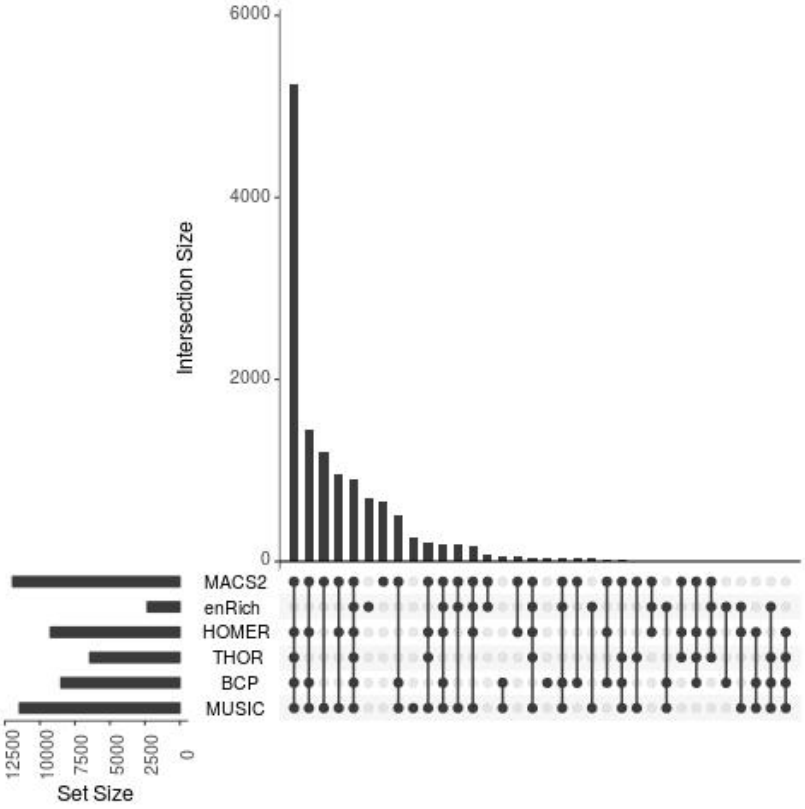


FIGURE 2.1: UpSetR plot for the intersection of genes found by peak calling programs. The "Intersection Size" gives the number of genes within that intersect, while "Set Size" shows the number of genes for each program as listed. The filled dots indicate which programs were included in each intersection. The target of the ChIP-seq experiment was H3K36me3.

Program	Number of Genes
MACS2	12599
MUSIC	12068
HOMER	9613
BCP	8867
THOR	6642
enRich	2539

TABLE 2.2: The number of genes found to be bound by H3K36me3 by each program. Programs are listed in decreasing number of genes.

2.3 Latent Class Analysis of ChIP-seq Peak Calling Programs

2.3.1 The Simple LCA Model

The LCA was performed using randomLCA (Beath, 2008). Since I was trying to determine whether any given gene was being bound or not bound within the sample, two different models were tested, one class or two classes. If the two class LCA was not found to be a significantly better fit than the one class model, this would imply that the genes did not partition into two classes. Using the Bayesian information criterion (BIC), the two class model was found to be the best model.

The two class model can be interpreted as having partitioned the genes into a binding class and a non-binding class. Figure 2.2 shows the calling probabilities of each class and program, or the probability that the program has a peak for a gene, given it is in that class (Schwarz, 1978). Thus Class 2 has genes that are classified as binding, while Class 1 includes the genes that are classified as non-binding. Confidence intervals at the 0.95 level were also calculated using a parametric bootstrap. It is clear that there is a high confidence on these binding probabilities, given by the small range over the confidence interval. Notably, enRich has a low probability of genes binding in either class. This is partly due to the low number of binding genes associated with the enRich peaks, as well as the low level of agreement, as observed in Figure 2.1.

The LCA can be used to estimate the number of binding genes. Binding or non-binding status is determined by finding the class for each gene that has the maximum posterior probability, based on the ”profile” of that gene (the outcome for each program of whether it calls the gene or not). In this case, if Class 2 has the highest posterior probability, the gene is considered binding under the model. Based on the

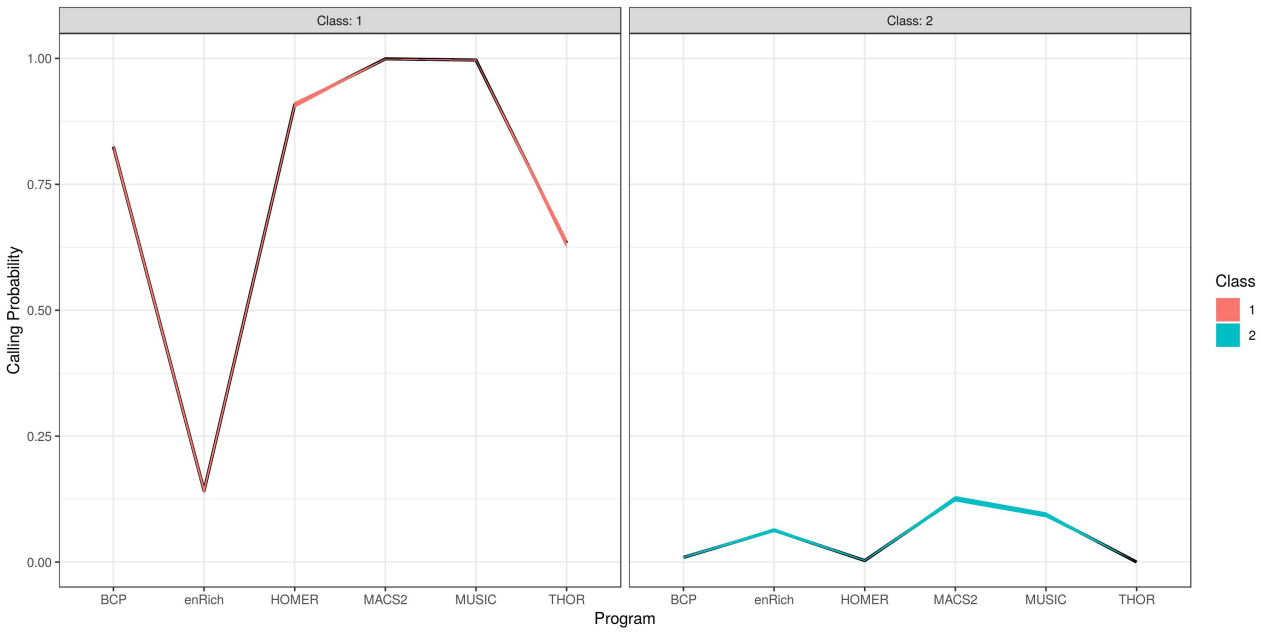


FIGURE 2.2: Calling probabilities for the LCA model, including 0.95 confidence intervals. The calling probabilities give the probability that the program has a peak for a gene, given it is in that class. The confidence interval is shown in colour differing by class, while the darker line indicates the outcome value.

LCA, 9,824 of the genes were found to be bound. This is higher than the average number of genes found by each of the programs individually (Table 2.2). Using GO enrichment analysis, I more closely investigated the function of these genes. The results are given in Figure 2.3. The top GO terms found were primarily associated with regulatory functions. This is consistent with the target H3K36me3 being associated with transcriptionally active genes.

The observed frequencies were compared to the expected frequencies based on the 2 Class LCA model (see Table 2.3). These expected frequencies tended to differ quite significantly from the observed frequencies. The difference between observed and expected indicated a generally poor fit to the data. A poor fit may be indicative of a violation of one of the key assumptions of the LCA model; that the programs are calling independently. This suggested a more complex model may be necessary.

Profile	Observed	Expected	Probability of Binding	Profile	Observed	Expected	Probability of Binding
000000	11592	10644.44	0.00	100000	666	1528.74	0.00
000001	266	1103.46	0.00	100001	1204	209.93	0.25
000010	43	99.41	0.00	100010	35	15.06	0.05
000011	50	10.50	0.02	100011	510	243.69	0.99
000100	0	0.00	0.11	100100	0	0.29	1.00
000101	0	0.07	1.00	100101	4	88.95	1.00
000110	0	0.00	0.98	100110	0	1.36	1.00
000111	0	0.34	1.00	100111	10	418.59	1.00
001000	0	31.33	0.00	101000	51	6.15	0.27
001001	0	3.67	0.11	101001	962	509.84	1.00
001010	1	0.30	0.02	101010	11	7.83	0.99
001011	2	2.00	0.98	101011	1458	2396.92	1.00
001100	0	0.00	1.00	101100	2	2.86	1.00
001101	1	0.72	1.00	101101	213	880.30	1.00
001110	0	0.01	1.00	101110	2	13.45	1.00
001111	2	3.40	1.00	101111	5281	4142.36	1.00
010000	704	715.29	0.00	110000	71	102.75	0.00
010001	31	74.16	0.00	110001	181	19.15	0.44
010010	1	6.68	0.00	110010	0	1.09	0.12
010011	3	0.73	0.05	110011	37	40.11	1.00
010100	0	0.00	0.24	110100	0	0.05	1.00
010101	0	0.01	1.00	110101	0	14.70	1.00
010110	0	0.00	0.99	110110	0	0.22	1.00
010111	1	0.06	1.00	110111	0	69.15	1.00
011000	0	2.11	0.00	111000	3	0.58	0.47
011001	1	0.29	0.24	111001	173	84.18	1.00
011010	1	0.02	0.05	111010	0	1.29	1.00
011011	0	0.33	0.99	111011	201	395.97	1.00
011100	0	0.00	1.00	111100	1	0.47	1.00
011101	0	0.12	1.00	111101	44	145.42	1.00
011110	0	0.00	1.00	111110	0	2.22	1.00
011111	1	0.56	1.00	111111	908	684.31	1.00

TABLE 2.3: Observed and Expected frequencies for LCA model of called genes from different programs for H3K36me3 data (without random elements). The expected frequencies demonstrate the goodness of fit of the model. The probability of binding gives the probability of any given gene in that profile belonging to Class 2 (rounded to 2 decimal places). The order of the programs in the profile is MACS2, enRich, HOMER, THOR, BCP, and MUSIC. The profile 000101 is highlighted to indicate where one would not ordinarily expect to see such a high probability of binding.

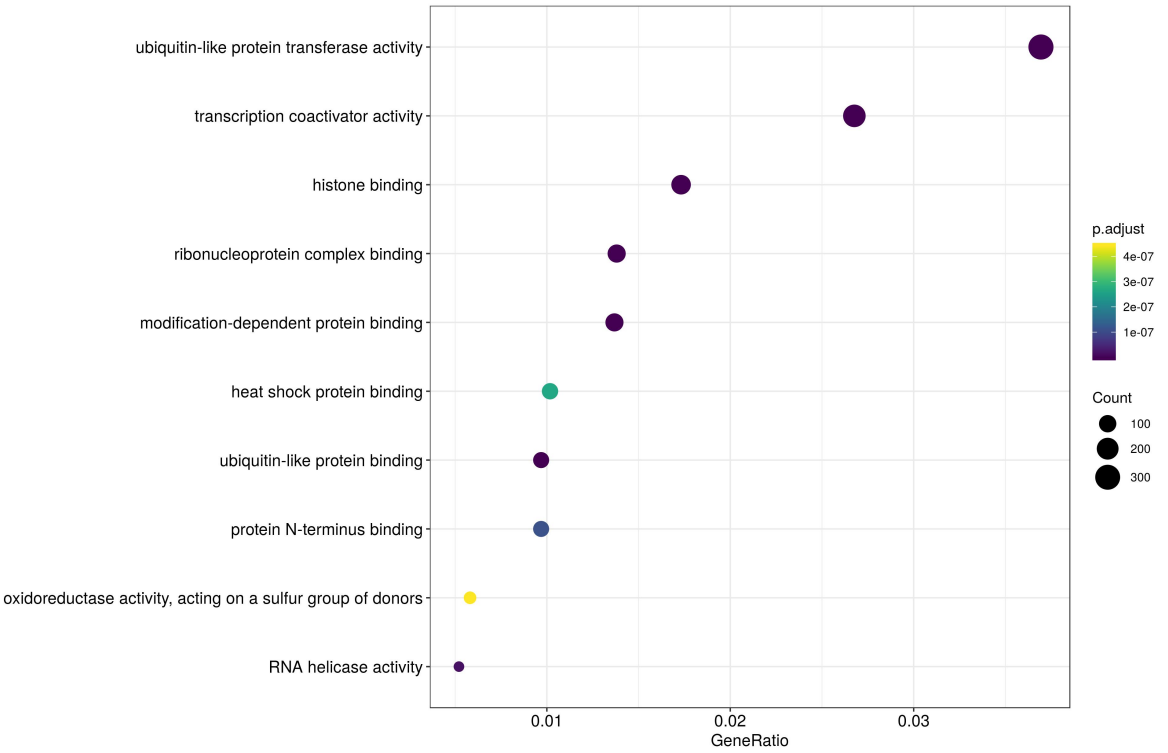


FIGURE 2.3: Significant GO terms for the combined gene set determined using the LCA model as a dotplot. The y-axis gives the significant GO terms, while the x-axis shows the ratio of genes with this GO term. The size of the points indicates the number of genes, and the colour indicates the adjusted p-value for that term.

Also included in the table is the posterior probability of each profile. Many of the profiles had a highly polarised probability of binding, in the sense that many of the posterior probabilities are approximately 1 or 0. While this was expected for profiles where all the programs agreed (000000 and 111111), others, such as profile 000101 (highlighted in Table 2.3), also have a posterior probability of 1. This indicated that if only THOR and MUSIC called a peak within the region of the gene, the LCA was confident that binding would occur. This may be due to the low number of genes that THOR called, and its high agreement to the other programs. It appears that the LCA model was more likely to weight this program more highly because of these factors.

2.3.2 LCA with a random effect

As mentioned in the Introduction, an LCA with a random effect (LCRE) is an alternative model for the data when there is evidence of correlation between programs. This could be due to some other factor about particular peaks that causes dependence between the programs. For example, an unusually strong signal in the data will be called by all programs, regardless of the different underlying models. The model for a LCRE is described in Chapter 1 as containing an unobserved continuous random variable $\lambda \sim N(0,1)$, which is incorporated as part of $\pi_{ti|c}$ (the probability that program t calls a peak near a particular gene i, given that it is in class c):

$$\pi_{it|c} = \Phi^{-1}(a_{t|c} + b_{t|c}\lambda_i)$$

This probability then has two parts, a fixed effect $a_{t|c}$ and the random effect λ_i, specific to gene i. Finally, $b_{t|c}$ is a scaling or loading effect that can either be kept constant for each program t or may vary.

Using the randomLCA package, I analysed the data above using an LCRE, testing for both a single or two class model, as well as with constant loading or non-constant loading. The BIC was used to determine the best fit. It was found that when constant loading was used, a two class LCRE was the best fit, but when non-constant loading was used, a 1 class LCRE was the best fit. The resulting BICs are shown in Table 2.4. This indicated that the inclusion of a random effect in the model could significantly change the class classification. In the case of the LCRE with non-constant loading, the random effect has accounted for the associations between the calling programs without the inclusion of an additional class.

Classes	LCRE (Constant Loading)	LCRE (No Constant Loading)
1	110194.62	91049.05
2	91622.69	91057.04

TABLE 2.4: BIC for different LCRE models for different class numbers and with or without constant loading. A lower BIC is preferred.

Notable is the size of the BIC for all four models. This is explained by examining the equation for the BIC:

$$BIC = \ln(n)k - 2\ln(\hat{L})$$

Where L is the maximised value of the likelihood function of the model and describes the goodness of fit for the data, n is the number of data points (in this case, the number of genes) and k is the number of parameters estimated by the model. Thus, the first term will mean a lower complexity model is preferred when the improvement in the fit is small. Because there is a such a large number of genes, the BIC values will always be large because of the complexity term, even with relatively simple models. Based on these results, I further investigated the two class LCRE model and the single class with non-constant loading.

Two Class LCRE with constant loading

The results from the two class LCRE with constant loading were used to generate calling probabilities for each program and class with percentiles. These are plotted alongside the equivalent probabilities for the LCA in order to compare the two methods in Figure 2.4. Class 1 is labelled here as the non-binding class and Class 2 as the binding class. The methods showed different calling probabilities for the different programs, with the LCA tending to have higher extremes in the probabilities relative to the LCRE. For the LCA, the calling probability for most of the programs

in Class 1 was greater than 0.5 with the exception of enRich, and all the programs were lower than 0.5 in Class 2. In contrast, both classes had calling probabilities closer to 0.5 for the LCRE for most programs, except for enRich. This implied that the LCRE with constant loading found fewer clear differences between the classes for most of the programs.

The confidence intervals were much wider for the LCRE, and in particular Class 2 showed a much wider range of possible calling probabilities, especially for programs HOMER, MACS2 and MUSIC. This is due to the addition of the random effect; because there is an additional term in the model that varies per gene, the calling probabilities for each program will vary per gene also, leading to the intervals observed.

While in general the LCA showed a greater difference in calling probabilties between the two classes, there was a much more extreme result for enRich. This was not observed in the calling probabilities of the LCRE,which instead found a higher calling probability for enRich in Class 1 and a much lower calling probability for enRich in Class 2. This implied that enRich was more likely to call a gene in Class 1 than Class 2 compared to the other programs.

Based on the LCRE, 2,320 of the genes were estimated to be bound, a number much lower than that found using LCA. This number agreed with previous results, which found that enRich was the most influential program, and the number of genes found by enRich was 2539 (see Table 2.2). When the gene lists between the LCA and the LCRE were compared, it was found that the majority of the genes found by the LCRE were common to both the LCRE and the LCA. This is demonstrated with the Venn diagram in Figure 2.5. Of the genes found to be binding by LCRE, 58% of these were also found to be binding by LCA.

To determine functional differences in the genes found by the LCRE compared

to the LCA, I investigated the gene set using a GO enrichment analysis. Due to the low number of binding genes identified, no significant GO terms were found.

Finally, I compared the expected and observed results for the LCRE. These results, along with the matching profile and probability of binding, are given in Table 2.5. Clearly, the expected values matched more closely compared to the expected values in Table 2.3. However, many of the values still had significant differences, which was particularly noticeable when the observed value was low. For example, for the profile 000010 the LCRE was expected to observe only 9 genes, but instead 43 were observed. Examining the probabilities of binding calculated by the LCRE, the biggest impact appeared to be the presence or absence of enRich. When enRich was present, the probability of binding was 1, and otherwise was 0. This extreme set of probabilities matched the results from the calling probabilities and the Venn diagram, which indicated that the LCRE was heavily biased towards the results of

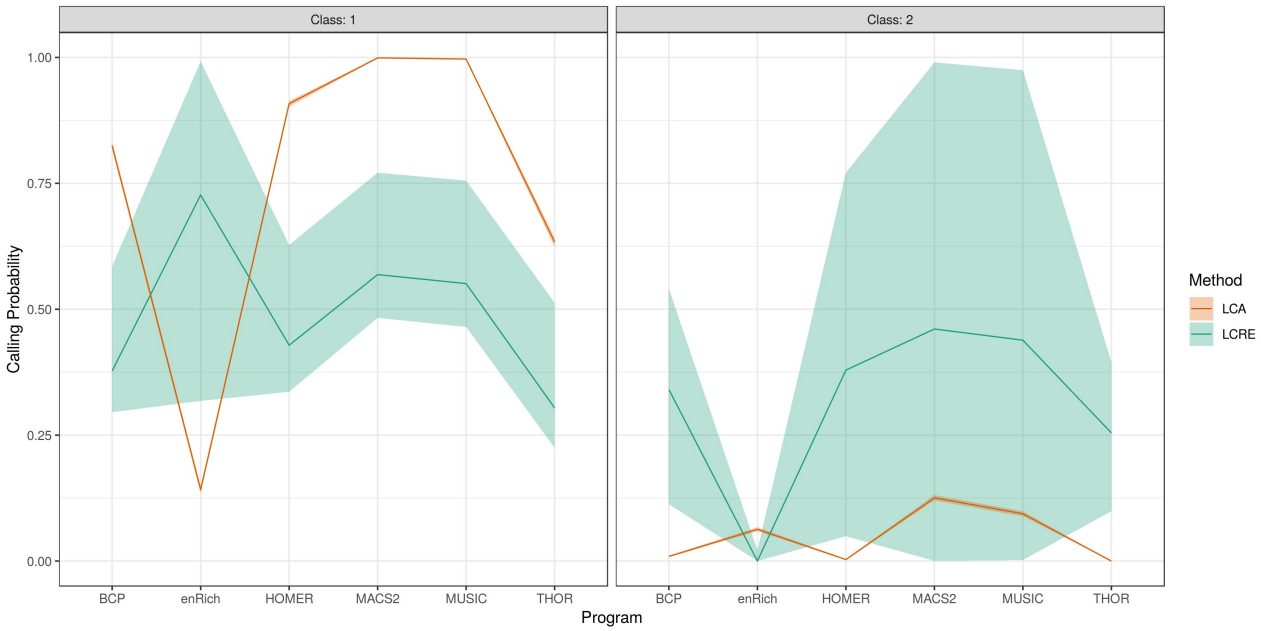


FIGURE 2.4: Calling probabilities for the LCRE with constant loading and the LCA for each program, including 2.5% and 97.5% quantiles. Ranges are shown in colour differing by model, while the line indicates the outcome value.

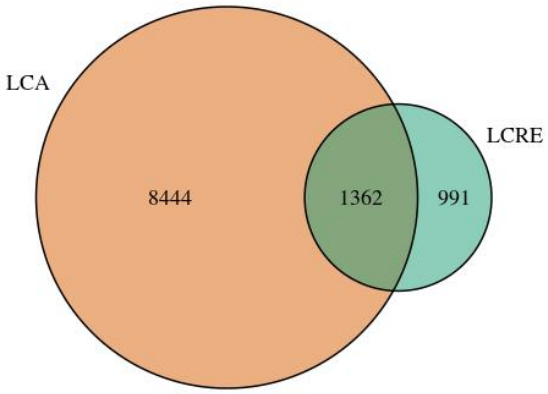


FIGURE 2.5: Venn diagram of Entrezgene IDs based on binding genes based on the LCA and LCRE models.

enRich. This was likely due to the random effect incorporated into the model lowering the influence of the other programs as they had a high agreement between them.

One Class LCRE with non-constant loading

Calling probabilities were generated for each program and class along with 95% confidence intervals, as calculated for the LCA. The results are given in Figure 2.6. As mentioned earlier, a one-class model implies that the genes did not classify into two classes. The overall low calling probability observed across the programs is likely because most of the genes were not called by any of the programs (this was also observed when using the simple LCA model). The small range indicates that the random effect had a small influence on the program calling probabilities, especially compared to 2.4.

The observed vs expected results were also calculated, and can be seen in Table

Profile	Observed	Expected	Probability of Binding	Profile	Observed	Expected	Probability of Binding
000000	11592	11508.69	0.02	100000	666	879.65	0.00
000001	266	464.23	0.00	100001	1204	1230.89	0.00
000010	43	8.55	0.00	100010	35	35.71	0.00
000011	50	17.89	0.00	100011	510	359.10	0.00
000100	0	0.11	0.00	100100	0	1.02	0.00
000101	0	0.48	0.00	100101	4	19.80	0.00
000110	0	0.02	0.00	100110	0	0.45	0.00
000111	0	0.19	0.00	100111	10	39.60	0.00
001000	0	22.93	0.00	101000	51	80.53	0.00
001001	0	40.70	0.00	101001	962	752.72	0.00
001010	1	1.24	0.00	101010	11	18.92	0.00
001011	2	8.34	0.00	101011	1458	1417.27	0.00
001100	0	0.03	0.00	101100	2	0.92	0.00
001101	1	0.38	0.00	101101	213	91.27	0.00
001110	0	0.01	0.00	101110	2	1.02	0.00
001111	2	0.36	0.00	101111	5281	5361.45	0.00
010000	704	698.62	1.00	110000	71	94.04	1.00
010001	31	57.54	1.00	110001	181	184.36	1.00
010010	1	0.15	1.00	110010	0	1.00	1.00
010011	3	0.57	1.00	110011	37	23.49	1.00
010100	0	0.00	1.00	110100	0	0.05	1.00
010101	0	0.03	1.00	110101	0	2.34	1.00
010110	0	0.00	1.00	110110	0	0.01	1.00
010111	1	0.00	1.00	110111	0	2.53	1.00
011000	0	2.10	1.00	111000	3	8.66	1.00
011001	1	5.03	1.00	111001	173	155.02	1.00
011010	1	0.03	1.00	111010	0	0.72	1.00
011011	0	0.36	1.00	111011	201	173.49	1.00
011100	0	0.00	1.00	111100	1	0.06	1.00
011101	0	0.03	1.00	111101	44	21.19	1.00
011110	0	0.00	1.00	111110	0	0.03	1.00
011111	1	0.01	1.00	111111	908	932.08	0.99

TABLE 2.5: Observed and Expected frequencies for genes called from programs from H3K36me3 data based on a two class LCRE model with constant loading. The expected frequencies demonstrate the goodness of fit of the model. The order of the programs in the profile is MACS2, enRich, HOMER, THOR, BCP, and MUSIC. The posterior probabilities indicate the probability for a gene with that profile being bound, according to the model.

2.6. Note that the probability of binding is not included, as the one-class model assumes all genes have a probability of 1 of being present in the only class within the model. The expected values in the table indicate that the fit appears improved compared to the equivalent values given by the LCA. For example, the observed number of genes that had calling profile 000000 was 11592. The LCA expected 10644.44 genes, which was almost 1000 less than actually observed, while the LCRE with non-constant loading expected 11540.72 genes, a closer fit. Similar results are seen throughout the table.

It is interesting that such an improvement in the fit was observed; based on the agreement within the data, some amount of clustering was expected. Specifically, when the observed results are revisited, some of the largest profiles are those with total agreement (000000 and 111111) or close to total agreement (101111). The results from this are not useful for the purpose of the identification of binding genes.

Since there is only one class, the binding genes and GO terms were not examined,

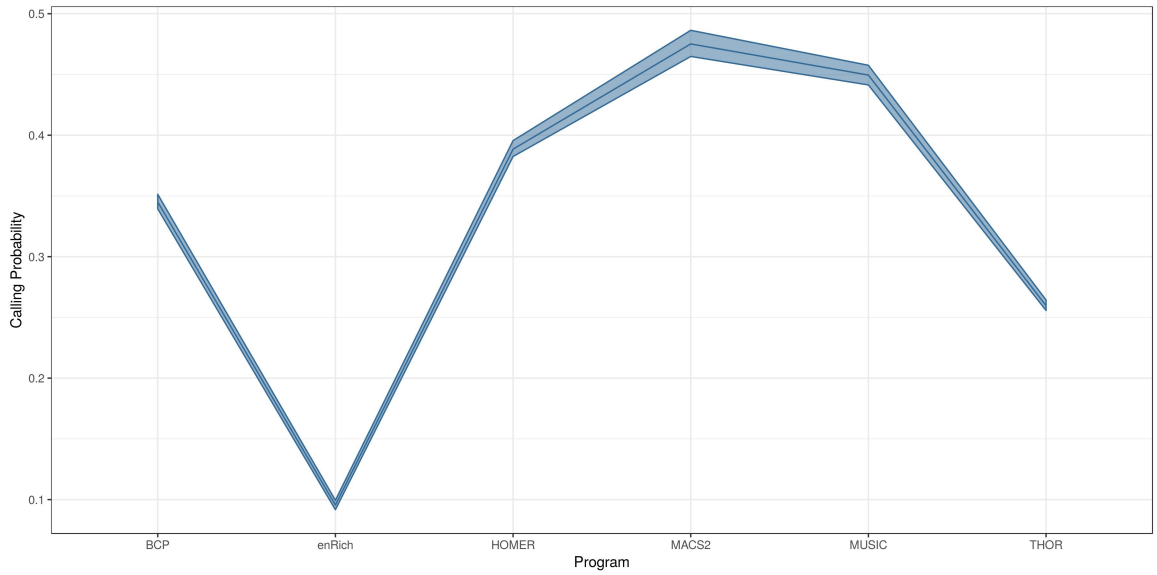


FIGURE 2.6: Calling probabilities for the LCRE without constant loading for each program, including 95% confidence intervals. The line indicates the mean calling probability for each program.

Profile	Observed	Expected	Profile	Observed	Expected	Profile	Observed	Expected
000000	11592	11540.72	010110	0	0.00	101100	2	0.13
000001	266	303.16	010111	1	0.00	101101	213	218.88
000010	43	55.23	011000	0	0.55	101110	2	0.08
000011	50	27.52	011001	1	1.13	101111	5281	5225.37
000100	0	0.02	011010	1	0.07	110000	71	68.13
000101	0	0.07	011011	0	0.26	110001	181	136.98
000110	0	0.00	011100	0	0.00	110010	0	7.04
000111	0	0.02	011101	0	0.00	110011	37	47.40
001000	0	5.40	011110	0	0.00	110100	0	0.03
001001	0	10.63	011111	1	0.00	110101	0	1.39
001010	1	0.66	100000	666	685.55	110110	0	0.01
001011	2	2.44	100001	1204	1270.23	110111	0	1.59
001100	0	0.00	100010	35	68.28	111000	3	3.41
001101	1	0.02	100011	510	421.27	111001	173	117.51
001110	0	0.00	100100	0	0.23	111010	0	0.96
001111	2	0.01	100101	4	11.74	111011	201	189.73
010000	704	749.66	100110	0	0.08	111100	1	0.01
010001	31	30.00	100111	10	13.00	111101	44	29.51
010010	1	5.25	101000	51	31.83	111110	0	0.01
010011	3	2.81	101001	962	981.82	111111	908	968.39
010100	0	0.00	101010	11	8.72			
010101	0	0.01	101011	1458	1483.05			

TABLE 2.6: Observed and Expected frequencies for LCRE without constant loading of programs from H3K36me3 data. The expected frequencies demonstrate the goodness of fit of the model. The order of the programs in the profile is MACS2, enRich, HOMER, THOR, BCP, and MUSIC.

as this would include all the genes within the genome. These results therefore would not be meaningful for identifying binding activity.

When these results are compared to the LCA and the LCRE with constant loading, the usefulness of using the BIC to determine the class of the model is questionable. Based on the result for the LCRE with non-constant loading, I would conclude that there is no binding in any of the programs, however this seems unlikely given

the number of peaks identified by the programs. Additionally, the other LCA models also preferred the two class model. Thus, the one class BIC may not be appropriate to use in this way. To assess this, I also decided to analyse the two class LCRE with non-constant loading.

Two class LCRE with non-constant loading

To compare the results of the one class LCRE with non-constant loading, I assessed the two class LCRE with non-constant loading, despite the higher BIC value. Returning to the Table 2.4, this model had a lower BIC compared to the two LCRE with constant loading models, indicating an improved fit.

Calling probabilities for each program and class with percentiles were generated, and are shown in Figure 2.7, alongside the LCRE and LCA results. The LCRE models showed very similar calling probabilities for the different programs compared to the LCA. Both gave enRich a high calling probability, and BCP, HOMER and THOR a low probability for Class 2. Similarly, MACS2 had a higher calling probability than the others in Class 1. The LCRE with non-constant loading tended to lower probabilities for Class 2 than the other two models, particularly for HOMER.

The number of genes identified in the binding class was 2320, using the Ensembl gene IDS. This was the same number of genes identified using the LCRE with constant loading, indicating that the same genes were identified. This was confirmed using Figure 2.8, as the LCRE models have coincident sets. The enRich gene set appeared to be the most influential of the 6 programs.

Similarly to the LCRE with constant loading, no significant GO terms were found, when the putative binding genes were analysed.

Table 2.7 was generated showing the observed and expected frequencies, as well as the posterior probabilities for the model. The expected results indicated a better

fit than either the LCRE with constant loading or the LCA models. For example, expected values for highly observed profiles, such as Profile 000000 were at least as close as those found by the other LCRE two class model, and the values for profiles with fewer observed genes, such as 0010000 also had lower expected values. Comparing the expected values to those found by the one class LCRE with non-constant loading indicated that the two models found similar results. The posterior probabilities were very similar to those esimated by the LCRE with constant loading, with presence or absence of enRich being the predictive factor. Thus, while the expected values fitted better, the probability of binding indicated that the model was not combining the results of the programs in the anticipated way.

To directly compare the posterior probabilities of the different models, I produced pairwise posterior probability plots, shown in Figure 2.9. These plots indicate that the posterior probabilities were polarised, as had been observed in the previous

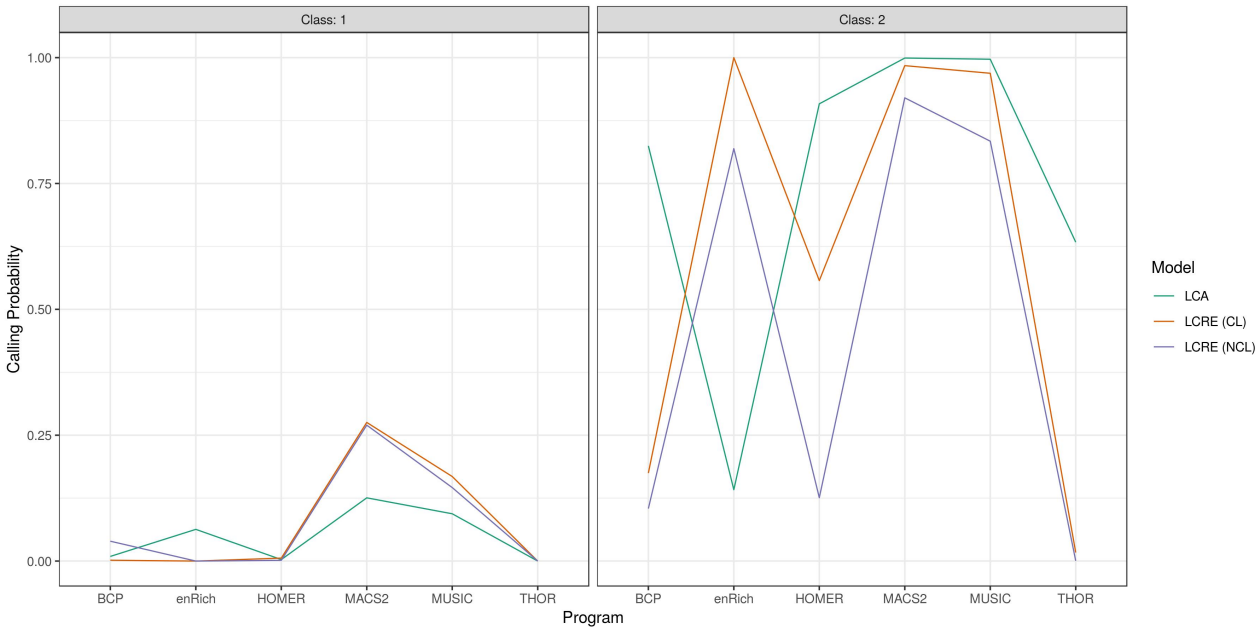


FIGURE 2.7: Calling probabilities for the 3 two class models, LCA, LCRE (CL) and LCRE (NCL), split by class. The line indicates the calling probability for each program.

Profile	Observed	Expected	Probability of Binding	Profile	Observed	Expected	Probability of Binding
000000	11592	11580.81	0.05	100000	666	680.00	0.03
000001	266	294.63	0.04	100001	1204	1232.10	0.03
000010	43	57.57	0.01	100010	35	71.47	0.01
000011	50	28.34	0.02	100011	510	432.11	0.01
000100	0	0.02	0.01	100100	0	0.26	0.01
000101	0	0.08	0.01	100101	4	12.11	0.02
000110	0	0.00	0.01	100110	0	0.10	0.01
000111	0	0.02	0.01	100111	10	14.16	0.01
001000	0	5.40	0.03	101000	51	31.39	0.03
001001	0	10.45	0.03	101001	962	927.88	0.03
001010	1	0.70	0.01	101010	11	9.10	0.01
001011	2	2.54	0.02	101011	1458	1481.74	0.01
001100	0	0.00	0.01	101100	2	0.14	0.01
001101	1	0.03	0.02	101101	213	203.77	0.02
001110	0	0.00	0.01	101110	2	0.09	0.01
001111	2	0.01	0.01	101111	5281	5289.17	0.01
010000	704	702.67	1.00	110000	71	72.80	1.00
010001	31	38.31	1.00	110001	181	174.15	1.00
010010	1	2.50	1.00	110010	0	3.79	1.00
010011	3	1.80	1.00	110011	37	35.39	1.00
010100	0	0.00	1.00	110100	0	0.02	1.00
010101	0	0.01	1.00	110101	0	1.29	1.00
010110	0	0.00	1.00	110110	0	0.00	1.00
010111	1	0.00	1.00	110111	0	0.96	1.00
011000	0	0.61	1.00	111000	3	3.63	1.00
011001	1	1.48	1.00	111001	173	174.35	1.00
011010	1	0.04	1.00	111010	0	0.56	1.00
011011	0	0.19	1.00	111011	201	192.97	1.00
011100	0	0.00	1.00	111100	1	0.01	1.00
011101	0	0.00	1.00	111101	44	42.69	1.00
011110	0	0.00	1.00	111110	0	0.00	1.00
011111	1	0.00	1.00	111111	908	911.59	1.00

TABLE 2.7: Observed and Expected frequencies for genes called from programs from H3K36me3 data based on a two class LCRE model with non-constant loading. The expected frequencies demonstrate the goodness of fit of the model. The order of the programs in the profile is MACS2, enRich, HOMER, THOR, BCP, and MUSIC. The posterior probabilities indicate the probability for a gene with that profile being bound, according to the model.

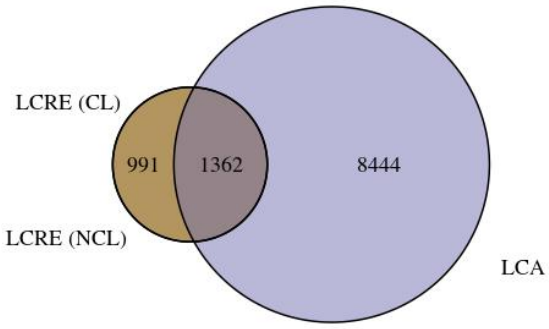


FIGURE 2.8: Venn diagram of Ensembl Gene IDs based on binding genes based on the three models (LCA, LCRE with constant loading and LCRE with non-constant loading).

posterior probability tables. The LCRE with constant loading and the LCRE without constant loading appeared to have the most similar posterior probabilities per gene; this is not suprising given that the models are similar as well. There was a number of genes which were found to have very different posterior probabilities between the LCA and the other models; these are indicated by points in the bottom-right and top-left of the plot. While there was some disagreement expected based on Figure 2.8, such an extreme difference is surprising, given these parameters were fitted using the same data. Finally, the LCA model appeared more likely to given a range of posterior probabilities, compared to the extremes of the two LCRE models. This is again expected, based on Table 2.5. Overall this figure indicates that the LCA found different results compared to the LCRE models.

Based on these results, the LCA method appeared the most promising. I used a "sum of scores" method as a comparison to the results from the LCA model for further investigation. This was performed by summing up the number of programs

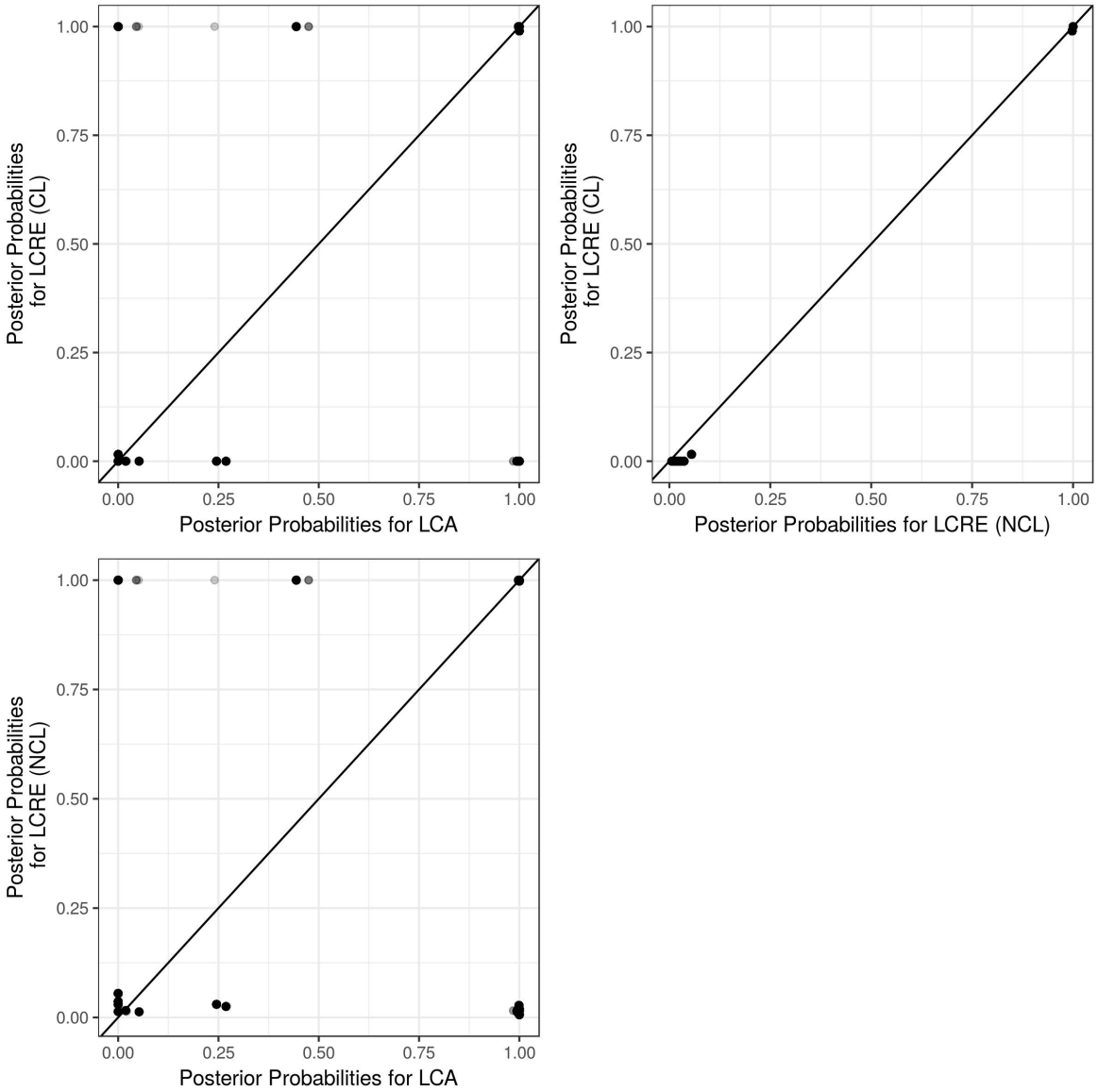


FIGURE 2.9: Pairwise plots of the posterior probabilities of genes from the three models. The posterior probabilities indicate the likelihood of a gene being bound under each of the 3 models. Points close to the $x = y$ line indicate when posterior probabilities were the same for the two models for a particular gene, indicating agreement between models. The posterior probabilities are very polarised.

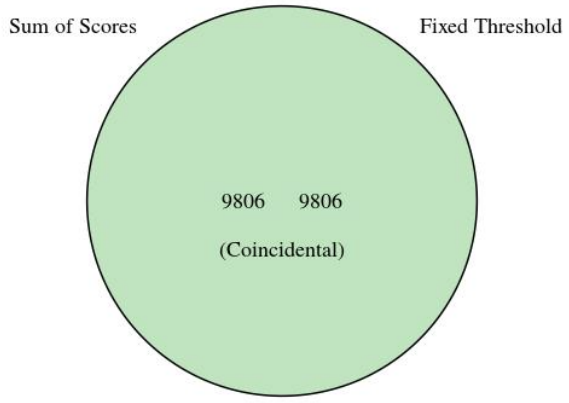


FIGURE 2.10: Venn diagram comparing the LCA gene set with the gene set generated from the sum of scores method. There is a high amount of overlap.

that found a peak associated with each gene, and dividing this by the number of programs to get a score. The scores that were greater than 0.5 were considered binding in this method. When these genes were compared to the genes found by the LCA method, it was found that there was a large amount of overlap. This was interesting because when we reconsider Table 2.3, while the number of programs that find binding is clearly an important factor, there are example of programs with high probability of binding that have fewer than 3 programs calling within the profile. One example of this is 001100, or when HOMER and THOR are the only programs that call a gene. For most of these examples, few genes are observed with that profile.

Classes	LCA	LCRE (Constant Loading)	LCRE (No Constant Loading)
1	161573	76470	75874
2	83738	75887	75900

TABLE 2.8: BIC for different LCA models and class number when not including enRich in the programs. A lower BIC is preferred.

2.3.3 LCA with a random effect: without enRich

Since the program enRich had such an influential affect on the LCRE with constant loading, I repeated the analysis without including the enRich data. This would allow a greater understanding of how much enRich affected the results. The BIC from the LCA, the LCRE with constant loading and the LCRE with non-constant loading is given in Table 2.8. The BIC indicated that for the LCA and the LCRE with constant loading a two class model was preferred, but for the LCRE non-constant loading, a one class model was still preferred. The remaining analysis compared the two class models.

The calling probabilities for the remaining 5 programs were compared side-by-side in Figure 2.11. The calling probabilities were very similar for Class 1, (which in this case was the non-binding class) but had clear differences in Class 2. Programs BCP and THOR showed very low calling probabilities for the LCRE with constant loading for Class 2, but remained relatively high for the LCA (although still lower than the other programs). This indicated that HOMER, MACS2 and MUSIC were the most influential for both models. The calling probabilities for the LCRE were similar to those observed in Figure 2.7, though the BCP had dropped in probability. This indicated that removing the program enRich did not make a significant change to this model.

Next, I compared the putative binding genes found by the LCA and the LCRE

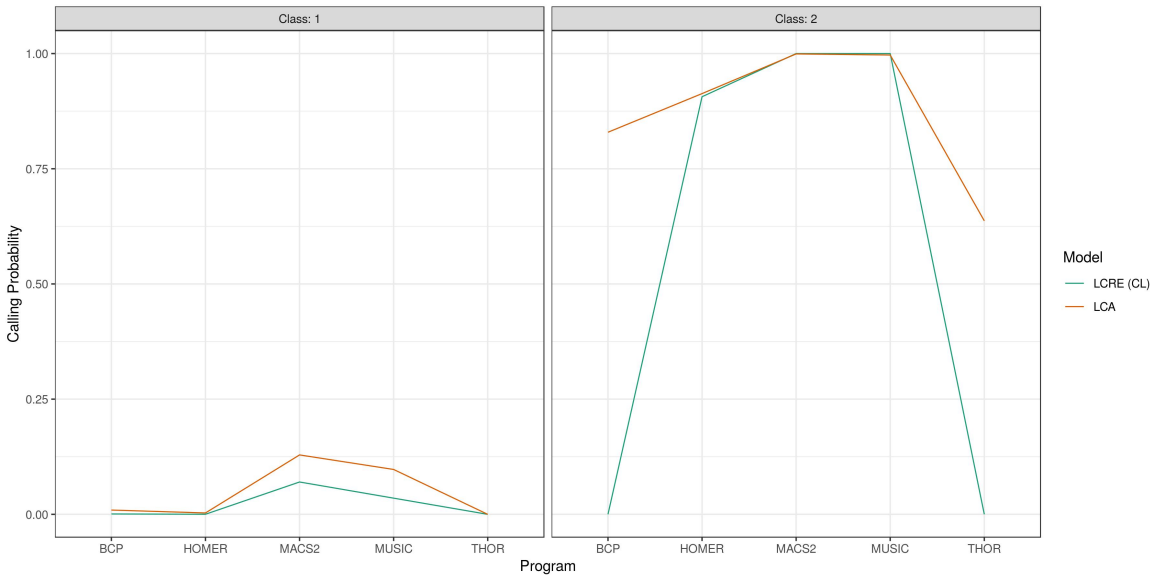


FIGURE 2.11: Calling probabilities for the LCRE and the LCA for each program. Ranges are shown in colour differing by model, while the line indicates the outcome value.

with constant loading as in previous sections. This indicated that the number of genes found by the two models were similar to the number found when the results from enRich were included. One major difference was that all of the genes found by the LCRE with constant loading were also found by the LCA. This indicated that the additional genes found previously were due to the influence of the enRich results, based on the Table 2.5.

The two LCA models, with and without using the results from the enRich program were compared using a venn diagram in Figure 2.13. It was found that there was no difference is the binding genes lists. This indicated that enRich had such a small effect on the resulting data that it made no difference to the putative binding genes. As this is the main desired outcome from this analysis, it was concluded that in terms of the LCA, the inclusion of enRich did not improve nor degrade the results.

Examining the observed vs the expected results for the LCA without enRich,

there were many similarities with Table 2.3, apart from the number of profiles, which was decreased due to the absence of enRich. This was particularly the case for the probability of binding, as expected given the results from Figure 2.13. One notable difference was that the fit of the model appeared improved, although this was in fact due to changes in the observed counts rather than changes to the expected.

Finally, I also calculated a new gene list based on a sum of scores method without enRich, and compared this to the results from LCA in Figure 2.14. Surprisingly, this indicated that the gene lists were also coincident. Returning to Table 2.9, this again would not always occur according to the probability of binding; for example, 00110 has a probability of binding of 0.98 but would not be considered binding using the sum of scores method. While in this case the two methods perform comparatively, this may be an irregular occurrence, raising questions regarding the similarity of the results from these two methods.

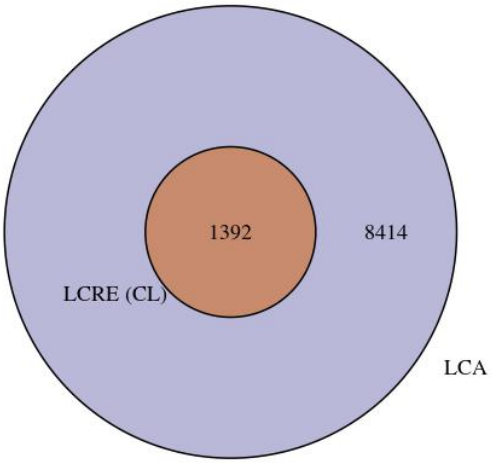


FIGURE 2.12: Venn diagram of the binding genes found by the models LCA and LCRE with constant loading, based on the Entrezgene ID. The genes found by the LCRE with constant loading model were also found by the LCA model.

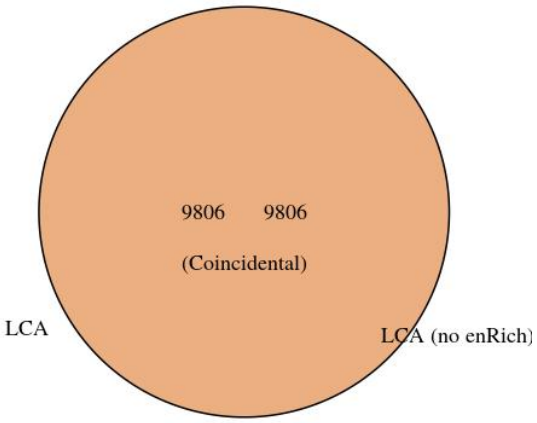


FIGURE 2.13: Ven diagram of the binding genes found by the LCA model with or without using the results from the program enRich using the Entrezgene ID. The genes are the same, and are therefore labelled "coincidental".

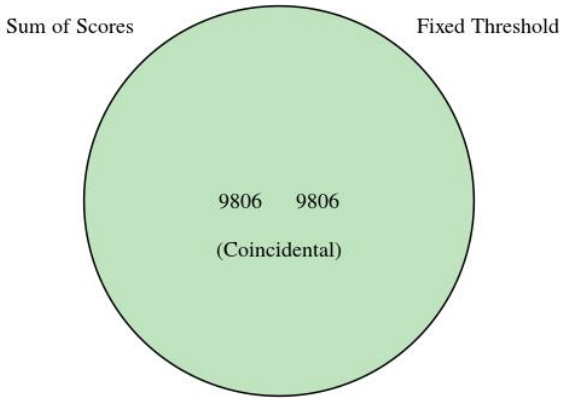


FIGURE 2.14: Venn diagram of the binding genes found by the LCA model without using the results from the program enRich and the gene set generated using the sum of scores method using Entrezgene IDs. The genes are the same, and are therefore labelled "coincident".

Profile	Observed	Expected	Probability of Binding	Profile	Observed	Expected	Probability of Binding
00000	12296	11316.09	0.00	10000	737	1675.18	0.00
00001	297	1220.76	0.00	10001	1385	235.15	0.23
00010	44	105.64	0.00	10010	35	16.48	0.05
00011	53	11.61	0.02	10011	547	266.19	0.99
00100	0	0.00	0.11	10100	0	0.30	1.00
00101	0	0.08	1.00	10101	4	95.55	1.00
00110	0	0.00	0.98	10110	0	1.47	1.00
00111	1	0.37	1.00	10111	10	464.11	1.00
01000	0	34.03	0.00	11000	54	6.86	0.27
01001	1	4.13	0.11	11001	1135	573.16	1.00
01010	2	0.32	0.02	11010	11	8.89	0.99
01011	2	2.25	0.98	11011	1659	2781.26	1.00
01100	0	0.00	1.00	11100	3	3.19	1.00
01101	1	0.80	1.00	11101	257	1004.71	1.00
01110	0	0.01	1.00	11110	2	15.51	1.00
01111	3	3.90	1.00	11111	6189	4880.00	1.00

TABLE 2.9: Observed and Expected frequencies for genes called from programs from H3K36me3 data based on a two class LCA model, calculated without using the results from the program enRich. The expected frequencies demonstrate the goodness of fit of the model. The order of the programs in the profile is MACS2, HOMER, THOR, BCP, and MUSIC. The posterior probabilities indicate the probability for a gene with that profile being bound, according to the model.

2.4 Conclusions

The LCA appeared to perform best of the three models considered. Initially, six programs were used to generate the ChIP-seq data-set. While there was a good level of agreement for most of the programs, enRich had a very low level of agreement with the other programs, and had the smallest number of total genes. The BIC was used to identify when the two class model was appropriate, and this led to the assessment of three models; the two class LCA, the two class LCRE with constant loading, and the one class LCRE without constant loading. While these models were assessed, additionally the two class LCRE with non-constant loading was also analysed, as the BIC was still competitive with the other models.

Of the three models considered, the LCA had the most consistent posterior probabilities, as well as the most confident calling probabilities for the 6 programs. This model also generated GO terms consistent with the original binding protein, a marker of transcriptionally active genes. In contrast, the two LCRE models had better expected values compared to the observed. The one class model was not assessed further, as the posterior probabilities were the same for all genes (as they belonged to the same single class). The LCRE with constant loading and the LCRE with non-constant loading models had very similar results, and did not combine the results of the programs in the way anticipated. Instead, enRich appeared to be the dominant factor in determining gene classification.

The method "sum of scores" was used as a simplistic method of using the level of agreement between programs to generate a result. This was compared in particular to the simple LCA as this appeared to be the best performing model. There was a high degree of similarity between the two in terms of the putative binding genes lists.

Since enRich appeared influential, I removed this program from the results and repeated the analysis. This had a small effect on the results, and in particular the LCA was almost unaffected, particularly when the binding genes were examined. In terms of BIC, the LCRE with non-constant loading still preferred a one-class model, so the other models were investigated only. The putative binding genes list was unaffected by the change in programs included. The LCRE models were more affected; removing enRich reduced the number of genes found, and made the model largely dependent on HOMER, MACS and MUSIC. A new gene list was generated using the sum of scores method but while not including enRich. This was found to find the same genes as the LCA. Based on the probability of binding of the different profiles for the LCA model without enRich, this may not always be the case.

The results of the three models raised questions about the validity of using the BIC to determine appropriate fit, as well as how appropriate the use of the LCRE model is over the simpler LCA model. Furthermore, the results of the LCRE models indicate undesirable sensitivity to programs with small gene sets. The sum of scores method proved to be competitive with the LCA model, however it was unclear which circumstances this occurred under. Additional research was needed to resolve these issues and develop new techniques for more robust identification of putative binding genes.

Chapter 3

Simulation Study

3.1 Introduction

To understand how the different LCA models are affected by changes to the underlying data, I performed a series of simulations. The simulations changed a range of conditions in the underlying data, and allowed the performance of the different models to be measured. Ideally, a better understanding of these factors will inform the application of LCA to real data.

I used the simulated data to investigate the LCA models and understand the results in Chapter 2. Firstly, using the underlying knowledge of the simulated data, I compared the three models using different measures. This was used to determine the most accurate model for each of the scenarios. Next, I investigated the BIC, both as a means to determine preferred class number for the model, and as a method to compare the three models to each other when the true answer is not known. Using the conclusions from each of these investigations, I was able to determine the best ways of analysing ChIP-seq data using LCA.

3.2 Methods

Using R, I generated test data for the three different models (LCA, LCRE with constant loading, LCRE with non-constant loading) over a number of different scenarios.

3.2.1 Simulating ChIP-seq data

In order to generate data, the simulation uses the concept of a score from which a binary threshold is derived. This is a simple model of how real ChIP-seq data is also generated. All of the programs will identify a great number of putative peaks, but some will be considered unlikely to be actual peaks given the profile of the surrounding noise. Thus, some sort of filter is required to reduce the number of peaks to be more manageable. Every program used in this study, and almost all of the tools considered for ChIP-seq peak identification, will label the peaks with some sort of score. For MACS2 and THOR this score is a $-log_{10}$ p-value, while for BCP and HOMER the p-value is used. MUSIC uses the q-value as a score. These scores are given to each peak at part of the calculations within the programs, and this is used later on in order to reduce the number of putative peaks given to the user, and thus acts as a threshold for the "best" putative peaks. The user can set a particular score threshold by specifying, for example, the maximum p-value allowed for the peaks.

Ideally, I would be able to use these score directly to gain a better idea of the peaks, and use some sort of clustering classification method such as the Multivariate Gaussian Mixture model (MGMM). However, the peak scores for all of the peaks identified by each program are difficult to access, and are influenced by the threshold set. For example, when peaks are close together, the programs may combine the

peaks into one peak region, so a higher threshold may in fact reduce the number of peaks. Furthermore, because I am using genes rather than peaks as a means of comparing multiple programs, I would ideally have a score for each gene, based on the peak scores. However, some genes may still remain without a score, if no peak is found to be associated with it. Thus this investigation was beyond the scope of this thesis. However, this concept will be used with the generation the simulated data. In this case, the score will be directly associated with the genes, and a threshold will be used to generate binary data.

3.2.2 Generation of Test Data

The number of "genes" was set at $n = 3000$, and the number of peak calling programs was given as p. The proportion of binding genes was given by p_0, and was used to calculate the number of binding and non-binding genes for the data set; $n_1 = \lfloor p_0 n \rfloor$ and $n_2 = 1 - n_1$ respectively.

I made some assumptions about the underlying model of the data, in particular how scores for binding and non-binding genes may differ. If the genes were binding, the scores were randomly sampled from a normal distribution with mean of δ, and a standard deviation of 1 ($\mathbf{X_1} \sim N_p(\delta, 1)$), while non-binding gene scores were randomly sampled from a normal distribution with a mean of $-\delta$ and a standard deviation of 1 ($\mathbf{X_2} \sim N(-\delta, 1)$). Scores were calculated for each gene and each program, resulting in $3000p$ scores. These were allocated into lists for each program, such that each program had n_1 binding genes and n_2 non-binding genes. The randomly sampled scores were then listed together as X_{ij}, where i is the program number and j is the gene number.

The random effect Z was generated from a normal distribution with mean 0 and standard deviation σ_z, and for each simulation a distribution $B \sim U(0, 2)$ was used

to randomly determine the coefficients for the random effects for each program. The coefficients determine how much of the random effect to add to each program. This represents how each program will be more or less affected by the same random effect. If two or more programs are strongly affected by the random effect, this will result in a higher correlation between the two programs. The resulting values from adding the random effect are thus:

$$W_{ij} = X_{ij} + Z_{ij}b_j \quad i = 1, ..., p \quad \text{and} \quad j = 1...n$$

The resulting binary outcomes for each program were then calculated using a threshold T_i:

$$Y_{i,j} = \begin{cases} 1 & \text{if } W_{i,j} > T_i \\ 0 & \text{otherwise} \end{cases} \qquad i = 1, ..., p \quad \text{and} \quad j = 1, ..., n$$

To create different scenarios, a number of parameters were varied. The parameters and the values tested are given in Table 3.1. The LCA model was hypothesised to be the best model when σ_z is 0, while the LCRE models was hypothesised as the better model when σ_z was greater than 0. This is because a high value of σ_z increases the correlation between programs. The program threshold parameter, T_i, was also changed. This affected the threshold scores for each program to call a gene as binding or otherwise; constant keeps the threshold score at 0, while varied changed the threshold score to:

$$T_i = \frac{2i}{i} - 1 \qquad i = 1, ..., p$$

Such that the thresholds were evenly spread for all programs. All combinations were tested. In addition, data with only one cluster was generated by setting δ to 0

Parameter	Description	Scenario Values
p	Number of programs	{5, 7}
δ	Difference between binding and non-binding scores	{0.5, 1.5}
σ_z	Level of random effect for scores	{0, 0.5, 1, 2, 4}
p_0	Proportion of binding genes	{0.1, 0.3, 0.5, 0.7, 0.9}
Program thresholds	Whether programs had the same threshold or constant thresholds	{Constant, Varied}

TABLE 3.1: Values for the parameters of the simulation in the different scenarios. For each scenario, one of each of the values is selected for each parameter. All combination of scenarios were tested once. In total 200 different scenarios were tested.

while changing the other parameters as shown in Table 3.1.

After I observed that there was a high variability for the same parameters for both the LCA and the LCRE due to the randomness of both the scores and the random effect, I repeated each simulation scenario 20 times to obtain an average correlation, as well as a standard deviation.

3.2.3 Model Fit of Simulation Data

The matrix $Y_{i,j}$ was fit to six different models; an LCA, an LCRE with constant loading, and an LCRE with non-constant loading, with both 1 class and 2 classes. The posterior probabilities for being present in Class 1 were obtained for the two class models. These were associated with the original genes. In a few cases, the model fitting procedure failed as the adaptive Gauss Hermite quadrature did not converge for some replications in some scenarios. When this occurred, posterior probabilities could not be obtained for these replications, and were not used for further analysis.

An MGMM was used to fit the original scores in W_{ij}. This model categorises data points into a set number of clusters, where each cluster is made up of points that are randomly sampled from a Gaussian distribution. Since the MGMM makes full use of the underlying scores of each gene, the posterior probabilities the model generates are a "gold standard" to which the posterior probabilities of the LCA and LCRE can be compared. After fitting the data, I obtained the posterior probability for each gene of being in one of the clusters.

3.2.4 Method Assesment

The correlation between the posterior probabilities of the three LCA models and the MGMM across the genes were calculated. A higher correlation indicates a better fit to the data. An average correlation for each scenario was calculated using the results from the 20 replications. These results were compared for each model for each scenario, and was used as an assessment of the accuracy of the posterior probabilities.

Another statistic calculated was the root mean square error (RMSE) between posterior probabilities of the MGMM and the model:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{p}_{M,i} - \hat{p}_{L,i})^2}$$

Where $\hat{p_{X,j}}$ are the posterior probabilities for model X, M is the MGMM model and L is one of the three LCA models being tested. This was used to confirm that the posterior probabilities were truly similar between the MGMM and the LCA and LCRE, since the correlation would still be high even if the posterior probabilities were different, as long as the difference in the posterior probability remained the same across all of the genes. This was calculated using one of the replicates.

To assess how changes in the scenario parameters might affect whether a one or two class scenario was observed, I calculated the BIC for all replicates and scenarios, and then averaged the BIC across the replicates. The BIC was used to compare the different models over the different scenarios. This was useful as these measurements do not rely on the MGMM, and thus give an indication of how correlated they are to a good posterior probability. For example, if a model with a poor posterior probability resulted in a good fit or the best BIC, this indicated that testing these for real data is not necessarily informative.

A "sum of scores" approach was also performed, and the correlation with MGMM calculated, as a simple method to compare to the other models. To find the sum of scores, the outcome of each program was summed together, to get a score for each gene:

$$S_j = \sum_{i=1}^{p} Y_{i,j}$$

Thus, this calculates "votes" from each program directly, as was done during the analysis of the ChIP-seq data in Chapter 2. This was performed for one replicate of each scenario, similar to the RMSE. Note that one difference between this method and that in Chapter 2 was that the score was not divided by the total number of programs in this case, as the sum of scores is used to find a correlation with the MGMM, rather than to calculate the number of binding genes.

I also investigated how often the BIC preferred the one class model. I analysed the same data using a one class model for the LCA, and the two LCRE models and calculated the BIC for each. The difference between the BIC for two class model and the one class model was then calculated to generate ΔBIC. This was then assessed across all 20 simulations to check how often a one class model was preferred over a two class model for the same data.

3.3 Results

3.3.1 Comparing the Simple LCA and LCRE models

Correlation to MGMM

Initially, the correlation of the posterior probabilities for each gene was compared for the three methods; LCA, LCRE with constant loading and the LCRE with non-constant loading, without considering the effect of averaging over different threshold values. A higher correlation to the MGMM was considered to indicate a closer fit. An example of the results for one scenario is given in Table 3.2 with parameters $\{p, \delta, \sigma, p_0, \text{Program Threshold}\} = \{5, 0.5, 0, 0.1, \text{Varied}\}$ (Scenario 1).

Full results, including the correlation tables for each scenario, can be found in Appendix B.1.

To compare the three methods, pairwise scatter plots were generated, comparing the average correlation compared to the MGMM for each scenario. This resulted in the three graphs in Figure 3.1. Each correlation is coloured based on the value of σ_z. The LCRE models were expected to show the most improvement over the LCA model when σ_z was high.

Most of the time, the models showed a high correlation to the MGMM, indicated by having a correlation above 0.75. Furthermore, the LCA was more likely to have high correlations, as there are a greater number of points found above 0.75 for the LCA compared to the other models. Overall, under low values of σ_z, all of the models were competitive, as most of the points with σ_z at 0 to 0.5 lie close to the $x = y$ line. Notably, these values also have some of the highest correlations for any of the models. This may be because there is less likely to be overlap between the scores of the binding and non-binding genes when the random effect is low for all of the programs, allowing all the models to more correctly classify the genes.

	LCA	LCRE (Constant Loading)	LCRE (Non-Constant Loading)
	0.76	0.74	0.72
	0.74	0.74	0.74
	0.71	0.64	0.63
	0.76	0.76	0.72
	0.75	0.33	0.55
	0.77	0.38	0.05
	0.76	0.68	0.62
	0.75	0.75	0.75
	0.75	0.75	0.75
	0.77	0.31	0.03
	0.76	0.35	0.06
	0.71	0.68	0.59
	0.75	0.55	0.75
	0.74	0.74	0.74
	0.74	0.74	0.74
	0.77	0.22	0.08
	0.74	0.45	0.06
	0.74	0.74	0.74
	0.70	0.70	0.70
	0.75	0.25	0.01
Average	**0.75**	**0.57**	**0.50**
Standard Deviation	**0.02**	**0.19**	**0.30**

TABLE 3.2: Correlation results for the LCA and LCRE (constant loading) and LCRE (non-constant loading) for Scenario 1. Correlation compares the posterior probability for all genes for the LCA and LCRE to the posterior probability for all genes for the MGMM. The average and standard deviations were used to compare Scenarios.

When σ_z was high, at 2 or 4, the LCRE with non-constant loading tended to perform the best over both of the models. These values also tended to have lower correlation to the MGMM, especially for the LCA and the LCRE with constant loading model. However, for smaller values, the LCRE with non-constant loading model was more likely to be outperformed by the LCA model. In some cases, the LCA model performs much better than either model, even for very high values of σ_z. For those points, the other parameters in those scenarios may indicate why that is the case.

To identify if there was any other trend to which scenarios had higher correlations within the models, I collated the results into Table 3.3. This only included scenarios where the difference between the parameters was greater than 0.01, and anything smaller was considered an equivalent correlation (and the scenario the methods are competitive). In total, 113 of the 200 scenarios had a model that performed significantly better. Most of these scenarios found that the LCA was the best model (74), followed by the LCRE with no contant loading (34).

The parameters can be examined to identify influential factors for the correlations. The number of programs, the proportion of binding sites (p and p_0, respectively) and the program thresholds had approximately equal number of scenarios within each row, indicating that these parameters did not affect the correlation to the MGMM for any of the models.

There is a clear difference in the frequencies of δ and σ_z for the three models. As observed in Figure 3.1, the LCA model performed better when the values of σ_z were low (0,0.5,1), while the LCRE with non-constant loading performed better when the values of σ_Z were high (2,4). The parameter option with the lowest frequency for σ_z was 0, which was expected; when there was no random effect all of the models should perform equally well.

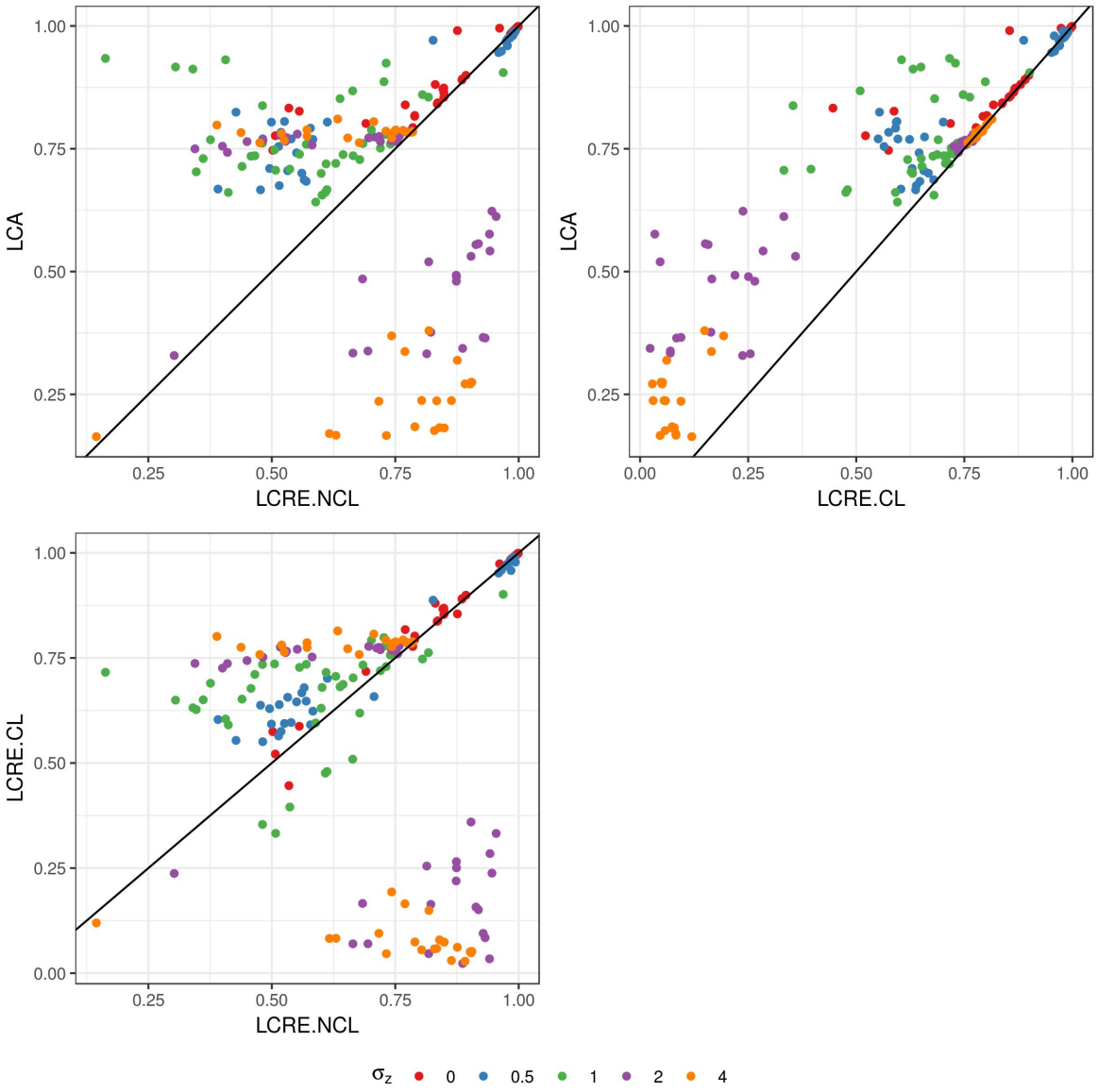


FIGURE 3.1: Average correlation to the MGMM model for models LCA, LCRE (constant loading) and LCRE (non-constant loading) for different scenarios, coloured by degree of random effect (σ). A higher value on both axes indicates a better fit to the MGMM. Points above the $x = y$ line indicate that the model on the y-axis performed better, while points below indicate that the model on the x-axis performed better. When the points lie close to the $x = y$ line the models were competitive. When the value of σ_z were high, the LCRE (non-constant loading) performed better; otherwise the LCA performed better.

The value of δ also influenced the model performance; when $\delta = 0.5$, the LCA model tended to perform better, while when $\delta = 1.5$, the LCRE with non-constant loading performed better. Since δ controls the level of overlap between the binding and non-binding genes, this indicated that the LCA was better able to distinguish the classification when the clustering was less clear. When δ was large, there was a greater correlation between the programs across different genes (since two genes that are both either binding or non-binding are more likely to be identified by multiple programs as binding). This may explain why the LCRE with non-constant loading was able to perform better in these scenarios.

I also investigated why some of the scenarios with very high values of σ_z had the best correlations when the model LCA was used. These are given in Table 3.4. The LCA was the best model to use in terms of correlation if both the number of programs was higher and δ was low, or that the number of programs was lower and δ was high. These are scenarios where the classification of the clusters was more distinct. The results from Table 3.4 showed that while the trends observed based on Table 3.3 are useful, they do not hold for all scenarios.

RMSE

In order to directly compare the LCA and LCRE models, the RMSE for each of the scenarios is given in pairwise plots in Figure 3.2. The RMSE looks at the actual difference per gene of the posterior probability found by the MGMM and the different LCA models. A lower value indicates a smaller difference between the posterior probabilities. In general, most of the points lie close to the $x = y$ line in the graphs. This implies that there were similar differences to the MGMM posterior probabilities across all three models. Interestingly, the LCA and the LCRE with constant loading appeared to be the most similar, even though in Figure 3.1, the two LCRE

	p		δ		σ_z					p_0					Program Thresholds		**Total**
	5	7	0.5	1.5	0	0.5	1	2	4	0.1	0.3	0.5	0.7	0.9	Constant	Varied	
LCA	37	39	53	23	10	21	34	8	3	16	14	17	16	13	48	36	**76**
LCRE (Constant Loading)	5	0	4	1	0	0	1	2	2	0	2	0	0	3	4	1	**5**
LCRE (Non-Constant Loading)	23	22	0	45	0	6	1	19	19	8	9	9	8	11	22	23	**45**

TABLE 3.3: Frequencies of scenarios with the best correlation (rounded to 2 decimal places) to the MGMM across the 3 models. The number of scenarios with particular parameter values are given in each column. Columns are grouped by parameter type. Total number of scenarios with a higher correlation is given at the end.

LCA	LCRE (Constant Loading)	LCRE (Non-Constant Loading)	p	δ	σ_z	p_0	Program Threshold
0.3290255	0.2371867	0.3022940	5	1.5	2	0.1	Varied
0.7575153	0.7523142	0.5816146	7	0.5	2	0.1	Varied
0.7645865	0.7437737	0.4498586	7	0.5	2	0.3	Varied
0.7550873	0.7258098	0.4000522	7	0.5	2	0.5	Varied
0.7498009	0.7369426	0.3443155	7	0.5	2	0.7	Varied
0.7798217	0.7706741	0.5516350	7	0.5	2	0.1	Constant
0.7674148	0.7516951	0.4820158	7	0.5	2	0.5	Constant
0.7650746	0.7626214	0.5278362	7	0.5	2	0.9	Constant
0.1639648	0.1195358	0.1446827	5	1.5	4	0.1	Varied
0.7756224	0.7747874	0.5714927	7	0.5	4	0.3	Constant
0.7669850	0.7641006	0.5252295	7	0.5	4	0.7	Constant

TABLE 3.4: Parameter details for scenarios where the value of σ_z was high but LCA had the highest correlation to MGMM.

models had more similar correlations overall. Most of the RMSE values appeared to be less than 0.4, although could be as high as 0.8 for some of the scenarios. The LCRE models tended to have a greater range of RMSE values, especially the model with constant loading.

The smaller values of σ_z generally had very low differences across all three methods, which is consistent with the results found in Figure 3.1. Similarly, for high values of σ_z, the LCRE with non-constant loading performed best. For medium values of σ_z, such as 1 and 0.5, the LCA tended to outperform the other two models. Again, differences between the scenarios other than the random effect must cause of some of the differences observed, in particular for those scenarios that had high values of σ_z but had similar RSMEs for all models.

To identify trends in the RMSE, I generated a new table similar to Table 3.3. The results are given in Table 3.5. In total, 140 of the 200 scenarios had a lower RMSE in one of the models compared to the others. More scenarios had a lower RMSE in the LCRE with non-constant loading model (72) compared to the LCRE with constant loading (23) and the LCA (45) models. This reversal in the performance frequencies was surprising because the LCA appeared competitive in Figure 3.5.

The most influential parameters observed in Table 3.3 were also observed in Table 3.5. The parameters that had the biggest effect on the model performance were σ_z and δ. The LCA and LCRE with non-constant loading models were better able to perform with low values of σ_z, while the LCRE with constant loading model performed best under high values of σ_z. Specifically, the LCRE with non-constant loading model performed best under the middle range σ_z values (0.5-2), perhaps because the loading in this case had less of an effect on the final binding or not binding classification of the genes.

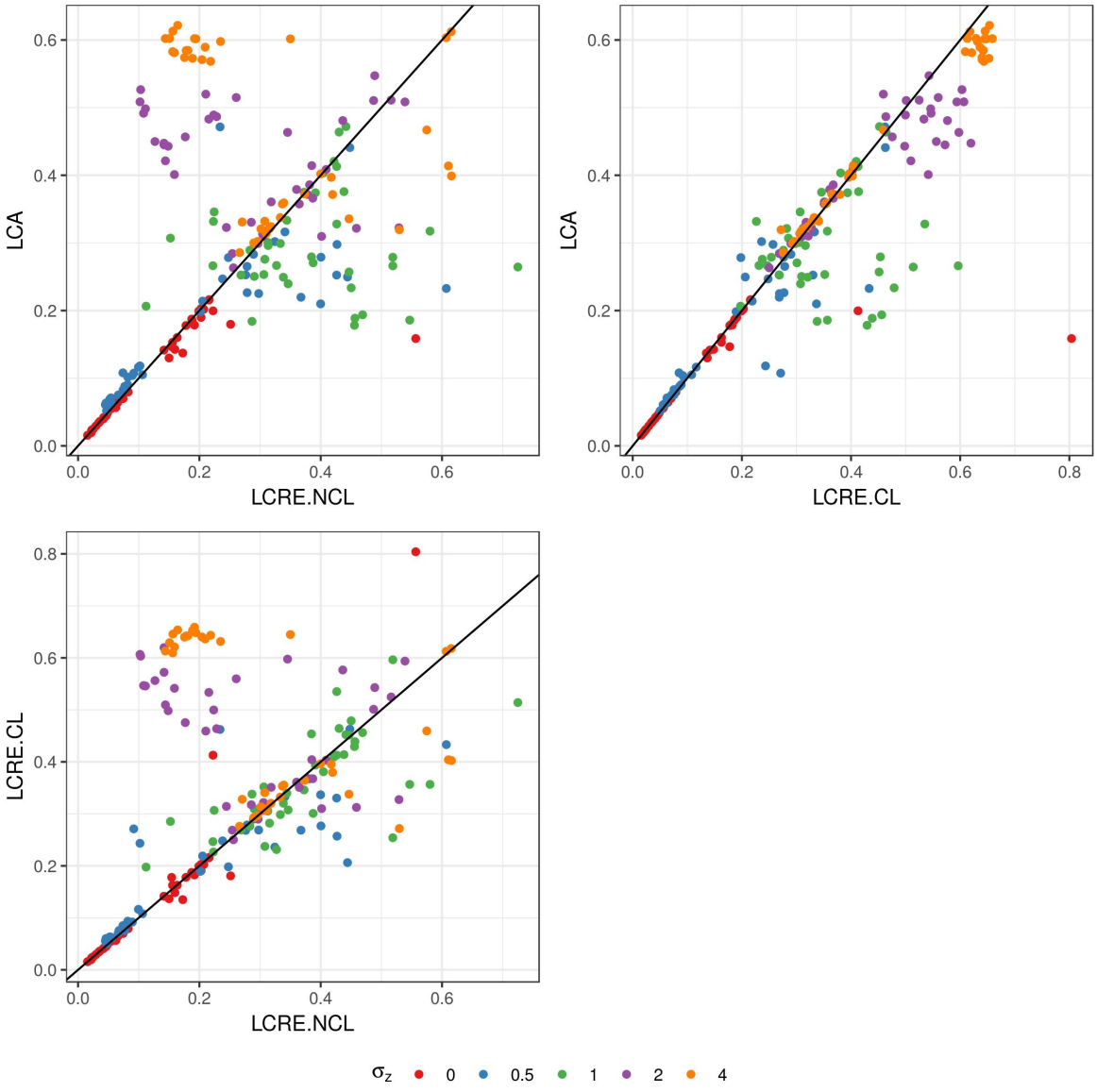


FIGURE 3.2: RMSE for the posterior probabilities for the MGMM model for LCA and LCRE (constant loading) and the LCRE (non-constant loading). A smaller value indicates a lower RMSE, which is preferred. Points above the $y = x$ line indicate scenarios where the model on the x-axis performed better while points below indicate that the model on could be expected that the correlation and the RMSE would give similar results, but the y-axis performed better. For most of the scenarios, the models appear equivalent (points lie close to the $x = y$ line).

	p		δ		σ_z					p_0					Program Thresholds		**Total**
	5	7	0.5	1.5	0	0.5	1	2	4	0.1	0.3	0.5	0.7	0.9	Constant	Varied	
LCA	19	26	26	19	5	11	23	3	3	8	8	11	11	7	23	22	**45**
LCRE (Constant Loading)	12	11	20	3	1	5	7	6	4	4	3	6	5	5	9	14	**23**
LCRE (Non-Constant Loading)	35	37	21	51	0	16	7	26	23	17	15	12	11	17	38	34	**72**

TABLE 3.5: Frequencies of scenarios that found a lower RMSE (rounded to 2 decimal places) to the MGMM across the 3 models. The number of scenarios with particular parameters are given in each column. Columns are grouped by parameter type. Total number of scenarios with a higher correlation is given at the end.

BIC

I obtained the average BIC for the 20 replicates in each scenario to determine the effect the different parameters had on the BIC, and which models performed best. These are shown in a pair-wise fashion in Figure 3.3. For most of the scenarios, the values appeared similar as the points lay close to the $x = y$ line. This is in part because the scenario points are distributed across a wide range of BIC values, so the differences are small in comparison. Thus, any differences that are noticeable indicated a large actual difference in BIC. There did not appear to be any clear trend in the BIC value and the value of σ_z.

A lower value for the BIC is desirable, so the LCRE with non-constant loading and the LCRE with constant loading appear to perform better than the LCA. This was generally the case for high values of σ_z, which was consistent with the results from the correlation to MGMM and the RMSE. The two LCRE models appeared to be competitive, with most points lying very close to the $x = y$ in the pairwise plot of LCRE (CL) and LCRE (NCL).

The frequencies of the scenarios with significantly lower BIC are summarised in

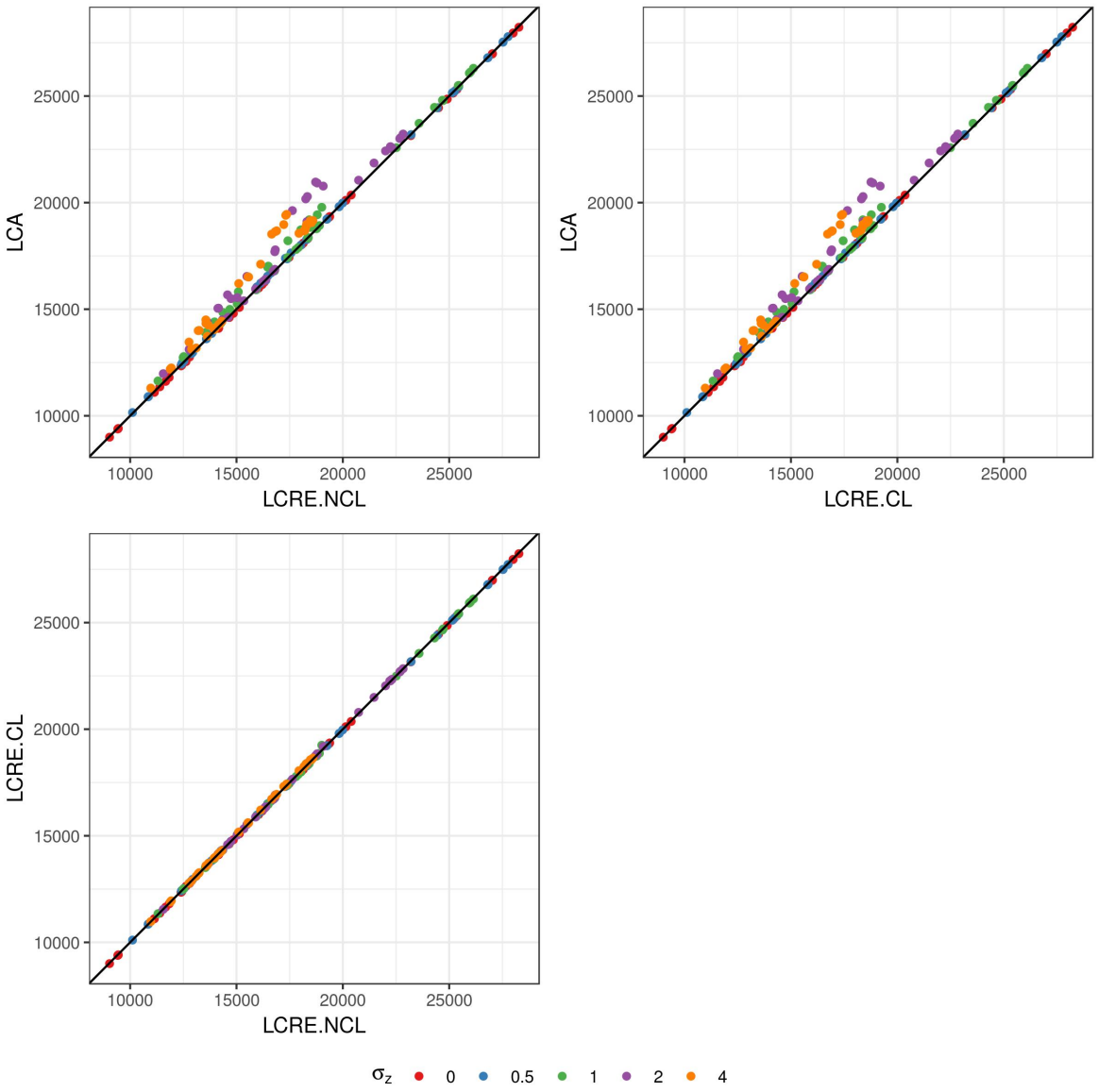


FIGURE 3.3: Pairwise plots of the average BIC for 20 replicates of each scenarios for the LCA, LCRE with constant loading and LCRE with non-constant loading models. A smaller BIC is desirable, so points that lie above the $y = x$ line indicate that the model on the x-axis is preferred, while points that lie below the $y = x$ line indicate that the model on the y-axis is preferred. Relative to the other scenarios, BICs for each individual scenario are similar.

Table 3.6. This indicated that 186 of the 200 scenarios had a better BIC for one model compared to the others. This was mostly found to be either of the two LCRE models with 68 and 77 scenarios preferring the constant loading and non-constant loading model, respectively. Similar trends in the parameters were observed compared to the correlation and RMSE tables. The parameters that appeared to influence the BIC are σ_z, δ and p, with the other parameters having similar numbers across the three models. The value of σ_z was 0 for almost all of the scenarios where the LCA was preferred, and was greater than 0 for the LCRE with constant loading and for LCRE with non-constant loading. These results was originally hypothesised based on the model parameters and indicates that, in terms of model fit, the models are working as expected. Parameters p and δ behaved similarly, with LCRE with constant loading model preferred for low values and the LCRE with non-constant loading model preferred for high values.

	p		δ		σ_z					p_0					Program Thresholds		**Total**
	5	7	0.5	1.5	0	0.5	1	2	4	0.1	0.3	0.5	0.7	0.9	Constant	Varied	
LCA	21	20	21	20	40	1	0	0	0	9	8	8	8	8	21	20	**41**
LCRE (Constant Loading)	43	25	51	17	0	18	31	13	6	13	13	17	14	11	34	34	**68**
LCRE (Non-Constant Loading)	24	53	22	55	0	15	6	25	31	15	14	15	17	16	39	38	**77**

TABLE 3.6: Frequencies of scenarios that found a significantly lower BIC (difference greater than 3) to the other 2 models. The number of scenarios with particular parameters are given in each column. Columns are grouped by parameter type. Total number of scenarios with a lower BIC associated with each model is given at the end.

Sum of Scores

In addition to the other measures, I calculated the correlation of the results of the sum of scores method (described in Methods) to the posterior probabilities of the

MGMM. These correlations were compared to the correlations of the other 3 methods. The sum of scores was used as a simplistic method, where positive responses for each program are counted, and the higher the number of positive responses the greater the likelihood a gene is binding. Thus the level of improvement given by the 3 LCA models over this method could be investigated. It should be noted, however, that the sum of scores should not be considered an alternative method, because it does not generate posterior probabilities.

The pairwise correlations are given in Figure 3.4. The LCA models gave an improved result for many scenarios, and this appeared to be largely influenced by the value of σ_z. When the value of σ_z was 0 or 0.5, all of the models showed a moderate improvement over the sum of scores approach. However, for medium to large values of σ_z, the result varied depending on the LCA model. The simple LCA model performed best for many of the moderate σ_z scenarios, but performed worse for the scenarios with higher (2 and 4) values of σ_z. This was consistent with the previous performance of the LCA. In contrast, the LCRE models performed poorly compared to the sum of scores for most scenarios where σ_z was equal to 1. The LCRE with constant loading performed poorly for higher values of σ_z, but the LCRE with non-constant loading performed much better for high values of σ_z, again consistent with previous observations.

The sum of scores method was competitive with the models. This was also observed in Chapter 2, when this method was compared to the results of the LCA. However, there are a number of cases where the LCA models, in particular the LCA and the LCRE with non-constant loading, do show improvement over this method.

To identify trends in the results, Table 3.7 was generated. This shows the cases

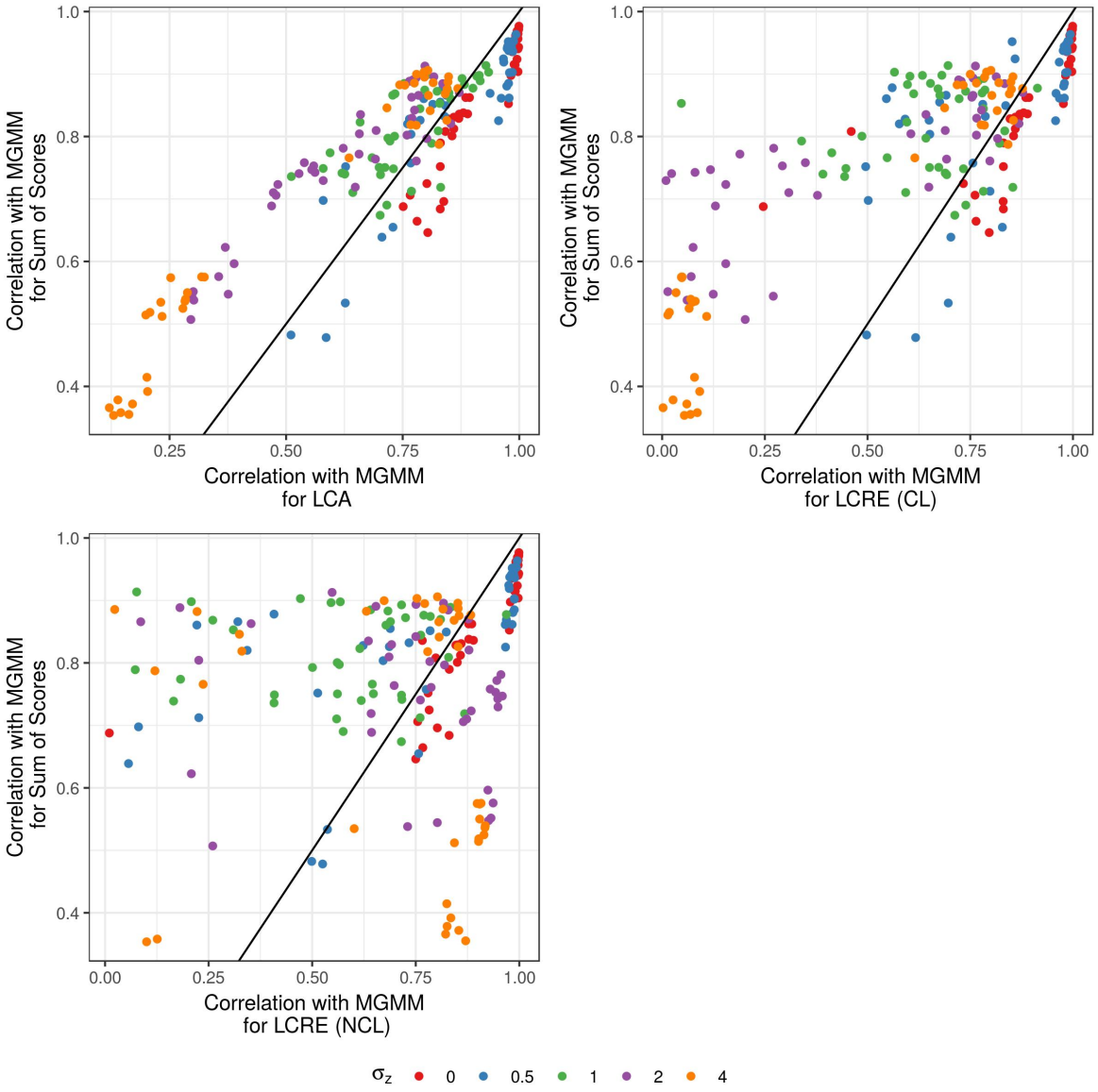


FIGURE 3.4: Comparisons of each LCA model to the Sum of Scores using correlation to the MGMM. A point above the $x = y$ axis indicates when the sum of scores performs better, while a point below the $x = y$ axis indicates when the LCA model performs better. The models tend to perform better when the σ_z value is low.

	p		δ		σ_z					p_0					Program Thresholds		**Total**
	5	7	0.5	1.5	0	0.5	1	2	4	0.1	0.3	0.5	0.7	0.9	Constant	Varied	
Sum of Scores	38	40	60	18	0	12	27	19	20	18	14	15	16	15	39	39	**78**

TABLE 3.7: Frequencies of scenarios that found a higher correlation to the MGMM (difference greater than 3) for the sum of scores method compared to the 3 LCA models. The number of scenarios with particular parameters are given in each column. Columns are grouped by parameter type. Total number of scenarios with a better correlation to the MGMM associated with the sum of scores model is given at the end.

where the sum of scores performed better than any of the 3 LCA models. This occurred for 78 out of the 200 scenarios. As noted in Figure 3.4, this was predominantly for the scenarios with a large values of σ_z. Additionally, the scenarios were also much more likely to have a δ of 0.5, compared to higher values of δ. This indicated that when the difference between the binding and non-binding genes scores were small, meaning that the thresholds were more likely to include genes that were not binding, this simpler method tends to perform better.

Summary of the measures

Finally, I collated a summary of the different results across the average correlation to the MGMM, the RMSE and the BIC in Table 3.8. This identified key patterns within the data, and the level of agreement across the three measures. The number of scenarios with each combination of results was counted (for example, preferred LCA for correlation, LCA for RMSE and LCA for BIC) and is given in the far right hand side of the table under "Frequency". As highlighted in the table, certain combinations appeared disproportionately frequently compared to the others, with three combinations representing 95 of the 200 scenarios. That these combinations were so frequent implies that most parameters had a small effect on the performance of individual models. This follows from the trends observed in Tables 3.3, 3.5 and

3.6, that most of the time σ_z and δ correlated with the preferred model while other parameters were distributed approximately equally across all the models.

The three most common combinations, highlighted in Table 3.8, were:

- The LCA was preferred for the correlation and RMSE but the LCRE with constant loading was preferred for BIC.
- The LCRE with non-constant loading was preferred for all three measures.
- The average correlation and RMSE had no clear preference (indicating that two or all three of the models performed equally well for that measure) but that the LCA performed best on BIC.

Further examining the parameters from these combinations, the first combination contained scenarios with medium values of σ_z, and slightly more scenarios with a lower value of δ. In contrast, the combination with preferences for LCRE with non-constant loading contained scenarios with exclusively a higher value of δ, and higher values of σ_z. Finally, the last combination with only a preference for the LCA model when using the BIC measure also had more scenarios with higher value of delta, and exclusively scenarios with a value of 0 in σ_z. This was expected, as it means no random effect was present, and thus all of the models should perform competitively. When all the models perform competitively, the simplest model in terms of the number of parameters (LCA) is preferred by BIC.

Conclusions

Based on these results, the models LCA and LCRE with non-constant loading appeared to be the two most competitive models, but the one that was preferred largely depended on the parameters σ_z and δ. A large δ and a large σ_z meant that the LCRE with non-constant loading was more likely to be preferred; otherwise the

LCA was preferred. Additionally, the LCA may be a better choice for practical reasons, as estimations for the LCRE with non-constant loading generally take a lot longer to calculate.

The sum of scores was considered to evaluate the benefit of using LCA rather than a naive method. The results indicated that, in terms of correlation with the MGMM, the LCA models offer on average a small improvement, although there were many scenarios where the sum of scores produced a higher correlation. In particular, the LCA model was preferred to the sum of scores method for scenarios with low σ_z and high δ. However, the sum of scores cannot be used to effectively rank

Average Correlation	RMSE	BIC	Frequency	Average Correlation	RMSE	BIC	Frequency
LCA	LCA	LCA	3	LCRE.CL	-	-	1
LCA	LCA	LCRE.CL	**31**	LCRE.NCL	LCA	LCRE.NCL	2
LCA	LCA	LCRE.NCL	3	LCRE.NCL	LCRE.CL	LCRE.NCL	1
LCA	LCA	-	2	LCRE.NCL	LCRE.NCL	LCRE.CL	2
LCA	LCRE.CL	LCA	1	LCRE.NCL	LCRE.NCL	LCRE.NCL	**36**
LCA	LCRE.CL	LCRE.CL	9	LCRE.NCL	LCRE.NCL	-	2
LCA	LCRE.CL	LCRE.NCL	3	LCRE.NCL	-	LCRE.CL	2
LCA	LCRE.CL	-	2	-	LCA	LCA	2
LCA	LCRE.NCL	LCRE.CL	3	-	LCA	LCRE.NCL	2
LCA	LCRE.NCL	LCRE.NCL	4	-	LCRE.CL	LCRE.CL	5
LCA	LCRE.NCL	-	1	-	LCRE.CL	LCRE.NCL	2
LCA	-	LCA	7	-	LCRE.NCL	LCRE.CL	6
LCA	-	LCRE.CL	2	-	LCRE.NCL	LCRE.NCL	12
LCA	-	LCRE.NCL	4	-	LCRE.NCL	-	3
LCA	-	-	1	-	-	LCA	**28**
LCRE.CL	LCRE.NCL	LCRE.CL	2	-	-	LCRE.CL	6
LCRE.CL	LCRE.NCL	LCRE.NCL	1	-	-	LCRE.NCL	6
LCRE.CL	-	LCRE.NCL	1	-	-	-	2

TABLE 3.8: Summary of results for model simulations. For each of the three different analyses presented, Average Correlation, RMSE and BIC, the best performing model was found for each scenario, and the frequency of the different combinations was counted. For example, LCA performed best in all three analyses for only 3 different scenarios (first row). The most common 3 results are highlighted in blue.

genes like the LCA model, because the sum of scores are not posterior probabilities, and the resulting scores were coarse in comparison, since there are only $p + 1$ possible values.

Given that the LCA model covered a wider range of σ_z values, is a simpler model, and is quick to calculate with the use of randomLCA, this would be the model of preference if no further information about the data is known. However, other methods of model selection, such as the use of BIC in Chapter 2, could be considered.

3.3.2 Should the BIC be used to select the best model?

Rather than determining whether there was a preferred model for all scenarios, another method would be to use the BIC to determine the best model. In previous sections, the BIC was used as a measure of model fit and the results were compared to the highest average correlation, and lowest RMSE, such as in Table 3.8. This indicated that the results for the average correlation and lowest RMSE did not necessarily match the BIC.

To further investigate whether using the BIC as a way to choose a model led to a reliable result, I returned to the original simulation data set, and identified the model for each scenario that had the lowest BIC. When the BIC was equivalent for multiple models, I chose the simplest model (LCA before LCRE (CL) before LCRE (NCL)). This meant that the LCA and LCRE with constant loading models performed best in terms of BIC more often than previously calculated. The LCA was chosen 44 times, the LCRE with constant loading 79 times and the LCRE with non-constant loading 77 times. I then collated the associated average correlation to the MGMM and the RMSE for the model with the best BIC, and used this to generate a 4th set of correlations and RMSEs.

Correlation to the MGMM

I compared the new average correlations in a pairwise fashion to the original average correlations by model. This is shown in Figure 3.5. The LCA and LCRE with constant loading showed mixed results. This was particularly observed when the model chosen using the BIC was the LCRE with non-constant loading; about half of the points indicated that the model based on BIC was superior, while the other half indicated that the other model was superior. When examining at the pairwise plot for the LCRE with non-constant loading, the points were more consistently found above the $x = y$ line, indicating that choosing the model based on the BIC improved the correlation. This was consistent with the observation that the LCRE with non-constant loading more often had the lowest BIC, but only improved the correlation when σ_z was high.

The correlation of the preferred model based on BIC in general had higher correlations compared to just choosing one model. This is shown in Figure 3.5, which compared the distribution of points along the x-axis to the distribution of points along the y-axis. There are many more points with a lower average correlation for the model on the x-axis compared to the BIC based model.

Investigating the correlations more closely, I collated a table with the frequencies of scenarios that preferred a different model to the BIC based model. This was split by model and parameters, and the results are given in Table 3.9. The LCA was the most common preference compared to the BIC based model, and found a higher correlation for 80 of the 200 scenarios. The LCRE models were preferred less often, for only 25 and 16 scenarios for the constant and non-constant loading models, respectively. This was in part because the LCRE was more often chosen as the best model based on the BIC. It was also possible that more than one of the models performed better for the same scenario compared to the BIC based model, so the

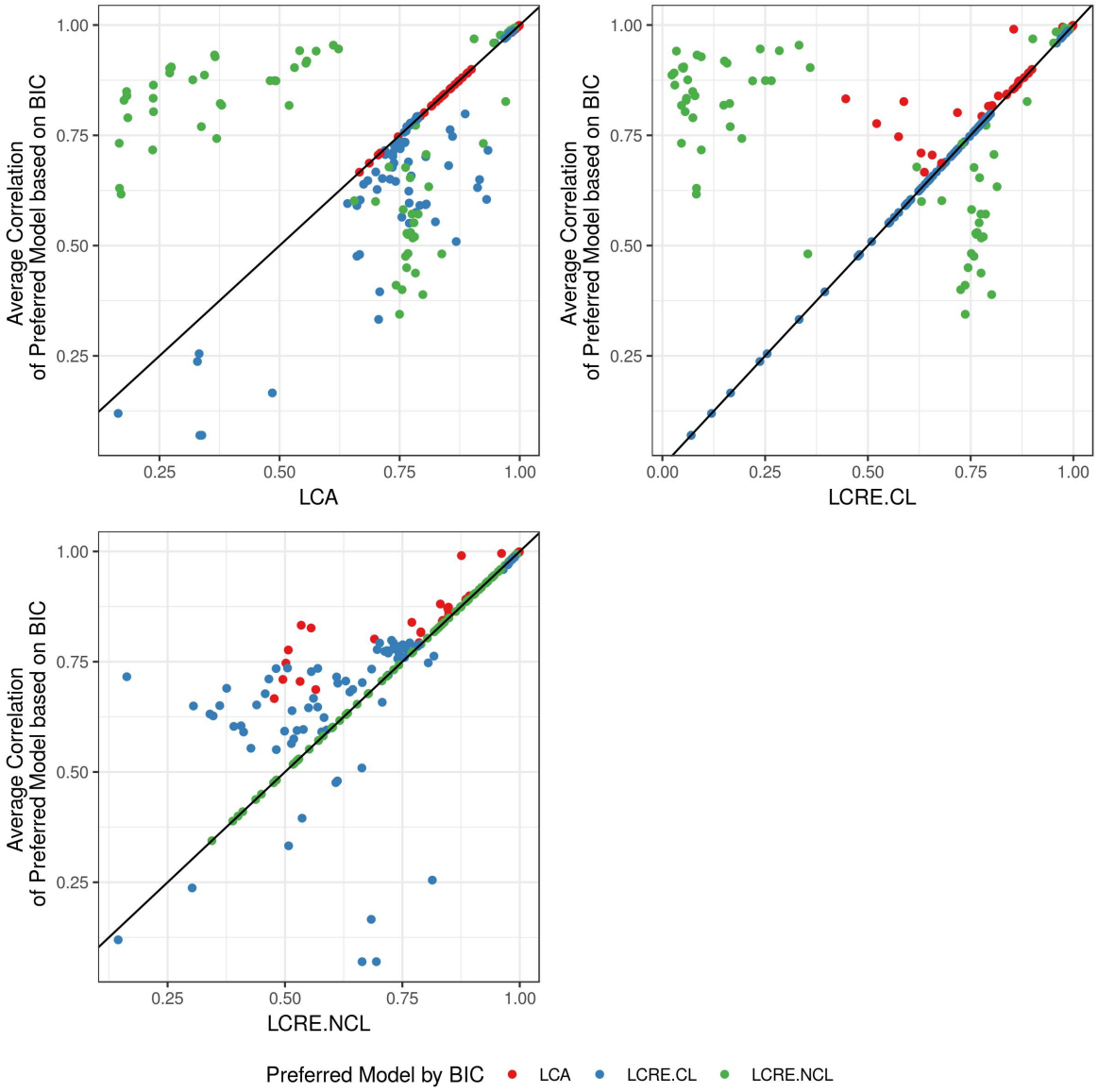


FIGURE 3.5: Average correlation to the MGMM model for models LCA, LCRE (constant loading) and LCRE (non-constant loading) for different scenarios, coloured by degree of random effect (σ), compared to the average correlation to the MGMM model for the model with the lowest BIC for that scenario. A higher value on both axes indicates a better fit to the MGMM. Points above the $x = y$ line indicate that the BIC model performed better, while points below indicate that the model on the x-axis performed better.

	p		δ		σ_z					p_0					Program Thresholds		**Total**
	5	7	0.5	1.5	0	0.5	1	2	4	0.1	0.3	0.5	0.7	0.9	Constant	Varied	
LCA	31	49	54	26	0	17	35	15	13	16	16	16	17	15	41	39	**80**
LCRE (Constant Loading)	4	21	22	3	0	1	2	10	12	6	4	4	5	6	14	11	**25**
LCRE (Non-Constant Loading)	13	3	1	15	0	3	7	5	1	5	3	2	1	5	8	8	**16**

TABLE 3.9: Frequencies of scenarios that had a higher correlation when the model on the left was chosen naively rather than when using the model with the lowest BIC (difference greater than 3) for that scenario. This does not include scenarios where the model on the left was the best model by the BIC. The LCA had the best chance of outperforming the BIC based model, while the LCRE with non-constant loading had the worst chance.

scenarios found for each model were not mutually exclusive.

In general, the BIC based model appeared to perform best when the σ_z was low, and when the δ was high. This is likely because all of the models perform equally well, as observed in earlier analysis. When δ was low, and σ_z moderate, the LCA model would have been a better choice than the BIC based model. Similar observations were made for the LCRE with constant loading, although a larger σ_z was needed for the model to perform the best. Finally, the LCRE with non-constant loading was more likely to outperform the BIC based model when the δ value is high. The other parameters did not seem to influence the average correlation, as noted previously.

One consideration for this method is how much better the other model performs in comparison to the BIC based model. Considering Figure 3.5, most of the points where the BIC based model performed better compared to the LCA were less than 0.5 for the LCA, and over 0.75 for the BIC based model. In contrast, when the LCA performed better, the BIC based model in general had an average correlation higher

than 0.5. Similar results were also observed when comparing the BIC based model to the LCRE models. Thus, while other models may give some improvement in terms of the average correlation this level of improvement may not necessarily be great.

RMSE

I performed the same analysis for the RMSE results (see Figure 3.6). The results were similar to those for Figure 3.5. When compared to the LCA, the RMSE of the BIC based model performed inconsistently in comparison. The range in magnitude of the RMSE also tended to be the same for both models. The LCRE with constant loading performed worse than the BIC based model with either equivalent or higher RMSE values. The LCRE with non-constant loading had very similar RMSE values to the BIC based model, with most values lying close to the $x = y$ line. This is again due to this model having smaller BIC values compared to the other models.

The trends across the different scenarios in terms of the lowest RMSE were also investigated, and are given in Table 3.10. Overall, fewer scenarios preferred the naive model over the BIC based model, with the LCA preferred the most (53 out of 200 scenarios). This indicated that the BIC based model more often gave the best RMSE result. This was particularly the case when the value of σ_z was 0, as before, while more often the LCA was preferred if the σ_z value was 1. This agreed with the general results from Table 3.9.

Sum of Scores

The sum of scores results were also compared to the BIC based model directly. The pairwise comparison for the sum of scores correlation to the MGMM, and the BIC based model correlation to the MGMM is given in Figure 3.7.

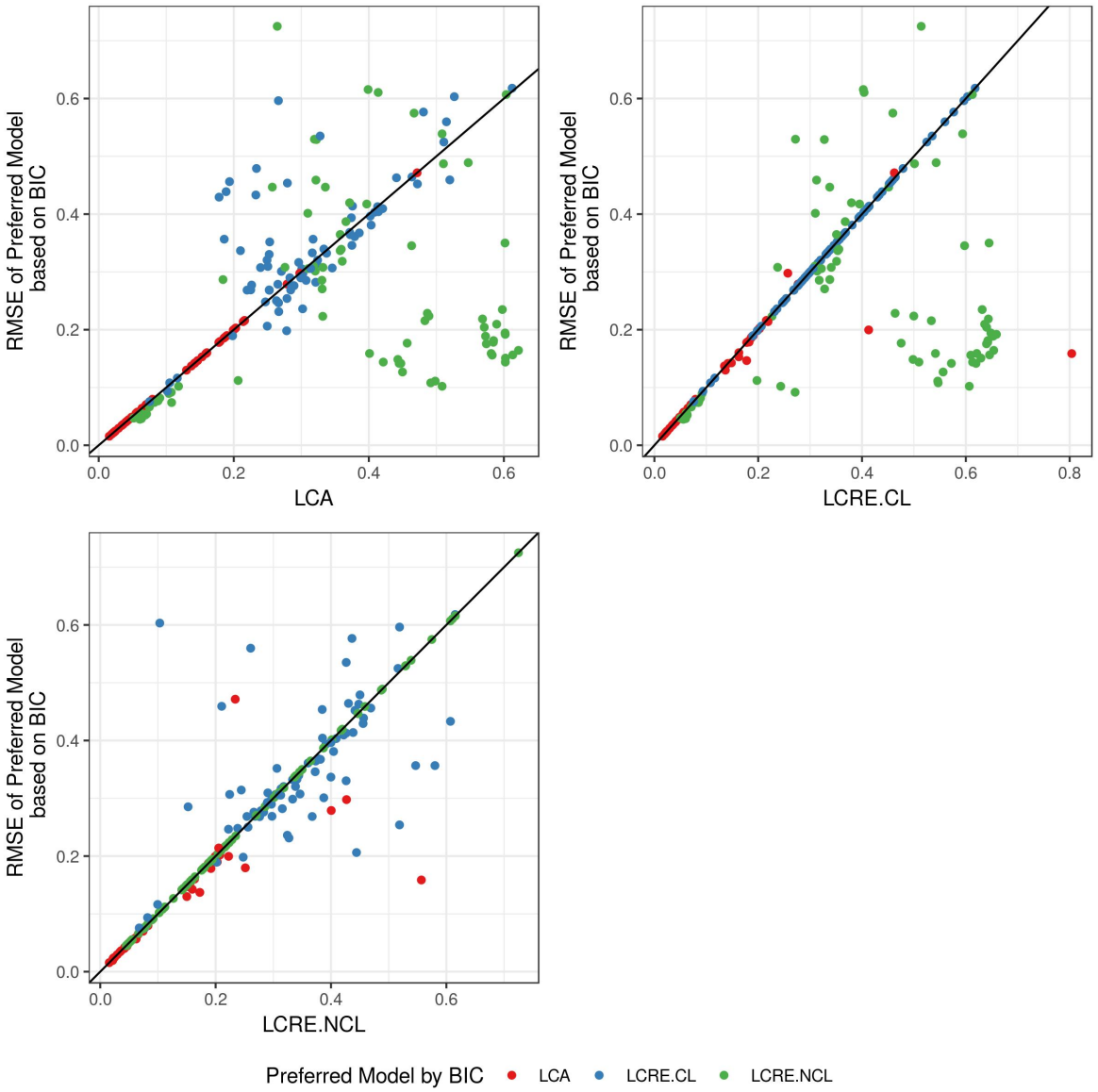


FIGURE 3.6: RMSE for models LCA, LCRE (constant loading) and LCRE (non-constant loading) for different scenarios, coloured by degree of random effect (σ), compared to the RMSE for the model with the lowest BIC for that scenario. A higher value on both axes indicates a better fit to the MGMM. Points above the $x = y$ line indicate that the model on the y-axis performed better, while points below indicate that the model on the x-axis performed better. When the value of σ_z were high, the LCRE (non-constant loading) performed better; otherwise the LCA performed better, or the models were equivalent (close to the $x = y$ line).

	p		δ		σ_z					p_0					Program Thresholds		**Total**
	5	7	0.5	1.5	0	0.5	1	2	4	0.1	0.3	0.5	0.7	0.9	Constant	Varied	
LCA	21	32	29	24	0	11	24	9	9	8	12	13	12	8	26	27	**53**
LCRE (Constant Loading)	5	12	14	3	1	2	2	5	7	3	5	2	3	4	10	7	**17**
LCRE (Non-Constant Loading)	21	6	14	13	0	6	12	8	1	8	7	1	2	9	15	12	**27**

TABLE 3.10: Frequencies of scenarios that had a lower RMSE when the model on the left was chosen naively rather than when using the model with the lowest BIC (difference greater than 3) for that scenario. This does not include scenarios where the model on the left was the best model by the BIC. The LCA had the best chance of outperforming the BIC based model, while the LCRE with non-constant loading had the worst chance.

The results showed that there were a large number of scenarios where the sum of scores method was superior to the BIC based model. When the LCA was chosen as the best model, the correlation was almost always superior to the sum of scores result. When the LCRE with constant loading was chosen however, the result was much more likely to be superior for the sum of scores method. The LCRE with non-constant loading model appeared to be almost evenly split across the $x = y$ line, indicating that for some scenarios it was preferred, while for others the sum of scores was preferred. This is most likely due to the LCRE with non-constant loading being preferred for the higher values of σ_z. These results are consistent with the results in the previous section, which indicated that the LCA was the most competitive with the sum of scores method. As the BIC-based model was less likely to choose the LCA method, it was less competitive overall compared to the LCA.

The trends for when the BIC based model was superior to the sum of scores method are given in Table 3.11. The BIC based model was preferred to the sum of scores for about half of the scenarios (102 out of 200). The results were consistent

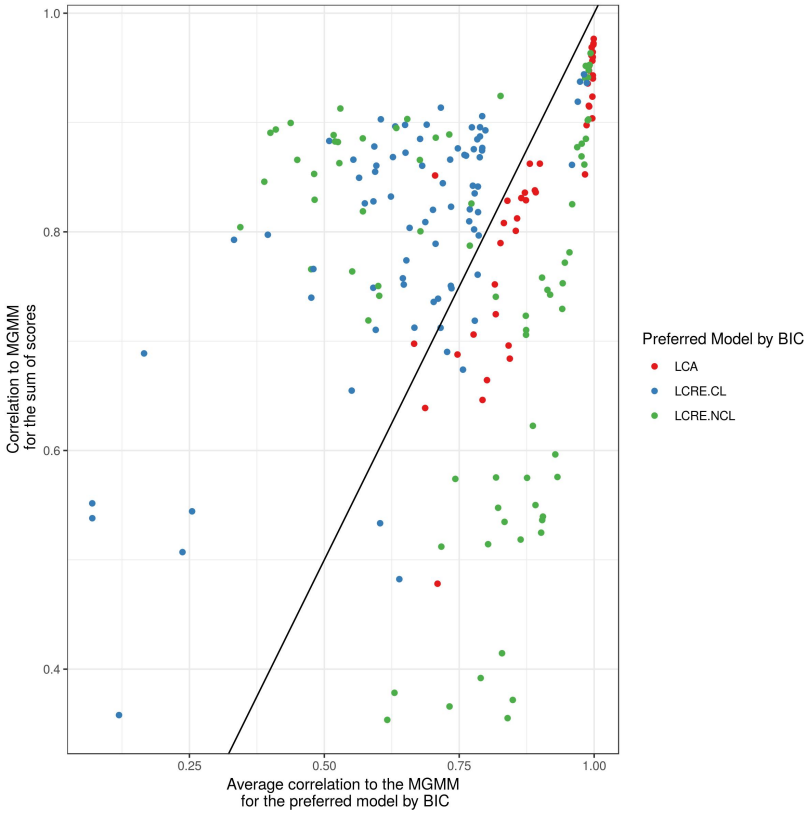


FIGURE 3.7: Correlation to the MGMM for Sum of Scores compared to the average correlation to the MGMM for the BIC based model. Points above the $x = y$ line indicate points where the Sum of Scores had a higher correlation, points below the $x = y$ line indicate points where the BIC based model performed better

	p		δ		σ_z					p_0					Program Thresholds		**Total**
	5	7	0.5	1.5	0	0.5	1	2	4	0.1	0.3	0.5	0.7	0.9	Constant	Varied	
BIC Based Model	48	54	29	73	40	23	5	16	18	21	20	20	19	22	51	51	**102**

TABLE 3.11: Frequencies of scenarios that found a higher correlation to the MGMM (difference greater than 3) for the BIC based model compared to the sum of scores method. The number of scenarios with particular parameters are given in each column. Columns are grouped by parameter type. Total number of scenarios with a better correlation to the MGMM associated with the BIC based model is given at the end.

with previous findings and Figure 3.7, as the BIC based model was preferred to sum of scores method for low to moderate values of σ_z and higher values of δ; most likely the occasions when the BIC based model was the LCA. This suggested that the LCA should be preferred over the BIC based model, because it was more likely to improve over the results given by the sum of scores method.

Conclusion

Using the BIC as a way to determine the best model had mixed results. Comparing the average correlation and RMSE of the BIC based model to the other models indicated that an alternative model was superior for most scenarios. The LCA had the lowest BIC for 41 scenarios, and additionally performed better in the average correlation than the BIC based model for 80 scenarios and performed better for the RMSE than the BIC based model for 53 scenarios. Thus choosing the LCA without considering the BIC would have been the better choice for 61% of the scenarios in terms of average correlation, and 47% of the scenarios in terms of the RMSE. Similar rates can be found for the LCRE models also. When comparing the results of the correlation of the BIC based model and the sum of scores method, the results indicated that the BIC based model performed better for only 51% of scenarios. This

implied that the LCA would have been a better choice, since it was competitive with the sum of scores in the previous section.

One consideration when assessing this method is that it is more time expensive. In order to calculate the BIC, all three models must be generated, which would take longer than any single model calculation. Thus overall it is more practical and gives better results to choose the LCA model rather than consider all three.

3.3.3 Investigating preference of One Class vs Two Class models using the BIC

One of the other issues discovered while applying LCA to real data was that sometimes the BIC would prefer a model with one class rather than two. This occurred for the more complex LCRE models. For ChIP-seq data, it is reasonable to assume that a two class model is the most appropriate, since the underlying signal would create a class of binding and a class of non-binding genes and also that the calling programs can, to the same extent, differentiate between binding and non-binding genes. While in the previous section, I only examined the results from two class models, I also investigated how often the BIC preferred the one class model.

Analysis of two class data

Initially, I analysed the same data using a one class model for the LCA, and the two LCRE models and calculated the difference in the BIC between a two class and one class model, ΔBIC. The two class model was preferred if the difference was greater than 3, otherwise the one class model was preferred. This was performed across the 20 replications. One of the initial observations was that while there were some scenarios where the one class model was preferred for the majority of the replications(at least 10), none of the scenarios found that a one class model was

always preferred. In contrast, there were a number of scenarios where the two class model was always preferred. Additionally, while often the difference in BIC was large when the two class model was preferred, the difference when the one class model was preferred was generally small. These observations indicated that model preference in terms of class is more dependent on random difference within the data than on scenario parameters. This was unexpected, and implies that the class preference calculated by BIC is not reliably indicative of the actual number of classes within the data.

There was a clear difference in the behaviour of ΔBIC for the three models. To demonstrate this, Figure 3.8 shows the results across the 20 replications for 10 random scenarios across each of the three models. Clearly, the LCA model found some of the greatest differences between the one class and two class model, and the two class was always preferred even if the magnitude of ΔBIC was small. In contrast, the LCRE models found in some cases that the one class model was preferred, but not consistently across the 20 replications. There also tended to be a greater variation in the BIC for the different replications, implying that these models were more sensitive to random changes in the data for the same scenario.

To investigate overall trends in the data and to better understand the behaviour observed in Figure 3.8, Table 3.12 was generated similarly to the tables in the previous section. This table shows the frequency of scenarios where the one class model was preferred for the majority of replications, split by parameter. The LCA model did not show any preference for the one class model, the LCRE with constant loading preferred the one class model for some of the scenarios (18), and the LCRE with non-constant loading preferred the one class model for almost half of the scenarios (83) out of the 200 scenarios total. This is therefore due to the random effect of the LCRE models, and more specifically the constant loading parameter.

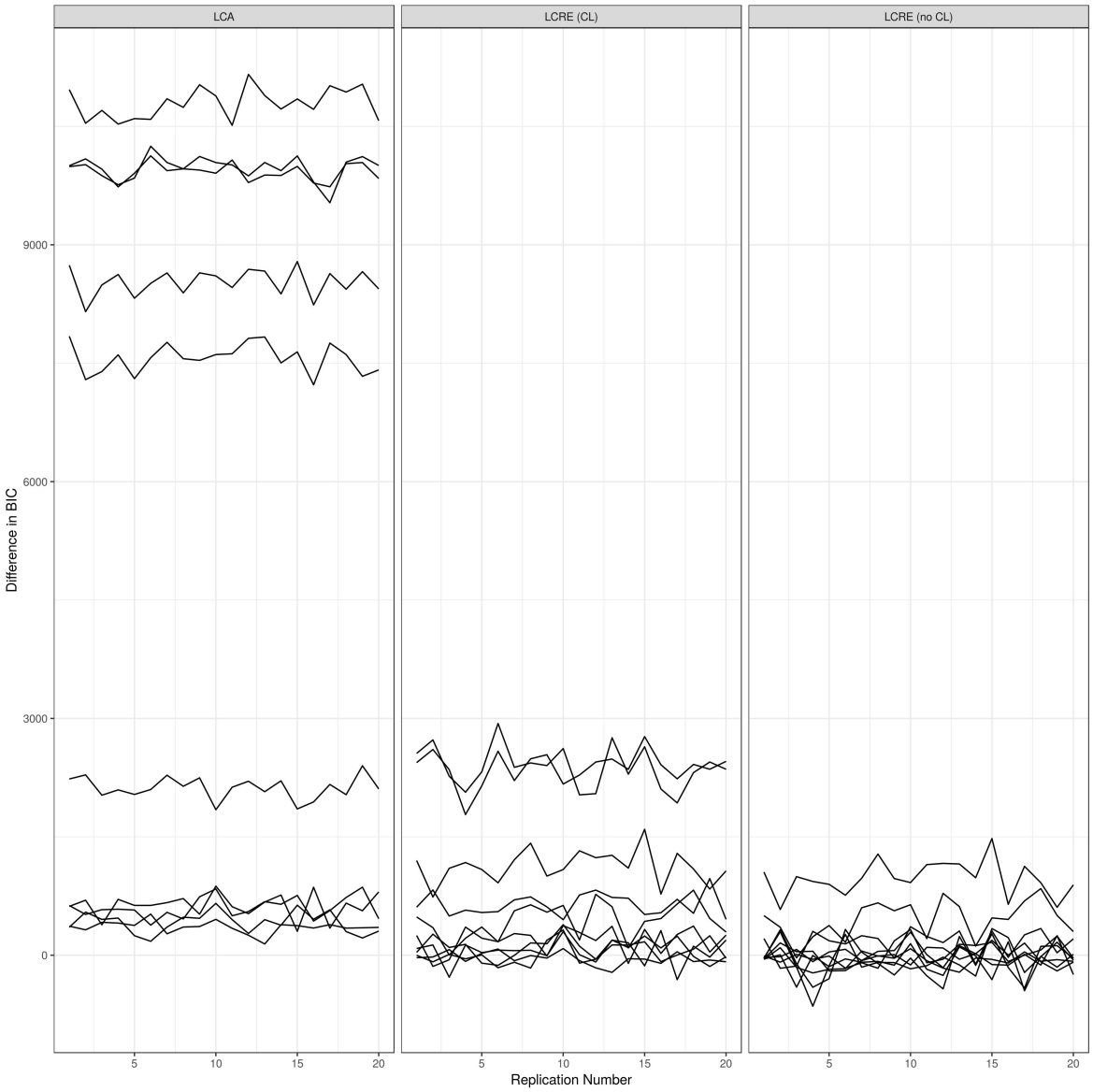


FIGURE 3.8: Plots of the difference in BIC (One Class-Two Class) for 20 replicates for 10 randomly selected scenarios for the LCA, LCRE with constant loading and LCRE with non-constant loading models. A smaller BIC is desirable, so points that lie above the $y = 0$ line indicate that two classes are preferred, while points that lie below the $y = 0$ line indicate that one class is preferred. More variation is seen in the LCRE models compared to the LCA models.

	p		δ		σ_z					p_0					Program Thresholds		**Total**
	5	7	0.5	1.5	0	0.5	1	2	4	0.1	0.3	0.5	0.7	0.9	Constant	Varied	
LCA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
LCRE (Constant Loading)	10	8	18	0	15	3	0	0	0	3	3	5	4	3	10	8	**18**
LCRE (Non-Constant Loading)	37	46	82	1	16	18	17	17	15	19	17	15	16	16	45	38	**83**

TABLE 3.12: Frequencies of scenarios that found a significantly lower BIC (difference greater than 3) for a one class model for the same scenario and model for the majority of the replications ($\geq$ 10 out of 20 replications). The overwhelming majority of these are found in scenarios with a δ of 0.5, and only for the LCRE (CL) and LCRE (no CL) models.

When a one class model was tested, the random effect of the LCRE models accounted for any difference between the binding and non-binding genes. This allowed the model to have a good fit to the data with the lower complexity model, resulting in a lower BIC. The LCA model cannot as readily account for the difference, and so has a poorer fit, leading to a higher BIC. Thus the LCA was more likely to correctly identify a two class model as preferred when using the BIC.

The most influential parameter was δ, with almost all of the scenarios with a preference for the one class model showing a low δ. A low value of δ was more likely to generate a single large group of values with overlap between binding and non-binding scores, which was reflected in the results of the programs. The LCRE with constant loading was also more likely to prefer the one class model when the value of σ_z was low, while no such trend was observed in the LCRE with non-constant loading. This is counter-intuitive and occurs because when a random effect exists, it is recognised by the LCRE with constant loading, leading to a better fit for the two class model.

Finally, I also examined the average correlation to the MGMM for some of the

scenarios for the two LCRE models. The scenarios that found an overall preference for a one class model over the 20 replicates were each split into replicates that preferred the one class model and replicates that preferred the two class model. The average correlation was then found for each group, and the results compared in a pairwise fashion. This result was given in Figure 3.9.

There are a greater number of scenarios in the LCRE with non-constant loading plot because a greater number of scenarios preferred a one class model. Overall, there was no apparent trend between the correlation of scenarios that preferred either class. The scenario average correlation appear randomly scattered around the $x = y$ line. The average difference between the correlations was calculated across all of the scenarios; this was found to be -0.0007 for the LCRE with non-constant loading and 0.01 for the LCRE with constant loading. This was evidence that the preference for one class or two class based on the BIC had little to no affect on the posterior probabilities of the model.

Analysis of one class data

I performed the same analysis using the data with only one cluster. In this case, the BIC should find the one class models to be preferred. The results were much more uniform across the 20 replications for the one cluster data, with almost all of the scenarios preferring either a one class or two class model exclusively. When only one cluster was present, the models were less sensitive to random changes, because instead of two smaller clusters with difference means, there was one large cluster with the same mean.

As before, there was a clear difference in the performance of the three models, and a similar distribution of ΔBIC was observed. This can be seen in Figure 3.10, which shows the ΔBIC across the 20 replicates for 10 randomly selected scenarios.

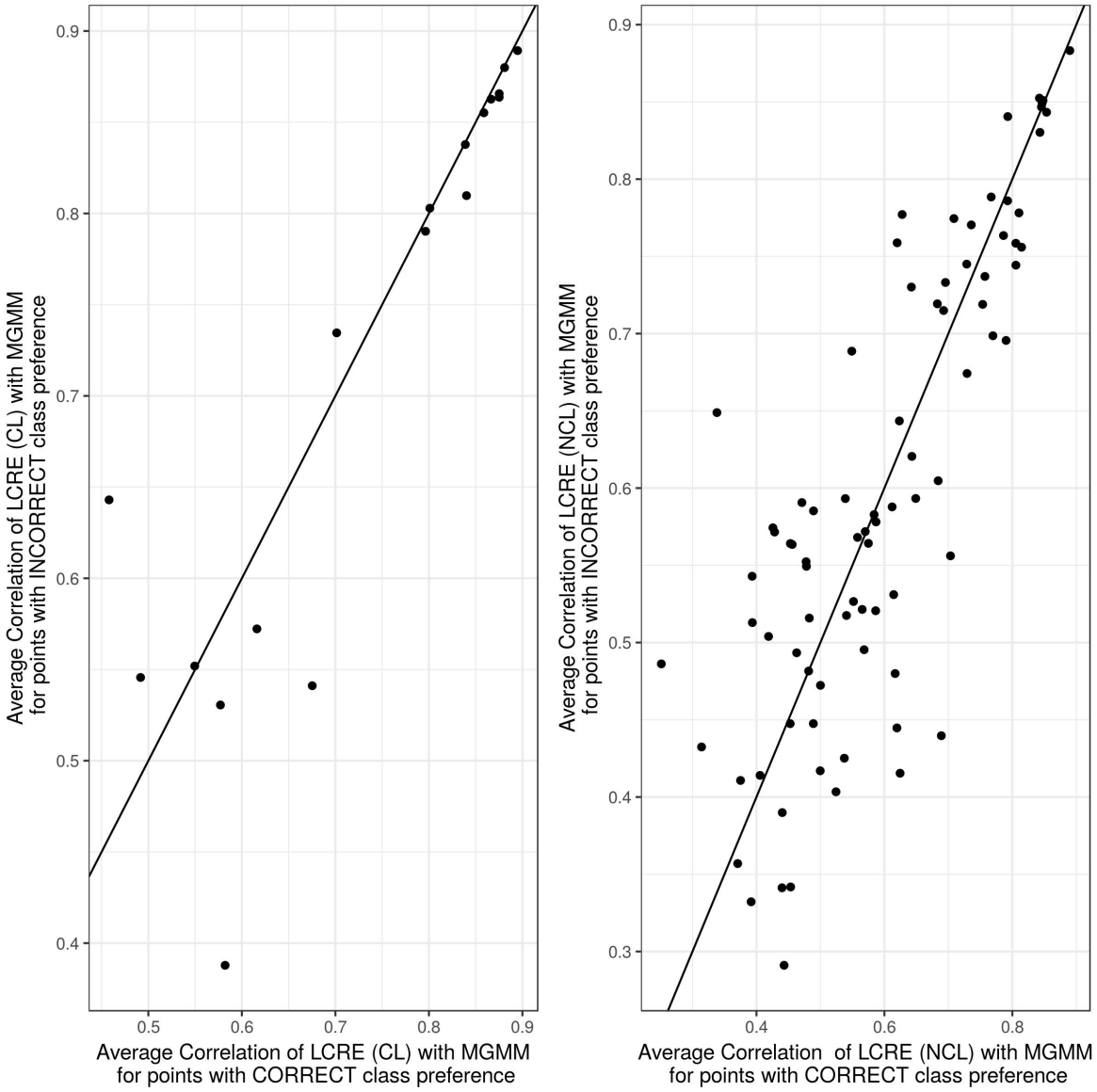


FIGURE 3.9: Plots of the average correlation to the MGMM for scenarios where there was an overall preference for a one class model. The 20 replicates for each scenario were divided into two groups by their class preference, and the average correlation was calculated for each group. Points above the $x = y$ line indicate scenarios where the replicates with incorrect class preference performed better, and points below the line indicate scenarios where the replicates with correct class preference performed better. There is no clear trend for either model.

As discussed earlier, the amount of variation across the 20 replications was lower in all three models, particularly for the two LCRE models. Additionally, all three models showed a negative ΔBIC for some scenarios, indicating a preference for the one class model. The LCRE with non-constant loading performed the best, as it consistently generated a lower BIC for the one class model compared to the two class model. In contrast, the other models, LCA and LCRE with constant loading, showed a preference for the two class models for a number of scenarios. Interestingly, the difference in ΔBIC when a one class model was preferred was still small compared to when a two class model is preferred, especially for the LCA model (although the difference was still significant).

Trends were identified and collated into Table 3.13. Clearly, the LCRE without constant loading performed the best compared to the other two models, with 100% of the scenarios correctly preferring the one class model. There was also a very clear predictor for when the LCA and the LCRE with constant loading would prefer the one class model, with all of the scenarios with a σ_z value of 0 having the correct model preferred. This stark contrast was unexpected, but indicated that these models were much more likely to explain slight changes in variation (such as those caused by σ_z) by changing the clustering, rather than from random effect.

Conclusion

This investigation indicated that the model BIC was not a reliable way to identify clustering. The LCA and the LCRE with constant loading tended to have a higher number of false positive in terms of preferring a two class model, especially in the case when there was some level of random effect present. In contrast, the LCRE with non-constant loading was more likely to have false negatives, in terms of preferring a two class model, particularly when the differences between the two underlying

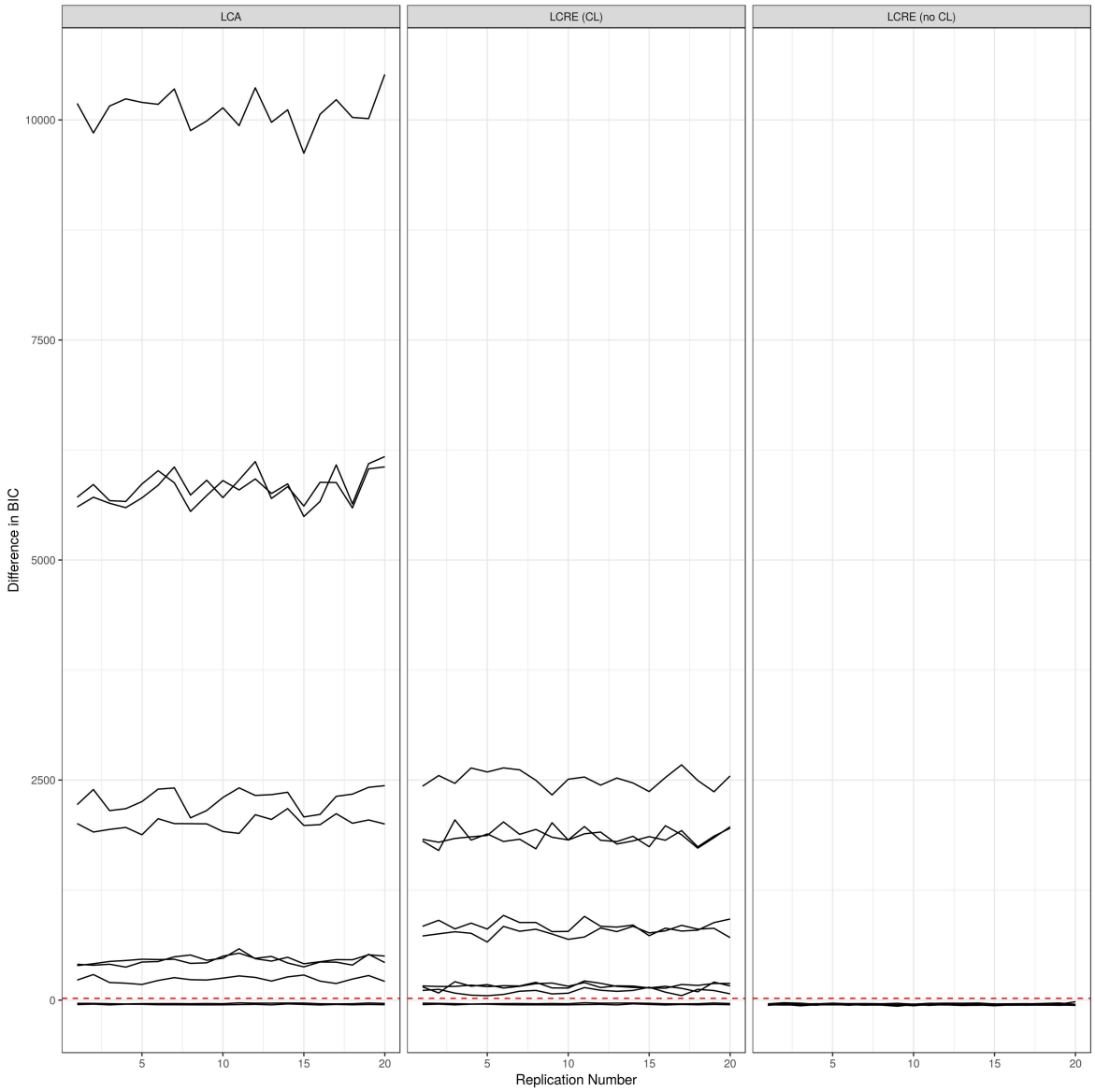


FIGURE 3.10: Plots of the difference in BIC (One Class-Two Class) for 20 replicates for 10 randomly selected scenarios for the LCA, LCRE with constant loading and LCRE with non-constant loading models, in the case where there are not two groups in the underlying data. A smaller BIC is desirable, so points that lie above the $y = 0$ line indicate that two classes are preferred, while points that lie below the $y = 0$ line indicate that one class is preferred. More variation is seen in the LCRE models compared to the LCA models.

	p		σ_z					p_0					Program Thresholds		**Total**
	5	7	0	0.5	1	2	4	0.1	0.3	0.5	0.7	0.9	Constant	Varied	
LCA	10	10	20	0	0	0	0	4	4	4	4	4	10	10	**20**
LCRE (Constant Loading)	10	10	20	0	0	0	0	4	4	4	4	4	10	10	**20**
LCRE (Non-Constant Loading)	50	50	20	20	20	20	20	20	20	20	20	20	50	50	**100**

TABLE 3.13: Frequencies of scenarios that found a significantly lower BIC (difference greater than 3) for a one class model for the same scenario and model for the majority of the replications ($\geq$ 10 out of 20 replications) for data where there was not two clusters in the underlying data. The LCRE (no CL) model preferred the one class model for every scenario, while the other models preferred the correct model when σ_z was 0.

clusters was small (such as when there was a small δ). Additionally, when scenarios with preference for a one class model were further examined, it was found that this preference did not affect the posterior probability correlation to the MGMM. This indicated that the preference for a one class model did not affect the accuracy of the model and the posterior probabilities could still be trusted.

When considering the analysis of ChIP-seq, it is reasonable to assume that the data will be a two class model, but considering the BIC may be helpful in identifying the level of noise within the data. For the application of LCA, unless all of the models agree that a one-class model is preferred, a conclusion on clustering can not be reached. Furthermore, examining the difference in the BIC is not necessarily helpful, as this was not found to be correlated to clustering.

Based on the results of this section, the BIC should not be used to determine class preference.

3.4 Conclusions

Simulations are useful for gauging the performance of different analysis methods when the underlying truth is not normally known. In this Chapter, simulation data was constructed using a normal distribution and then identified by different "programs" using thresholds on the scores in order to generate a set of binary results. The aim was to answer some of the questions that arose when the LCA models were applied to real ChIP-seq data in Chapter 2.

Firstly, each of the models was assessed to determine which of the three models most accurately calculated the correct posterior probabilities, and had the most competitive BIC. While the LCRE models were expected to perform best when there was a random effect present ($\sigma_z > 0$) , it was found that the LCA tended to perform the best for the average correlation to the MGMM and the RMSE when the random effect was low to moderate ($\sigma_z \leq 1$). The LCRE with non-constant loading was preferred for higher values of the random effect for the average correlation and the RMSE, and both of the LCRE models performed well when the BICs of the models were compared when a random effect was present. The sum of scores was also used as a comparison to the 3 models. It was found that the 3 models performed better when σ_z was low, and δ was high. Given that this is when the LCA performed best, this was further evidence that the LCA should be the preferred model.

The next sections investigated the BIC as a method of determining the best model. When a two class model is assumed, the BIC can be used to determine the best model to continue with, across the three models. This method was tested against the more "naive" approach of selecting the same model each time. The results indicated that this would result in a reasonable analysis, even if there may have been a more appropriate model to choose. However, when the sum of scores method was considered, the BIC based model did not necessarily improve on this score. One

consideration before using this method is that since all models must be calculated to determine the BIC, it is more time consuming than just selecting a model without the BIC.

It was observed when applying the models to real data that sometimes a one class model was preferred by the BIC. When this was assessed using the simulation data, it was found that the LCRE models were more likely to falsely determine that the one class model was appropriate when the underlying data had two clusters, but that the LCA was more likely to prefer a two class model when a one class model would be more appropriate. Additionally, it was found that class preference by BIC did not have an effect on the accuracy of the posterior probability. While in reality the underlying data of ChIP-seq is always expected to form two clusters, this assessment indicated that the BIC was not reliable for determining the number of clusters (although comparing the BICs of both models may be informative for the level of noise in the underlying data).

There are some limitations to the simulations designed here. The main one is that the conclusions are based on simulations that are simplistic model of actual ChIP-seq peak finding software results. Since I cannot obtain this real data, in practise it may not be well approximated by this kind of model, and thus might have parameters different to those I considered here. This may be improved by considering a wider range of simulations, or a different type of simulation. Another limitation was the number of genes chosen for the simulation.

Here, only 3000 were used compared to the approximately 25,000 present in the Ensembl database. The full effects of this change is unknown, but it may, for example, change the results for the BIC section. Since the BIC is calculated using the number of data points n, the number of genes may change the results of the BIC, which may affect the conclusions. Again, this may be improved by considering a

wider range of simulations, in this case with different numbers of genes.

Clearly, the selection of the "best" model is influenced by the distribution of the underlying data. In reality this knowledge cannot be obtained, and these simulations are useful in determining the method that has the best results. In general, *the LCA was the model that gave the best results*. Therefore, based on the results from the simulations, using this model would be the best choice.

Chapter 4

Changing Threshold Method

4.1 Introduction

In Chapter 3, I used the results from a series of simulations to compare the models LCA, LCRE with constant loading and LCRE with non-constant loading. I found that the LCA performed best for a large number of the scenarios, particularly when the value of the random effect was low and the difference between the scores of binding and non-binding genes was high. In this chapter, I developed a new approach that uses different thresholds on the data to generate multiple posterior probabilities for each gene.

The threshold used to generate the binary data can have a large effect on the performance of the LCA models. To illustrate, a series of simulations were performed using the methods described in Chapter 3, where the scenario was kept the same (5, 1.5, 1, 0.3, Constant) with 20 replications, but the threshold T_i was changed. The output was then analysed using the LCA model. The results were summarised in Figure 4.1, which shows the average correlation to the MGMM for simulated data when a constant threshold is changed. Since the optimal threshold is known to be 0, it is clear that increasing this threshold beyond the optimal value degrades performance.

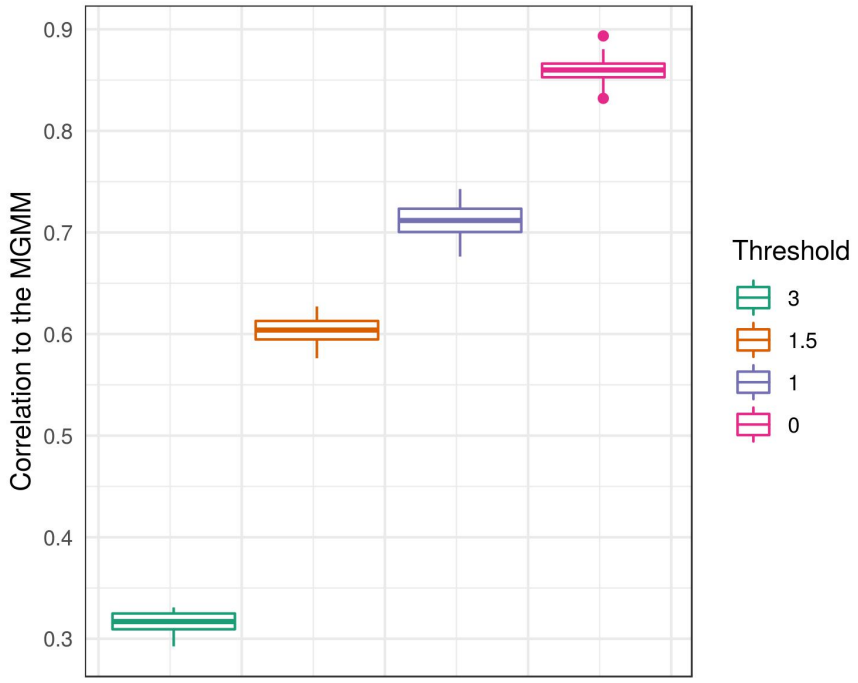


FIGURE 4.1: Boxplot of correlations to MGMM for the LCA model for the same scenario (5, 1.5, 1, 0.3, Constant) for four different thresholds over 20 replications. A threshold of 0 is considered optimal. Thresholds closer to this value have increased correlation to the MGMM.

When the LCA model is applied to real ChIP-seq data, the optimal threshold is not known, and is very unlikely to used by all of the programs. A new approach was investigated to determine if using multiple different thresholds would improve the performance. It was hypothesised that even if some of the thresholds are non-optimal, the overall performance would be improved compared to a single set of non-optimal thresholds. In this chapter, this idea was tested using simulated data.

4.2 Method

I developed a new method for obtaining more robust posterior probabilities using different thresholds for the same program. See the method described in Chapter 3

for a description of the data generation, although note that some alterations were made and are described below.

Instead of using the constant or varied threshold, three values were used in T_i, representing low, medium and a high threshold value (when a 'high' threshold value is discussed in this context, this means a less stringent threshold).

To generate a posterior probability for each gene from these thresholds, a gene set was generated with each program using the three thresholds. This is in contrast to the usual method of generating a gene set with the programs, when a single threshold is used. Each combinations of the gene sets for each program was analysed using the LCA models, resulting in 3^p analyses for each model, and the same number of posterior probabilities for each gene. The average posterior probability for each gene was then calculated, generating a single posterior probability for each gene.

This approach was performed for the scenarios described in Table 3.1 with a program number of 5. This was done as the number of permutations that would be required if 7 programs was tested is impractical for most settings ($3^7 = 2187$) compared to the number required for 5 programs ($3^5 = 243$). Each scenario analysis was repeated 7 times.

The average posterior probabilities were compared to the original results from the three different models across the same scenarios. The analysis of this approach was limited to the assessment of the posterior probability, as the expected and observed fit changes with each LCA, so no consensus expected and observed fit would be made. The correlation of the average posterior probability to the MGMM was calculated to compare to the posterior probabilities for the original LCA and LCRE results. Additionally, the RMSE for the genes was also measured.

The BIC could not be used to assess the average threshold method, as the BIC

varied with each gene set combination. An alternative method of assessment was instead used to approximately measure the accuracy of both methods as follows. Since the underlying true number of binding genes was known, the number of binding genes calculated could be compared to the scenario's p_0 parameter, and used to determine how accurately the model determined the number of genes. The number of binding genes found by the model was calculated using the posterior probability, making is possible to measure this for both the average and fixed threshold method.

The average posterior probabilities for each gene were added together, and averaged across the 7 replications, to get the approximate number of binding genes. This was then compared to the true number of binding genes ($3000 \times p_0$) for each scenario. A metric for the accuracy is the absolute proportion of estimated binding genes over the true binding genes, so a perfectly accurate estimation would have the result of 1. The closer to 1, the more accurate the estimation.

4.3 Results

4.3.1 Correlation to MGMM

The correlation to the MGMM for the scenario with parameters $\{p, \delta, \sigma_z, p_0, \text{Program Threshold}\} = \{5, 0.5, 0, 0.1, \text{Varied}\}$ (Scenario 1) for the correlation are shown in Table 4.1. Full results can be found in Appendix B.2. In contrast to similar results in Table 3.2, the standard deviation of the correlations was lower for all of the models, particularly for the LCRE models. Additionally, the average correlation was much higher as well, with all three models having an average greater than 0.9.

To compare the performance of the averaged threshold results for each model, pairwise plots were generated. These plots compared the average correlation to the MGMM for the "fixed threshold" method to the average correlation for the average

	LCA	LCRE (Constant Loading)	LCRE (No Constant Loading)
	0.920	0.918	0.913
	0.915	0.910	0.880
	0.930	0.925	0.927
	0.920	0.909	0.903
	0.916	0.911	0.910
	0.925	0.921	0.923
	0.922	0.916	0.901
Average	**0.921**	**0.916**	**0.908**
Standard Deviation	**0.005**	**0.006**	**0.015**

TABLE 4.1: Correlation results for the LCA and LCRE (Constant Loading) and LCRE (No Constant Loading) for Scenario 1, averaged across different thresholds. Correlation compares the posterior probability for all genes for the LCA and LCRE to the posterior probability for all genes for the MGMM. The average and standard deviations were used to compare Scenarios.

threshold method for the same model. The results are given in Figure 4.2, coloured by the value of σ_z for that scenario, and indicate that for most scenarios, there was at least some improvement found by using the average thresholds method. For the LCA, this was mostly a minor improvement, particularly for scenarios with high σ_z that had very low average correlation for the fixed threshold model. Similarly, the LCRE with constant loading showed moderate improvement when the average threshold method was used. Some of the low correlation, high σ_z points did however show a higher average correlation for the fixed threshold method, but only for 6 scenarios. For the LCRE with non-constant thresholds, many of the scenarios had a large improvement when average threshold method was used, across all of the values of σ_z. There was only one scenario for which the fixed threshold method performed better.

While the σ_z value of the scenario seemed to have little effect on the level of improvement seen when using the average threshold method, it did have an effect of the actual average correlation value. Lower values of σ_z had higher average correlation, and the average correlation tended to decrease as the σ_z increased. A similar effect was noted in Figure 3.1.

To further assess this new method in terms of the average correlation to the MGMM, I analysed the trends of scenarios that preferred either the fixed threshold or average thresholds method (see Table 4.2). This was only performed for the LCA method, based on the results in previous sections that demonstrated this to be the preferred model for most scenarios. The results indicated that the average threshold method was preferred for 44 of the 50 scenarios, while the fixed threshold method was preferred for only 3 scenarios. The remaining scenarios were competitive. Interestingly, the scenarios that preferred the fixed threshold method were those with a low value of δ. This may be due to certain combinations of gene sets resulting in

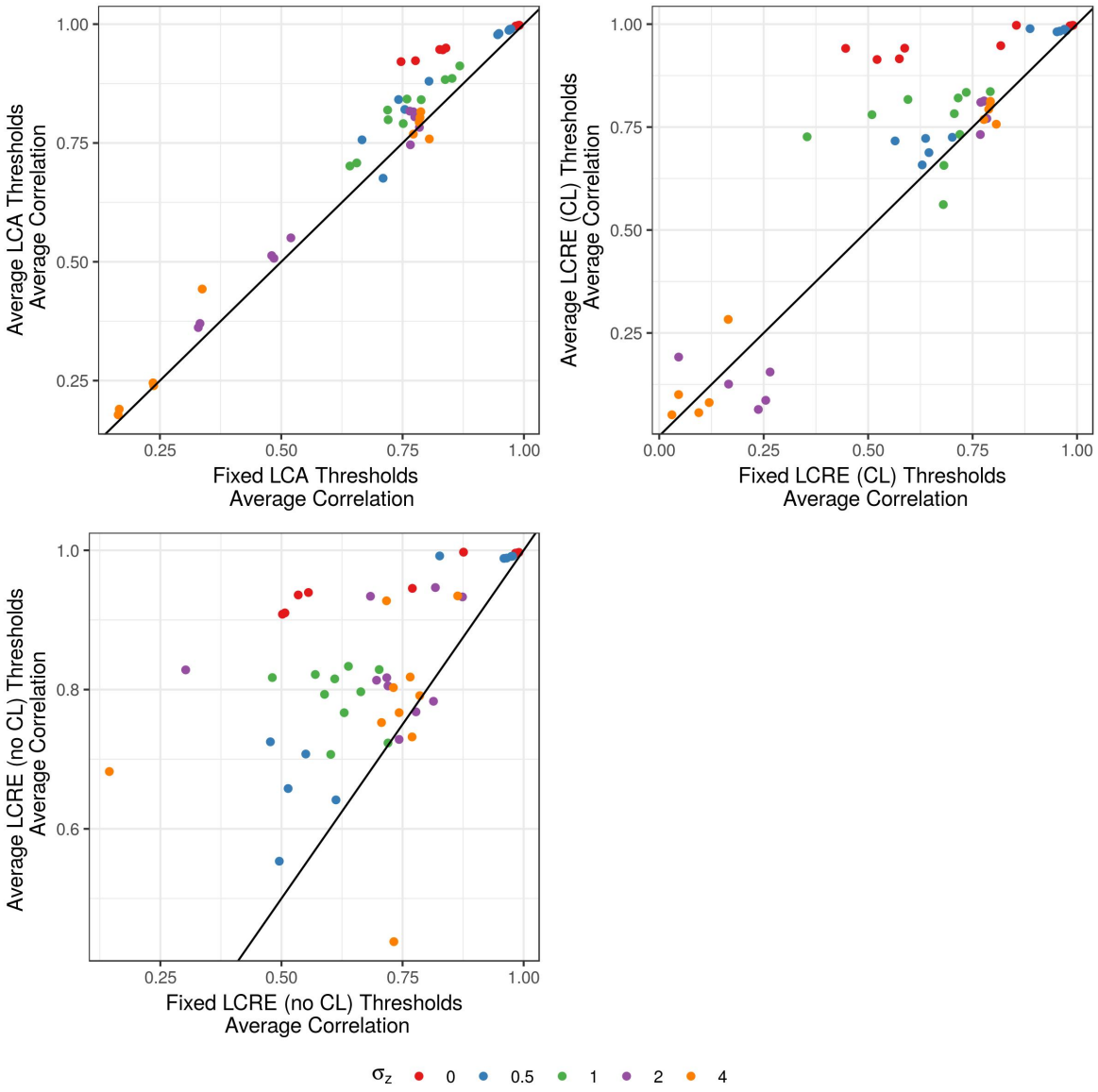


FIGURE 4.2: Average correlation to the MGMM model for fixed thresholds against average correlation to the MGMM model for averaged thresholds, for models LCA, LCRE (constant loading) and LCRE (non-constant loading) for different scenarios, coloured by degree of random effect (σ). A higher value on both axes indicates a better fit to the MGMM. Points above the $x = y$ line indicate that the average threshold method performed better, while points below indicate that the fixed threshold method performed better. Most of the time, the average threshold method appears to perform better.

	δ		σ_z					p_0					**Total**
	0.5	1.5	0	0.5	1	2	4	0.1	0.3	0.5	0.7	0.9	
Fixed	3	0	0	1	0	1	1	1	0	0	1	1	**3**
Average	20	24	10	9	10	8	7	9	9	9	8	9	**44**

TABLE 4.2: Frequencies of scenarios that found a higher average correlation for one of the two methods (fixed vs average thresholds) for the LCA model. Most found that the average was preferred.

a much lower average posterior probability for genes compared to the other combinations, reducing the average posterior probability. In this case, the fixed threshold method would perform better in terms of posterior probability. This may be more likely to occur when the clustering is less clear, such as when δ is low.

4.3.2 RMSE

I compared the different methods for each model with the RMSE, as shown in Figure 4.3. As before, the average threshold method performed better for most of the scenarios, or at least was competitive. The LCA model had the smallest improvement for the average thresholds method, but was also the most consistent, with only 9 scenarios performing better with the fixed threshold method. The LCRE models were more likely to perform better using a fixed threshold method, but in general, improvements in the RMSE were low. In contrast, using the average threshold method was more likely to lead to significant improvement, especially for the LCRE with constant loading model.

The highest RMSE for the average threshold method was less than 0.4, while the highest with the fixed threshold method was greater than 0.6. While most scenarios had some level of improvement under the average thresholds method, the scenarios with a lower value of σ_z were more likely to be improved compare to higher values.

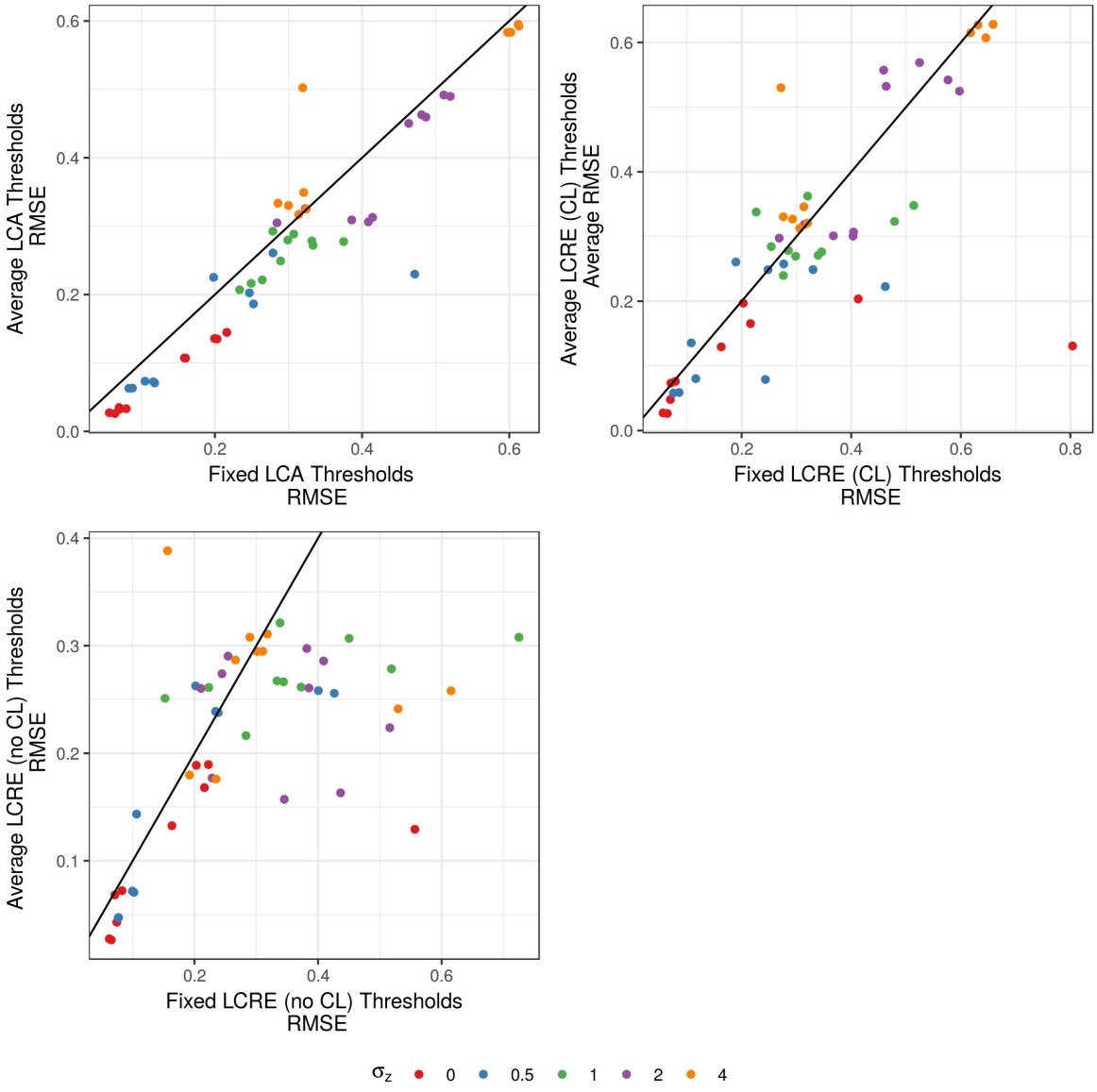


FIGURE 4.3: RMSE for the posterior probabilities for the MGMM model for LCA and LCRE (constant loading) and the LCRE (non-constant loading) for fixed and average threshold methods. A smaller value indicates a lower RMSE, which is preferred. Points above the $y = x$ line indicate scenarios where fixed threshold method performed better while points below indicate that the average threshold method performed better. For most of the scenarios, the average threshold method appears to perform better.

	δ		σ_z					p_0					**Total**
	0.5	1.5	0	0.5	1	2	4	0.1	0.3	0.5	0.7	0.9	
Fixed	8	1	0	1	1	2	5	2	3	1	2	1	**9**
Average	16	24	10	9	9	8	4	8	7	8	8	9	**40**

TABLE 4.3: Frequencies of scenarios that found a lower RMSE for one of the two methods (fixed vs average thresholds) for the LCA model. Most found that the average was preferred.

The overall trends for the LCA model for the RMSE results were collated in Table 4.3. While there were a greater number of scenarios that performed better under the fixed thresholds method (9 of the 50), most still found a lower RMSE for the average threshold method (40 scenarios). Similar trends were observed to those in Table 4.2. The two parameters that influenced the RMSE performance in either the fixed or average threshold method were δ and σ_z. Lower values of δ and higher values of σ_z in scenarios were more likely to prefer the fixed threshold method than other scenarios. This is likely a result of less distinct clusters within the underlying data. In this case, gene sets are more likely to contain incorrectly classified genes and generate a lower RMSE. A single fixed threshold containing more genes is thus more likely to be better than the average threshold results.

4.3.3 Binding Accuracy using p_0

Pairwise plots of the binding accuracy were generated for each model comparing the fixed and average threshold methods, and the results are given in Figure 4.4, coloured by the value of σ_z. The dashed lines indicate the ideal results for each method. While there were some points with binding accuracy greater than 2, these were omitted from the figure to gain a higher level of detail from the remaining scenarios. For all three models, there was a cluster of results close to the point (1,1),

	δ		σ_z					p_0					**Total**
	0.5	1.5	0	0.5	1	2	4	0.1	0.3	0.5	0.7	0.9	
Fixed	10	6	1	4	4	4	3	8	4	1	1	2	**16**
Average	11	7	4	2	4	4	4	1	5	4	5	3	**18**

TABLE 4.4: Frequencies of scenarios that were closer to the actual binding proportion(p_0) for one of the two methods (fixed vs average thresholds) for the LCA model. Both methods had approximately the same number of scenarios. The fixed method performed better for $p_0 = 0.1$.

indicating that most of the results were close to the correct estimation. Many of the points far from (1,1) show a positive correlation between the average threshold and fixed threshold methods, indicating that for most points with a lower binding accuracy, the two methods were competitive. The value of σ_z did not appear to have an effect on the binding accuracy. Furthermore, all three models and methods appeared competitive for binding accuracy.

To further investigate the binding accuracy, Table 4.4 was generated. This table identifies which method had the higher binding accuracy for each scenario (within 2 decimal places) and then looks at the frequencies of the different parameters within all the scenarios identified. Each of the two methods found an approximately equal number of scenarios with a higher binding accuracy, with 16 being competitive. This result suggests that both of the methods were reasonably competitive in terms of binding accuracy. There are few trends to be observed amongst the parameters, with the most notable that the lowest p_0 value, 0.1, was more accurately estimated by the fixed threshold method, while higher p_0 values were more accurately estimated by the average threshold method. This may have been because a smaller number of binding genes are harder to classify correctly using the average thresholds.

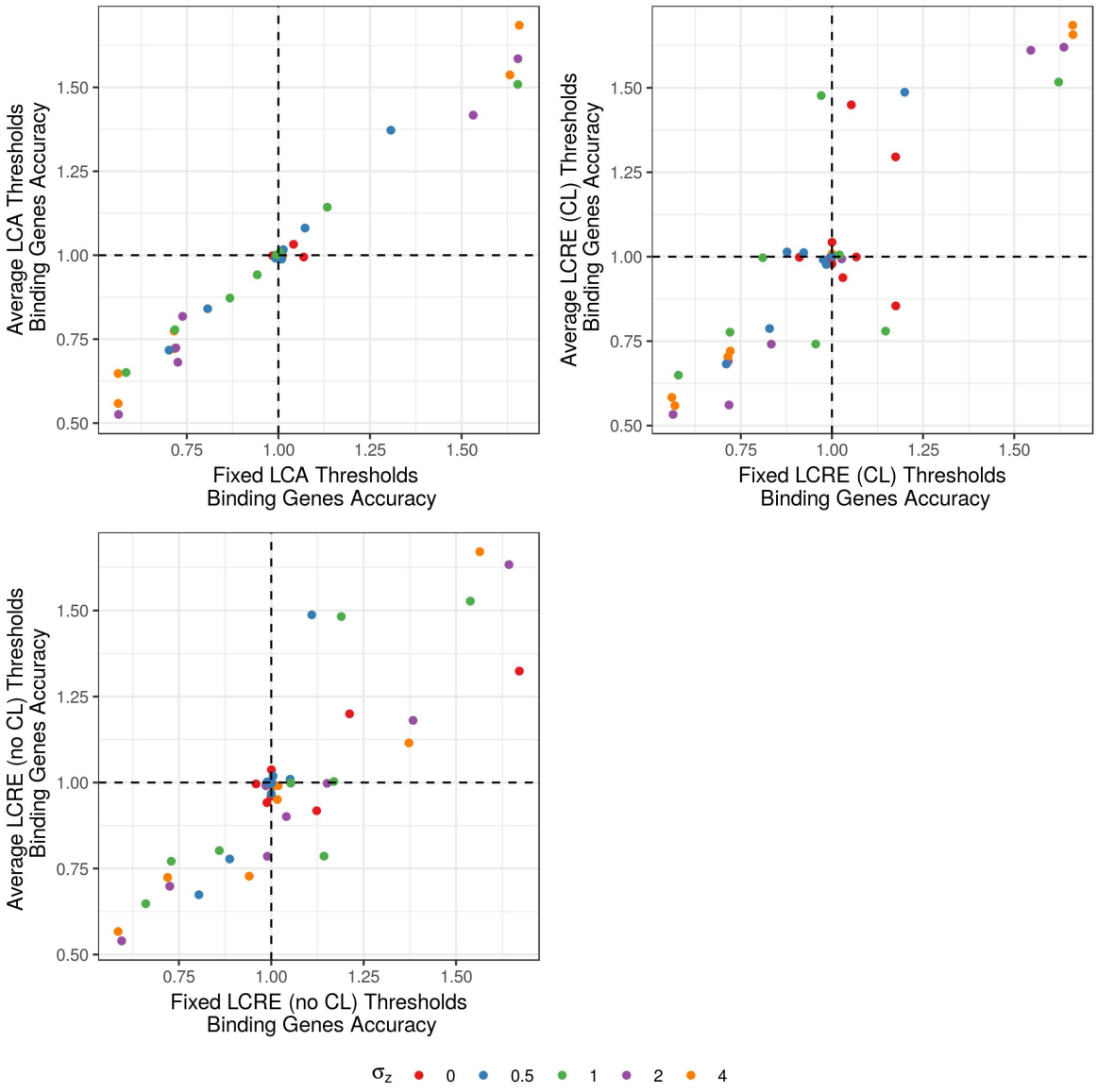


FIGURE 4.4: Average accuracy of binding proportion compared to set p_0 for fixed thresholds against averaged thresholds, for models LCA, LCRE with constant loading and LCRE with non-constant loading across different scenarios, coloured by degree of random effect (σ). Accuracy is measured as proportion to correct p_0 value. A value closer to 1 (dashed line) is thus preferred. The $x = y$ line indicates similarity of values between fixed and average threshold methods. In general, the methods were comparative, with most values lying close to (1,1), although in cases when the accuracy was poor, the fixed threshold method performed better. Points with binding accuracy greater than 2 were omitted.

	δ		σ_z					p_0					**Total**
	0.5	1.5	0	0.5	1	2	4	0.1	0.3	0.5	0.7	0.9	
Sum of Scores	11	12	0	1	4	8	10	5	4	3	6	5	**23**
Average	14	13	10	9	6	2	0	5	6	7	4	5	**27**

TABLE 4.5: Frequencies of scenarios that had higher correlations to the MGMM for one of the two methods (sum of scores vs average thresholds) for the LCA model. Both methods had approximately the same number of scenarios. The average method performed better for $\sigma_z < 1$.

4.3.4 Sum of Scores

The sum of scores was used again to compare to the results of the average threshold method. The pairwise correlation plots are given in Figure 4.5. The results indicated that the average threshold method generally showed improvement over the sum of scores method, with most scenarios showing an improved correlation to the MGMM. The LCRE with non-constant loading showed the best improvement over the sum of scores, particularly for σ_z values of 2 and 4. However, the LCA was consistently better for lower values of σ_z. This was consistent with previous results.

Additionally, trends were identified using Table 4.5 for when the sum of scores and the average threshold method with the LCA model performed better. These resulted was consistent with the observation above, and indicated that the average threshold method performed better for low values of σ_z. In total the average threshold method performed better for 54% of the scenarios.

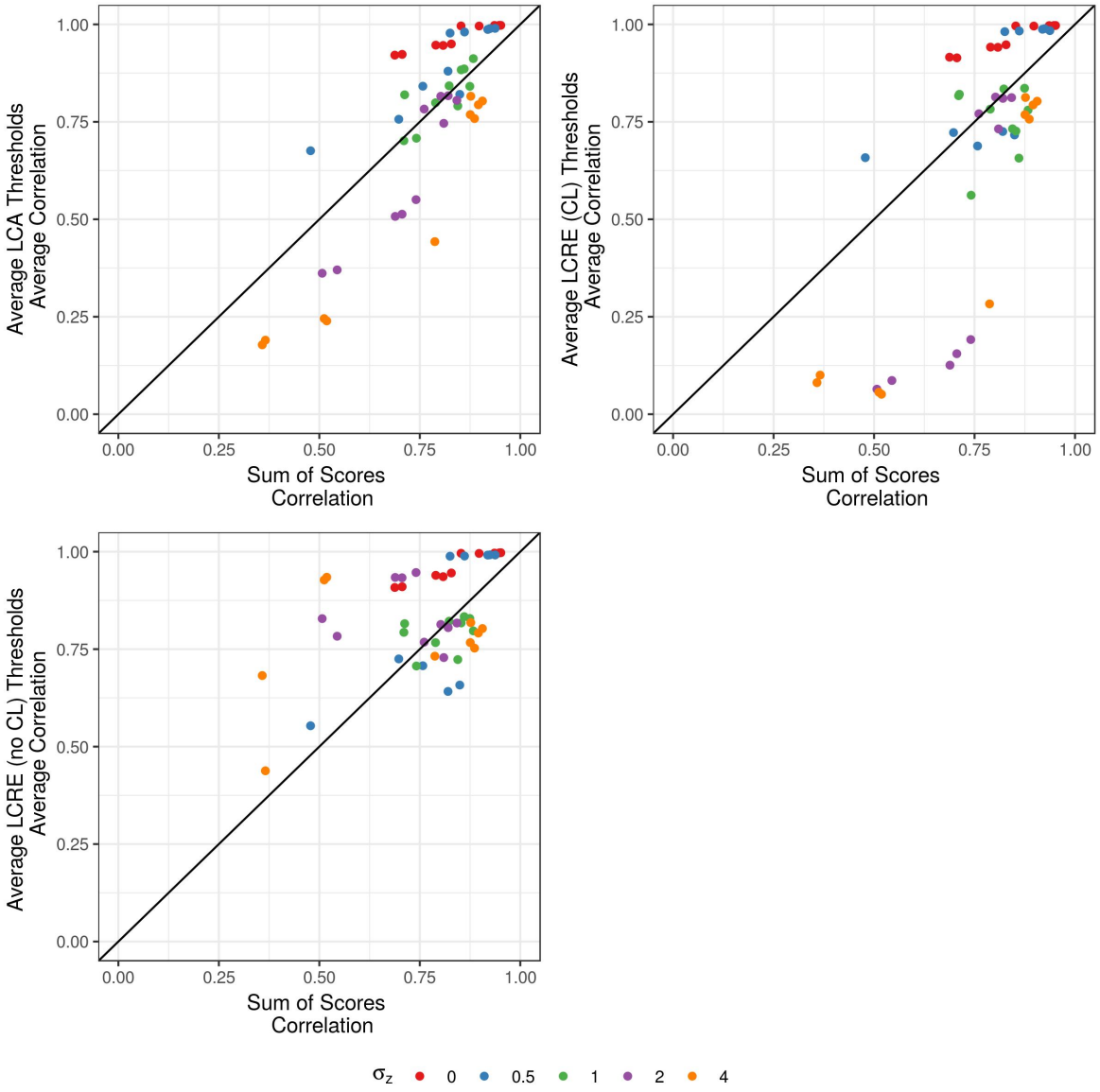


FIGURE 4.5: Correlation to MGMM for sum of scores compared to average correlation to MGMM for the average threshold method for each model. Points above the $x = y$ line indicate when the average threshold method performed the best, while points below indicate when the sum of scores method performed the best.

4.4 Conclusions

The proposed method of using thresholds to create multiple genes sets, and combine them to generate average posterior probabilities, appeared to improve upon the results from using a single fixed threshold. Examining the average correlation to the MGMM, as well as the RMSE indicated that for most scenarios, a modest to high improvement could be obtained using this method for all three models. This was particularly true for parameters that generated more distinct clusters of data points, such as a high δ. While the BIC of the average threshold method could not be calculated, an assessment of the accuracy of the estimated binding genes indicated that the average threshold method was competitive with the fixed threshold method. Furthermore, comparing the sum of scores and the average threshold method indicated that the LCA model in particular was more competitive than used the fixed models, as analysed in Chapter 3.

Based on the simulation results, the average thresholds method was a good approach to increase the accuracy of any of the LCA models, and is recommended in conjunction with the LCA model in particular.

Chapter 5

Applying new LCA method to data

5.1 Introduction

Based on the results from Chapter 4, I decided to apply the changing thresholds method to the original H3K36me3 data. When this method was applied to simulations, the posterior probabilities were improved and the LCA model gave the most consistent results. To use this method, new thresholds were applied to each of the programs, and combinations of thresholds were fitted with the different models. Another change to the original method was to omit the program enRich, since it had very little agreement with other programs as observed in Chapter 2. Furthermore, reducing the number of programs decreased the time taken to perform the analysis, and based on Chapter 3, including more than 5 programs did not improve the results.

Each model generated average posterior probabilities for each gene; these were compared, along with the standard deviations of the posterior probabilities across each gene. The posterior probabilities were then used to identify putative binding genes for each model, and the sets were compared. Finally I focused on genes found by the LCA model and identified significant GO terms. The results of this method were compared to the original results from Chapter 1.

5.2 Methods

In order to generate thresholds for each of the programs, I initially attempted to rerun the programs from the command line. All of the programs allow some degree of control over the final threshold of peak significance using the p-value, as can be seen in Table 5.1. However, I found that this was not a feasible method due to time constraints, as many of the peaks had extreme p-values that meant even very stringent thresholds did not lower the number of identified peaks.

Program	Threshold Measure	Threshold Modifier Flag
BCP	p-value	*–pval*
HOMER	p-value	*-poisson*
MACS2	$-log_{10}$ p-value	*-p*
MUSIC	q-value	*-q_val*
THOR	$-log_{10}$ p-value	*-p*

TABLE 5.1: Summary of thresholds for the different programs. Different programs use different measures for the threshold of peaks to retain, and this can be modified by the user using the threshold modifier flags during the command line input.

Instead, I ordered the already generated peaks identified by each program by score. The ordered peaks were used to create 4 overlapping sets of genes, where the number was constrained to 1000 or 4000 genes. In this way, the smallest gene lists should contain genes from each program that are the most likely to be binding, while the larger gene lists are less stringent. I confirmed that this would have the same effect as having extremely stringent controls by comparing the results from different program outputs. For MACS2, BCP, HOMER and THOR, the manual approach was equivalent in terms of the p-values observed. The program MUSIC,

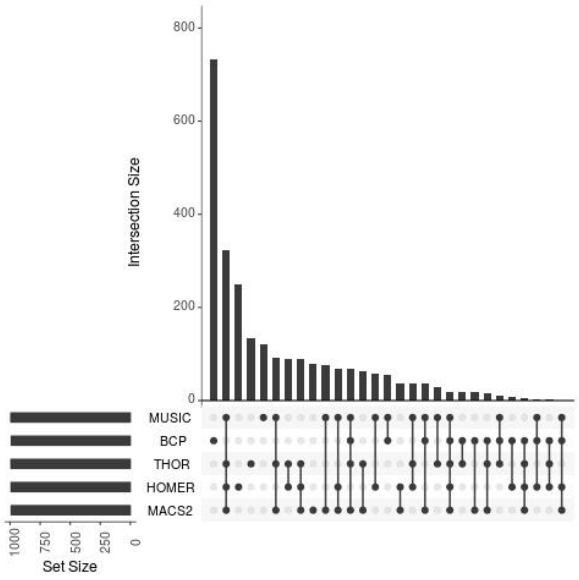
Program	1000 Gene List	4000 Gene List	Default Gene List
MACS2	1000	4001	12032
HOMER	1000	4006	9321
THOR	1002	4002	6470
BCP	1000	4009	8558
MUSIC	1000	4001	11542

TABLE 5.2: Gene lists for the different thresholds for the peak calling programs. Note that the number of genes are approximately close to the limit.

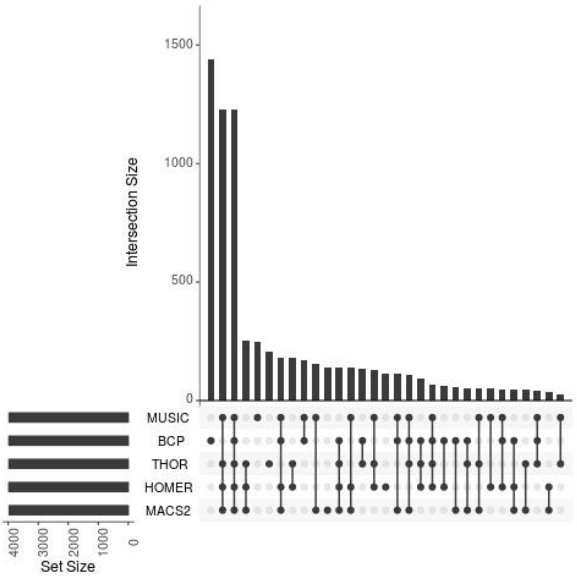
however, retained different peaks depending on the q-value specified. For simplicity, I continued to use the manual method for MUSIC. An additional list with all genes was also kept for each program. The number of genes in each list is given in Table 5.2. Note that for some of the gene sets, the number of retained genes is slightly higher than the threshold limit. This occurred when reducing the number of scored peaks reduced the number of genes to below the limit for that gene set.

I compared the genes found in each list to all the programs, resulting in the UpSet plots in Figure 5.1. As the number of genes increased there was a higher level of agreement between programs. The program BCP in particular had a large number of genes that it found uniquely, as did HOMER. This is most noticeable in Figure 5.1a.

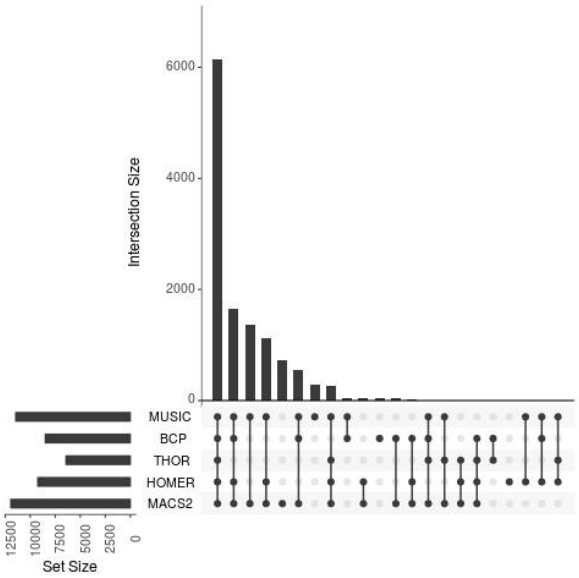
For the three models, I used the different thresholds in different combinations, as in the Simulation methods section. Only three of the gene sets generated were used in this case; 1000, 4000 and the default, to reduce the number of permutations and capture a range of stringency within the thresholds. This meant that there were 243 different permutations to perform. I obtained the posterior probabilities for each permutation, and averaged them to attain an overall posterior probability.



(A) Intersections of 1000 genes lists



(B) Intersections of 4000 genes lists



(C) Intersections of default genes lists

FIGURE 5.1: UpSet plots showing the intersections and therefore level of agreement for the gene lists with different thresholds. There appears to be a greater level of agreement in the larger gene lists

5.3 Results

I performed an LCA, LCRE with constant loading and LCRE with non-constant loading model fit in order to compare the results and determine if similar results as the simulation were observed. The average posterior probability was plotted for each model and the results are given side-by-side in Figure 5.2. Similar distributions were observed across all three models, with a higher density section found at an average posterior probability of 0.3. The LCA model found a second high density point at approximately 0.7, indicating that there were two distinct groups of genes with different posterior probabilities, with few genes in between. In contrast, the LCRE models did not have any other high density sections, but instead several medium density regions between 0.1 to 0.5. All three models also had a low range of values, with no posterior probability greater than 0.75. This implied that many of the genes had low posterior probabilities for at least some of the combinations. This may have been due to the lack of agreement observed in the 1000 gene set, which may have decreased the posterior probability for some of the genes.

To directly compare the average posterior probabilities for each gene, pairwise plots comparing the three models were generated (see Figure 5.3). The amount of correlation between the LCA and the two LCRE models was low, particularly for genes that were given a low average posterior probability by the LCA model. In comparison, the LCRE models had a much higher level of agreement, with most points falling close to the $x = y$ line. The LCRE posterior probabilities were also more evenly distributed across both axes compared to the posterior probabilities of the LCA model, where most points have an average posterior probability of approximately 0.7. Additionally, the LCA model found a number of genes with posterior probabilities of approximately 0.3. The LCRE models tended to have a wide range of posterior probabilities, extending from 0.35 to 0.9 for the constant loading model.

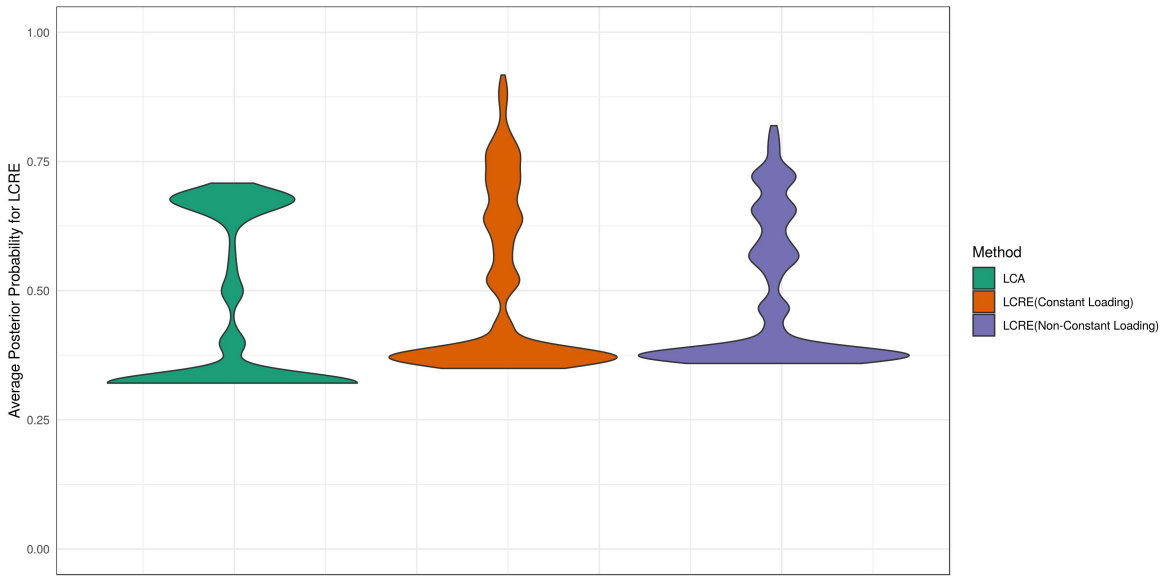


FIGURE 5.2: Average posterior probabilities over different thresholds for genes from H3K36me3 data, separated by model (LCA, LCRE (CL) and LCRE (NCL)) and displayed as a violin plot. The y-axis gives the average posterior probability, and the width of the violins along the x-axis determines the number of genes with that average posterior probability. There are a large number of genes with a posterior probability of 0.7 across the three models.

In contrast, the LCA range extended from 0.35 to 0.7. This indicated less variation for the combinations of gene sets tested for these models.

Each gene has 243 posterior probabilities, one for each combination of thresholds. To further investigate the differences between the models, I calculated the standard deviation of these posterior probabilities for each gene. This is given in Figure 5.4. The genes ranged in standard deviation from 0.2 to 0.5. The LCRE models generally had lower average standard deviations, and a greater range of standard deviations. The LCA model results had the lowest range, and the highest average standard deviations. This high level of variability in the posterior probability are most likely due to the lack of agreement in the smaller gene lists. This meant some permutations of thresholds gave genes very different profiles, resulting in different

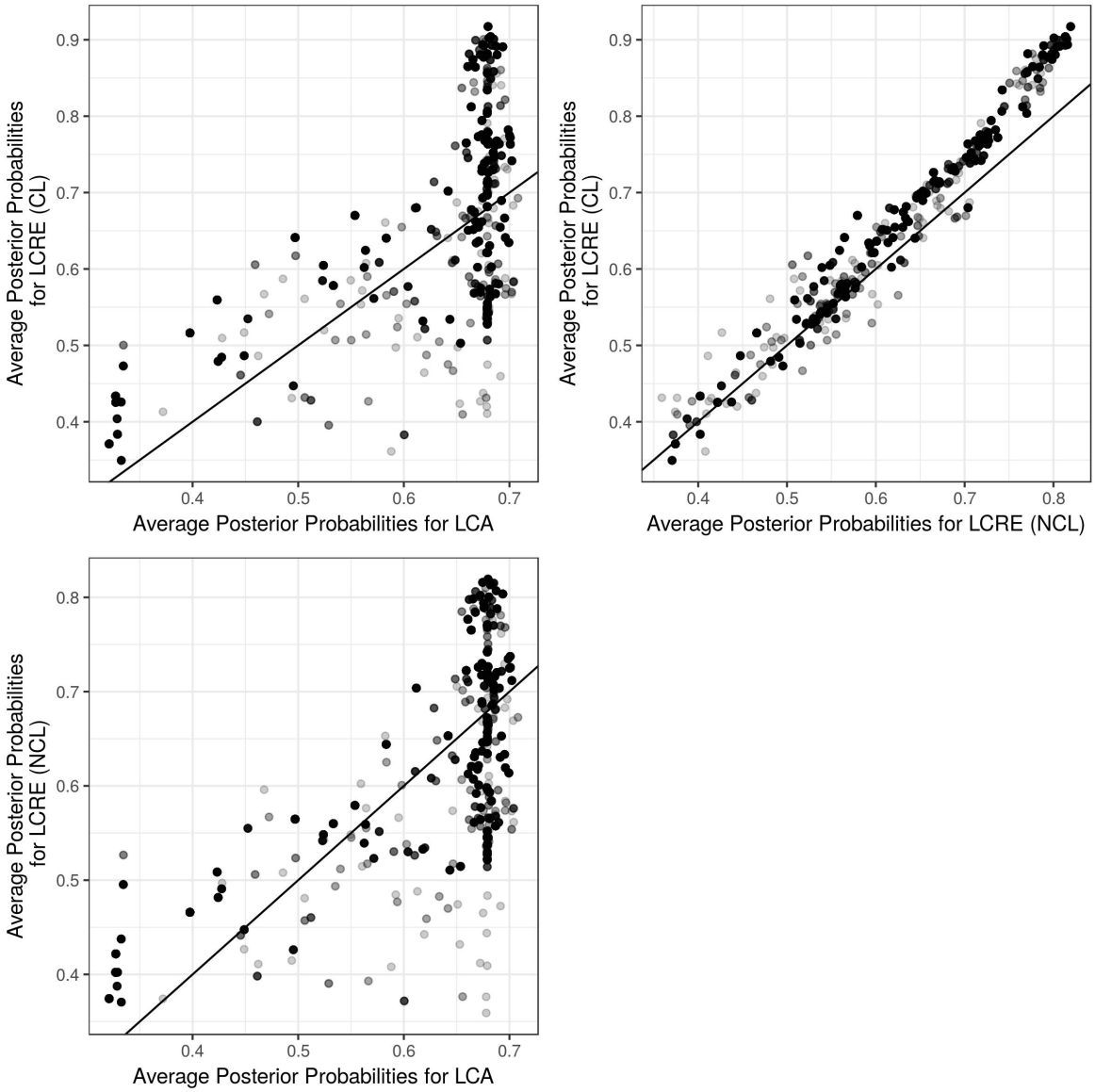


FIGURE 5.3: Average posterior probabilities over different thresholds for genes from H3K36me3 data, plotted pairwise for the 3 models. An $x = y$ line is included. Points are given a light colour to indicate overlapping; black indicates more than five superimposed points. There is a positive correlation between the LCRE (CL) and LCRE (NCL) results.

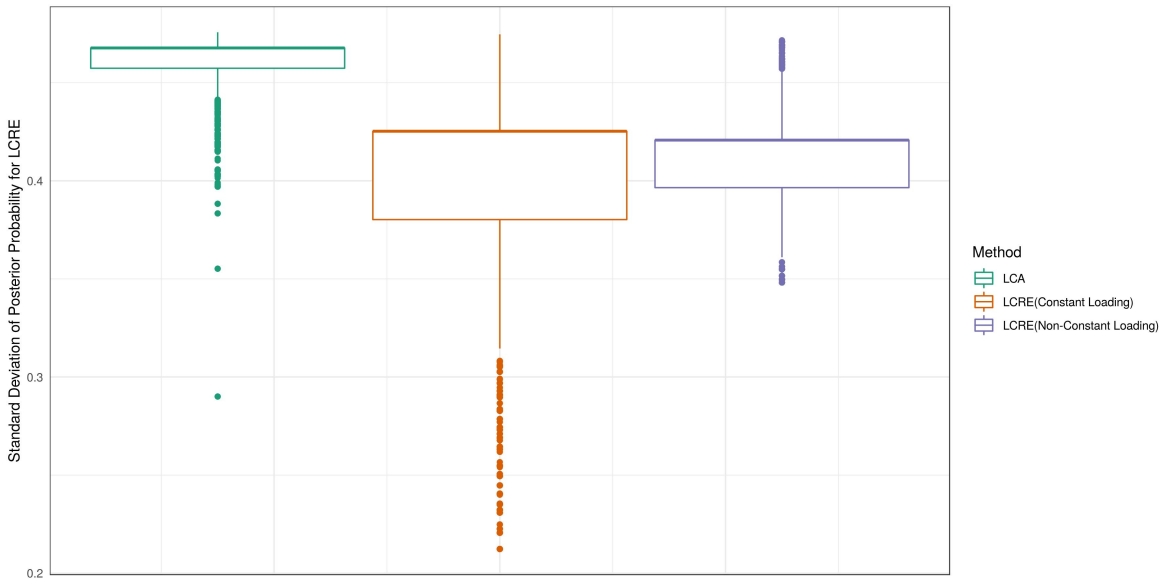


FIGURE 5.4: Standard deviations of posterior probabilities over different thresholds for genes from H3K36me3 data by model. Standard deviations are collated as boxplots. The LCA model has the highest average standard deviation across the genes.

posterior probabilities for presence in a particular class. Furthermore, this also explained why the maximum posterior probability for all three models was low, and the minimum posterior probability was high, relative to the possible range. If differences within genes are large, this results in the average moving away from the extremes.

Next, I examined the genes that were associated with binding for each model. This was calculated by classifying all genes with a posterior probability greater than 0.5 as binding, and those with a posterior probability less than 0.5 as non-binding. The resulting gene sets for each model were compared using the Entrezgene ID, and the Venn diagram shown in Figure 5.5 was constructed. The figure indicates that there was a large amount of correlation across all three models, and the LCRE with non-constant loading in particular found no unique genes compared to the other models. In contrast, the LCRE models in Chapter 2 found almost identical gene sets.

The LCA model found 8779 genes were binding, the LCRE model with constant loading found 11,198 and the LCRE with non-constant loading found 9862. The majority of these (8733) were found by all three models. Notably, these numbers are comparable with the original gene lists identified by the fixed thresholds method with the models.

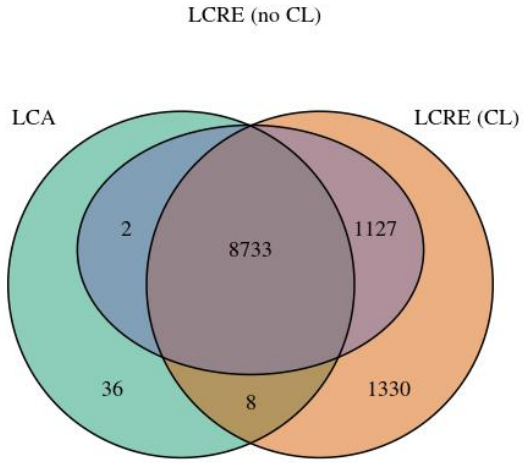


FIGURE 5.5: Venn diagram of binding genes based on the three LCA models; LCA, LCRE with constant loading and LCRE with non-constant loading. The results indicate a high level of agreement (8733 were found by all genes) the LCRE with non-constant loading found no unique genes.

Focusing on the results for the LCA, I obtained the significant GO terms for these genes, as I did with the original LCA and LCRE models (see Figure 5.6). The significantly enriched GO terms are different to those found in Chapter 2 when using the fixed threshold method. Thus, despite the similarities in the number of genes identified as enriched, there was a difference in the set of genes found by the two different methods. These GO terms still appeared to be associated with transcriptional activity, as noted in Chapter 2.

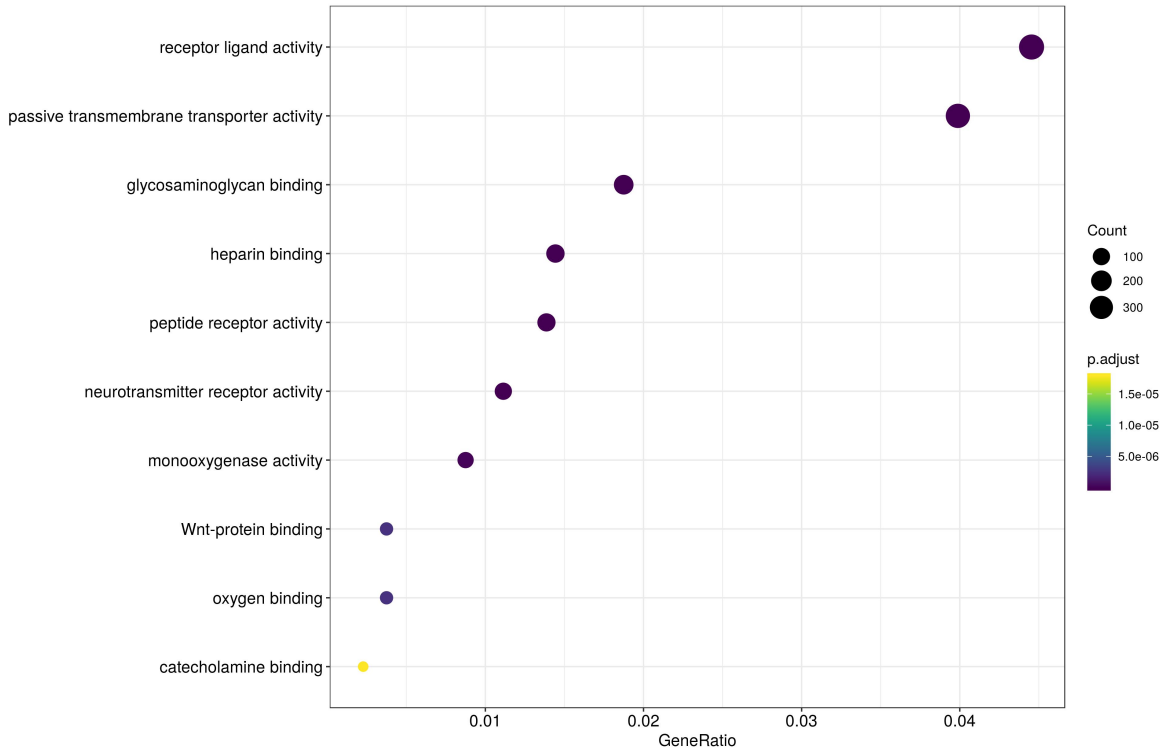


FIGURE 5.6: Significant GO terms for the LCA model using the average threshold method. While a comparative number of genes were identified as binding overall, different significant GO terms were found.

I then compared the gene list from the average threshold method LCA to the fixed threshold method LCA. Despite the differences in the GO terms found, the genes had a large amount of overlap, with around 8746 genes being in common. This also indicated that the average threshold resulted in a more stringent number of genes, and gives high confidence for the resulting set. Notably, using a sum of scores method will give the same results as the fixed threshold LCA, so this is also the overlap between the average threshold method and the sum of scores method as well.

Due to constraints with this method, the actual and expected number of genes could not be calculated, as the profile of each gene varied in the combinations depending on its score for each program. However, it is clear that this method is useful

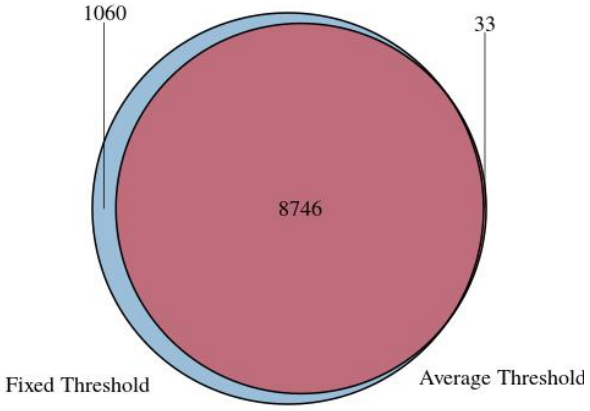


FIGURE 5.7: Venn diagram for Entrezgene IDs for the genes found with the fixed threshold LCA method (from Chapter 2) and the genes found with the average threshold LCA method

and can be used to combine the results from multiple ChIP-seq programs in a more reliable way.

5.4 Conclusions

In this chapter, I applied a new method developed as part of Chapter 4 to the experimental ChIP-seq data. This method used varying thresholds to create several sets of genes for each program, representing low, medium and high stringency, where a high stringency would only retain the most likely genes for binding. Each combination of the gene sets for all the programs was analysed using the three different LCA models, with the resulting posterior probabilities from each combination averaged for each gene.

The resulting average posterior probabilities were compared for the three models. All three models had a relatively small range of posterior probabilities, particularly the LCA model. A large number of genes were found to have low posterior probabilities for all three models. The LCA model had a similar cluster of genes with high posterior probabilities, while the LCRE models tended to have more genes along the range of posterior probabilities, with only a few gaining the maximum value. Analysis of the standard deviation indicated that the LCA also had the highest standard deviation, which may have resulted in the smaller maximum value of the posterior probabilities.

I generated putative binding genes for each model using the average posterior probabilities. Comparisons of these sets indicated that about 9000 genes were found by all three models. Based on the results of Chapters 2, 3 and 4, I selected the LCA model for further analysis. I identified significant GO terms for the LCA putative gene binding set, and found that the results were different to those found in Chapter 2. However, these GO terms still confirmed that this gene set was associated with basal transcriptional activity, as expected.

When comparing the fixed threshold LCA gene set and the average LCA gene set, it was found that there was a large amount of overlap in the genes identified as putatively binding. This meant that the genes found by the average LCA gene set can be considered high confidence.

While all three models are useful, the LCA model is the most practical in conjunction with the average thresholds method, because it estimates the parameters much more quickly.

There are some limitations to this application of the threshold methods. The threshold was kept at the default for the 5 programs, so this makes the assumption

that the optimal threshold is at the default or lower, which may not be the case. Similarly, if the thresholds chosen are all poor, then this would degrade the performance further. An extension to this study would be to change the default threshold value using the threshold modifiers described in Table 5.1, and then generate thresholds using the method described in 5.2. The results would be useful as a comparison and would provide more insight into the gene lists.

Another limitation is that the target protein of the ChIP-seq data set used identifies transcriptionally active genes, but no condition was changed during the experiment. An experiment that has a known outcome (for example, the activation of ion channels) prior to the use of LCA may be helpful for measuring the effectiveness of this method.

Overall, the results of this chapter indicated that the average threshold method provides useful results.

Chapter 6

Conclusions and Future Directions

ChIP-seq is a popular tool for identifying binding regions of proteins within the genomes. Applications of this technology include finding transcriptionally active regions with histone marks, as well as the effects of an added treatment over time for particular proteins. The results of ChIP-seq vary in both peak characteristics and quality, and therefore subsequent analysis is complex.

It is difficult to determine the best program to use on ChIP-seq data, even when accounting for the expected type of peak. The surprising amount of variation between programs, and even within programs using different settings, means there is often a low level of agreement on genes associated with binding sites. This thesis explored methods of identifying putative binding genes for the purposes of further investigation, in particular through the application of LCA and LCRE (with constant loading and with non-constant loading) models.

Conclusions

In Chapter 2, a data set obtained from ENCODE was analysed using a number of different ChIP-seq programs. Peaks were associated with genes in order to compare between programs easily. The programs had varying levels of agreement, making this data set ideal for testing the LCA models. Three different models were

compared, increasing in complexity; LCA (without random effects), LCRE with constant loading, and LCRE with non-constant loading. The BIC was used to identify whether the two class or one class fit was more appropriate for each of the models, with the assumption that a one class fit would occur when the data could not sufficiently be classified into two classes.

It was initially expected that the LCRE would more closely fit the data, as this model accounts for the anticipated correlation found between programs as a random effect. While the LCRE models improved the fit to the data, the result tended to rely too heavily on particular programs, such as enRich. Furthermore, the BIC of the LCRE with non-constant loading models indicated that a one class model should be preferred. Despite the assumed dependence between programs, the LCA provided the most reliable results, and was the most practical to use when considering the time to estimate the parameters. Questions about the use of BIC to identify the correct number of classes remained.

In Chapter 3, simulated data was generated to investigate the 3 LCA models across a range of scenarios. The parameters varied across these scenarios included the extent to which genes were separated into two groups, the amount of random effect present within the data, the number of programs used, and the proportion of binding genes within the data. To assess the models, an MGMM was used as a "gold-standard" using the true underlying scores. Measures of the performance of the three models included the average correlation to the MGMM for each scenario, the RMSE compared to the MGMM, and the BIC.

The LCA model was found to perform best even when moderate levels of random effect were included. The LCA generally had higher average correlations, and lower RMSE, than the other two models, unless the random effect was large in magnitude, in which case the LCRE with non-constant loading was the better choice of

model. The LCRE models tended to attain better BIC values for most scenarios.

Next, the method of using the BIC to select a model, rather than always using the same model was assessed. The BIC for all the models for each scenarios were compared, and the average correlation to the MGMM and RMSE were retained for the preferred model. This was found to produce competitive results for most scenarios, however did not improve the results found when using the sum of scores. The drawback of this approach is that it involves estimating the parameters for all three models, which is more time consuming than selecting a single model. Furthermore, in most cases the LCA was still an appropriate model, unless the random effect was large in magnitude. Thus using the simple LCA is preferred over comparing the models' BIC (when using a two class model) to identify the best model for a particular data set.

To assess the reliability of the BIC for choosing a one class or two class model, these different models were also tested with the simulated data. When the simulated data was generated with two classes, the LCA always identified the two class model as best, but the LCRE models were less consistent, particularly the model with non-constant loading. In contrast, when the simulated data was generated with a single class, the LCRE model with non-constant loading most accurately identified this correctly, while the LCA and LCRE with constant loading showed mixed results. It was concluded that comparing a one class and two class model BIC was found to be an unreliable way to determine whether the two class model was appropriate.

When the LCRE BIC suggested that a one-class model is more appropriate for the data, it was considered that this may degrade the performance of the two class posterior probabilities. However, the simulation studied showed no such effect and I concluded that an LCRE BIC that favours a one class model is not diagnostic of poor performance. Overall when analysing multiple ChIP-seq programs, it is more

realistic to assume a two class model should be used.

In Chapter 4, I developed a new method of using the LCA model. In this method, different thresholds were used to create new gene lists with varying levels of stringency for each program. The LCA models were then used to analyse each combination of the gene lists, and the resulting posterior probabilities were averaged. When tested on the simulated data, this led to very consistent posterior probabilities that relied less on random changes to the data. Furthermore, the results showed improvement for most scenarios for each of the three models. However, this method is also time consuming, and is most appropriate when used in conjunction with the LCA model, which is fast compared to the LCRE models.

Finally, in Chapter 5, the changing threshold method was applied to the original data, based on the conclusions from Chapter 3. Comparisons between the three models indicated that the LCA tended to have greater variation in the probabilities for each gene, and smaller posterior probabilities overall. In comparison to the LCRE models, the LCA had two distinct groups of genes, with either high or low posterior probabilities.

The average posterior probabilities were used to identify a number of binding genes, which showed general agreement between the three models. The binding genes that were found by the LCA model were used to identify significant GO terms. The results indicated that these genes had different functions to those found in Chapter 2. I concluded that this method provided consistent results, and generates a set of putative binding genes that can be used for further analysis.

Future Directions

Further research is required to fully take advantage of LCA in conjunction with ChIP-seq data.

The LCA model should be applied to a greater range of data sets. The H3K36me3 data used in this thesis was chosen because it was high quality and was well documented, however testing with a number of other data sets, including those with different peak characteristics such as DNA binding proteins will give greater insight into the affects of different types of peaks on DNA. Furthermore, data sets where there is a known change in the condition of cells between different samples would allow a better understanding of the differences in gene sets between the fixed threshold LCA method and the average threshold LCA method. For example, if the only genes found are associated with depolarisation activity after the addition of potassium depolarisation solution, this would provide greater confidence for the LCA model. Applying the LCA model to a ChIP-seq data set to identify changes under different condition is also the most likely application of this method. Applying LCA to a number of different data sets would therefore allow for better testing of the model in a range of conditions. Additionally, as mentioned in Chapter 5, using the same dataset but changing the default threshold may also provide greater insight into the method outcomes.

More extensive simulations are needed to fully explore this model. In Chapter 3 and 4, simulations were used to test the accuracy of the 3 LCA models when the underlying truth about the data was known. As mentioned in that section, there are some limitations in the results because the model makes assumptions about the distribution of the underlying data that may not be accurate. Therefore, testing a wider variety of parameters and different extensions to the model may lead to more relevant results. In particular, increasing the number of genes would be beneficial, since this affected the results of the BIC, one of the measures tested in Chapter 3. In addition, alternative measures to the BIC should be explored, such as a bootstrap likelihood ratio test (McLachlan, 1987).

A greater variety of parameter options would also be beneficial. Testing different simulation data generation models would also be useful, for example testing alternative score distributions. Furthermore, the use of the program enRich in Chapter 2 indicated that a low agreement and small gene set result can influence the LCRE models in particular. This was not explored in the simulations, which assumed that all of the programs were equally accurate. Thus testing the results of the LCA models when one or multiple programs has low agreement may provide further insight into these effects.

Finally, the merits of using an ordinal response LCA should be considered as an alternative to the average threshold method. The original proposal to use LCA as a means to integrate the results from several calling programs or across several studies assumes a binary outcome from each program (Cantarel et al., 2014; Elsik et al., 2007; Chen et al., 2007). For example, each gene is classified as either binding or non-binding by each calling program in the present context. When multiple thresholds are introduced, as introduced here, the outcomes can then be considered as ordinal rather than binary. A more standard statistical approach is then to consider LCA models for ordinal rather than binary data, see Agresti and Lang, (1993). In principle, that approach could be expected to make optimal use of the data and produce more efficient estimates of the posterior probabilities than those obtained from the averaging method considered in this thesis. The application of LCA with ordinal responses was not undertaken in this thesis due to difficulties with implementation. Nevertheless, the evaluation of such methods is a logical next step and an important area for future research.

Final Recommendations

The results of the thesis indicate that the LCA is a promising model for the purpose of combining multiple ChIP-seq peak finding identification programs to create

a set of putative binding genes. While three different LCA models of increasing complexity were considered, the simplest LCA model without random effects not only had the most reliable performance when applied to real ChIP-seq data, but also had the most accurate results for most of the scenarios in Chapter 3.

A further recommendation would be to use the average threshold method developed in Chapter 4 and applied in Chapter 5. The results indicated that this, when used in conjunction with the LCA model, was more accurate than the fixed threshold method.

In addition, further research is needed to fully explore the application of LCA to ChIP-seq data, including more extensive simulations, the applications to a greater variety of ChIP-seq data sets and the investigation into an ordinal response LCA.

Appendix A

Software

This appendix describes all of the software and other resources used in this thesis, and includes the scripts used to generate and analyse the data, where possible. Code used can be found in the digital appendix on GitHub at https://github.com/catisha/Thesis_Code/ under the subheadings given here:

A.1 ChIP-seq Peak Identification Software

MACS2

MACS2 version 2.1.0.20151222

See `macs2_all_samples.sh` for commands.

HOMER

HOMER v4.10.1

See `homer_analysis.sh` for commands.

THOR

THOR version 0.11.3

See `THOR_analysis.sh` and `H3K36me3_THOR.config` for commands.

enRich

enRich version 3.0

See:

- `enrich_all_chromosomes.sh`
- `enrich_mycounts_allchromosomes_human_neutrophil_pheonix.R`
- `combine_chroms_enRich.R`

for commands.

Note `enrich_mycounts_allchromosomes_human_neutrophil_pheonix.R` was run on R version 3.4.1 while `combine_chroms_enRich.R` was run on R version 3.5.1.

MUSIC

See `MUSIC_analysis.sh` for commands.

BCP

BCP version 1.18

See `BCP_analysis.sh` for commands.

A.2 R Software and Scripts

A.2.1 Chapter 2

R version 3.5.1 (2018-07-02)

Platform: x86_64-pc-linux-gnu (64-bit)

Running under: Ubuntu 16.04.5 LTS

R scripts (see Digital Appendix):

- `macs_analysis_slim.Rmd`: Takes peak information generated by the MACS2 program and identifies associated peaks. Also generates thresholds for the program for Chapter 4.
- `HOMER_analysis_slim.Rmd`: Takes peak information generated by the HOMER program and identifies associated peaks. Also generates thresholds for the program for Chapter 4.
- `THOR_analysis_slim.Rmd`: Takes peak information generated by the THOR program and identifies associated peaks. Also generates thresholds for the program for Chapter 4.
- `enRich_analysis.Rmd`: Takes peak information generated by the MUSIC program and identifies associated peaks.
- `MUSIC_analysis_slim.Rmd`: Takes peak information generated by the MUSIC program and identifies associated peaks. Also generates thresholds for the program for Chapter 4.
- `BCP_analysis_slim.Rmd`: Takes peak information generated by the BCP program and identifies associated peaks. Also generates thresholds for the program for Chapter 4.
- `lca_random_programs.Rmd`: Combines genes lists from the programs and tests the LCA, LCRE (CL) and LCRE (NCL) models with enRich and without enRich. Generates figures used in Chapter 2.

A.2.2 Chapter 3

Simulation Generation

R version 3.4.1 (2017-06-30)

Platform: x86_64-pc-linux-gnu (64-bit)

Running under: Ubuntu 16.04.5 LTS

R scripts (see Digital Appendix):

- `LCA_sim_with_lca_record_commandline_conLoad_pcoef.R`: Generates simulation data based on Chapter 3 Methods with a varied threshold and two clusters of scores.
- `LCA_sim_with_lca_record_contThresh_commandline_conLoad_pcoef.R`: Generates simulation data based on Chapter 3 Methods with a constant threshold and two clusters of scores.
- `LCA_sim_with_lca_record_commandline_conLoad_pcoef_oneclass.R`: Generates simulation data based on Chapter 3 Methods with a varied threshold and one cluster of scores.
- `LCA_sim_with_lca_record_contThresh_commandline_conLoad_pcoef_oneclass.R`: Generates simulation data based on Chapter 3 Methods with a constant threshold and one cluster of scores.

Commandline Scripts (see Digital Appendix):

- `simulate_LCA_commandline_3methods.sh`: Used to run the top two scripts above with a range of parameters.
- `simulate_LCA_commandline_1cluster_3methods.sh` Used to run the bottom two scripts above with a range of parameters.

Analysis

R version 3.5.1 (2018-07-02)

Platform: x86_64-pc-linux-gnu (64-bit)

Running under: Ubuntu 16.04.5 LTS

R scripts (see Digital Appendix):

- `combined_comparing_parameters_simulation.R`: Takes the simulation data for both varied and constant thresholds, and generates results for the correlation MGMM, the RMSE, the BIC and the sum of scores, as well as additional analyses that were not used. (Section 3.3.1)

- `choose_by_BIC.R`: Analysis comparing the results when using the BIC based model compared to the results using one of th other LCA models. (Section 3.3.2)

- `one_class_vs_two_class_analysis.R`: Compares the BIC for a one class model vs a two class model and generates results used for assessment of the BIC. (Section 3.3.3)

- `nocluster_one_class_vs_two_class_analysis.R`: Compares the BIC for a one class model vs a two class model when the underlying only has one cluster and generates results used for assessment of the BIC. (Section 3.3.3)

A.2.3 Chapter 4

Average Threshold Method Simulation Generation

R version 3.4.4 (2018-03-15) on Pheonix

Platform: x86_64-pc-linux-gnu (64-bit)

Running under: Red Hat Enterprise Linux Server 7.5 (Maipo)

R Scripts (see Digital Appendix):

- `LCA_sim_with_lca_record_change_Thresh_commandline_conLoad_pcoef.R`: Generates simulation data for motivating example used in section 4.1.

- `LCA_sim_3_thresholds_commandline_20rep.R`: Generates simulation data based on Chapter 4 Methods for the average threshold method.

Commandline Scripts (see Digital Appendix):

- `simulate_LCA_3methods_thresholds_commandline.sh`: Used to run the above R script for a variety of parameters.

Analysis

R version 3.5.1 (2018-07-02)

Platform: x86_64-pc-linux-gnu (64-bit)

Running under: Ubuntu 16.04.5 LTS

R Scripts (see Digital Appendix):

- `one_scenario_different_thresholds_compare.R`: Generates plot for motivating example used in section 4.1.
- `ave_threshold_results.R`: Takes the simulation data of the average thresholds and generates results using the correlation to the MGMM, the RMSE, the sum of scores and the binding gene accuracy for comparison.
- `compare_thresholds_ave_fixed.R`: Takes the results from the previous R script and compares them to the results from `combined_comparing_parameters_simulation.R`.

A.2.4 Chapter 5

R version 3.5.1 (2018-07-02)

Platform: x86_64-pc-linux-gnu (64-bit)

Running under: Ubuntu 16.04.5 LTS

R Scripts (see Digital Appendix):

- `lca_peak_thresholds.Rmd`: Takes the threshold data from the 5 programs (MACS2, HOMER, THOR, MUSIC and BCP) and applies the average threshold data described in Section 5.2. Analyses and produces the results shown in Section 5.3.

A.3 Other Software

ENCODE

This data is originally from www.encodeproject.org. It was chosen because it was a histone mark, was ansiogenic and has no known significant issues, according to the website.

- The H3K36me3 data can be found at: https://www.encodeproject.org/experiments/ENCSR373WCB/ at the bottom under Processed Data. I selected the filtered alignments. 1 or 2 indicates which anisogenic replicate it is.
- The Control data can be found at: https://www.encodeproject.org/experiments/ENCSR557RDB/ at the bottom under Processed Data. I selected the filtered alignments. 1 or 2 indicates which anisogenic replicate it is.
- Details of how the ChIP-seq reads were mapped can be found at: https://www.encodeproject.org/pipelines/ENCPL220NBH/
- Details of processing after mapping can be found at: https://www.encodeproject.org/pipelines/ENCPL272XAE/
- General information on the histone analysis can be found at: https://www.encodeproject.org/chip-seq/histone/#histone

Ensembl BioMart

To identify genes, the R package `biomaRt` was used to access the Ensembl BioMart database. To match the time of the H3K36me3 generation and mapping, an older version of the gene database is used, http://mar2016.archive.ensembl.org/index.html. More information about the package can be found at https://bioconductor.org/packages/release/bioc/html/biomaRt.html.

Phoenix

A portion of this thesis was completed using supercomputing resources provided by the Phoenix HPC service at the University of Adelaide, in particular the generation of data used in Chapter 4, and the analysis of H3K36me3 using the program enRich.

Appendix B

Chapter 3 Full Results

B.1 Comparing the Simple LCA and LCRE models Results

B.1.1 Average Correlation to MGMM and Standard Deviation of Correlation to MGMM for three models

p	δ	σ_z	p_0	Threshold	LCA ave	LCA sd	LCRE.CL ave	LCRE.CL sd	LCRE.NCL ave	LCRE.NCL sd
5	0.5	0	0.1	Varying over Programs	0.74679	0.01902	0.57464	0.19805	0.50201	0.30988
5	0.5	0	0.3	Varying over Programs	0.82643	0.00870	0.58748	0.25154	0.55566	0.30562
5	0.5	0	0.5	Varying over Programs	0.83928	0.00813	0.81738	0.09347	0.77021	0.11122
5	0.5	0	0.7	Varying over Programs	0.83272	0.00994	0.44612	0.22474	0.53459	0.27419
5	0.5	0	0.9	Varying over Programs	0.77655	0.02448	0.52135	0.27828	0.50712	0.34413
5	1.5	0	0.1	Varying over Programs	0.98309	0.00377	0.98308	0.00383	0.98300	0.00397
5	1.5	0	0.3	Varying over Programs	0.98813	0.00201	0.98815	0.00197	0.98773	0.00241
5	1.5	0	0.5	Varying over Programs	0.99054	0.00235	0.85483	0.25762	0.87583	0.23458
5	1.5	0	0.7	Varying over Programs	0.99046	0.00283	0.99044	0.00277	0.99021	0.00294
5	1.5	0	0.9	Varying over Programs	0.98598	0.00382	0.98594	0.00382	0.98557	0.00410
7	0.5	0	0.1	Varying over Programs	0.79315	0.02497	0.77719	0.05929	0.78561	0.03325
7	0.5	0	0.3	Varying over Programs	0.85479	0.00621	0.85332	0.00719	0.84839	0.00990
7	0.5	0	0.5	Varying over Programs	0.86499	0.00483	0.86454	0.00469	0.84839	0.02035

p	δ	σ_z	p_0	Threshold	LCA ave	LCA sd	LCRE.CL ave	LCRE.CL sd	LCRE.NCL ave	LCRE.NCL sd
7	0.5	0	0.7	Varying over Programs	0.85723	0.00740	0.85679	0.00724	0.84974	0.01511
7	0.5	0	0.9	Varying over Programs	0.80142	0.01445	0.71795	0.19056	0.69055	0.24374
7	1.5	0	0.1	Varying over Programs	0.99658	0.00138	0.99659	0.00137	0.99644	0.00155
7	1.5	0	0.3	Varying over Programs	0.99677	0.00090	0.99674	0.00092	0.99674	0.00090
7	1.5	0	0.5	Varying over Programs	0.99745	0.00108	0.99746	0.00108	0.99741	0.00110
7	1.5	0	0.7	Varying over Programs	0.99743	0.00116	0.99743	0.00116	0.99740	0.00122
7	1.5	0	0.9	Varying over Programs	0.99677	0.00297	0.99676	0.00295	0.99663	0.00296
5	0.5	0	0.1	Constant Over Programs	0.81597	0.01097	0.79332	0.04808	0.78928	0.05049
5	0.5	0	0.3	Constant Over Programs	0.87166	0.00655	0.86654	0.01833	0.84639	0.03872
5	0.5	0	0.5	Constant Over Programs	0.88080	0.00480	0.88014	0.00533	0.83097	0.10500
5	0.5	0	0.7	Constant Over Programs	0.87363	0.00486	0.86851	0.01490	0.84830	0.03960
5	0.5	0	0.9	Constant Over Programs	0.81770	0.01684	0.80222	0.03962	0.78942	0.05073
5	1.5	0	0.1	Constant Over Programs	0.99084	0.00231	0.99078	0.00227	0.99062	0.00246

p	δ	σ_z	p_0	Threshold	LCA ave	LCA sd	LCRE.CL ave	LCRE.CL sd	LCRE.NCL ave	LCRE.NCL sd
5	1.5	0	0.3	Constant Over Programs	0.99431	0.00139	0.99427	0.00133	0.99402	0.00132
5	1.5	0	0.5	Constant Over Programs	0.99545	0.00158	0.97391	0.07194	0.96129	0.15253
5	1.5	0	0.7	Constant Over Programs	0.99479	0.00121	0.99477	0.00118	0.99464	0.00136
5	1.5	0	0.9	Constant Over Programs	0.99012	0.00233	0.99001	0.00241	0.98985	0.00239
7	0.5	0	0.1	Constant Over Programs	0.84371	0.01237	0.83835	0.01618	0.83605	0.02323
7	0.5	0	0.3	Constant Over Programs	0.89181	0.00555	0.89148	0.00548	0.88553	0.01029
7	0.5	0	0.5	Constant Over Programs	0.89942	0.00360	0.89923	0.00373	0.89314	0.01443
7	0.5	0	0.7	Constant Over Programs	0.89017	0.00533	0.89004	0.00554	0.88513	0.00789
7	0.5	0	0.9	Constant Over Programs	0.84145	0.01037	0.83719	0.01148	0.83492	0.01375
7	1.5	0	0.1	Constant Over Programs	0.99759	0.00131	0.99757	0.00132	0.99740	0.00168
7	1.5	0	0.3	Constant Over Programs	0.99885	0.00067	0.99884	0.00068	0.99882	0.00069
7	1.5	0	0.5	Constant Over Programs	0.99922	0.00054	0.99921	0.00053	0.99919	0.00056
7	1.5	0	0.7	Constant Over Programs	0.99862	0.00082	0.99861	0.00080	0.99861	0.00080

p	δ	σ_z	p_0	Threshold	LCA ave	LCA sd	LCRE.CL ave	LCRE.CL sd	LCRE.NCL ave	LCRE.NCL sd
7	1.5	0	0.9	Constant Over Programs	0.99773	0.00123	0.99772	0.00122	0.99767	0.00136
5	0.5	0.5	0.1	Varying over Programs	0.66638	0.08954	0.63753	0.10695	0.47712	0.23338
5	0.5	0.5	0.3	Varying over Programs	0.74167	0.06921	0.64549	0.13542	0.55018	0.22793
5	0.5	0.5	0.5	Varying over Programs	0.80434	0.02927	0.70134	0.10721	0.61248	0.21973
5	0.5	0.5	0.7	Varying over Programs	0.75458	0.04064	0.56444	0.21651	0.51373	0.24744
5	0.5	0.5	0.9	Varying over Programs	0.70981	0.07548	0.62938	0.16752	0.49555	0.23232
5	1.5	0.5	0.1	Varying over Programs	0.94580	0.01058	0.95195	0.00735	0.95929	0.00697
5	1.5	0.5	0.3	Varying over Programs	0.96900	0.00496	0.96970	0.00449	0.97463	0.00350
5	1.5	0.5	0.5	Varying over Programs	0.97339	0.00328	0.97388	0.00241	0.97721	0.00287
5	1.5	0.5	0.7	Varying over Programs	0.97080	0.00454	0.88746	0.18575	0.82657	0.30815
5	1.5	0.5	0.9	Varying over Programs	0.94861	0.00971	0.95893	0.00700	0.96501	0.00774
7	0.5	0.5	0.1	Varying over Programs	0.67500	0.08452	0.63897	0.08915	0.51505	0.19438
7	0.5	0.5	0.3	Varying over Programs	0.77407	0.03251	0.65805	0.11564	0.70709	0.10047

p	δ	σ_z	p_0	Threshold	LCA ave	LCA sd	LCRE.CL ave	LCRE.CL sd	LCRE.NCL ave	LCRE.NCL sd
7	0.5	0.5	0.5	Varying over Programs	0.80557	0.03252	0.59416	0.12247	0.52564	0.25860
7	0.5	0.5	0.7	Varying over Programs	0.77012	0.04037	0.55074	0.13026	0.48156	0.18475
7	0.5	0.5	0.9	Varying over Programs	0.66794	0.08429	0.60337	0.15180	0.39104	0.22484
7	1.5	0.5	0.1	Varying over Programs	0.97525	0.00586	0.97793	0.00486	0.98184	0.00430
7	1.5	0.5	0.3	Varying over Programs	0.98625	0.00253	0.98644	0.00214	0.98893	0.00161
7	1.5	0.5	0.5	Varying over Programs	0.98762	0.00241	0.98795	0.00224	0.98987	0.00208
7	1.5	0.5	0.7	Varying over Programs	0.98570	0.00254	0.98693	0.00215	0.98866	0.00174
7	1.5	0.5	0.9	Varying over Programs	0.97840	0.00715	0.98270	0.00525	0.98474	0.00541
5	0.5	0.5	0.1	Constant Over Programs	0.68684	0.09770	0.67931	0.11481	0.56555	0.26254
5	0.5	0.5	0.3	Constant Over Programs	0.76910	0.08044	0.62341	0.12450	0.58320	0.20063
5	0.5	0.5	0.5	Constant Over Programs	0.80418	0.05030	0.59264	0.14309	0.49922	0.21814
5	0.5	0.5	0.7	Constant Over Programs	0.76994	0.05426	0.59642	0.11185	0.53912	0.22082
5	0.5	0.5	0.9	Constant Over Programs	0.70527	0.12296	0.65642	0.16652	0.53215	0.25431

p	δ	σ_z	p_0	Threshold	LCA ave	LCA sd	LCRE.CL ave	LCRE.CL sd	LCRE.NCL ave	LCRE.NCL sd
5	1.5	0.5	0.1	Constant Over Programs	0.95959	0.01135	0.96989	0.00654	0.97702	0.00615
5	1.5	0.5	0.3	Constant Over Programs	0.97817	0.00395	0.98115	0.00341	0.98331	0.00378
5	1.5	0.5	0.5	Constant Over Programs	0.97984	0.00272	0.95803	0.05126	0.98446	0.00286
5	1.5	0.5	0.7	Constant Over Programs	0.97625	0.00467	0.98032	0.00348	0.98332	0.00301
5	1.5	0.5	0.9	Constant Over Programs	0.96075	0.01233	0.97054	0.00737	0.97664	0.00476
7	0.5	0.5	0.1	Constant Over Programs	0.68366	0.07143	0.64725	0.10927	0.56960	0.20535
7	0.5	0.5	0.3	Constant Over Programs	0.78351	0.06331	0.57505	0.13382	0.51870	0.25260
7	0.5	0.5	0.5	Constant Over Programs	0.82448	0.02138	0.55382	0.11512	0.42747	0.22912
7	0.5	0.5	0.7	Constant Over Programs	0.79194	0.02786	0.59101	0.11471	0.57821	0.18089
7	0.5	0.5	0.9	Constant Over Programs	0.70032	0.08237	0.66694	0.09036	0.56072	0.23059
7	1.5	0.5	0.1	Constant Over Programs	0.98374	0.00352	0.98721	0.00274	0.98944	0.00208
7	1.5	0.5	0.3	Constant Over Programs	0.98697	0.00291	0.98904	0.00257	0.99137	0.00256
7	1.5	0.5	0.5	Constant Over Programs	0.98953	0.00234	0.97790	0.04188	0.99338	0.00131

p	δ	σ_z	p_0	Threshold	LCA ave	LCA sd	LCRE.CL ave	LCRE.CL sd	LCRE.NCL ave	LCRE.NCL sd
7	1.5	0.5	0.7	Constant Over Programs	0.98827	0.00190	0.99021	0.00138	0.99177	0.00165
7	1.5	0.5	0.9	Constant Over Programs	0.98055	0.00488	0.98468	0.00292	0.98808	0.00237
5	0.5	1	0.1	Varying over Programs	0.71917	0.17721	0.71546	0.17383	0.60997	0.23855
5	0.5	1	0.3	Varying over Programs	0.72020	0.08046	0.70616	0.11215	0.62949	0.21942
5	0.5	1	0.5	Varying over Programs	0.75903	0.05365	0.73484	0.07483	0.56985	0.26470
5	0.5	1	0.7	Varying over Programs	0.75134	0.07324	0.71957	0.09185	0.71992	0.08776
5	0.5	1	0.9	Varying over Programs	0.78835	0.06727	0.79232	0.08078	0.70167	0.23159
5	1.5	1	0.1	Varying over Programs	0.64146	0.01977	0.59527	0.06555	0.58889	0.07931
5	1.5	1	0.3	Varying over Programs	0.83778	0.01604	0.35389	0.19973	0.48112	0.17589
5	1.5	1	0.5	Varying over Programs	0.86786	0.01457	0.50910	0.13254	0.66376	0.10472
5	1.5	1	0.7	Varying over Programs	0.85200	0.01652	0.68137	0.04851	0.63819	0.16111
5	1.5	1	0.9	Varying over Programs	0.65559	0.03023	0.67985	0.07119	0.60177	0.31587
7	0.5	1	0.1	Varying over Programs	0.76143	0.08160	0.73444	0.10154	0.48116	0.24670

p	δ	σ_z	p_0	Threshold	LCA ave	LCA sd	LCRE.CL ave	LCRE.CL sd	LCRE.NCL ave	LCRE.NCL sd
7	0.5	1	0.3	Varying over Programs	0.66108	0.11687	0.59070	0.16117	0.41192	0.22755
7	0.5	1	0.5	Varying over Programs	0.73000	0.09843	0.65045	0.15195	0.36104	0.25496
7	0.5	1	0.7	Varying over Programs	0.70312	0.11013	0.62714	0.16907	0.34705	0.24576
7	0.5	1	0.9	Varying over Programs	0.73873	0.07100	0.72755	0.09484	0.55606	0.21606
7	1.5	1	0.1	Varying over Programs	0.69983	0.02393	0.63039	0.10345	0.59994	0.17301
7	1.5	1	0.3	Varying over Programs	0.90497	0.01106	0.90124	0.01062	0.96860	0.00275
7	1.5	1	0.5	Varying over Programs	0.93114	0.00577	0.60474	0.05426	0.40636	0.09852
7	1.5	1	0.7	Varying over Programs	0.92430	0.00883	0.72945	0.05708	0.73153	0.34648
7	1.5	1	0.9	Varying over Programs	0.72794	0.05103	0.61885	0.18026	0.67811	0.15334
5	0.5	1	0.1	Constant Over Programs	0.75946	0.08555	0.75693	0.09920	0.74009	0.09908
5	0.5	1	0.3	Constant Over Programs	0.73582	0.08273	0.70263	0.11253	0.66463	0.18370
5	0.5	1	0.5	Constant Over Programs	0.73815	0.08841	0.68713	0.13090	0.64438	0.20786
5	0.5	1	0.7	Constant Over Programs	0.76071	0.09875	0.73320	0.12617	0.68458	0.17791

p	δ	σ_z	p_0	Threshold	LCA ave	LCA sd	LCRE.CL ave	LCRE.CL sd	LCRE.NCL ave	LCRE.NCL sd
5	0.5	1	0.9	Constant Over Programs	0.77909	0.06631	0.77883	0.07530	0.73018	0.13559
5	1.5	1	0.1	Constant Over Programs	0.66672	0.02095	0.47979	0.13940	0.61177	0.08552
5	1.5	1	0.3	Constant Over Programs	0.86011	0.01404	0.74739	0.03788	0.80515	0.02545
5	1.5	1	0.5	Constant Over Programs	0.88647	0.01269	0.79849	0.02519	0.72704	0.07642
5	1.5	1	0.7	Constant Over Programs	0.85511	0.01460	0.76268	0.02897	0.81728	0.01532
5	1.5	1	0.9	Constant Over Programs	0.66114	0.03066	0.47589	0.13931	0.60829	0.10520
7	0.5	1	0.1	Constant Over Programs	0.73589	0.08077	0.71077	0.10980	0.46568	0.26670
7	0.5	1	0.3	Constant Over Programs	0.71402	0.11528	0.65203	0.17146	0.43967	0.25416
7	0.5	1	0.5	Constant Over Programs	0.76832	0.08637	0.68972	0.12470	0.37578	0.23392
7	0.5	1	0.7	Constant Over Programs	0.73489	0.08306	0.67745	0.12039	0.45787	0.26678
7	0.5	1	0.9	Constant Over Programs	0.74770	0.11023	0.73571	0.13063	0.50550	0.28509
7	1.5	1	0.1	Constant Over Programs	0.70626	0.02358	0.33263	0.15507	0.50776	0.07024
7	1.5	1	0.3	Constant Over Programs	0.91644	0.00870	0.64962	0.03900	0.30502	0.19141

p	δ	σ_z	p_0	Threshold	LCA ave	LCA sd	LCRE.CL ave	LCRE.CL sd	LCRE.NCL ave	LCRE.NCL sd
7	1.5	1	0.5	Constant Over Programs	0.93394	0.00757	0.71595	0.04557	0.16308	0.12252
7	1.5	1	0.7	Constant Over Programs	0.91216	0.01158	0.63142	0.06297	0.34016	0.21054
7	1.5	1	0.9	Constant Over Programs	0.70854	0.02327	0.39528	0.23944	0.53645	0.11015
5	0.5	2	0.1	Varying over Programs	0.77508	0.06199	0.77521	0.06937	0.71736	0.18556
5	0.5	2	0.3	Varying over Programs	0.76479	0.09226	0.76957	0.10445	0.71990	0.20571
5	0.5	2	0.5	Varying over Programs	0.78464	0.04626	0.78417	0.05516	0.77767	0.04944
5	0.5	2	0.7	Varying over Programs	0.76609	0.09116	0.76852	0.10114	0.74287	0.11551
5	0.5	2	0.9	Varying over Programs	0.77215	0.07322	0.77754	0.07618	0.69632	0.20680
5	1.5	2	0.1	Varying over Programs	0.32903	0.01804	0.23719	0.03054	0.30229	0.05514
5	1.5	2	0.3	Varying over Programs	0.48492	0.02404	0.16596	0.04484	0.68375	0.16941
5	1.5	2	0.5	Varying over Programs	0.51983	0.01622	0.04642	0.03818	0.81776	0.12266
5	1.5	2	0.7	Varying over Programs	0.48032	0.01437	0.26528	0.09241	0.87377	0.04933
5	1.5	2	0.9	Varying over Programs	0.33257	0.01693	0.25490	0.07837	0.81382	0.03081

p	δ	σ_z	p_0	Threshold	LCA ave	LCA sd	LCRE.CL ave	LCRE.CL sd	LCRE.NCL ave	LCRE.NCL sd
7	0.5	2	0.1	Varying over Programs	0.75752	0.08359	0.75231	0.09814	0.58161	0.23463
7	0.5	2	0.3	Varying over Programs	0.76459	0.07088	0.74377	0.08952	0.44986	0.23436
7	0.5	2	0.5	Varying over Programs	0.75509	0.06349	0.72581	0.07885	0.40005	0.17762
7	0.5	2	0.7	Varying over Programs	0.74980	0.09443	0.73694	0.12014	0.34432	0.18049
7	0.5	2	0.9	Varying over Programs	0.77284	0.06190	0.76595	0.06986	0.52972	0.24471
7	1.5	2	0.1	Varying over Programs	0.37651	0.01529	0.16374	0.03214	0.82207	0.25381
7	1.5	2	0.3	Varying over Programs	0.57619	0.01677	0.03417	0.02573	0.94088	0.00590
7	1.5	2	0.5	Varying over Programs	0.62285	0.03262	0.23802	0.07682	0.94572	0.00490
7	1.5	2	0.7	Varying over Programs	0.54188	0.01495	0.28449	0.07833	0.94175	0.00608
7	1.5	2	0.9	Varying over Programs	0.34366	0.02120	0.02309	0.02230	0.88645	0.15983
5	0.5	2	0.1	Constant Over Programs	0.77527	0.09237	0.77859	0.09914	0.75801	0.11105
5	0.5	2	0.3	Constant Over Programs	0.78614	0.08173	0.78610	0.09419	0.78121	0.09363
5	0.5	2	0.5	Constant Over Programs	0.78322	0.05637	0.78347	0.06513	0.77718	0.06372

p	δ	σ_z	p_0	Threshold	LCA ave	LCA sd	LCRE.CL ave	LCRE.CL sd	LCRE.NCL ave	LCRE.NCL sd
5	0.5	2	0.7	Constant Over Programs	0.76352	0.09965	0.75966	0.11074	0.75482	0.10935
5	0.5	2	0.9	Constant Over Programs	0.77325	0.09937	0.77344	0.10753	0.71235	0.17081
5	1.5	2	0.1	Constant Over Programs	0.33402	0.01961	0.06993	0.03493	0.66422	0.17091
5	1.5	2	0.3	Constant Over Programs	0.49250	0.02170	0.21950	0.09484	0.87338	0.02996
5	1.5	2	0.5	Constant Over Programs	0.53107	0.01471	0.35962	0.09576	0.90345	0.01841
5	1.5	2	0.7	Constant Over Programs	0.48963	0.01504	0.25066	0.06515	0.87401	0.03808
5	1.5	2	0.9	Constant Over Programs	0.33822	0.01558	0.07001	0.03402	0.69464	0.12996
7	0.5	2	0.1	Constant Over Programs	0.77982	0.06473	0.77067	0.07626	0.55164	0.24623
7	0.5	2	0.3	Constant Over Programs	0.77743	0.08638	0.77575	0.09941	0.51724	0.23361
7	0.5	2	0.5	Constant Over Programs	0.76741	0.07726	0.75170	0.09587	0.48202	0.21226
7	0.5	2	0.7	Constant Over Programs	0.74261	0.09204	0.73662	0.11003	0.41009	0.24084
7	0.5	2	0.9	Constant Over Programs	0.76507	0.06728	0.76262	0.07376	0.52784	0.26327
7	1.5	2	0.1	Constant Over Programs	0.36462	0.01545	0.08437	0.02591	0.93200	0.01195

p	δ	σ_z	p_0	Threshold	LCA ave	LCA sd	LCRE.CL ave	LCRE.CL sd	LCRE.NCL ave	LCRE.NCL sd
7	1.5	2	0.3	Constant Over Programs	0.55481	0.01584	0.15736	0.06824	0.91344	0.17531
7	1.5	2	0.5	Constant Over Programs	0.61203	0.02230	0.33249	0.10139	0.95431	0.00480
7	1.5	2	0.7	Constant Over Programs	0.55668	0.01358	0.15095	0.06554	0.91843	0.13701
7	1.5	2	0.9	Constant Over Programs	0.36592	0.01996	0.09468	0.02736	0.92803	0.00907
5	0.5	4	0.1	Varying over Programs	0.80508	0.05419	0.80674	0.05889	0.70650	0.23979
5	0.5	4	0.3	Varying over Programs	0.77216	0.09187	0.77699	0.09595	0.74291	0.15468
5	0.5	4	0.5	Varying over Programs	0.78588	0.04991	0.79256	0.05232	0.73106	0.19793
5	0.5	4	0.7	Varying over Programs	0.78422	0.05878	0.78854	0.06444	0.78551	0.06389
5	0.5	4	0.9	Varying over Programs	0.78748	0.05889	0.79267	0.06230	0.76573	0.12283
5	1.5	4	0.1	Varying over Programs	0.16396	0.01895	0.11954	0.02145	0.14468	0.01863
5	1.5	4	0.3	Varying over Programs	0.23584	0.02457	0.09469	0.02366	0.71684	0.24430
5	1.5	4	0.5	Varying over Programs	0.33716	0.20968	0.16509	0.28940	0.76962	0.26150
5	1.5	4	0.7	Varying over Programs	0.23727	0.01962	0.03025	0.01814	0.86381	0.06485

p	δ	σ_z	p_0	Threshold	LCA ave	LCA sd	LCRE.CL ave	LCRE.CL sd	LCRE.NCL ave	LCRE.NCL sd
5	1.5	4	0.9	Varying over Programs	0.16624	0.01980	0.04633	0.02795	0.73200	0.16172
7	0.5	4	0.1	Varying over Programs	0.81054	0.04139	0.81429	0.04027	0.63343	0.22892
7	0.5	4	0.3	Varying over Programs	0.79829	0.05459	0.80114	0.06551	0.38880	0.25690
7	0.5	4	0.5	Varying over Programs	0.78290	0.05154	0.77537	0.05902	0.43765	0.27418
7	0.5	4	0.7	Varying over Programs	0.78103	0.06537	0.78076	0.07240	0.51996	0.27661
7	0.5	4	0.9	Varying over Programs	0.78855	0.04673	0.78631	0.05597	0.57130	0.29660
7	1.5	4	0.1	Varying over Programs	0.18213	0.02105	0.07928	0.02514	0.83961	0.05538
7	1.5	4	0.3	Varying over Programs	0.27110	0.01494	0.05097	0.02119	0.90182	0.00858
7	1.5	4	0.5	Varying over Programs	0.31933	0.11783	0.06163	0.16533	0.87569	0.13473
7	1.5	4	0.7	Varying over Programs	0.27129	0.01546	0.02840	0.01638	0.89128	0.01706
7	1.5	4	0.9	Varying over Programs	0.17615	0.02670	0.05785	0.02050	0.82929	0.02238
5	0.5	4	0.1	Constant Over Programs	0.78267	0.08505	0.78487	0.08927	0.74505	0.17998
5	0.5	4	0.3	Constant Over Programs	0.78470	0.06742	0.78805	0.07053	0.78517	0.07240

p	δ	σ_z	p_0	Threshold	LCA ave	LCA sd	LCRE.CL ave	LCRE.CL sd	LCRE.NCL ave	LCRE.NCL sd
5	0.5	4	0.5	Constant Over Programs	0.78351	0.06185	0.78467	0.06513	0.78246	0.06302
5	0.5	4	0.7	Constant Over Programs	0.78823	0.08542	0.78848	0.09114	0.75078	0.17331
5	0.5	4	0.9	Constant Over Programs	0.78337	0.10235	0.78751	0.10722	0.77255	0.11364
5	1.5	4	0.1	Constant Over Programs	0.16677	0.01950	0.08300	0.02688	0.62997	0.26682
5	1.5	4	0.3	Constant Over Programs	0.23688	0.02414	0.05871	0.02600	0.83396	0.10290
5	1.5	4	0.5	Constant Over Programs	0.36912	0.23906	0.19325	0.32069	0.74276	0.28334
5	1.5	4	0.7	Constant Over Programs	0.23753	0.02105	0.05544	0.02500	0.80348	0.16877
5	1.5	4	0.9	Constant Over Programs	0.17012	0.01829	0.08266	0.02172	0.61666	0.30173
7	0.5	4	0.1	Constant Over Programs	0.76230	0.06660	0.75826	0.07445	0.67706	0.18204
7	0.5	4	0.3	Constant Over Programs	0.77562	0.06971	0.77479	0.07689	0.57149	0.25678
7	0.5	4	0.5	Constant Over Programs	0.76228	0.07206	0.75811	0.08381	0.47563	0.25795
7	0.5	4	0.7	Constant Over Programs	0.76698	0.05062	0.76410	0.05222	0.52523	0.30822
7	0.5	4	0.9	Constant Over Programs	0.77200	0.07112	0.77185	0.08127	0.65372	0.24527

p	δ	σ_z	p_0	Threshold	LCA ave	LCA sd	LCRE.CL ave	LCRE.CL sd	LCRE.NCL ave	LCRE.NCL sd
7	1.5	4	0.1	Constant Over Programs	0.18159	0.02140	0.07376	0.02337	0.84921	0.01491
7	1.5	4	0.3	Constant Over Programs	0.27468	0.01493	0.04850	0.02361	0.90373	0.00918
7	1.5	4	0.5	Constant Over Programs	0.37961	0.19946	0.14924	0.29054	0.81822	0.22436
7	1.5	4	0.7	Constant Over Programs	0.27451	0.01491	0.05186	0.01681	0.90508	0.00826
7	1.5	4	0.9	Constant Over Programs	0.18409	0.02756	0.07421	0.02245	0.78977	0.18354

B.1.2 RMSE for three models

p	δ	σ_z	p_0	Threshold	LCA	LCRE.CL	LCRE.NCL	p	δ	σ_z	p_0	Threshold	LCA	LCRE.CL	LCRE.NCL
5	0.5	0	0.1	Varying over Programs	0.15866	0.80402	0.55664	7	0.5	0.5	0.1	Varying over Programs	0.44093	0.46290	0.44806
5	0.5	0	0.3	Varying over Programs	0.20296	0.20296	0.20296	7	0.5	0.5	0.3	Varying over Programs	0.26527	0.27857	0.27865
5	0.5	0	0.5	Varying over Programs	0.21613	0.21614	0.21614	7	0.5	0.5	0.5	Varying over Programs	0.22646	0.27723	0.27875
5	0.5	0	0.7	Varying over Programs	0.19961	0.41280	0.22250	7	0.5	0.5	0.7	Varying over Programs	0.27831	0.19813	0.24780
5	0.5	0	0.9	Varying over Programs	0.16019	0.16271	0.16348	7	0.5	0.5	0.9	Varying over Programs	0.30197	0.23610	0.32432
5	1.5	0	0.1	Varying over Programs	0.06412	0.06439	0.06544	7	1.5	0.5	0.1	Varying over Programs	0.06258	0.06109	0.05552
5	1.5	0	0.3	Varying over Programs	0.07959	0.07915	0.08270	7	1.5	0.5	0.3	Varying over Programs	0.07534	0.07565	0.06756
5	1.5	0	0.5	Varying over Programs	0.06991	0.06994	0.07424	7	1.5	0.5	0.5	Varying over Programs	0.09047	0.08889	0.08175
5	1.5	0	0.7	Varying over Programs	0.07071	0.07072	0.07113	7	1.5	0.5	0.7	Varying over Programs	0.08248	0.07948	0.07410
5	1.5	0	0.9	Varying over Programs	0.05657	0.05652	0.06218	7	1.5	0.5	0.9	Varying over Programs	0.06338	0.06017	0.04604
7	0.5	0	0.1	Varying over Programs	0.14260	0.14836	0.15959	5	0.5	0.5	0.1	Constant Over Programs	0.29774	0.25706	0.42689
7	0.5	0	0.3	Varying over Programs	0.20186	0.20259	0.20726	5	0.5	0.5	0.3	Constant Over Programs	0.22514	0.26886	0.29773
7	0.5	0	0.5	Varying over Programs	0.21550	0.21550	0.21550	5	0.5	0.5	0.5	Constant Over Programs	0.20991	0.33672	0.39982
7	0.5	0	0.7	Varying over Programs	0.19921	0.19910	0.19904	5	0.5	0.5	0.7	Constant Over Programs	0.23269	0.43316	0.60708
7	0.5	0	0.9	Varying over Programs	0.15303	0.16258	0.15572	5	0.5	0.5	0.9	Constant Over Programs	0.21399	0.21892	0.20535
7	1.5	0	0.1	Varying over Programs	0.02213	0.02213	0.02213	5	1.5	0.5	0.1	Constant Over Programs	0.06890	0.06331	0.05223
7	1.5	0	0.3	Varying over Programs	0.03414	0.03414	0.03414	5	1.5	0.5	0.3	Constant Over Programs	0.10162	0.09362	0.08194
7	1.5	0	0.5	Varying over Programs	0.03624	0.03629	0.03616	5	1.5	0.5	0.5	Constant Over Programs	0.10746	0.27109	0.09176
7	1.5	0	0.7	Varying over Programs	0.04209	0.04191	0.04222	5	1.5	0.5	0.7	Constant Over Programs	0.10391	0.09191	0.08959
7	1.5	0	0.9	Varying over Programs	0.02940	0.02940	0.02940	5	1.5	0.5	0.9	Constant Over Programs	0.10794	0.08492	0.07392
5	0.5	0	0.1	Constant Over Programs	0.13717	0.13494	0.17261	7	0.5	0.5	0.1	Constant Over Programs	0.31627	0.33295	0.34099
5	0.5	0	0.3	Constant Over Programs	0.17969	0.18076	0.25149	7	0.5	0.5	0.3	Constant Over Programs	0.25271	0.26822	0.27630
5	0.5	0	0.5	Constant Over Programs	0.18735	0.18734	0.18734	7	0.5	0.5	0.5	Constant Over Programs	0.21968	0.26851	0.36742
5	0.5	0	0.7	Constant Over Programs	0.17854	0.18240	0.19186	7	0.5	0.5	0.7	Constant Over Programs	0.28311	0.28992	0.28917
5	0.5	0	0.9	Constant Over Programs	0.14655	0.17774	0.15428	7	0.5	0.5	0.9	Constant Over Programs	0.24957	0.20619	0.44420
5	1.5	0	0.1	Constant Over Programs	0.04004	0.04059	0.04172	7	1.5	0.5	0.1	Constant Over Programs	0.05183	0.05027	0.04692
5	1.5	0	0.3	Constant Over Programs	0.05521	0.05521	0.05521	7	1.5	0.5	0.3	Constant Over Programs	0.07094	0.06279	0.05423
5	1.5	0	0.5	Constant Over Programs	0.04857	0.04857	0.04857	7	1.5	0.5	0.5	Constant Over Programs	0.05938	0.05654	0.04900
5	1.5	0	0.7	Constant Over Programs	0.05750	0.05677	0.05955	7	1.5	0.5	0.7	Constant Over Programs	0.07507	0.07113	0.06628
5	1.5	0	0.9	Constant Over Programs	0.04437	0.04499	0.04636	7	1.5	0.5	0.9	Constant Over Programs	0.06082	0.05570	0.04498
7	0.5	0	0.1	Constant Over Programs	0.14157	0.14149	0.14149	5	0.5	1	0.1	Varying over Programs	0.30712	0.28531	0.15225
7	0.5	0	0.3	Constant Over Programs	0.17786	0.17786	0.17786	5	0.5	1	0.3	Varying over Programs	0.27879	0.25401	0.51879
7	0.5	0	0.5	Constant Over Programs	0.18986	0.19068	0.20278	5	0.5	1	0.5	Varying over Programs	0.33344	0.33957	0.34413
7	0.5	0	0.7	Constant Over Programs	0.18557	0.18613	0.18790	5	0.5	1	0.7	Varying over Programs	0.37496	0.34598	0.37256
7	0.5	0	0.9	Constant Over Programs	0.12997	0.13671	0.15004	5	0.5	1	0.9	Varying over Programs	0.28927	0.27617	0.28323
7	1.5	0	0.1	Constant Over Programs	0.01562	0.01562	0.01562	5	1.5	1	0.1	Varying over Programs	0.29916	0.29856	0.33349
7	1.5	0	0.3	Constant Over Programs	0.02363	0.02357	0.02275	5	1.5	1	0.3	Varying over Programs	0.26439	0.51402	0.72499
7	1.5	0	0.5	Constant Over Programs	0.01932	0.01954	0.02152	5	1.5	1	0.5	Varying over Programs	0.23362	0.47896	0.45023
7	1.5	0	0.7	Constant Over Programs	0.02479	0.02485	0.02488	5	1.5	1	0.7	Varying over Programs	0.24929	0.32042	0.33845
7	1.5	0	0.9	Constant Over Programs	0.02213	0.02213	0.02213	5	1.5	1	0.9	Varying over Programs	0.33176	0.22657	0.22313
5	0.5	0.5	0.1	Varying over Programs	0.27892	0.27677	0.40042	7	0.5	1	0.1	Varying over Programs	0.46386	0.46424	0.43030
5	0.5	0.5	0.3	Varying over Programs	0.24687	0.24807	0.23823	7	0.5	1	0.3	Varying over Programs	0.37414	0.39367	0.39126
5	0.5	0.5	0.5	Varying over Programs	0.25248	0.33031	0.42622	7	0.5	1	0.5	Varying over Programs	0.25246	0.26883	0.26851
5	0.5	0.5	0.7	Varying over Programs	0.19831	0.18945	0.20173	7	0.5	1	0.7	Varying over Programs	0.31742	0.35661	0.58016
5	0.5	0.5	0.9	Varying over Programs	0.47151	0.46227	0.23402	7	0.5	1	0.9	Varying over Programs	0.34581	0.30672	0.22434
5	1.5	0.5	0.1	Varying over Programs	0.08791	0.08589	0.07680	7	1.5	1	0.1	Varying over Programs	0.27583	0.23743	0.30793
5	1.5	0.5	0.3	Varying over Programs	0.11639	0.11634	0.09950	7	1.5	1	0.3	Varying over Programs	0.20647	0.19757	0.11207
5	1.5	0.5	0.5	Varying over Programs	0.10509	0.10803	0.10629	7	1.5	1	0.5	Varying over Programs	0.17825	0.42939	0.45563
5	1.5	0.5	0.7	Varying over Programs	0.11819	0.24335	0.10204	7	1.5	1	0.7	Varying over Programs	0.18404	0.33789	0.28674
5	1.5	0.5	0.9	Varying over Programs	0.08313	0.07561	0.07711	7	1.5	1	0.9	Varying over Programs	0.25707	0.45140	0.44671

p	δ	σ_z	p_0	Threshold	LCA	LCRE.CL	LCRE.NCL	p	δ	σ_z	p_0	Threshold	LCA	LCRE.CL	LCRE.NCL
5	0.5	1	0.1	Constant Over Programs	0.47196	0.45217	0.44155	7	0.5	2	0.1	Constant Over Programs	0.51050	0.50110	0.48717
5	0.5	1	0.3	Constant Over Programs	0.41298	0.41285	0.42627	7	0.5	2	0.3	Constant Over Programs	0.35761	0.35084	0.36467
5	0.5	1	0.5	Constant Over Programs	0.32131	0.28209	0.31562	7	0.5	2	0.5	Constant Over Programs	0.33050	0.31756	0.28557
5	0.5	1	0.7	Constant Over Programs	0.30074	0.30541	0.31281	7	0.5	2	0.7	Constant Over Programs	0.36063	0.35105	0.31842
5	0.5	1	0.9	Constant Over Programs	0.26632	0.24654	0.22230	7	0.5	2	0.9	Constant Over Programs	0.32165	0.31242	0.45887
5	1.5	1	0.1	Constant Over Programs	0.27938	0.45373	0.38476	7	1.5	2	0.1	Constant Over Programs	0.50856	0.60670	0.10209
5	1.5	1	0.3	Constant Over Programs	0.25324	0.35182	0.30624	7	1.5	2	0.3	Constant Over Programs	0.44988	0.55626	0.12677
5	1.5	1	0.5	Constant Over Programs	0.23960	0.30746	0.34628	7	1.5	2	0.5	Constant Over Programs	0.42137	0.50964	0.14398
5	1.5	1	0.7	Constant Over Programs	0.25045	0.30940	0.29051	7	1.5	2	0.7	Constant Over Programs	0.44491	0.57220	0.14173
5	1.5	1	0.9	Constant Over Programs	0.32787	0.53513	0.42639	7	1.5	2	0.9	Constant Over Programs	0.49166	0.54668	0.10802
7	0.5	1	0.1	Constant Over Programs	0.40342	0.38100	0.40444	5	0.5	4	0.1	Varying over Programs	0.32070	0.31350	0.30140
7	0.5	1	0.3	Constant Over Programs	0.37580	0.41387	0.43831	5	0.5	4	0.3	Varying over Programs	0.29995	0.29284	0.28964
7	0.5	1	0.5	Constant Over Programs	0.27056	0.30089	0.38750	5	0.5	4	0.5	Varying over Programs	0.32414	0.32033	0.31797
7	0.5	1	0.7	Constant Over Programs	0.29614	0.31642	0.31334	5	0.5	4	0.7	Varying over Programs	0.31384	0.30584	0.31029
7	0.5	1	0.9	Constant Over Programs	0.42050	0.40949	0.42193	5	0.5	4	0.9	Varying over Programs	0.28559	0.27604	0.26627
7	1.5	1	0.1	Constant Over Programs	0.26611	0.59631	0.51885	5	1.5	4	0.1	Varying over Programs	0.61223	0.61793	0.61520
7	1.5	1	0.3	Constant Over Programs	0.18868	0.43883	0.45640	5	1.5	4	0.3	Varying over Programs	0.59773	0.63156	0.23485
7	1.5	1	0.5	Constant Over Programs	0.18585	0.35664	0.54672	5	1.5	4	0.5	Varying over Programs	0.31945	0.27178	0.52961
7	1.5	1	0.7	Constant Over Programs	0.19370	0.45620	0.46902	5	1.5	4	0.7	Varying over Programs	0.60200	0.65899	0.19197
7	1.5	1	0.9	Constant Over Programs	0.26666	0.23129	0.32715	5	1.5	4	0.9	Varying over Programs	0.61310	0.64578	0.15634
5	0.5	2	0.1	Varying over Programs	0.32283	0.31432	0.24449	7	0.5	4	0.1	Varying over Programs	0.33091	0.32795	0.27055
5	0.5	2	0.3	Varying over Programs	0.28431	0.26862	0.25426	7	0.5	4	0.3	Varying over Programs	0.39692	0.39534	0.41743
5	0.5	2	0.5	Varying over Programs	0.38588	0.36742	0.38169	7	0.5	4	0.5	Varying over Programs	0.33209	0.34076	0.30777
5	0.5	2	0.7	Varying over Programs	0.40857	0.40332	0.40897	7	0.5	4	0.7	Varying over Programs	0.37156	0.37968	0.41958
5	0.5	2	0.9	Varying over Programs	0.41427	0.40420	0.38525	7	0.5	4	0.9	Varying over Programs	0.39894	0.40249	0.61559
5	1.5	2	0.1	Varying over Programs	0.51109	0.52477	0.51610	7	1.5	4	0.1	Varying over Programs	0.60213	0.61302	0.14391
5	1.5	2	0.3	Varying over Programs	0.48088	0.57666	0.43637	7	1.5	4	0.3	Varying over Programs	0.58471	0.64243	0.18113
5	1.5	2	0.5	Varying over Programs	0.46338	0.59757	0.34535	7	1.5	4	0.5	Varying over Programs	0.56838	0.64344	0.21854
5	1.5	2	0.7	Varying over Programs	0.48667	0.46377	0.22839	7	1.5	4	0.7	Varying over Programs	0.57283	0.65272	0.18875
5	1.5	2	0.9	Varying over Programs	0.51991	0.45923	0.21052	7	1.5	4	0.9	Varying over Programs	0.58119	0.62112	0.15883
7	0.5	2	0.1	Varying over Programs	0.54722	0.54286	0.48905	5	0.5	4	0.1	Constant Over Programs	0.40207	0.39644	0.39993
7	0.5	2	0.3	Varying over Programs	0.30966	0.31005	0.40145	5	0.5	4	0.3	Constant Over Programs	0.33745	0.33231	0.33341
7	0.5	2	0.5	Varying over Programs	0.31014	0.32173	0.30564	5	0.5	4	0.5	Constant Over Programs	0.37239	0.36441	0.37354
7	0.5	2	0.7	Varying over Programs	0.36609	0.36762	0.38695	5	0.5	4	0.7	Constant Over Programs	0.30219	0.29656	0.29601
7	0.5	2	0.9	Varying over Programs	0.32242	0.32739	0.52900	5	0.5	4	0.9	Constant Over Programs	0.31822	0.30915	0.31001
7	1.5	2	0.1	Varying over Programs	0.49832	0.54590	0.11106	5	1.5	4	0.1	Constant Over Programs	0.62161	0.65373	0.16422
7	1.5	2	0.3	Varying over Programs	0.44742	0.61963	0.14167	5	1.5	4	0.3	Constant Over Programs	0.60165	0.64478	0.35002
7	1.5	2	0.5	Varying over Programs	0.40104	0.54150	0.15882	5	1.5	4	0.5	Constant Over Programs	0.58932	0.63628	0.20952
7	1.5	2	0.7	Varying over Programs	0.44297	0.49830	0.14851	5	1.5	4	0.7	Constant Over Programs	0.60176	0.64819	0.19433
7	1.5	2	0.9	Varying over Programs	0.50848	0.59382	0.53888	5	1.5	4	0.9	Constant Over Programs	0.60342	0.61262	0.60689
5	0.5	2	0.1	Constant Over Programs	0.37168	0.36837	0.37802	7	0.5	4	0.1	Constant Over Programs	0.35920	0.35541	0.33913
5	0.5	2	0.3	Constant Over Programs	0.37903	0.36104	0.36021	7	0.5	4	0.3	Constant Over Programs	0.41384	0.40400	0.61053
5	0.5	2	0.5	Constant Over Programs	0.29918	0.28974	0.29672	7	0.5	4	0.5	Constant Over Programs	0.46700	0.45940	0.57479
5	0.5	2	0.7	Constant Over Programs	0.26321	0.25014	0.25612	7	0.5	4	0.7	Constant Over Programs	0.33562	0.33779	0.44668
5	0.5	2	0.9	Constant Over Programs	0.31322	0.30656	0.30414	7	0.5	4	0.9	Constant Over Programs	0.35755	0.35330	0.33675
5	1.5	2	0.1	Constant Over Programs	0.52657	0.60325	0.10319	7	1.5	4	0.1	Constant Over Programs	0.60230	0.62916	0.15085
5	1.5	2	0.3	Constant Over Programs	0.48301	0.53359	0.21553	7	1.5	4	0.3	Constant Over Programs	0.58441	0.64214	0.17819
5	1.5	2	0.5	Constant Over Programs	0.45675	0.47546	0.17672	7	1.5	4	0.5	Constant Over Programs	0.57093	0.64010	0.20422
5	1.5	2	0.7	Constant Over Programs	0.48907	0.49975	0.22359	7	1.5	4	0.7	Constant Over Programs	0.57404	0.63987	0.17554
5	1.5	2	0.9	Constant Over Programs	0.51493	0.55975	0.26073	7	1.5	4	0.9	Constant Over Programs	0.58273	0.60943	0.15597

B.1.3 BIC for each of the three models

p	δ	σ_z	p_0	Threshold	Preferred Model	LCA	LCRE.CL	LCRE.NCL
5	0.5	0	0.1	Varying over Programs	LCA	18028.997	18039.847	18065.845
5	0.5	0	0.3	Varying over Programs	LCA	18312.155	18326.782	18351.493
5	0.5	0	0.5	Varying over Programs	LCA	18103.646	18111.817	18141.146
5	0.5	0	0.7	Varying over Programs	LCA	17429.707	17444.435	17469.465
5	0.5	0	0.9	Varying over Programs	LCA	16269.319	16281.586	16306.435
5	1.5	0	0.1	Varying over Programs	LCA	11799.375	11806.764	11834.252
5	1.5	0	0.3	Varying over Programs	LCA	12756.800	12763.921	12792.266
5	1.5	0	0.5	Varying over Programs	LCA	12539.798	12631.404	12623.773
5	1.5	0	0.7	Varying over Programs	LCA	11361.300	11368.093	11395.812
5	1.5	0	0.9	Varying over Programs	LCA	8996.169	9003.612	9032.024
7	0.5	0	0.1	Varying over Programs	LCA	24858.771	24868.895	24909.444
7	0.5	0	0.3	Varying over Programs	LCA	25314.861	25322.106	25364.161
7	0.5	0	0.5	Varying over Programs	LCA	25135.791	25142.982	25184.779
7	0.5	0	0.7	Varying over Programs	LCA	24450.073	24457.736	24500.232
7	0.5	0	0.9	Varying over Programs	LCA	23142.362	23156.603	23197.070
7	1.5	0	0.1	Varying over Programs	LCA	15084.657	15092.032	15134.565
7	1.5	0	0.3	Varying over Programs	LCA	16152.976	16160.598	16203.675
7	1.5	0	0.5	Varying over Programs	LCA	15993.287	16000.742	16044.408
7	1.5	0	0.7	Varying over Programs	LCA	14804.138	14811.817	14854.302
7	1.5	0	0.9	Varying over Programs	LCA	12352.946	12360.354	12402.578
5	0.5	0	0.1	Constant Over Programs	LCA	19308.561	19315.830	19346.074
5	0.5	0	0.3	Constant Over Programs	LCA	20109.356	20116.820	20144.549
5	0.5	0	0.5	Constant Over Programs	LCA	20354.906	20362.577	20391.872
5	0.5	0	0.7	Constant Over Programs	LCA	20105.623	20112.809	20140.625
5	0.5	0	0.9	Constant Over Programs	LCA	19347.006	19354.752	19383.452
5	1.5	0	0.1	Constant Over Programs	LCA	9381.105	9388.800	9418.828
5	1.5	0	0.3	Constant Over Programs	LCA	11097.949	11104.768	11132.674
5	1.5	0	0.5	Constant Over Programs	LCA	11613.527	11653.878	11669.090
5	1.5	0	0.7	Constant Over Programs	LCA	11107.442	11114.287	11142.660
5	1.5	0	0.9	Constant Over Programs	LCA	9407.272	9414.900	9444.119
7	0.5	0	0.1	Constant Over Programs	LCA	26973.203	26980.514	27023.346
7	0.5	0	0.3	Constant Over Programs	LCA	27956.482	27964.080	28005.230
7	0.5	0	0.5	Constant Over Programs	LCA	28227.258	28234.856	28279.094
7	0.5	0	0.7	Constant Over Programs	LCA	27954.485	27962.394	28004.125
7	0.5	0	0.9	Constant Over Programs	LCA	26986.087	26993.692	27036.458
7	1.5	0	0.1	Constant Over Programs	LCA	12344.203	12351.986	12394.128
7	1.5	0	0.3	Constant Over Programs	LCA	14100.635	14108.308	14151.928
7	1.5	0	0.5	Constant Over Programs	LCA	14609.459	14617.097	14660.726
7	1.5	0	0.7	Constant Over Programs	LCA	14111.857	14119.356	14163.667
7	1.5	0	0.9	Constant Over Programs	LCA	12379.660	12386.973	12431.906
5	0.5	0.5	0.1	Varying over Programs	LCA	18263.331	18266.373	18291.537
5	0.5	0.5	0.3	Varying over Programs	LCRE.CL	18341.159	18336.285	18363.005
5	0.5	0.5	0.5	Varying over Programs	LCRE.CL	18000.745	17995.688	18022.217
5	0.5	0.5	0.7	Varying over Programs	LCRE.CL	17362.196	17352.755	17379.252
5	1.5	0.5	0.5	Varying over Programs	LCRE.CL	13609.161	13577.343	13583.198
5	1.5	0.5	0.9	Varying over Programs	LCRE.CL	10153.125	10107.295	10110.975
7	0.5	0.5	0.1	Varying over Programs	LCRE.CL	25235.557	25227.327	25265.446
7	0.5	0.5	0.3	Varying over Programs	LCRE.CL	25436.810	25409.769	25447.293
7	0.5	0.5	0.5	Varying over Programs	LCRE.CL	25153.254	25110.100	25148.070
7	0.5	0.5	0.7	Varying over Programs	LCRE.CL	24446.563	24404.767	24444.314
7	0.5	0.5	0.9	Varying over Programs	LCRE.CL	23192.460	23173.796	23210.369
5	0.5	0.5	0.3	Constant Over Programs	LCRE.CL	19807.908	19797.525	19823.642
5	0.5	0.5	0.5	Constant Over Programs	LCRE.CL	19981.307	19962.484	19989.637
5	0.5	0.5	0.7	Constant Over Programs	LCRE.CL	19815.444	19804.865	19830.555
7	0.5	0.5	0.1	Constant Over Programs	LCRE.CL	26789.760	26772.894	26807.145
7	0.5	0.5	0.3	Constant Over Programs	LCRE.CL	27540.468	27498.804	27537.264
7	0.5	0.5	0.5	Constant Over Programs	LCRE.CL	27785.977	27727.500	27767.998
7	0.5	0.5	0.7	Constant Over Programs	LCRE.CL	27523.225	27488.627	27527.450
7	0.5	0.5	0.9	Constant Over Programs	LCRE.CL	26804.118	26791.751	26828.294
5	0.5	1	0.1	Varying over Programs	LCRE.CL	17889.306	17877.344	17901.942
5	0.5	1	0.3	Varying over Programs	LCRE.CL	17801.155	17771.205	17795.704
5	0.5	1	0.5	Varying over Programs	LCRE.CL	17428.183	17385.941	17412.158
5	0.5	1	0.7	Varying over Programs	LCRE.CL	16794.478	16760.484	16786.449
5	0.5	1	0.9	Varying over Programs	LCRE.CL	15909.193	15900.768	15925.838
5	1.5	1	0.1	Varying over Programs	LCRE.CL	14336.348	14238.816	14251.605
5	1.5	1	0.5	Varying over Programs	LCRE.CL	14992.656	14672.302	14685.994
5	1.5	1	0.7	Varying over Programs	LCRE.CL	13903.900	13507.096	13532.453
7	0.5	1	0.1	Varying over Programs	LCRE.CL	24789.650	24706.180	24718.266
7	0.5	1	0.3	Varying over Programs	LCRE.CL	24802.801	24652.525	24683.379
7	0.5	1	0.5	Varying over Programs	LCRE.CL	24469.037	24279.117	24307.365
7	0.5	1	0.7	Varying over Programs	LCRE.CL	23719.299	23556.591	23584.337
7	0.5	1	0.9	Varying over Programs	LCRE.CL	22576.612	22492.899	22515.167
7	1.5	1	0.5	Varying over Programs	LCRE.CL	19431.484	18776.098	18797.582
5	0.5	1	0.1	Constant Over Programs	LCRE.CL	18358.968	18349.987	18376.119
5	0.5	1	0.3	Constant Over Programs	LCRE.CL	18789.501	18751.648	18776.178
5	0.5	1	0.5	Constant Over Programs	LCRE.CL	18923.550	18872.116	18898.094
5	0.5	1	0.7	Constant Over Programs	LCRE.CL	18774.433	18733.232	18758.812
5	0.5	1	0.9	Constant Over Programs	LCRE.CL	18338.865	18327.482	18352.522
5	1.5	1	0.3	Constant Over Programs	LCRE.CL	14405.646	13940.504	13966.746
5	1.5	1	0.5	Constant Over Programs	LCRE.CL	14853.621	14334.476	14358.063
5	1.5	1	0.7	Constant Over Programs	LCRE.CL	14362.791	13898.407	13923.829
7	0.5	1	0.1	Constant Over Programs	LCRE.CL	25490.971	25406.389	25427.672
7	0.5	1	0.3	Constant Over Programs	LCRE.CL	26130.066	25968.613	26000.534
7	0.5	1	0.5	Constant Over Programs	LCRE.CL	26302.405	26108.018	26140.849
7	0.5	1	0.7	Constant Over Programs	LCRE.CL	26073.394	25917.515	25947.465
7	0.5	1	0.9	Constant Over Programs	LCRE.CL	25500.735	25426.647	25447.668
7	1.5	1	0.1	Constant Over Programs	LCRE.CL	17019.665	16477.672	16493.932
7	1.5	1	0.3	Constant Over Programs	LCRE.CL	18724.055	17988.416	18025.807
7	1.5	1	0.5	Constant Over Programs	LCRE.CL	19193.223	18381.449	18415.483
7	1.5	1	0.7	Constant Over Programs	LCRE.CL	18714.391	17991.819	18027.227
5	0.5	2	0.1	Varying over Programs	LCRE.CL	16397.185	16350.929	16359.656
5	0.5	2	0.3	Varying over Programs	LCRE.CL	16305.249	16238.287	16252.960
5	0.5	2	0.5	Varying over Programs	LCRE.CL	15949.876	15877.113	15895.006
5	0.5	2	0.7	Varying over Programs	LCRE.CL	15394.839	15329.059	15346.520
5	0.5	2	0.9	Varying over Programs	LCRE.CL	14622.662	14590.475	14608.797
5	1.5	2	0.1	Varying over Programs	LCRE.CL	14508.988	14324.312	14340.351
5	1.5	2	0.9	Varying over Programs	LCRE.CL	11976.806	11549.128	11557.013
5	0.5	2	0.1	Constant Over Programs	LCRE.CL	16408.024	16368.981	16384.952
5	0.5	2	0.3	Constant Over Programs	LCRE.CL	16781.182	16707.341	16726.524
5	0.5	2	0.5	Constant Over Programs	LCRE.CL	16870.746	16786.399	16805.973
5	0.5	2	0.7	Constant Over Programs	LCRE.CL	16787.199	16712.136	16731.435
5	0.5	2	0.9	Constant Over Programs	LCRE.CL	16372.227	16332.311	16347.626
5	1.5	2	0.1	Constant Over Programs	LCRE.CL	13131.786	12784.512	12788.771
5	0.5	4	0.5	Varying over Programs	LCRE.CL	14126.635	13994.092	13997.893
5	0.5	4	0.9	Varying over Programs	LCRE.CL	13180.434	13101.198	13104.337
5	1.5	4	0.1	Varying over Programs	LCRE.CL	13126.204	12884.898	12890.985
5	0.5	4	0.3	Constant Over Programs	LCRE.CL	14339.194	14224.716	14233.361
5	0.5	4	0.5	Constant Over Programs	LCRE.CL	14432.245	14304.792	14311.600
5	0.5	4	0.7	Constant Over Programs	LCRE.CL	14343.102	14222.673	14226.672
5	1.5	0.5	0.1	Varying over Programs	LCRE.NCL	12914.087	12883.976	12879.314
5	1.5	0.5	0.7	Varying over Programs	LCRE.NCL	12419.404	12419.100	12411.547
7	1.5	0.5	0.1	Varying over Programs	LCRE.NCL	16697.048	16624.914	16616.616
7	1.5	0.5	0.5	Varying over Programs	LCRE.NCL	17388.987	17311.247	17299.035
7	1.5	0.5	0.7	Varying over Programs	LCRE.NCL	16221.843	16142.770	16136.511
7	1.5	0.5	0.9	Varying over Programs	LCRE.NCL	13820.515	13732.596	13711.631
5	1.5	0.5	0.1	Constant Over Programs	LCRE.NCL	10885.652	10846.994	10841.132
5	1.5	0.5	0.5	Constant Over Programs	LCRE.NCL	12975.416	12958.678	12940.775
5	1.5	0.5	0.7	Constant Over Programs	LCRE.NCL	12495.348	12458.579	12454.383
5	1.5	0.5	0.9	Constant Over Programs	LCRE.NCL	10905.337	10869.115	10862.538
7	1.5	0.5	0.1	Constant Over Programs	LCRE.NCL	14371.189	14285.819	14263.463
7	1.5	0.5	0.3	Constant Over Programs	LCRE.NCL	16034.394	15953.918	15949.520
7	1.5	0.5	0.5	Constant Over Programs	LCRE.NCL	16549.146	16501.413	16461.554
7	1.5	0.5	0.7	Constant Over Programs	LCRE.NCL	16057.002	15982.041	15976.869
7	1.5	0.5	0.9	Constant Over Programs	LCRE.NCL	14398.725	14309.187	14294.772
5	1.5	1	0.3	Varying over Programs	LCRE.NCL	15259.410	15051.039	15031.780
5	1.5	1	0.9	Varying over Programs	LCRE.NCL	11637.090	11345.796	11309.278
7	1.5	1	0.1	Varying over Programs	LCRE.NCL	18836.827	18517.002	18482.108
7	1.5	1	0.3	Varying over Programs	LCRE.NCL	19782.002	19249.383	19009.049

p	δ	σ_z	p_0	Threshold	Preferred Model	LCA	LCRE.CL	LCRE.NCL
7	1.5	1	0.7	Varying over Programs	LCRE.NCL	18210.930	17452.210	17415.918
7	1.5	1	0.9	Varying over Programs	LCRE.NCL	15815.135	15143.548	15085.458
5	1.5	2	0.5	Varying over Programs	LCRE.NCL	15496.917	14744.816	14741.720
5	1.5	2	0.7	Varying over Programs	LCRE.NCL	14399.594	13636.741	13632.598
7	0.5	2	0.1	Varying over Programs	LCRE.NCL	22539.047	22262.705	22205.271
7	0.5	2	0.3	Varying over Programs	LCRE.NCL	22626.680	22266.527	22235.866
7	0.5	2	0.5	Varying over Programs	LCRE.NCL	22424.151	22033.319	22006.809
7	0.5	2	0.7	Varying over Programs	LCRE.NCL	21858.937	21492.227	21465.573
7	0.5	2	0.9	Varying over Programs	LCRE.NCL	21052.447	20789.469	20739.909
7	1.5	2	0.1	Varying over Programs	LCRE.NCL	19111.509	18398.887	18312.204
7	1.5	2	0.3	Varying over Programs	LCRE.NCL	20775.660	19193.260	19076.625
7	1.5	2	0.5	Varying over Programs	LCRE.NCL	20920.210	18845.522	18791.165
7	1.5	2	0.7	Varying over Programs	LCRE.NCL	19626.990	17652.850	17627.725
7	1.5	2	0.9	Varying over Programs	LCRE.NCL	16539.497	15525.618	15487.963
5	1.5	2	0.3	Constant Over Programs	LCRE.NCL	15050.243	14162.927	14153.969
5	1.5	2	0.5	Constant Over Programs	LCRE.NCL	15675.137	14583.906	14577.432
5	1.5	2	0.7	Constant Over Programs	LCRE.NCL	15040.331	14131.433	14125.280
7	0.5	2	0.1	Constant Over Programs	LCRE.NCL	22573.221	22338.076	22307.831
7	0.5	2	0.3	Constant Over Programs	LCRE.NCL	23045.071	22721.200	22708.759
7	0.5	2	0.5	Constant Over Programs	LCRE.NCL	23217.528	22840.176	22834.088
7	0.5	2	0.7	Constant Over Programs	LCRE.NCL	23003.078	22684.801	22675.941
7	0.5	2	0.9	Constant Over Programs	LCRE.NCL	22568.403	22335.185	22300.698
7	1.5	2	0.1	Constant Over Programs	LCRE.NCL	17785.805	16915.643	16824.055
7	1.5	2	0.3	Constant Over Programs	LCRE.NCL	20284.965	18387.279	18331.594
7	1.5	2	0.5	Constant Over Programs	LCRE.NCL	20965.011	18759.359	18718.277
7	1.5	2	0.7	Constant Over Programs	LCRE.NCL	20174.512	18312.994	18255.239
7	1.5	2	0.9	Constant Over Programs	LCRE.NCL	17695.294	16875.884	16799.202
5	0.5	4	0.1	Varying over Programs	LCRE.NCL	14310.403	14207.085	14199.749
5	1.5	4	0.3	Varying over Programs	LCRE.NCL	14253.944	13708.091	13676.432
5	1.5	4	0.5	Varying over Programs	LCRE.NCL	14331.064	13599.719	13563.604
5	1.5	4	0.7	Varying over Programs	LCRE.NCL	13457.305	12773.513	12767.570
5	1.5	4	0.9	Varying over Programs	LCRE.NCL	11299.440	10974.081	10968.751
7	0.5	4	0.1	Varying over Programs	LCRE.NCL	18869.035	18370.226	18267.580
7	0.5	4	0.3	Varying over Programs	LCRE.NCL	19134.707	18567.644	18483.958
7	0.5	4	0.5	Varying over Programs	LCRE.NCL	19152.358	18559.678	18470.070
7	0.5	4	0.7	Varying over Programs	LCRE.NCL	18976.873	18387.901	18294.564
7	0.5	4	0.9	Varying over Programs	LCRE.NCL	18560.967	18054.131	17937.331
7	1.5	4	0.1	Varying over Programs	LCRE.NCL	17109.763	16207.579	16134.156
7	1.5	4	0.3	Varying over Programs	LCRE.NCL	18970.088	17314.155	17221.254
7	1.5	4	0.5	Varying over Programs	LCRE.NCL	19448.807	17427.187	17356.764
7	1.5	4	0.7	Varying over Programs	LCRE.NCL	18522.334	16721.059	16653.638
7	1.5	4	0.9	Varying over Programs	LCRE.NCL	16205.147	15178.769	15106.958
5	0.5	4	0.9	Constant Over Programs	LCRE.NCL	14045.296	13951.959	13946.694
5	1.5	4	0.1	Constant Over Programs	LCRE.NCL	12244.441	11952.154	11934.550
5	1.5	4	0.3	Constant Over Programs	LCRE.NCL	14003.017	13268.418	13246.882
5	1.5	4	0.5	Constant Over Programs	LCRE.NCL	14498.674	13579.095	13559.495
5	1.5	4	0.7	Constant Over Programs	LCRE.NCL	13993.960	13221.179	13199.711
5	1.5	4	0.9	Constant Over Programs	LCRE.NCL	12189.568	11894.748	11884.752
7	0.5	4	0.1	Constant Over Programs	LCRE.NCL	18669.067	18242.436	18168.137
7	0.5	4	0.3	Constant Over Programs	LCRE.NCL	19079.519	18581.831	18528.626
7	0.5	4	0.5	Constant Over Programs	LCRE.NCL	19159.237	18641.341	18593.640
7	0.5	4	0.7	Constant Over Programs	LCRE.NCL	19057.709	18564.379	18515.992
7	0.5	4	0.9	Constant Over Programs	LCRE.NCL	18702.217	18294.128	18223.127
7	1.5	4	0.1	Constant Over Programs	LCRE.NCL	16517.742	15613.752	15540.397
7	1.5	4	0.3	Constant Over Programs	LCRE.NCL	18677.154	16944.853	16886.883
7	1.5	4	0.5	Constant Over Programs	LCRE.NCL	19406.954	17368.654	17315.810
7	1.5	4	0.7	Constant Over Programs	LCRE.NCL	18638.880	16888.906	16819.342
7	1.5	4	0.9	Constant Over Programs	LCRE.NCL	16508.297	15601.387	15570.667
5	0.5	0.5	0.9	Varying over Programs	LCA	16356.983	16357.353	16381.739
5	1.5	0.5	0.3	Varying over Programs	LCRE.CL	13861.994	13831.720	13833.652
7	1.5	0.5	0.3	Varying over Programs	LCRE.CL	17638.889	17566.864	17563.979
5	0.5	0.5	0.1	Constant Over Programs	LCA	19218.581	19220.299	19247.320
5	0.5	0.5	0.9	Constant Over Programs	LCA	19234.291	19233.230	19259.768
5	1.5	0.5	0.3	Constant Over Programs	LCRE.CL	12568.638	12535.368	12534.084
5	1.5	1	0.1	Constant Over Programs	LCRE.CL	12734.590	12492.102	12490.875
5	1.5	1	0.9	Constant Over Programs	LCRE.CL	12777.799	12519.563	12519.091
7	1.5	1	0.9	Constant Over Programs	LCRE.CL	16969.527	16465.210	16465.467
5	1.5	2	0.3	Varying over Programs	LCRE.CL	15542.633	15027.556	15027.169
5	1.5	2	0.9	Constant Over Programs	LCRE.CL	13111.870	12773.070	12772.430
5	0.5	4	0.3	Varying over Programs	LCRE.CL	14274.565	14158.264	14159.499
5	0.5	4	0.7	Varying over Programs	LCRE.CL	13734.602	13625.051	13626.569
5	0.5	4	0.1	Constant Over Programs	LCRE.CL	14061.092	13973.061	13974.866

B.1.4 Sum of Scores with matching replicate results

p	δ	σ_z	p_0	Threshold	LCA	LCRE (CL)	LCRE (No CL)	Sum of Scores
5	0.5	0	0.1	Varying over Programs	0.75108	0.24617	0.01034	0.68779
5	0.5	0	0.3	Varying over Programs	0.83091	0.83091	0.83091	0.78973
5	0.5	0	0.5	Varying over Programs	0.84685	0.84685	0.84685	0.82845
5	0.5	0	0.7	Varying over Programs	0.84163	0.46025	0.79841	0.80806
5	0.5	0	0.9	Varying over Programs	0.76597	0.76156	0.75463	0.70606
5	1.5	0	0.1	Varying over Programs	0.97687	0.97670	0.97591	0.85253
5	1.5	0	0.3	Varying over Programs	0.98481	0.98497	0.98356	0.93569
5	1.5	0	0.5	Varying over Programs	0.99017	0.99016	0.98891	0.95152
5	1.5	0	0.7	Varying over Programs	0.98801	0.98800	0.98787	0.94754
5	1.5	0	0.9	Varying over Programs	0.98202	0.98206	0.97832	0.89758
7	0.5	0	0.1	Varying over Programs	0.80379	0.79618	0.74993	0.64622
7	0.5	0	0.3	Varying over Programs	0.85588	0.85539	0.85034	0.80090
7	0.5	0	0.5	Varying over Programs	0.85959	0.85959	0.85959	0.83092
7	0.5	0	0.7	Varying over Programs	0.85962	0.85966	0.85800	0.81231
7	0.5	0	0.9	Varying over Programs	0.78081	0.76413	0.76707	0.66437
7	1.5	0	0.1	Varying over Programs	0.99728	0.99728	0.99728	0.90381
7	1.5	0	0.3	Varying over Programs	0.99722	0.99722	0.99722	0.95650
7	1.5	0	0.5	Varying over Programs	0.99737	0.99736	0.99738	0.96444
7	1.5	0	0.7	Varying over Programs	0.99577	0.99581	0.99575	0.95988
7	1.5	0	0.9	Varying over Programs	0.99519	0.99519	0.99519	0.92367
5	0.5	0	0.1	Constant Over Programs	0.83063	0.83069	0.77977	0.75191
5	0.5	0	0.3	Constant Over Programs	0.87124	0.87053	0.76590	0.83585
5	0.5	0	0.5	Constant Over Programs	0.88526	0.88526	0.88526	0.86232
5	0.5	0	0.7	Constant Over Programs	0.87035	0.86803	0.84986	0.82891
5	0.5	0	0.9	Constant Over Programs	0.80184	0.73322	0.78246	0.72466
5	1.5	0	0.1	Constant Over Programs	0.99105	0.99081	0.99027	0.91465
5	1.5	0	0.3	Constant Over Programs	0.99270	0.99270	0.99270	0.96256
5	1.5	0	0.5	Constant Over Programs	0.99527	0.99527	0.99527	0.96876
5	1.5	0	0.7	Constant Over Programs	0.99213	0.99231	0.99155	0.96154
5	1.5	0	0.9	Constant Over Programs	0.98898	0.98869	0.98796	0.91533
7	0.5	0	0.1	Constant Over Programs	0.83055	0.83062	0.83062	0.68402
7	0.5	0	0.3	Constant Over Programs	0.88925	0.88925	0.88925	0.83616
7	0.5	0	0.5	Constant Over Programs	0.89320	0.89278	0.87778	0.86237
7	0.5	0	0.7	Constant Over Programs	0.88014	0.87982	0.87809	0.83795
7	0.5	0	0.9	Constant Over Programs	0.83810	0.83011	0.80212	0.69601
7	1.5	0	0.1	Constant Over Programs	0.99864	0.99864	0.99864	0.94307
7	1.5	0	0.3	Constant Over Programs	0.99867	0.99868	0.99877	0.97231
7	1.5	0	0.5	Constant Over Programs	0.99925	0.99924	0.99907	0.97653
7	1.5	0	0.7	Constant Over Programs	0.99854	0.99853	0.99853	0.97117
7	1.5	0	0.9	Constant Over Programs	0.99728	0.99728	0.99728	0.94025
5	0.5	0.5	0.1	Varying over Programs	0.57951	0.50165	0.08076	0.69766
5	0.5	0.5	0.3	Varying over Programs	0.76713	0.75644	0.77517	0.75753
5	0.5	0.5	0.5	Varying over Programs	0.76040	0.57728	0.34301	0.82014
5	0.5	0.5	0.7	Varying over Programs	0.84181	0.83622	0.82370	0.84945
5	0.5	0.5	0.9	Varying over Programs	0.58581	0.61705	0.52517	0.47817
5	1.5	0.5	0.1	Varying over Programs	0.95558	0.95775	0.96672	0.82527
5	1.5	0.5	0.3	Varying over Programs	0.96679	0.96665	0.97576	0.91902
5	1.5	0.5	0.5	Varying over Programs	0.97736	0.97624	0.97681	0.93736

p	δ	σ_z	p_0	Threshold	LCA	LCRE (CL)	LCRE (No CL)	Sum of Scores
5	1.5	0.5	0.7	Varying over Programs	0.96602	0.85910	0.97456	0.92430
5	1.5	0.5	0.9	Varying over Programs	0.96128	0.96722	0.96603	0.86128
7	0.5	0.5	0.1	Varying over Programs	0.51043	0.49771	0.49888	0.48231
7	0.5	0.5	0.3	Varying over Programs	0.76772	0.65157	0.67129	0.80363
7	0.5	0.5	0.5	Varying over Programs	0.83040	0.67549	0.68838	0.85499
7	0.5	0.5	0.7	Varying over Programs	0.72915	0.82818	0.75695	0.65478
7	0.5	0.5	0.9	Varying over Programs	0.62690	0.69646	0.53689	0.53347
7	1.5	0.5	0.1	Varying over Programs	0.97822	0.97903	0.98271	0.86154
7	1.5	0.5	0.3	Varying over Programs	0.98632	0.98623	0.98902	0.93617
7	1.5	0.5	0.5	Varying over Programs	0.98345	0.98401	0.98648	0.94788
7	1.5	0.5	0.7	Varying over Programs	0.98360	0.98473	0.98674	0.94132
7	1.5	0.5	0.9	Varying over Programs	0.97708	0.97929	0.98787	0.88509
5	0.5	0.5	0.1	Constant Over Programs	0.70550	0.70278	0.05657	0.63888
5	0.5	0.5	0.3	Constant Over Programs	0.83960	0.78664	0.73375	0.83227
5	0.5	0.5	0.5	Constant Over Programs	0.83629	0.55999	0.40784	0.87806
5	0.5	0.5	0.7	Constant Over Programs	0.79509	0.54585	0.22158	0.86056
5	0.5	0.5	0.9	Constant Over Programs	0.81456	0.78044	0.78469	0.85154
5	1.5	0.5	0.1	Constant Over Programs	0.97324	0.97737	0.98452	0.88063
5	1.5	0.5	0.3	Constant Over Programs	0.97500	0.97852	0.98359	0.94405
5	1.5	0.5	0.5	Constant Over Programs	0.97644	0.85194	0.98280	0.95180
5	1.5	0.5	0.7	Constant Over Programs	0.97400	0.97944	0.98051	0.94091
5	1.5	0.5	0.9	Constant Over Programs	0.93741	0.95879	0.96877	0.86903
7	0.5	0.5	0.1	Constant Over Programs	0.62759	0.49543	0.51338	0.75177
7	0.5	0.5	0.3	Constant Over Programs	0.78744	0.64975	0.68623	0.82606
7	0.5	0.5	0.5	Constant Over Programs	0.84349	0.69127	0.32055	0.86602
7	0.5	0.5	0.7	Constant Over Programs	0.76442	0.59148	0.62306	0.82787
7	0.5	0.5	0.9	Constant Over Programs	0.76883	0.79802	0.22631	0.71235
7	1.5	0.5	0.1	Constant Over Programs	0.98507	0.98586	0.98768	0.90284
7	1.5	0.5	0.3	Constant Over Programs	0.98794	0.99054	0.99294	0.95264
7	1.5	0.5	0.5	Constant Over Programs	0.99290	0.99356	0.99517	0.96341
7	1.5	0.5	0.7	Constant Over Programs	0.98644	0.98779	0.98941	0.95265
7	1.5	0.5	0.9	Constant Over Programs	0.97894	0.98217	0.98841	0.90209
5	0.5	1	0.1	Varying over Programs	0.76864	0.78187	0.76058	0.71224
5	0.5	1	0.3	Varying over Programs	0.81208	0.82165	0.07285	0.78908
5	0.5	1	0.5	Varying over Programs	0.65866	0.59568	0.61503	0.82296
5	0.5	1	0.7	Varying over Programs	0.78700	0.78206	0.76206	0.84449
5	0.5	1	0.9	Varying over Programs	0.79736	0.78570	0.78516	0.87443
5	1.5	1	0.1	Varying over Programs	0.64306	0.59286	0.55900	0.71036
5	1.5	1	0.3	Varying over Programs	0.82921	0.04676	0.30965	0.85298
5	1.5	1	0.5	Varying over Programs	0.88234	0.59715	0.68288	0.88315
5	1.5	1	0.7	Varying over Programs	0.85255	0.72293	0.67808	0.86045
5	1.5	1	0.9	Varying over Programs	0.61975	0.68903	0.71657	0.74151
7	0.5	1	0.1	Varying over Programs	0.69947	0.67329	0.64777	0.75063
7	0.5	1	0.3	Varying over Programs	0.57944	0.44727	0.40872	0.74885
7	0.5	1	0.5	Varying over Programs	0.82384	0.73880	0.72535	0.87240
7	0.5	1	0.7	Varying over Programs	0.73353	0.61230	0.25995	0.86838
7	0.5	1	0.9	Varying over Programs	0.71599	0.73908	0.57486	0.69019
7	1.5	1	0.1	Varying over Programs	0.71194	0.65079	0.56092	0.75045
7	1.5	1	0.3	Varying over Programs	0.90066	0.91357	0.96883	0.87740
7	1.5	1	0.5	Varying over Programs	0.93428	0.56549	0.47154	0.90295
7	1.5	1	0.7	Varying over Programs	0.91621	0.76991	0.83425	0.88903
7	1.5	1	0.9	Varying over Programs	0.73261	0.48640	0.56041	0.80049
5	0.5	1	0.1	Constant Over Programs	0.70163	0.71235	0.71534	0.67395
5	0.5	1	0.3	Constant Over Programs	0.51115	0.44415	0.40792	0.73590
5	0.5	1	0.5	Constant Over Programs	0.82617	0.83392	0.82926	0.80895
5	0.5	1	0.7	Constant Over Programs	0.72858	0.67602	0.68889	0.86607
5	0.5	1	0.9	Constant Over Programs	0.83147	0.85367	0.86790	0.71870
5	1.5	1	0.1	Constant Over Programs	0.68430	0.54774	0.64521	0.76603
5	1.5	1	0.3	Constant Over Programs	0.84558	0.67160	0.76942	0.87635
5	1.5	1	0.5	Constant Over Programs	0.87729	0.78289	0.71573	0.89283
5	1.5	1	0.7	Constant Over Programs	0.84914	0.77883	0.80983	0.86967
5	1.5	1	0.9	Constant Over Programs	0.62584	0.39137	0.61845	0.73980
7	0.5	1	0.1	Constant Over Programs	0.70243	0.69274	0.16541	0.73884
7	0.5	1	0.3	Constant Over Programs	0.59428	0.41288	0.18218	0.77390
7	0.5	1	0.5	Constant Over Programs	0.80072	0.68086	0.20827	0.89803
7	0.5	1	0.7	Constant Over Programs	0.75319	0.65317	0.64115	0.88499
7	0.5	1	0.9	Constant Over Programs	0.73016	0.73346	0.71569	0.74838
7	1.5	1	0.1	Constant Over Programs	0.72268	0.33935	0.50074	0.79271
7	1.5	1	0.3	Constant Over Programs	0.91421	0.63359	0.56787	0.89780
7	1.5	1	0.5	Constant Over Programs	0.92852	0.69592	0.07623	0.91361
7	1.5	1	0.7	Constant Over Programs	0.90891	0.60314	0.54556	0.89644
7	1.5	1	0.9	Constant Over Programs	0.71822	0.64209	0.56555	0.79734
5	0.5	2	0.1	Varying over Programs	0.77538	0.77760	0.75010	0.84219
5	0.5	2	0.3	Varying over Programs	0.85309	0.86861	0.87854	0.82060
5	0.5	2	0.5	Varying over Programs	0.77923	0.79772	0.78712	0.76082
5	0.5	2	0.7	Varying over Programs	0.69306	0.68911	0.68546	0.80963

p	δ	σ_z	p_0	Threshold	Preferred Model	LCA	LCRE.CL	LCRE.NCL
5	0.5	2	0.9	Varying over Programs	0.75912	0.76506	0.78399	0.80221
5	1.5	2	0.1	Varying over Programs	0.29573	0.20218	0.25991	0.50705
5	1.5	2	0.3	Varying over Programs	0.46888	0.12972	0.64336	0.68881
5	1.5	2	0.5	Varying over Programs	0.52770	0.02298	0.76116	0.74061
5	1.5	2	0.7	Varying over Programs	0.47832	0.37817	0.86495	0.70583
5	1.5	2	0.9	Varying over Programs	0.29785	0.27053	0.80208	0.54437
7	0.5	2	0.1	Varying over Programs	0.64830	0.64920	0.64229	0.71896
7	0.5	2	0.3	Varying over Programs	0.78394	0.75672	0.08610	0.86584
7	0.5	2	0.5	Varying over Programs	0.77192	0.72185	0.65380	0.89062
7	0.5	2	0.7	Varying over Programs	0.65713	0.60546	0.22621	0.80427
7	0.5	2	0.9	Varying over Programs	0.79764	0.76259	0.54810	0.91282
7	1.5	2	0.1	Varying over Programs	0.37588	0.12440	0.92583	0.54760
7	1.5	2	0.3	Varying over Programs	0.57954	0.00933	0.94854	0.72950
7	1.5	2	0.5	Varying over Programs	0.65640	0.18928	0.94570	0.77182
7	1.5	2	0.7	Varying over Programs	0.55920	0.29265	0.94277	0.75296
7	1.5	2	0.9	Varying over Programs	0.36959	0.07529	0.20828	0.62258
5	0.5	2	0.1	Constant Over Programs	0.65959	0.64219	0.63574	0.83516
5	0.5	2	0.3	Constant Over Programs	0.80173	0.81653	0.81939	0.79669
5	0.5	2	0.5	Constant Over Programs	0.83571	0.83342	0.82836	0.88463
5	0.5	2	0.7	Constant Over Programs	0.87132	0.87781	0.87732	0.87039
5	0.5	2	0.9	Constant Over Programs	0.81684	0.81325	0.81704	0.89559
5	1.5	2	0.1	Constant Over Programs	0.30094	0.01373	0.93167	0.55167
5	1.5	2	0.3	Constant Over Programs	0.48232	0.15488	0.88384	0.72322
5	1.5	2	0.5	Constant Over Programs	0.53907	0.34859	0.92972	0.75812
5	1.5	2	0.7	Constant Over Programs	0.47264	0.30831	0.87292	0.71017
5	1.5	2	0.9	Constant Over Programs	0.30207	0.06068	0.73034	0.53807
7	0.5	2	0.1	Constant Over Programs	0.69153	0.69235	0.69793	0.76378
7	0.5	2	0.3	Constant Over Programs	0.76454	0.72796	0.18074	0.88850
7	0.5	2	0.5	Constant Over Programs	0.76981	0.76561	0.69163	0.82936
7	0.5	2	0.7	Constant Over Programs	0.77853	0.75686	0.75058	0.89361
7	0.5	2	0.9	Constant Over Programs	0.76923	0.75387	0.35251	0.86285
7	1.5	2	0.1	Constant Over Programs	0.35546	0.07063	0.93678	0.57577
7	1.5	2	0.3	Constant Over Programs	0.55488	0.11760	0.95905	0.74692
7	1.5	2	0.5	Constant Over Programs	0.62247	0.27150	0.95563	0.78123
7	1.5	2	0.7	Constant Over Programs	0.56192	0.08002	0.94795	0.74254
7	1.5	2	0.9	Constant Over Programs	0.38850	0.15479	0.92418	0.59642
5	0.5	4	0.1	Varying over Programs	0.81539	0.81961	0.81575	0.88613
5	0.5	4	0.3	Varying over Programs	0.84515	0.85030	0.85465	0.87558
5	0.5	4	0.5	Varying over Programs	0.80409	0.80029	0.80239	0.90583
5	0.5	4	0.7	Varying over Programs	0.84839	0.85396	0.85169	0.89567
5	0.5	4	0.9	Varying over Programs	0.86819	0.87732	0.88285	0.87676
5	1.5	4	0.1	Varying over Programs	0.14542	0.08555	0.12600	0.35799
5	1.5	4	0.3	Varying over Programs	0.23407	0.10833	0.84335	0.51205
5	1.5	4	0.5	Varying over Programs	0.82789	0.84196	0.12036	0.78736
5	1.5	4	0.7	Varying over Programs	0.20800	0.01756	0.90227	0.51852
5	1.5	4	0.9	Varying over Programs	0.12094	0.00222	0.82211	0.36587
7	0.5	4	0.1	Varying over Programs	0.79661	0.78494	0.77146	0.89500
7	0.5	4	0.3	Varying over Programs	0.71612	0.68761	0.32437	0.84585
7	0.5	4	0.5	Varying over Programs	0.77966	0.75007	0.67367	0.89960
7	0.5	4	0.7	Varying over Programs	0.74357	0.71802	0.63093	0.88264
7	0.5	4	0.9	Varying over Programs	0.77497	0.76500	0.02330	0.88551
7	1.5	4	0.1	Varying over Programs	0.16286	0.06902	0.87050	0.35520
7	1.5	4	0.3	Varying over Programs	0.27864	0.06560	0.91471	0.52484
7	1.5	4	0.5	Varying over Programs	0.32460	0.04681	0.89829	0.57499
7	1.5	4	0.7	Varying over Programs	0.28829	0.03377	0.90425	0.55008
7	1.5	4	0.9	Varying over Programs	0.20151	0.07876	0.82496	0.41455
5	0.5	4	0.1	Constant Over Programs	0.77920	0.78445	0.77881	0.81802
5	0.5	4	0.3	Constant Over Programs	0.84123	0.84524	0.84254	0.86821
5	0.5	4	0.5	Constant Over Programs	0.80898	0.81558	0.80653	0.84140
5	0.5	4	0.7	Constant Over Programs	0.84711	0.84905	0.85249	0.88725
5	0.5	4	0.9	Constant Over Programs	0.84455	0.85497	0.85227	0.82588
5	1.5	4	0.1	Constant Over Programs	0.13923	0.02656	0.82573	0.37837
5	1.5	4	0.3	Constant Over Programs	0.23125	0.07369	0.60140	0.53472
5	1.5	4	0.5	Constant Over Programs	0.25252	0.04791	0.90433	0.57401
5	1.5	4	0.7	Constant Over Programs	0.19917	0.01384	0.90168	0.51431
5	1.5	4	0.9	Constant Over Programs	0.13001	0.05399	0.10030	0.35361
7	0.5	4	0.1	Constant Over Programs	0.80527	0.80257	0.80571	0.86573
7	0.5	4	0.3	Constant Over Programs	0.76715	0.77638	0.33022	0.81875
7	0.5	4	0.5	Constant Over Programs	0.63508	0.61506	0.23675	0.76580
7	0.5	4	0.7	Constant Over Programs	0.75464	0.73283	0.22206	0.88218
7	0.5	4	0.9	Constant Over Programs	0.79572	0.78968	0.75291	0.90318
7	1.5	4	0.1	Constant Over Programs	0.17065	0.06002	0.85391	0.37184
7	1.5	4	0.3	Constant Over Programs	0.28308	0.08058	0.91703	0.53637
7	1.5	4	0.5	Constant Over Programs	0.31882	0.04851	0.90771	0.57532
7	1.5	4	0.7	Constant Over Programs	0.28433	0.06957	0.91779	0.53955
7	1.5	4	0.9	Constant Over Programs	0.20293	0.09142	0.83473	0.39188

B.2 Investigating preference of One Class vs Two Class models using the BIC

B.2.1 Frequencies of one class preference for data with two clusters

p	δ	σ_z	p_0	Threshold	LCA	LCRE.CL	LCRE.NCL	p	δ	σ_z	p_0	Threshold	LCA	LCRE.CL	LCRE.NCL
5	0.5	0.0	0.1	varied	9	15	15	7	0.5	0.0	0.7	varied	0	11	12
5	0.5	0.0	0.3	varied	1	14	14	7	0.5	0.0	0.9	varied	2	10	11
5	0.5	0.0	0.5	varied	0	15	15	7	0.5	2.0	0.1	varied	0	0	12
5	0.5	0.0	0.7	varied	0	14	14	7	0.5	2.0	0.3	varied	0	0	11
5	0.5	0.0	0.9	varied	6	11	11	7	0.5	2.0	0.5	varied	0	0	14
5	0.5	2.0	0.1	varied	0	0	12	7	0.5	2.0	0.7	varied	0	0	13
5	0.5	2.0	0.3	varied	0	0	14	7	0.5	2.0	0.9	varied	0	0	14
5	0.5	2.0	0.5	varied	0	0	10	7	0.5	4.0	0.1	varied	0	0	10
5	0.5	2.0	0.7	varied	0	0	10	7	0.5	4.0	0.3	varied	0	0	11
5	0.5	2.0	0.9	varied	0	0	11	7	0.5	4.0	0.5	varied	0	0	15
5	0.5	4.0	0.1	varied	0	0	13	7	0.5	4.0	0.7	varied	0	0	10
5	0.5	4.0	0.3	varied	0	0	9	7	0.5	4.0	0.9	varied	0	0	14
5	0.5	4.0	0.5	varied	0	0	13	7	1.5	0.0	0.1	varied	0	0	0
5	0.5	4.0	0.7	varied	0	0	7	7	1.5	0.0	0.3	varied	0	0	0
5	0.5	4.0	0.9	varied	0	0	13	7	1.5	0.0	0.5	varied	0	0	0
5	1.5	0.0	0.1	varied	0	0	1	7	1.5	0.0	0.7	varied	0	0	0
5	1.5	0.0	0.3	varied	0	0	0	7	1.5	0.0	0.9	varied	0	0	0
5	1.5	0.0	0.5	varied	0	1	2	7	1.5	2.0	0.1	varied	0	0	2
5	1.5	0.0	0.7	varied	0	0	0	7	1.5	2.0	0.3	varied	0	0	0
5	1.5	0.0	0.9	varied	0	0	2	7	1.5	2.0	0.5	varied	0	0	0
5	1.5	2.0	0.1	varied	0	0	6	7	1.5	2.0	0.7	varied	0	0	0
5	1.5	2.0	0.3	varied	0	0	1	7	1.5	2.0	0.9	varied	0	0	1
5	1.5	2.0	0.5	varied	0	0	0	7	1.5	4.0	0.1	varied	0	0	3
5	1.5	2.0	0.7	varied	0	0	0	7	1.5	4.0	0.3	varied	0	0	0
5	1.5	2.0	0.9	varied	0	0	0	7	1.5	4.0	0.5	varied	0	0	0
5	1.5	4.0	0.1	varied	0	0	5	7	1.5	4.0	0.7	varied	0	0	0
5	1.5	4.0	0.3	varied	0	0	0	7	1.5	4.0	0.9	varied	0	0	4
5	1.5	4.0	0.5	varied	0	0	0	5	0.5	0.5	0.1	varied	0	5	14
5	1.5	4.0	0.7	varied	0	0	0	5	0.5	0.5	0.3	varied	0	6	11
5	1.5	4.0	0.9	varied	0	0	1	5	0.5	0.5	0.5	varied	0	5	9
7	0.5	0.0	0.1	varied	3	6	6	5	0.5	0.5	0.7	varied	0	4	10
7	0.5	0.0	0.3	varied	0	9	10	5	0.5	0.5	0.9	varied	0	7	9
7	0.5	0.0	0.5	varied	0	10	12	5	1.5	0.5	0.1	varied	0	1	2

p	δ	σ_z	p_0	Threshold	LCA	LCRE.CL	LCRE.NCL
5	1.5	0.5	0.3	varied	0	2	2
5	1.5	0.5	0.5	varied	0	2	2
5	1.5	0.5	0.7	varied	0	4	5
5	1.5	0.5	0.9	varied	0	2	2
7	0.5	0.5	0.1	varied	0	3	11
7	0.5	0.5	0.3	varied	0	7	13
7	0.5	0.5	0.5	varied	0	10	13
7	0.5	0.5	0.7	varied	0	10	15
7	0.5	0.5	0.9	varied	0	6	13
7	1.5	0.5	0.1	varied	0	0	1
7	1.5	0.5	0.3	varied	0	0	0
7	1.5	0.5	0.5	varied	0	0	0
7	1.5	0.5	0.7	varied	0	1	2
7	1.5	0.5	0.9	varied	0	1	1
5	0.5	1.0	0.1	varied	0	0	14
5	0.5	1.0	0.3	varied	0	0	12
5	0.5	1.0	0.5	varied	0	0	8
5	0.5	1.0	0.7	varied	0	0	13
5	0.5	1.0	0.9	varied	0	0	10
5	1.5	1.0	0.1	varied	0	3	7
5	1.5	1.0	0.3	varied	0	5	8
5	1.5	1.0	0.5	varied	0	2	2
5	1.5	1.0	0.7	varied	0	1	1
5	1.5	1.0	0.9	varied	0	6	3
7	0.5	1.0	0.1	varied	0	0	12
7	0.5	1.0	0.3	varied	0	0	15
7	0.5	1.0	0.5	varied	0	0	14
7	0.5	1.0	0.7	varied	0	0	15
7	0.5	1.0	0.9	varied	0	0	14
7	1.5	1.0	0.1	varied	0	8	10
7	1.5	1.0	0.3	varied	0	3	0
7	1.5	1.0	0.5	varied	0	5	5
7	1.5	1.0	0.7	varied	0	3	2
7	1.5	1.0	0.9	varied	0	9	6
5	0.5	0.0	0.1	constant	5	10	10
5	0.5	0.0	0.3	constant	0	15	15
5	0.5	0.0	0.5	constant	0	16	16
5	0.5	0.0	0.7	constant	0	14	13
5	0.5	0.0	0.9	constant	5	13	13
5	0.5	2.0	0.1	constant	0	0	15
5	0.5	2.0	0.3	constant	0	0	9
5	0.5	2.0	0.5	constant	0	0	8
5	0.5	2.0	0.7	constant	0	0	11
5	0.5	2.0	0.9	constant	0	0	9
5	0.5	4.0	0.1	constant	0	0	12
5	0.5	4.0	0.3	constant	0	0	10
5	0.5	4.0	0.5	constant	0	0	8
5	0.5	4.0	0.7	constant	0	0	8
5	0.5	4.0	0.9	constant	0	0	12
5	1.5	0.0	0.1	constant	0	0	0
5	1.5	0.0	0.3	constant	0	0	0
5	1.5	0.0	0.5	constant	0	2	2
5	1.5	0.0	0.7	constant	0	0	0
5	1.5	0.0	0.9	constant	0	0	0
5	1.5	2.0	0.1	constant	0	0	3
5	1.5	2.0	0.3	constant	0	0	0
5	1.5	2.0	0.5	constant	0	0	0
5	1.5	2.0	0.7	constant	0	0	0
5	1.5	2.0	0.9	constant	0	0	4
5	1.5	4.0	0.1	constant	0	0	4
5	1.5	4.0	0.3	constant	0	0	1
5	1.5	4.0	0.5	constant	0	0	0
5	1.5	4.0	0.7	constant	0	0	0
5	1.5	4.0	0.9	constant	0	0	3
7	0.5	0.0	0.1	constant	0	10	11
7	0.5	0.0	0.3	constant	0	12	13
7	0.5	0.0	0.5	constant	0	7	7
7	0.5	0.0	0.7	constant	0	9	9
7	0.5	0.0	0.9	constant	0	9	8
7	0.5	2.0	0.1	constant	0	0	13
7	0.5	2.0	0.3	constant	0	0	14
7	0.5	2.0	0.5	constant	0	0	13
7	0.5	2.0	0.7	constant	0	0	12
7	0.5	2.0	0.9	constant	0	0	13
7	0.5	4.0	0.1	constant	0	0	8
7	0.5	4.0	0.3	constant	0	0	12
7	0.5	4.0	0.5	constant	0	0	11
7	0.5	4.0	0.7	constant	0	0	13
7	0.5	4.0	0.9	constant	0	0	15
7	1.5	0.0	0.1	constant	0	0	0
7	1.5	0.0	0.3	constant	0	0	0
7	1.5	0.0	0.5	constant	0	0	0
7	1.5	0.0	0.7	constant	0	0	0
7	1.5	0.0	0.9	constant	0	0	0
7	1.5	2.0	0.1	constant	0	0	0
7	1.5	2.0	0.3	constant	0	0	0
7	1.5	2.0	0.5	constant	0	0	0
7	1.5	2.0	0.7	constant	0	0	0
7	1.5	2.0	0.9	constant	0	0	3
7	1.5	4.0	0.1	constant	0	0	2
7	1.5	4.0	0.3	constant	0	0	0
7	1.5	4.0	0.5	constant	0	0	0
7	1.5	4.0	0.7	constant	0	0	0
7	1.5	4.0	0.9	constant	0	0	4
5	0.5	0.5	0.1	constant	0	8	15
5	0.5	0.5	0.3	constant	0	6	13
5	0.5	0.5	0.5	constant	0	5	10
5	0.5	0.5	0.7	constant	0	2	10

p	δ	σ_z	p_0	Threshold	LCA	LCRE.CL	LCRE.NCL	p	δ	σ_z	p_0	Threshold	LCA	LCRE.CL	LCRE.NCL
5	0.5	0.5	0.9	constant	0	3	10	5	0.5	1.0	0.5	constant	0	0	10
5	1.5	0.5	0.1	constant	0	5	3	5	0.5	1.0	0.7	constant	0	0	8
5	1.5	0.5	0.3	constant	0	2	2	5	0.5	1.0	0.9	constant	0	0	9
5	1.5	0.5	0.5	constant	0	5	4	5	1.5	1.0	0.1	constant	0	4	5
5	1.5	0.5	0.7	constant	0	3	3	5	1.5	1.0	0.3	constant	0	2	2
5	1.5	0.5	0.9	constant	0	2	2	5	1.5	1.0	0.5	constant	0	2	2
7	0.5	0.5	0.1	constant	0	1	10	5	1.5	1.0	0.7	constant	0	1	1
7	0.5	0.5	0.3	constant	0	6	11	5	1.5	1.0	0.9	constant	0	5	6
7	0.5	0.5	0.5	constant	0	10	15	7	0.5	1.0	0.1	constant	0	0	11
7	0.5	0.5	0.7	constant	0	7	17	7	0.5	1.0	0.3	constant	0	0	14
7	0.5	0.5	0.9	constant	0	6	15	7	0.5	1.0	0.5	constant	0	0	13
7	1.5	0.5	0.1	constant	0	1	0	7	0.5	1.0	0.7	constant	0	0	15
7	1.5	0.5	0.3	constant	0	0	1	7	0.5	1.0	0.9	constant	0	0	15
7	1.5	0.5	0.5	constant	0	4	3	7	1.5	1.0	0.1	constant	0	8	7
7	1.5	0.5	0.7	constant	0	2	2	7	1.5	1.0	0.3	constant	0	4	5
7	1.5	0.5	0.9	constant	0	3	3	7	1.5	1.0	0.5	constant	0	5	6
5	0.5	1.0	0.1	constant	0	0	14	7	1.5	1.0	0.7	constant	0	4	5
5	0.5	1.0	0.3	constant	0	0	8	7	1.5	1.0	0.9	constant	0	7	7

B.2.2 Frequencies of one class preference for data with one clusters

p	σ_z	p_0	Threshold	LCA	LCRE.CL	LCRE.NCL	p	σ_z	p_0	Threshold	LCA	LCRE.CL	LCRE.NCL
5	0.0	0.1	varied	20	20	20	7	0.0	0.1	varied	20	20	20
5	0.0	0.3	varied	20	20	20	7	0.0	0.3	varied	20	20	20
5	0.0	0.5	varied	20	20	20	7	0.0	0.5	varied	20	20	20
5	0.0	0.7	varied	20	20	20	7	0.0	0.7	varied	20	20	20
5	0.0	0.9	varied	20	20	20	7	0.0	0.9	varied	20	20	20
5	2.0	0.1	varied	0	0	20	7	2.0	0.1	varied	0	0	20
5	2.0	0.3	varied	0	0	20	7	2.0	0.3	varied	0	0	20
5	2.0	0.5	varied	0	0	20	7	2.0	0.5	varied	0	0	20
5	2.0	0.7	varied	0	0	20	7	2.0	0.7	varied	0	0	20
5	2.0	0.9	varied	0	0	20	7	2.0	0.9	varied	0	0	20
5	4.0	0.1	varied	0	0	20	7	4.0	0.1	varied	0	0	20
5	4.0	0.3	varied	0	0	20	7	4.0	0.3	varied	0	0	19
5	4.0	0.5	varied	0	0	20	7	4.0	0.5	varied	0	0	20
5	4.0	0.7	varied	0	0	19	7	4.0	0.7	varied	0	0	20
5	4.0	0.9	varied	0	0	20	7	4.0	0.9	varied	0	0	20

p	σ_z	p_0	Threshold	LCA	LCRE.CL	LCRE.NCL	p	σ_z	p_0	Threshold	LCA	LCRE.CL	LCRE.NCL
5	0.5	0.1	varied	0	0	20	7	0.0	0.1	constant	20	20	20
5	0.5	0.3	varied	0	0	20	7	0.0	0.3	constant	20	20	20
5	0.5	0.5	varied	0	0	20	7	0.0	0.5	constant	20	20	20
5	0.5	0.7	varied	0	0	20	7	0.0	0.7	constant	20	20	20
5	0.5	0.9	varied	0	0	20	7	0.0	0.9	constant	20	20	20
7	0.5	0.1	varied	0	0	20	7	2.0	0.1	constant	0	0	20
7	0.5	0.3	varied	0	0	20	7	2.0	0.3	constant	0	0	20
7	0.5	0.5	varied	0	0	20	7	2.0	0.5	constant	0	0	20
7	0.5	0.7	varied	0	0	20	7	2.0	0.7	constant	0	0	20
7	0.5	0.9	varied	0	0	20	7	2.0	0.9	constant	0	0	20
5	1.0	0.1	varied	0	0	20	7	4.0	0.1	constant	0	0	20
5	1.0	0.3	varied	0	0	20	7	4.0	0.3	constant	0	0	20
5	1.0	0.5	varied	0	0	20	7	4.0	0.5	constant	0	0	20
5	1.0	0.7	varied	0	0	20	7	4.0	0.7	constant	0	0	20
5	1.0	0.9	varied	0	0	20	7	4.0	0.9	constant	0	0	20
7	1.0	0.1	varied	0	0	20	5	0.5	0.1	constant	0	0	20
7	1.0	0.3	varied	0	0	20	5	0.5	0.3	constant	0	0	20
7	1.0	0.5	varied	0	0	20	5	0.5	0.5	constant	0	0	20
7	1.0	0.7	varied	0	0	20	5	0.5	0.7	constant	0	0	20
7	1.0	0.9	varied	0	0	20	5	0.5	0.9	constant	0	0	20
5	0.0	0.1	constant	20	20	20	7	0.5	0.1	constant	0	0	20
5	0.0	0.3	constant	20	20	20	7	0.5	0.3	constant	0	0	20
5	0.0	0.5	constant	20	20	20	7	0.5	0.5	constant	0	0	20
5	0.0	0.7	constant	20	20	20	7	0.5	0.7	constant	0	0	20
5	0.0	0.9	constant	20	20	20	7	0.5	0.9	constant	0	0	20
5	2.0	0.1	constant	0	0	20	5	1.0	0.1	constant	0	0	20
5	2.0	0.3	constant	0	0	20	5	1.0	0.3	constant	0	0	20
5	2.0	0.5	constant	0	0	20	5	1.0	0.5	constant	0	0	20
5	2.0	0.7	constant	0	0	20	5	1.0	0.7	constant	0	0	20
5	2.0	0.9	constant	0	0	20	5	1.0	0.9	constant	0	0	20
5	4.0	0.1	constant	0	0	20	7	1.0	0.1	constant	0	0	20
5	4.0	0.3	constant	0	0	20	7	1.0	0.3	constant	0	0	20
5	4.0	0.5	constant	0	0	20	7	1.0	0.5	constant	0	0	19
5	4.0	0.7	constant	0	0	20	7	1.0	0.7	constant	0	0	20
5	4.0	0.9	constant	0	0	20	7	1.0	0.9	constant	0	0	20

Appendix C

Chapter 4 Full Results

C.1 Average correlation and standard deviation for threshold method results

p	δ	σ_z	p_0	LCA ave	LCA sd	LCRE (CL) ave	LCRE (CL) sd	LCRE (NCL) ave	LCRE (NCL) sd
5	0.5	0	0.1	0.9210450	0.0052021831	0.91581859	0.0060136757	0.9082160	0.0157013895
5	0.5	0	0.3	0.9467083	0.0025859088	0.94159459	0.0031424863	0.9394147	0.0057210198
5	0.5	0	0.5	0.9497352	0.0013777766	0.94767679	0.0016554670	0.9453488	0.0024873278
5	0.5	0	0.7	0.9459602	0.0034074725	0.94112468	0.0043508853	0.9358799	0.0077042266
5	0.5	0	0.9	0.9230366	0.0059138534	0.91408775	0.0070580042	0.9101225	0.0122211251
5	1.5	0	0.1	0.9960176	0.0013290765	0.99592476	0.0013713679	0.9959263	0.0013883680
5	1.5	0	0.3	0.9973350	0.0005521753	0.99678397	0.0005383734	0.9964452	0.0005917164
5	1.5	0	0.5	0.9975338	0.0004899443	0.99717180	0.0005381644	0.9973520	0.0005387138
5	1.5	0	0.7	0.9975700	0.0004814859	0.99708176	0.0006149491	0.9969824	0.0006481318
5	1.5	0	0.9	0.9957000	0.0008724407	0.99563041	0.0008559517	0.9955630	0.0008813812
5	0.5	0.5	0.1	0.7566163	0.0538400200	0.72246863	0.0799297638	0.7249823	0.0668977343
5	0.5	0.5	0.3	0.8413618	0.0481192188	0.68802887	0.1111258869	0.7074345	0.1140930287
5	0.5	0.5	0.5	0.8798677	0.0342857038	0.72515660	0.0590663197	0.6415557	0.2339536827

p	δ	σ_z	p_0	LCA ave	LCA sd	LCRE (CL) ave	LCRE (CL) sd	LCRE (NCL) ave	LCRE (NCL) sd
5	0.5	0.5	0.7	0.8205189	0.0681020234	0.71637492	0.1842178707	0.6578963	0.2365733022
5	0.5	0.5	0.9	0.6758140	0.2893874098	0.65810567	0.2452461425	0.5535085	0.2643385479
5	1.5	0.5	0.1	0.9776494	0.0052219156	0.98133773	0.0031414707	0.9883977	0.0022937979
5	1.5	0.5	0.3	0.9869499	0.0015741261	0.98746965	0.0015588774	0.9912890	0.0013702253
5	1.5	0.5	0.5	0.9897980	0.0017201913	0.98393920	0.0092502732	0.9915482	0.0024363052
5	1.5	0.5	0.7	0.9891669	0.0017751841	0.98897818	0.0015613336	0.9919249	0.0008689696
5	1.5	0.5	0.9	0.9803811	0.0031263458	0.98303307	0.0025224911	0.9888303	0.0012450638
5	0.5	1	0.1	0.8193276	0.0684860332	0.82071816	0.0717301064	0.8153657	0.0747185663
5	0.5	1	0.3	0.7988360	0.0799452409	0.78256470	0.0861812974	0.7667475	0.0922565723
5	0.5	1	0.5	0.8423897	0.0515687934	0.83432369	0.0849017508	0.8216569	0.0808455117
5	0.5	1	0.7	0.7907864	0.0716257886	0.73208171	0.0866323349	0.7232787	0.0836748024
5	0.5	1	0.9	0.8410177	0.0637841577	0.83615398	0.0723854802	0.8287194	0.0561723365
5	1.5	1	0.1	0.7015185	0.0283089446	0.81701719	0.0451759447	0.7931197	0.0374961761

p	δ	σ_z	p_0	LCA ave	LCA sd	LCRE (CL) ave	LCRE (CL) sd	LCRE (NCL) ave	LCRE (NCL) sd
5	1.5	1	0.3	0.8832773	0.0093420597	0.72639694	0.0850742809	0.8172183	0.0265313263
5	1.5	1	0.5	0.9121529	0.0089858638	0.78006267	0.0223188334	0.7968163	0.0298025384
5	1.5	1	0.7	0.8857962	0.0108183577	0.65688778	0.1962230594	0.8334154	0.0239191423
5	1.5	1	0.9	0.7079864	0.0150184774	0.56177816	0.1102217453	0.7068368	0.2083294898
5	0.5	2	0.1	0.8049998	0.0813330478	0.81212971	0.0893130510	0.8170143	0.1043762795
5	0.5	2	0.3	0.8169681	0.0607929406	0.81015811	0.0752018970	0.8052526	0.0729047010
5	0.5	2	0.5	0.7827958	0.0850617575	0.77063738	0.0929050083	0.7680095	0.0899696567
5	0.5	2	0.7	0.7461112	0.0596095664	0.73184616	0.0767000276	0.7283999	0.0775576698
5	0.5	2	0.9	0.8155942	0.0810438783	0.81356004	0.0878868569	0.8133957	0.0953779086
5	1.5	2	0.1	0.3616256	0.0259048362	0.06426162	0.0322319950	0.8282943	0.0226006780
5	1.5	2	0.3	0.5074093	0.0079378391	0.12580712	0.0561916605	0.9339908	0.0142506167
5	1.5	2	0.5	0.5503757	0.0159696869	0.19150295	0.0487745683	0.9465253	0.0067123762
5	1.5	2	0.7	0.5130870	0.0140305352	0.15521175	0.0802081234	0.9330809	0.0127884168

p	δ	σ_z	p_0	LCA ave	LCA sd	LCRE (CL) ave	LCRE (CL) sd	LCRE (NCL) ave	LCRE (NCL) sd
5	1.5	2	0.9	0.3702303	0.0102970945	0.08647111	0.0235851889	0.7831928	0.0690963985
5	0.5	4	0.1	0.7584104	0.1385894187	0.75698582	0.1450234123	0.7526559	0.1572601867
5	0.5	4	0.3	0.7685809	0.0938347205	0.76830886	0.1003129320	0.7668522	0.1158458682
5	0.5	4	0.5	0.8034393	0.0786714553	0.80287110	0.0855135582	0.8028647	0.0869718043
5	0.5	4	0.7	0.7937109	0.0827599207	0.79333823	0.0895683461	0.7912820	0.0911707998
5	0.5	4	0.9	0.8156653	0.0562516045	0.81248484	0.0607420783	0.8180433	0.0637758062
5	1.5	4	0.1	0.1780741	0.0303837033	0.08103485	0.0313474845	0.6823125	0.2412388312
5	1.5	4	0.3	0.2452922	0.0185541558	0.05646697	0.0196283896	0.9274955	0.0112654820
5	1.5	4	0.5	0.4427067	0.2967076989	0.28295458	0.4024994866	0.7320144	0.3615542586
5	1.5	4	0.7	0.2393488	0.0177527110	0.05126489	0.0333573689	0.9344888	0.0083970194
5	1.5	4	0.9	0.1899562	0.0187144640	0.10034642	0.0170066787	0.4379723	0.1640390744

C.2 RMSE results

p	delta	sigma.z	p.0	LCA	LCRE.CL	LCRE.NCL	p	delta	sigma.z	p.0	LCA	LCRE.CL	LCRE.NCL
5	0.5	0	0.1	0.10730668	0.13081023	0.12930045	5	1.5	1	0.1	0.27965565	0.26952918	0.26731402
5	0.5	0	0.3	0.13508049	0.19706200	0.18876046	5	1.5	1	0.3	0.22136950	0.34808192	0.30786441
5	0.5	0	0.5	0.14467158	0.16533182	0.16805354	5	1.5	1	0.5	0.20702649	0.32333085	0.30684556
5	0.5	0	0.7	0.13572640	0.20350745	0.18944506	5	1.5	1	0.7	0.21631999	0.36241587	0.32121563
5	0.5	0	0.9	0.10707998	0.12937579	0.13264158	5	1.5	1	0.9	0.27841276	0.33784804	0.26115239
5	1.5	0	0.1	0.02609860	0.02642228	0.02643204	5	0.5	2	0.1	0.32563326	0.31864393	0.27392419
5	1.5	0	0.3	0.03308827	0.07610076	0.07224225	5	0.5	2	0.3	0.30488188	0.29757291	0.29030030
5	1.5	0	0.5	0.03481727	0.04803807	0.04290975	5	0.5	2	0.5	0.30913955	0.30108607	0.29744778
5	1.5	0	0.7	0.03243031	0.07337678	0.06830176	5	0.5	2	0.7	0.30630063	0.30061650	0.28580554
5	1.5	0	0.9	0.02710538	0.02730255	0.02739059	5	0.5	2	0.9	0.31285019	0.30696189	0.26070463
5	0.5	0.5	0.1	0.26082702	0.25768556	0.25826198	5	1.5	2	0.1	0.49179696	0.56888809	0.22376954
5	0.5	0.5	0.3	0.20242846	0.24867932	0.23774761	5	1.5	2	0.3	0.46282322	0.54207907	0.16316764
5	0.5	0.5	0.5	0.18633304	0.24889735	0.25576147	5	1.5	2	0.5	0.45019494	0.52476892	0.15715663
5	0.5	0.5	0.7	0.22519690	0.26064805	0.26262157	5	1.5	2	0.7	0.45948407	0.53222006	0.17698647
5	0.5	0.5	0.9	0.22984598	0.22266777	0.23888659	5	1.5	2	0.9	0.48974718	0.55721830	0.26029850
5	1.5	0.5	0.1	0.06321602	0.05889048	0.04709133	5	0.5	4	0.1	0.34923426	0.34606314	0.29462755
5	1.5	0.5	0.3	0.07253349	0.08025064	0.07183332	5	0.5	4	0.3	0.33027021	0.32692657	0.30785688
5	1.5	0.5	0.5	0.07327892	0.13552756	0.14338971	5	0.5	4	0.5	0.32495605	0.32077305	0.31077112
5	1.5	0.5	0.7	0.07079419	0.07902630	0.07068208	5	0.5	4	0.7	0.31717516	0.31297815	0.29475622
5	1.5	0.5	0.9	0.06291766	0.05819404	0.04729235	5	0.5	4	0.9	0.33362023	0.33047051	0.28666790
5	0.5	1	0.1	0.28837864	0.27841756	0.25103724	5	1.5	4	0.1	0.59509803	0.61500072	0.25801961
5	0.5	1	0.3	0.29252543	0.28447443	0.27847931	5	1.5	4	0.3	0.58357771	0.62711803	0.17599866
5	0.5	1	0.5	0.27199083	0.27097124	0.26644069	5	1.5	4	0.5	0.50229595	0.53006827	0.24129245
5	0.5	1	0.7	0.27745443	0.27634065	0.26154798	5	1.5	4	0.7	0.58346451	0.62804594	0.17980566
5	0.5	1	0.9	0.24915428	0.23975969	0.21634978	5	1.5	4	0.9	0.59241597	0.60718005	0.38831727

C.3 Binding Accuracy

p	δ	σ_z	p_0	Binding Number	LCA	LCRE.CL	LCRE.NCL	MGMM
5	0.5	0.0	0.1	300	309.71	434.85	397.18	348.79
5	0.5	0.0	0.3	900	895.10	1165.88	1079.67	900.19
5	0.5	0.0	0.5	1500	1497.95	1498.88	1493.92	1511.79
5	0.5	0.0	0.7	2100	2097.18	1793.99	1927.74	2088.47
5	0.5	0.0	0.9	2700	2684.14	2532.52	2542.23	2637.94
5	0.5	2.0	0.1	300	1556.29	1522.75	1535.71	1401.36
5	0.5	2.0	0.3	900	1426.68	1458.09	1470.13	1389.50
5	0.5	2.0	0.5	1500	1494.18	1490.03	1486.15	1453.79
5	0.5	2.0	0.7	2100	1430.41	1449.99	1466.83	1573.20
5	0.5	2.0	0.9	2700	1419.34	1439.13	1457.34	1537.34
5	0.5	4.0	0.1	300	1502.94	1502.97	1493.04	1428.54
5	0.5	4.0	0.3	900	1516.44	1516.54	1503.92	1494.08
5	0.5	4.0	0.5	1500	1491.16	1489.78	1486.16	1312.97
5	0.5	4.0	0.7	2100	1515.09	1513.23	1520.41	1514.72
5	0.5	4.0	0.9	2700	1507.77	1509.48	1530.49	1546.82
5	1.5	0.0	0.1	300	299.80	299.44	299.51	300.17
5	1.5	0.0	0.3	900	899.90	880.71	863.79	900.01
5	1.5	0.0	0.5	1500	1500.45	1497.25	1500.71	1499.93
5	1.5	0.0	0.7	2100	2100.78	2189.84	2177.75	2100.07
5	1.5	0.0	0.9	2700	2699.27	2699.62	2699.65	2699.94
5	1.5	2.0	0.1	300	1039.25	1496.20	764.14	309.41
5	1.5	2.0	0.3	900	1275.44	1449.87	1062.59	897.34
5	1.5	2.0	0.5	1500	1494.17	1490.27	1496.33	1503.01
5	1.5	2.0	0.7	2100	1717.37	1556.45	1892.00	2100.89
5	1.5	2.0	0.9	2700	1954.89	1514.94	2120.98	2697.18

p	δ	σ_z	p_0	Binding Number	LCA	LCRE.CL	LCRE.NCL	MGMM
5	1.5	4.0	0.1	300	1327.60	1504.26	756.44	300.89
5	1.5	4.0	0.3	900	1383.37	1491.61	1003.60	902.58
5	1.5	4.0	0.5	1500	1496.66	1513.27	1482.88	1547.73
5	1.5	4.0	0.7	2100	1625.64	1477.80	1996.35	2103.64
5	1.5	4.0	0.9	2700	1748.51	1575.28	1965.06	2697.11
5	0.5	0.5	0.1	300	1053.36	1139.15	1144.17	1313.72
5	0.5	0.5	0.3	900	1235.13	1338.15	1338.68	1101.12
5	0.5	0.5	0.5	1500	1482.37	1518.06	1514.47	1437.10
5	0.5	0.5	0.7	2100	1765.16	1653.38	1633.29	1778.87
5	0.5	0.5	0.9	2700	1936.56	1842.33	1819.07	1807.84
5	0.5	1.0	0.1	300	1760.41	1725.59	1693.86	1479.32
5	0.5	1.0	0.3	900	1358.00	1365.34	1374.47	1422.96
5	0.5	1.0	0.5	1500	1512.67	1507.78	1504.56	1410.07
5	0.5	1.0	0.7	2100	1633.89	1630.74	1619.37	1668.13
5	0.5	1.0	0.9	2700	1756.54	1753.01	1749.04	1539.50
5	1.5	0.5	0.1	300	324.27	297.36	300.55	300.81
5	1.5	0.5	0.3	900	914.79	879.34	870.97	900.40
5	1.5	0.5	0.5	1500	1500.26	1501.55	1492.51	1499.43
5	1.5	0.5	0.7	2100	2087.10	2129.16	2139.36	2100.31
5	1.5	0.5	0.9	2700	2675.35	2702.45	2701.02	2700.42
5	1.5	1.0	0.1	300	646.87	870.52	864.66	304.29
5	1.5	1.0	0.3	900	1028.80	1329.28	1334.18	898.28
5	1.5	1.0	0.5	1500	1501.61	1495.46	1496.93	1497.84
5	1.5	1.0	0.7	2100	1977.83	1637.18	1650.59	2101.37
5	1.5	1.0	0.9	2700	2355.26	2001.61	2165.77	2699.79

Bibliography

Agresti, Alan and Joseph B Lang (1993). "A proportional odds model with subject-specific effects for repeated ordered categorical responses". In: *Biometrika* 80.3, pp. 527–534.

Allhoff, Manuel et al. (2014). "Detecting differential peaks in ChIP-seq signals with ODIN". In: *Bioinformatics* 30.24, pp. 3467–3475.

Anders, Simon and Wolfgang Huber (2010). "Differential expression analysis for sequence count data". In: *Genome biology* 11.10, R106.

Bao, Yanchun et al. (2014). "Joint modeling of ChIP-seq data via a Markov random field model". In: *Biostatistics* 15.2, p. 296. DOI: 10.1093/biostatistics/kxt047. eprint: /oup/backfile/content_public/journal/biostatistics/15/2/10.1093/biostatistics/kxt047/2/kxt047.pdf. URL: +http://dx.doi.org/10.1093/biostatistics/kxt047.

Beath, Ken J (2008). "RandomLCA. R package. Available at http://cran.r-project.org/package=
In: *CRAN*.

Beath, Ken J and Gillian Z Heller (2009). "Latent trajectory modelling of multivariate binary data". In: *Statistical Modelling: An International Journal* 9.3, pp. 199–213. DOI: 10.1177/1471082x0800900302.

Berger, Shelley L et al. (2009). "An operational definition of epigenetics". In: *Genes & development* 23.7, pp. 781–783.

Buenrostro, Jason D et al. (2013). "Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, DNA-binding proteins and nucleosome position". In: *Nature methods* 10.12, pp. 1213–1218.

Cantarel, Brandi L et al. (2014). "BAYSIC: a Bayesian method for combining sets of genome variants with improved specificity and sensitivity". In: *BMC Bioinformatics* 15.1, p. 104. DOI: 10.1186/1471-2105-15-104.

Chahwan, Richard, Sandeep N Wontakal, and Sergio Roa (2011). "The multidimensional nature of epigenetic information and its role in disease". In: *Discovery medicine* 11.58, pp. 233–243.

Chen, Feng et al. (2007). "Assessing Performance of Orthology Detection Strategies Applied to Eukaryotic Genomes". In: *PLoS ONE* 2.4. Ed. by Cecile Fairhead, e383. DOI: 10.1371/journal.pone.0000383.

Consortium, ENCODE Project (2012). "An integrated encyclopedia of DNA elements in the human genome." In: *Nature* 489 (7414), pp. 57–74. ISSN: 1476-4687. DOI: 10.1038/nature11247.

Elsik, Christine G et al. (2007). "Creating a honey bee consensus gene set". In: *Genome Biology* 8.1, R13. DOI: 10.1186/gb-2007-8-1-r13.

Even-Faitelson, Liron et al. (2016). "Coming to terms with chromatin structure". In: *Chromosoma* 125.1, pp. 95–110. ISSN: 1432-0886. DOI: 10.1007/s00412-015-0534-9. URL: http://dx.doi.org/10.1007/s00412-015-0534-9.

Giresi, Paul G et al. (2007). "FAIRE (Formaldehyde-Assisted Isolation of Regulatory Elements) isolates active regulatory elements from human chromatin". In: *Genome research* 17.6, pp. 877–885.

Harmanci, Arif, Joel Rozowsky, and Mark Gerstein (2014). "MUSIC: identification of enriched regions in ChIP-Seq experiments using a mappability-corrected multi-scale signal processing framework." In: *Genome biology* 15 (10), p. 474. ISSN: 1474-760X. DOI: 10.1186/s13059-014-0474-3.

Heinz, Sven et al. (2010). "Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and B cell identities". In: *Molecular cell* 38.4, pp. 576–589.

III, Robert J Sims and Danny Reinberg (2009). "Processing the H3K36me3 signature". In: *Nature Genetics* 41.3, pp. 270–271. DOI: 10.1038/ng0309-270.

Jeong, Hae Min et al. (2016). "Efficiency of methylated DNA immunoprecipitation bisulphite sequencing for whole-genome DNA methylation analysis". In: *Epigenomics* 8.8, pp. 1061–1077.

Johnson, David S et al. (2007). "Genome-wide mapping of in vivo protein-DNA interactions". In: *Science* 316.5830, pp. 1497–1502.

Ku, Chee Seng et al. (2011). "Studying the epigenome using next generation sequencing". In: *Journal of Medical Genetics* 48.11, pp. 721–730. ISSN: 0022-2593. DOI: 10.1136/jmedgenet-2011-100242. eprint: http://jmg.bmj.com/content/48/11/721.full.pdf. URL: http://jmg.bmj.com/content/48/11/721.

Lan, Xun et al. (2010). "W-ChIPeaks: a comprehensive web application tool for processing ChIP-chip and ChIP-seq data". In: *Bioinformatics* 27.3, pp. 428–430.

Lex, Alexander et al. (2014). "UpSet: Visualization of Intersecting Sets". In: *IEEE Transactions on Visualization and Computer Graphics* 20.12, pp. 1983–1992. DOI: 10.1109/tvcg.2014.2346248.

Li, Heng and Richard Durbin (2009). "Fast and accurate short read alignment with Burrows-Wheeler transform." In: *Bioinformatics (Oxford, England)* 25 (14), pp. 1754–1760. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btp324.

Linda M. Collins, Stephanie T. Lanza (2010). *Latent Class Analysis*. JOHN WILEY & SONS INC. 285 pp. ISBN: 0470228393. URL: https://www.ebook.de/de/product/9338185/linda_m_collins_stephanie_t_lanza_latent_class_analysis.html.

Liu, Bin et al. (2013). "QChIPat: a quantitative method to identify distinct binding patterns for two biological ChIP-seq samples in different experimental conditions". In: *BMC Genomics* 14.8, p. 1. ISSN: 1471-2164. DOI: 10.1186/1471-2164-14-S8-S3. URL: http://dx.doi.org/10.1186/1471-2164-14-S8-S3.

McLachlan, Geoffrey J (1987). "On bootstrapping the likelihood ratio test stastistic for the number of components in a normal mixture". In: *Applied statistics*, pp. 318–324.

Meyer, Clifford A and X Shirley Liu (2014). "Identifying and mitigating bias in next-generation sequencing methods for chromatin biology". In: *Nature Reviews Genetics* 15.11, pp. 709–721.

Narlikar, Leelavati and Raja Jothi (2012). "ChIP-Seq data analysis: identification of Protein–DNA binding sites with SISSRs peak-finder". In: *Next Generation Microarray Bioinformatics: Methods and Protocols*, pp. 305–322.

Phillips, Theresa and K Shaw (2008). "Chromatin remodeling in eukaryotes". In: *Nature Education* 1.1, p. 209.

Portela, Anna and Manel Esteller (2010). "Epigenetic modifications and human disease". In: *Nature biotechnology* 28.10, pp. 1057–1068.

Qu, Y, M Tan, and M H Kutner (1996). "Random effects models in latent class analysis for evaluating accuracy of diagnostic tests." In: *Biometrics* 52 (3), pp. 797–810. ISSN: 0006-341X.

Ranciati, Saverio, Cinzia Viroli, and Ernst Wit (2015). "Spatio-temporal model for multiple ChIP-seq experiments". In: *Statistical applications in genetics and molecular biology* 14.2, pp. 211–219.

Robinson, Mark D, Davis J McCarthy, and Gordon K Smyth (2010). "edgeR: a Bioconductor package for differential expression analysis of digital gene expression data". In: *Bioinformatics* 26.1, pp. 139–140.

Rozowsky, Joel et al. (2009). "PeakSeq enables systematic scoring of ChIP-seq experiments relative to controls". In: *Nature biotechnology* 27.1, pp. 66–75.

Schuster, Stephan C (2008). "Next-generation sequencing transforms today's biology". In: *Nature methods* 5.1, p. 16.

Schwarz, Gideon (1978). "Estimating the Dimension of a Model". In: *The Annals of Statistics*.

Shen, Li et al. (2013). "diffReps: detecting differential chromatin modification sites from ChIP-seq data with biological replicates". In: *PloS one* 8.6, e65598.

Song, Qiang and Andrew D Smith (2011). "Identifying dispersed epigenomic domains from ChIP-Seq data". In: *Bioinformatics* 27.6, pp. 870–871.

Steinhauser, Sebastian et al. (2016). "A comprehensive comparison of tools for differential ChIP-seq analysis". In: *Briefings in bioinformatics* 17.6, pp. 953–966.

Stueve, Theresa Ryan et al. (2016). "The importance of detailed epigenomic profiling of different cell types within organs". In:

Thomas, Reuben et al. (2016). "Features that define the best ChIP-seq peak calling algorithms". In: *Briefings in bioinformatics* 18.3, pp. 441–450.

Valouev, Anton et al. (2008). "Genome-wide analysis of transcription factor binding sites based on ChIP-Seq data". In: *Nature methods* 5.9, pp. 829–834.

Venkatesh, Swaminathan and Jerry L Workman (2015). "Histone exchange, chromatin structure and the regulation of transcription". In: *Nature reviews Molecular cell biology* 16.3, pp. 178–189.

Wang, Dan et al. (2012). "IMA: an R package for high-throughput analysis of Illumina's 450K Infinium methylation data". In: *Bioinformatics* 28.5, pp. 729–730.

Wang, Zhong, Mark Gerstein, and Michael Snyder (2009). "RNA-Seq: a revolutionary tool for transcriptomics". In: *Nature reviews genetics* 10.1, pp. 57–63.

Xing, Haipeng et al. (2012). "Genome-Wide Localization of Protein-DNA Binding and Histone Modification by a Bayesian Change-Point Method with ChIP-seq Data". In: *PLoS Computational Biology* 8.7. Ed. by Ilya Ioshikhes, e1002613. DOI: 10.1371/journal.pcbi.1002613.

Xu, Shiliyang et al. (2014). "Spatial Clustering for Identification of ChIP-Enriched Regions (SICER) to Map Regions of Histone Methylation Patterns in Embryonic Stem Cells". In: *Stem Cell Transcriptional Networks*. DOI: 10.1007/978-1-4939-0512-6_5. URL: http://dx.doi.org/10.1007/978-1-4939-0512-6_5.

Zang, Chongzhi et al. (2009). "A clustering approach for identification of enriched domains from histone modification ChIP-Seq data". In: *Bioinformatics* 25.15, pp. 1952–1958.

Zerbino, Daniel R et al. (2018). "Ensembl 2018." In: *Nucleic acids research* 46 (D1), pp. D754–D761. ISSN: 1362-4962. DOI: 10.1093/nar/gkx1098.

Zhang, Yan et al. (2011). "QDMR: a quantitative method for identification of differentially methylated regions by entropy". In: *Nucleic Acids Research* 39.9, e58. DOI: 10.1093/nar/gkr053. eprint: /oup/backfile/content_public/journal/nar/39/9/10.1093_nar_gkr053/2/gkr053.pdf. URL: +http://dx.doi.org/10.1093/nar/gkr053.

Zhang, Yong et al. (2008). "Model-based analysis of ChIP-Seq (MACS)". In: *Genome biology* 9.9, R137.