©Copyright 2016

Jean Morrison

Flexible strategies for association analysis with genomic phenotypes

Jean Morrison

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

Doctor of Philosophy

University of Washington

2016

Reading Committee:

Noah Simon, Chair

Daniela Witten

Bruce Weir

Program Authorized to Offer Degree:
Biostatistics

University of Washington

Abstract

Flexible strategies for association analysis with genomic phenotypes

Jean Morrison

Chair of the Supervisory Committee:
Dr. Noah Simon
Biostatistics

Advances in high throughput sequencing have lead to a proliferation of genomic assays that exploit sequencing to measure epigenetic traits with very high resolution, sometimes at every base-pair. Studying the relationship between these molecular traits or *genomic phenotypes* and cell or organismal level traits can lead to better understanding of genetic regulation and the biological processes underlying variation. The most basic approach to this task is to search for associations between genomic phenotypes and other traits such as experimental condition or disease status. This undertaking can be challenging. Functional genomic elements, such as promoters, exons, and transcription factor binding sites, are the unit of interest but annotations of the boundaries of these elements are far from complete. We therefore face the two-pronged problem of finding associations and identifying the boundaries of the underlying signal. We make two novel proposals that accomplish these tasks simultaneously, resulting in data adaptive region boundaries.

In our first proposal, *joint adaptive differential estimation* (JADE), we approach the problem through estimation of the mean genomic phenotype, or *profile*, at each trait level. We use penalized regression to impose structure on these estimates and recover regions of association. JADE is powerful and provides a useful descriptive summary of the results by clustering profiles within associated regions.

In our second proposal, *flexible robust excursion test* (FRET), we employ results for scanning statistics to construct a method that searches the genome for areas with non-zero regression coefficients. While less powerful than JADE in some circumstances, FRET is more computationally efficient, more robust to outliers, and provides control of the region-wise false discovery rate.

We compare FRET and several alternative strategies applied to the problem of identifying genomic regions in which chromatin accessibility differs between drug resistant and susceptible cancer cell lines. Our results suggest that FRET is more powerful than the alternatives and that methods that rely on assumptions about the distribution of the data are poorly calibrated for this problem.

TABLE OF CONTENTS

Page

List of Figures . . . iii

List of Tables . . . v

Chapter 1: Introduction . . . 1

1.1 Overview of sequence-based assays . . . 1

1.2 Statement of the problem . . . 5

1.3 Existing approaches: Fixed-region and adaptive methods . . . 5

1.4 Imposing structure . . . 9

1.5 Point-wise tests and false discovery rate control . . . 11

Chapter 2: Differential region detection via profile estimation . . . 13

2.1 Problem formulation . . . 14

2.2 Penalties to induce structure . . . 17

2.3 Joint smoothing and comparison with JADE . . . 19

2.4 Solving the JADE optimization problem . . . 20

2.5 Simulations . . . 24

2.6 Application to methylation data . . . 31

2.7 Discussion . . . 40

Chapter 3: An excursion procedure for genomic phenotypes . . . 42

3.1 Overview . . . 42

3.2 Background: Controlling the region-wise false discovery rate for excursion procedures . . . 46

3.3 Background: Robust test statistics . . . 53

3.4 FRET: The proposal . . . 61

3.5 Simulations . . . 67

3.6 Discussion 78

Chapter 4: Analysis of DNase 1 sensitivity in drug susceptible and resistant cancer cells 79

4.1 Description of the problem 79

4.2 Methods 80

4.3 Results 82

4.4 Summary 92

Chapter 5: Conclusions and future directions 94

Bibliography 97

Appendix A: Technical details and additional simulation results for Chapter 2 105

A.1 Trend filtering 105

A.2 Details of algorithm 1 106

A.3 Cross-validation of γ 107

A.4 Details of figures 2.3, 2.4, and 2.5 in Section 2.5 108

A.5 Additional simulation results 109

A.6 Read tiling in binomial simulations 111

Appendix B: Additional Discussion for Chapter 3 116

B.1 Smooth then test or test then smooth? 116

Appendix C: Additional details for Chapter 4 121

C.1 Selection of σ_0 and z^* 121

C.2 Hotspot calling and peak master list construction 122

C.3 Additional fixed-regions tests 123

LIST OF FIGURES

Figure Number Page

1.1 Examples of two types of sequence-based genomic data. 4

2.1 An illustration of the toy example described in Section 2.1.1 16
2.2 Profiles for simulated data 25
2.3 Results of simulations for the normal auto-regressive model 28
2.4 Results of simulations for the normal random effects model 29
2.5 Results of simulations for the binomial random effects model 32
2.6 Example results from analysis of methylation data 35
2.7 Methylation patterns in differential regions 37

3.1 Illustration of an excursion procedure 44
3.2 Relationship between the threshold, z, and $\lambda(z)$ 49
3.3 Motivation for merging 51
3.4 Squared error, L_1, and Huber Loss functions 56
3.5 An example in which a variable threshold is helpful 64
3.6 $\lambda(z)$ on intervals with different noise levels 66
3.7 An example of a profile for simulated data 70
3.8 Schematic for constructing profiles 72
3.9 Simulation results for a binary trait 74
3.10 Simulation results for a continuous trait 77

4.1 A region identified as differential by multiple methods 83
4.2 Quantile-quantile plots for fixed-region tests 84
4.3 A region identified as differential by DESeq2 and Wellington-Bootstrap only 86
4.4 Number of discoveries made by each method 87
4.5 Summary statistics for top ranked peaks for all methods 90
4.6 Summary statistics for all peaks by treatment group 91
4.7 A region identified as differential by FRET only 92

A.1 Study of the effect of sample size on simulation results 110
A.2 Region-wise results for the normal auto-regressive setting 112
A.3 Region-wise results for the normal random effects setting 113
A.4 Region-wise results for the binomial random effects setting 114
A.5 Read tiling for binomial simulations 115

B.1 Performance of ST and TS testing methods for normal data 118
B.2 Performance of ST and TS testing methods for Poisson data 120

C.1 Quantile-quantile plots for additional fixed-region tests 124

LIST OF TABLES

Table Number Page

2.1 Overlap between detected DMRs and genomic features 39

2.2 Overlap between gain-of-methylation and loss-of-methylation DMR sub-regions and genomic features 41

3.1 Numerical experiments comparing test statistics for over-dispersed data . . . 59

4.1 Rank correlation between analysis methods 88

ACKNOWLEDGMENTS

I am extremely lucky to have been able to spend the past five years as a graduate student learning and exploring new ideas in a field I am passionate about. There are many individuals who have made it possible for me to complete this journey and to whom I am deeply grateful.

I would like to thank my advisor, Noah Simon, for being always full of new, creative ideas (many more than I have been able to follow up on), encouraging words, and new perspectives. Noah has provided excellent mentorship, guidance, and been a great ally and friend.

Daniela Witten, in addition to serving on my committee, has provided invaluable feedback on several of these projects and made me a much better communicator. I am incredibly grateful for her patience, insights, and the time she has taken with this work.

I would like to thank Bruce Weir for mentorship, encouragement, and pushing me to achieve things I thought were out of reach.

Over the past five years I have found a wonderful community in the department of Biostatistics and in Seattle. I am grateful to my cohort-mates and fellow students, especially Jenn, Jason, Josh, Shizhe, and Ashley for making this treck with me and for being funny and supportive along the way. I would like to thank the beautiful people with whom I share my home, Jenna, Angie, Kamaria, and Tobi, for their love and support. I am also grateful to Sunday, the cat, for steadfast company on long nights, and for being generally friendly and gregarious.

Finally, I'd like to thank my family. My parents, Dan and Margaret, always encouraged my curiosity and supported my passions. My father has been a good ear and source of advice over the past five years. I know my mother would have liked to see me through this journey. I have found myself drawing on the values and love she instilled in me at many points along

the way and am grateful for the parts of her I carry with me. My sister, Ruth, has always been available for non-judgemental support and kindness (and also helped me with a few coding problems at various points). Klara, who is family in all but the biological sense, has carried me through difficult times, laughed with me in happy ones, and taken me on many adventures.

DEDICATION

This work is dedicated to Ruth and Klara.

Chapter 1
INTRODUCTION

Recent advances in high throughput sequencing technologies have facilitated the development of numerous sequencing-based genomic assays. These assays are used to measure local features of the chemical and molecular environment of DNA, such as DNA methylation, histone modifications, and RNA expression levels. These features, which we will refer to as *genomic phenotypes*, reflect the regulatory state of DNA and therefore comprise a critical component necessary to understanding the molecular foundations of cell and organismal level variation.

Genomic phenotypes are dynamic — they can change over the course of cell development and under different environmental conditions. Therefore, the scientific interest in these features is often in identifying associations between a genomic phenotype and a biological or experimental variable. In this paper we discuss strategies for accomplishing this goal and make two novel proposals suitable under different conditions. We also present two examples of real biological problems that can be solved using these tools and discuss the advantages and drawbacks of other alternatives.

1.1 Overview of sequence-based assays

Each type of sequence-based genomic assay has a set of unique technical features that must be understood and considered before an analysis is performed. However, there are several important commonalities that allow many methods to be (thoughtfully) adapted across applications.

Genomic assays share a common general structure. The first step typically involves a clever experimental protocol that allows the researcher to isolate, tag, or selectively amplify

DNA (or RNA) with desired features. Resulting fragments are then sequenced and mapped to a reference genome. The number of sequences mapping to each base-pair or a transformation of this count is a quantitative measure of the feature of interest. Here we briefly describe four of the most common sequence-based assays:

Chromatin immunoprecipitation sequencing (ChIP-seq): This assay is used to detect binding sites for protein-DNA interactions. Bound proteins are first crosslinked to the DNA. DNA is then fragmented and fragments with bound proteins are isolated and amplified. The amplified DNA fragments are sequenced and mapped to a reference. The number of fragments sequenced at each position is counted. Protein binding sites are indicated by peaks in these counts.

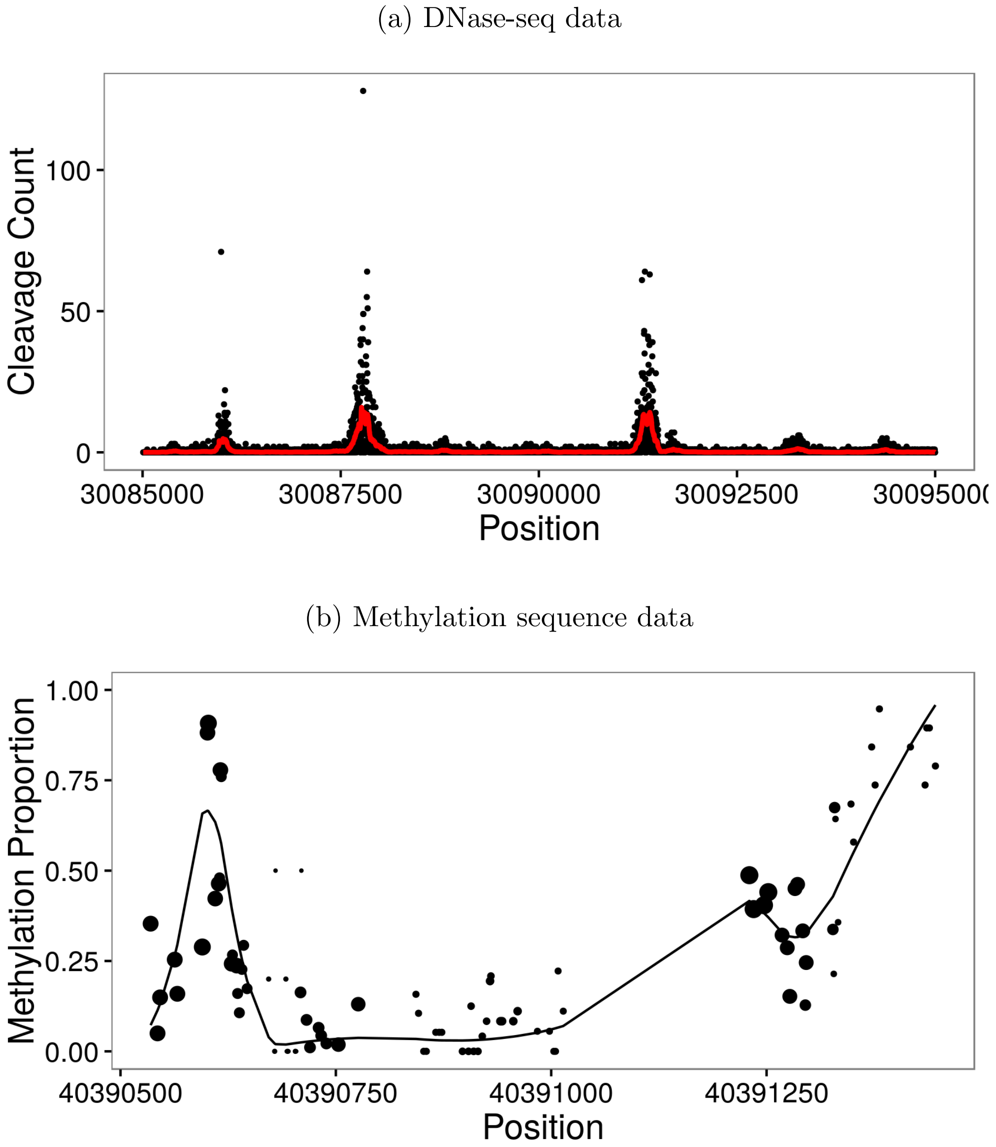
DNase-seq: This assay is used to measure chromatin accessibility. Inactive regions of DNA are packaged into tight chromatin conformations while active regions are more open and accessible. In a DNase-seq assay, DNA is first digested with the DNase 1 enzyme which preferentially cleaves DNA in open chromatin. The resulting fragments are sequenced and mapped. In this assay, only sequence endpoints are counted as these mark DNase 1 cleavage sites. The number of sequence endpoints observed at a particular base-pair can be taken as a quantitative measure of DNA accessibility and is described as the *DNase 1 sensitivity* at that location. Figure 1.1a shows an example of DNase-seq data for a small cell lung cancer cell line in a 10 killobase pair (kb) segment of chromosome 22. These data are discussed in greater detail in Chapter 4. Like ChIP-seq data, DNase-seq data often have dramatic peaks corresponding to functional elements.

RNA-seq: In this assay, RNA fragments are amplified and sequenced. The number of fragments aligning to a particular position indicates the expression level at that position. In these studies, the focus is often restricted to known exons.

Methylation bisulfite sequencing: DNA methylation is a chemical modification that can affect sequences of the form 5′-C-G-3′ (CpGs). In a bisulfite sequencing experiment, non-methylated cytosine residues are converted into uracil residues. Converted DNA is fragmented, sequenced, and mapped. At each CpG, the number of methylated and unmethylated fragments are counted to give an estimate of the proportion of cells with DNA that is methylated at that position. Figure 1.1b shows an example of bisulfite sequencing data for a skeletal muscle myotube in a 1kb segment of chromosome 22. These data are discussed in greater detail in Section 2.6.

The data from this assay differs slightly from the other examples in that the quantity of interest is a proportion rather than a count and measurements are only made at CpG sites rather than at every base-pair. Another key difference is that making no reads at a CpG should be interpreted as missing data. This is in contrast to the other assays for which an observation of 0 sequences at a particular position is indicative of a very low trait value (e.g. very low expression or accessibility).

Both of the examples in Figure 1.1 show data that possesses spatial structure, in that measurements at nearby locations tend to be similar. This type of structure is induced by the underlying biology of these data types — patterns in genomic phenotypes reflect the activity and regulation of functional elements spanning many base-pairs, such as promoters or exons. Associations between a genomic phenotype and a trait arise due to differences in the activity of these elements across levels of the trait. We can expect that, using a sequence-based assay, we will make many measurements of the genomic phenotype within each associated region. This is a potentially great advantage over technologies that make only one or two measurements in each functional element (such as array technologies) but also creates an analytical challenge: The boundaries of functional elements are usually unknown or uncertain and must be estimated from the data.


(a) DNase-seq data

(b) Methylation sequence data

Figure 1.1: Examples of two types of sequence-based genomic data: Figure 1.1a shows the number of DNase 1 cleavages from a DNase-seq experiment in a 10kb segment of chromosome 22. The red line shows a 50 base-pair moving average. Figure 1.1b shows the methylation proportion measured by bisulfite sequencing in a 1kb segment of chromosome 22. Point size is proportional to the number of reads at each CpG site. The black line shows a piecewise quadratic smooth fit of the data.

1.2 Statement of the problem

Here we formally describe the problem of identifying associations with genomic phenotypes. For a single sample, we observe measurements $\mathbf{y} = (y_1, \dots, y_p)^\top \in \mathbb{R}^p$ at genomic coordinates $s_1 \leq \cdots \leq s_p$. In the case of DNase-seq, y_j is the number of DNase 1 cleavages at base-pair s_j, while for methylation sequence data, y_j is the proportion of reads observed to be methylated. For simplicity, we assume that $s_j \in \{1, \dots, p\}$.

We also observe a trait $\mathbf{x} \in \mathbb{R}^q$. In many cases q will equal 1, but allowing an arbitrary dimension will allow us to accommodate categorical traits such as cell type. We assume that we have n independent observations of $(\mathbf{x}, \mathbf{y})$, denoted $(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_n, \mathbf{y}_n)$.We model

$$E[y_j|\mathbf{x}] = \alpha_j + \boldsymbol{\beta}_j^\top \mathbf{x} \tag{1.1}$$

and say that y_j is associated with $\mathbf{x}$ if $\boldsymbol{\beta}_j \neq 0$. We seek to identify *differential regions*: subsets of contiguous positions, $\mathcal{S} \subseteq \{1, \dots, p\}$ such that $\boldsymbol{\beta}_j \neq 0$ for all $j \in \mathcal{S}$. Here we slightly abuse notation and let 0 indicate the 0 vector in $\mathbb{R}^q$. The model in (1.1) assumes a linear relationship between the genomic phenotype and the trait which may not always be a good fit. In these cases, it may be possible to apply a transformation to the gnomic phenotype, the trait, or both so that the linear model is more appropriate.

1.3 Existing approaches: Fixed-region and adaptive methods

Numerous strategies have been applied to the problem of identifying regions in which a genomic phenotype is associated with a trait. Often, methods are developed for a specific technology and sometimes implemented only for a particular application, making the literature on this topic vast and occasionally redundant. Here, we seek to provide an overview of different types of approaches that have been considered, with particular attention paid to a handful of representative methods.

There are two broad classes of approaches to identifying differential regions: 1) Two stage approaches in which region boundaries are fixed and then each region is tested individually;

and 2) Joint methods that simultaneously test for association and identify region boundaries. Our proposals fall into the second category, though both strategies have advantages in certain situations.

In the first stage of a two stage approach, region boundaries can be determined in one of three ways: 1) The genome may be segmented arbitrarily; 2) Regions can be defined using existing annotations of functional regions; or 3) Regions can be learned from the data. Taking a two stage approach is natural when one of these strategies is likely to correctly recover the boundaries of all or most signal regions. This might be the case in a hypothesis-driven study in which the the scientific question is specific and relates to existing annotation sets (e.g. testing for differential expression at several well characterized exons).

Pre-determining boundaries can also be a practical approach for certain data types for which the boundaries of relevant regions are easy to determine. Madrigal and Krajewski [2012] suggest that this may be the case for ChIP-seq data. These data possess relatively clear peaks in sequence counts that indicate protein binding sites. Many ChIP-seq analysis strategies first identify the peaks in each sample and then test within each peak. DNase-seq data is often also analyzed according to this strategy, though peaks may be less clearly defined for DNase-seq data than for ChIP-seq data [Madrigal and Krajewski, 2012].

When two-stage strategies work well, the first stage can massively reduce the portion of the genome that is considered for testing, effectively screening out a large number of unpromising regions. This pre-screening can boost the power as well as reduce computational time.

There are, however, several drawbacks to using two stage approaches. If the boundary determination procedure performs poorly, important regions may be left out, signals may be split, and regions with signal in opposite directions may be combined. All of these events cause a reduction in power. Using annotation based region definitions may limit the potential of the study to discover new functional elements. Finally, learning region boundaries from the data could result in biased false discovery rate estimates if this pre-processing is not accounted for in the significance estimation procedure.

Single stage approaches are able to use more information in determining region boundaries and are inherently more adaptive than two stage methods. These methods can facilitate discoveries in areas of the genome with uncertain or unreliable annotations. They are especially useful for phenotypes for which functional elements are difficult or impossible to identify from a single sample such as DNA methylation sequencing data. They also allow the entire analysis to be done within a single framework and reduce the amount of pre-processing that must be done.

1.3.1 Fixed region testing methods

The boundary determination method used in the first stage of fixed-region methods is usually chosen based on the application. For example, numerous peak calling utilities have been developed for ChIP-seq and DNase-seq experiments. Annotations are more likely to be used for an RNA-seq experiments, while arbitrary binning may be the only choice for DNA methylation sequencing data.

Once region boundaries are fixed, testing for an association within each region can become a relatively simple problem. DESeq2 [Love et al., 2014] and edgeR [Robinson et al., 2009, McCarthy et al., 2012] are both fixed window testing methods developed for identifying differential expression with RNA-seq count data. These methods sum sequence counts within each region and test each region for a difference in expected total count across levels of a categorical variable. Both methods model the total number of sequences in the region in a single sample as arising from a negative binomial distribution, and take an empirical-Bayes approach to estimating the negative binomial dispersion parameter, sharing information across regions.

Shim and Stephens [2015] propose WaveQTL, a two-step testing procedure for identifying associations between a genomic phenotype and a trait. In this method, the data in each region are not combined with a simple sum. Instead, to accommodate spatial structure, they are first transformed using a discrete wavelet transformation. A hierarchical Bayesian regression is then performed in order to generate a bin-level test statistic, as well as estimates of association between the data and the outcome at different spatial scales. This flexibility allows WaveQTL to detect regions with complex association patterns that might be missed by simpler aggregating techniques.

The Wellington-Bootstrap method of Piper et al. [2015] is developed specifically for DNase-seq data. This method focuses only on *footprints* — patterns characteristic of transcription factor binding — identified using Wellington [Piper et al., 2013] and only performs binary comparisons. Wellington-Bootstrap first identifies footprints under each condition, pooling samples within groups. It then tests that each peak is found only in one group using a bootstrap approach to measure significance.

For all of these methods, controlling the false discovery rate is straight-forward if region boundaries are treated as fixed a priori, and the genomic phenotype is assumed to be independent across regions. Each method outputs a single p-value for each region considered. These p-values can then be transformed into false discovery rate estimates, for example using the method of Benjamini and Hochberg [1995].

1.3.2 Joint approaches

The simplest methods for identifying differential regions without pre-specification of boundaries rely on calculating a test statistic at every nucleotide and setting a significance threshold. Neighboring or closely spaced nucleotides with test statistics exceeding the threshold in absolute value are then merged into differential regions using post-processing rules. Two methods using this strategy for methylation sequence data are methylKit [Akalin et al., 2012] and BSmooth [Hansen et al., 2012]. BSmooth is able to improve power significantly by first smoothing the raw data in each sample. Both methylKit and BSmooth calculate one statistic per base-pair and rely on controlling the point-wise (base-pair-wise) false discovery rate which is not appropriate for this problem.

There have also been several proposals using hidden markov models (HMMs). DER Finder of Frazee et al. [2014] and Collado Torres et al. [2016] was developed to identify differentially expressed regions using RNA-seq data. This method models the association between expression level and a trait using a general linear model. The regression coefficients are then modeled as emissions from an HMM with three states — differential expression, no expression at all, and expression not associated with the trait. Differential regions are identified as contiguous blocks of nucleotides assigned to the first state. Significance estimates for each region are calculated by permuting trait values and recalculating test statistics. For each region, the empirical probability of observing an average coefficient (averaging over all base-pairs in the region) in the permuted data that is larger than the observed average is used as a p-value. Frazee et al. [2014] demonstrate the utility of the "annotation agnostic" approach for identifying differential regions with complex association patterns (for example if the signal changes sign within the region) and for making novel discoveries in areas that are poorly annotated.

Allhoff et al. [2014] and Xu et al. [2008] both propose HMM based methods for identification of differential peaks with ChIP-seq data. These methods are designed only for analysis of binary traits. Like DER finder, these two methods define differential regions as consecutive nucleotides assigned to one of the HMM states that include an association between the trait and the sequence count. In the method of Allhoff et al. [2014], regions are post-processed by merging close regions and eliminating small regions. Allhoff et al. [2014] assign a p-value to each region as the probability of observing summed counts within that region with an equal or greater difference under a beta-binomial model. Xu et al. [2008] do not attempt to assign significance to differential regions.

1.4 Imposing structure

Our first proposal for solving the problem described in Section 1.2, *joint adaptive differential estimation* (JADE), is based on estimating $E[y_j|x]$ as a function of position.

For this method, we consider only binary and categorical traits. Using (1.1), we can model the relationship between y_j and a categorical trait with M levels by coding the trait as an indicator variable in $\mathbb{R}^{M-1}$, setting one level as the reference coded by $\mathbf{x} = 0$.

For this discussion, we will use an equivalent formulation in which we code the trait numerically by letting x take on values in $1, \ldots, M$:

$$E\left[y_j | x = m\right] = f_m(s_j). \tag{1.2}$$

We refer to $f_m(s_j)$ as the *profile* for group m. A coefficient vector of $\boldsymbol{\beta}_j = 0$ in (1.1) corresponds to $f_1(s_j) = \cdots = f_M(s_j)$ in (1.2). Differential regions, defined in Section 1.2, can be equivalently defined in the context of (1.2) as sets of contiguous positions in which the group profiles are not identical across all levels of the trait.

We impose two structural constraints on the fitted values in (1.2). First, we require that group profiles be smooth functions of s_j (this assumption motivates our notation, $f_m(s_j)$, for the profile for group m at position s_j). This constraint is suggested by the spatial structure characteristic of genomic phenotypes, illustrated in Figure 1.1. As discussed in Section 1.1, we expect this type of structure in all phenotypes produced by sequence-based assays as a result of the density of measurements made using these techniques.

Second, we impose the condition that there is no association between y_j and x at most locations. This is equivalent to requiring $f_1(s_j) = \cdots = f_M(s_j)$ for most j and reflects a belief that most of the genome is involved in biological processes that are shared across all levels of the trait.

To fit the model in (1.2) at a single position with no constraints, we would typically minimize a loss function on the distance between the fitted values and the observed data:

$$\underset{f_1(s_j),\ldots f_M(s_j)}{\text{minimize}} \left\{ \sum_{m=1}^{M} \sum_{i:x_i=m} l\left(y_{ij} - f_m(s_j)\right) \right\}.$$

We can impose our two constraints — that $f_1(s_j), \ldots, f_M(s_j)$ are smooth functions of position and that $f_1(s_j) = \cdots = f_M(s_j)$ for most j — by adding two convex penalties to the original loss and solving:

$$\underset{f_1,\ldots,f_M}{\text{minimize}} \left\{ \sum_{m=1}^{m=M} \sum_{i:X=m} \sum_{j=1}^{p} l(y_{ij} - f_m(s_j)) + \lambda \sum_{m=1}^{M} P(f_m) + \gamma \sum_{j=1}^{p} \sum_{m<m'} |f_m(s_j) - f_{m'}(s_j)| \right\}. \tag{1.3}$$

Note that we now obtain estimates at all p positions simultaneously. The penalty $P(\cdot)$ in (1.3) is a convex function that penalizes roughness in the fitted profiles. We discuss the form of this penalty in Chapter 2. The associated penalty parameter, λ, controls the smoothness of the resulting profiles. The third term penalizes the difference between pairs of profiles leading us to choose profiles that tend to be similar across categories at most positions. The associated parameter, γ, controls how much the profiles are forced together.

Solving this problem gives an interesting result: We obtain M smooth fitted profiles that are identical at most positions. Differential regions are identified as stretches of positions at which the fitted profiles are not identical. The fact that we obtain regions rather than isolated points from this procedure is due to the smoothness imposed on the fitted profiles.

We describe our method for fitting this model and its behavior in Chapter 2. We also present an application of JADE to identifying differentially methylated regions between three cell types.

1.5 Point-wise tests and false discovery rate control

JADE is powerful but it is not appropriate in all settings. Although JADE can indicate interesting regions and rank them, it does not provide a way to estimate the false discovery rate. Since we are interested in identifying associated regions rather than individual points, the appropriate false discovery rate criterion is the *region-wise false discovery rate* (rFDR), which is the expected proportion of region-level discoveries that contain true signal. Our second proposal, the *flexible robust excursion test* (FRET) is an alternative approach to adaptively identifying differential regions while controlling the rFDR.

FRET is an adaptation of a simple strategy, which we refer to as an *excursion procedure*, described by Siegmund et al. [2011] and Chouldechova [2014]. Excursion procedures detect underlying signal in point-wise statistics by scanning for large values. Applied to the problem in Section 1.2, the simplest form of an excursion procedure has three steps: 1) Calculate a statistic testing the hypothesis $\boldsymbol{\beta}_j = 0$ in (1.1) for each value of j (ignoring the surrounding data). 2) Smooth the test statistics; and 3) Choose a threshold. Declare positions where the smoothed statistic exceeds the threshold in absolute value to have an association between the genomic phenotype and the trait. Combine contiguous associated points into single discoveries.

For an excursion procedure, the rFDR is determined by the the threshold in step 3. We can choose a threshold to control the rFDR by applying results of Siegmund et al. [2011].

We make two contributions in adapting this procedure for use with genomic data. First, we propose the use of a robust test statistic. Genomic phenotype data often contain outliers or are heterogenous within levels of the trait. This type of pattern can arise as a result of unmeasured biological and technical factors that influence the phentoype. We demonstrate that, when samples are heterogeneous or contain outliers, using the robust statistic results in large power gains. When there is little heterogeneity, the robust statistic is only slightly less powerful than non-robust alternatives.

Second, we introduce a procedure that allows the threshold to vary with position. This is important in genomic applications because, in these problems, we typically find that some regions are much noisier than other regions. Choosing high thresholds in noisy regions and lower thresholds in less variable regions allows us to make discoveries in "quiet" regions without incurring many false discoveries from the noisy regions.

This procedure is described in Chapter 3. In Chapter 4, we apply FRET and several alternative methods to detecting differences in DNase 1 sensitivity between small cell lung cancer cell lines that respond to a particular therapy and cell lines that are resistant.

Chapter 2

DIFFERENTIAL REGION DETECTION VIA PROFILE ESTIMATION

We first propose an approach to the problem described in Section 1.2 that is based on estimating the expected value of the genomic phenotype as a function of position. Rather than estimate $\boldsymbol{\beta}_j$ in (1.1) directly, we estimate the *profile*, $E[y_j|x]$, for each value of x (subject to some constraints) and look for regions in which profiles are identical across all trait values. We call this approach *joint adaptive differential estimation* (JADE) because we simultaneously estimate profiles and detect differential regions with data-adaptive boundaries.

In this discussion, we consider only categorical traits such as disease status or tissue type. Since we will not work directly with the linear model in (1.1), we slightly modify our previous notation for the trait x, which we now assume to be numerically coded, taking on values in $1, \ldots, M$ rather than an indicator variable.

The key observation that motivates our proposal is that genomic phenotype data possess structure that we can exploit to improve our estimates of $E\left[y_j|x\right]$. This structure allows us to "borrow" information about $E[y_j|x]$ in two directions. To illustrate this idea, let $\mathbf{Y}$ be the $n \times p$ matrix of genomic phenotype data whose rows correspond to samples and columns correspond to genomic positions. Without additional assumptions, the model in (1.1) requires us to estimate pM parameters separately — one for each position and each group. In the unstructured problem, to estimate $E[y_j|x = m]$, we use only those observations in the jth column of $\mathbf{Y}$ corresponding to samples in the mth trait group (for example, we could use the average of these observations).

Fortunately, there are several types of structure we can add to the model that will reduce the variance of our estimates. We expect the data to possess *spatial structure*, like the patterns observed in Figure 1.1. This suggests that we should make similar profile estimates at physically close locations. Therefore, for estimating $E[y_j|x]$, we can borrow information horizontally within rows of $\mathbf{Y}$, using observations at neighboring sites to inform our estimates at site j. This is the idea behind smoothing — considering data at neighboring locations can reduce the variance of the estimate of the mean. We discuss smoothing in more detail in Section 2.2.1.

The second type of structure we can employ is the typical high-dimensional assumption of sparsity. We assume that there is an association between y_j and $\mathbf{x}$ over only a small portion of the genome. This assumption reflects our belief that most of the genome is involved in critical biological functions shared by samples across all trait levels. This assumption means that, at most locations, $E[y_j|x = 1] = \cdots = E[y_j|x = M]$, so we can borrow information vertically in columns of $\mathbf{Y}$, using the observations made at position s_j across all trait values to inform our estimate of $E[y_j|x]$. This is similar to the idea behind penalized regression estimates, such as the LASSO [Tibshirani, 1996], which shrink regression coefficients towards zero, forcing fitted values towards an overall mean. We discuss this type of structure in Section 2.2.2.

Combining both structural assumptions gives our estimates very useful traits: They are smooth and identical in most places. As a consequence of smoothness, the base-pairs at which the profiles differ conveniently occur in clumps, allowing us to identify differential regions rather than isolated points. We now introduce some specific notation for this discussion and formally describe our approach to solving this problem.

2.1 Problem formulation

Recall that the categorical trait is coded numerically by $x \in \{1, \ldots, M\}$ and that $\mathbf{y} = (y_1, \ldots, y_p)^T$ denotes the genomic phenotype measured at positions $s_1 < s_2 < \ldots < s_p$ along the genome.

For a given value of x, we assume that $\mathbf{y}$ varies smoothly as a function of genomic position,

$$E\left[y_j|x = m\right] = f_m\left(s_j\right).$$

Here the function f_m is the genomic profile for the mth class. If all M profiles are identical at site s_j ($f_m(s_j) = f_{m'}(s_j)$ for all $m \neq m'$), then there is no association between the mean of y_j and x. If $f_m(s_j) \neq f_{m'}(s_j)$ for some $1 \leq m < m' \leq M$, then there is an association between the mean of y_j and x. Our goal is to identify *differential regions*, or contiguous blocks of associated sites.

In what follows, we assume that we have n independent observations of $(x, \mathbf{y})$, denoted $(x_1, \mathbf{y}_1), \ldots, (x_n, \mathbf{y}_n)$. We now introduce some notation that will be used throughout this chapter. Let N_m denote the number of observations with $x_i = m$, so that $N_1 + \ldots + N_M = n$. Let $\bar{y}_{mj} \equiv \sum_{i:x_i=m} y_{ij}/N_m$, and let $\bar{\mathbf{y}}_m \equiv (\bar{y}_{m1}, \ldots, \bar{y}_{mp})^\top$. Furthermore, we let $\theta_{mj} \equiv f_m(s_j)$, and $\boldsymbol{\theta}_m \equiv (\theta_{m1}, \ldots, \theta_{mp})^\top$. In what follows, unless otherwise specified, the letter i will index the n observations, m will index the M values of the categorical trait x, and j will index the p genomic positions of $\mathbf{y}$.

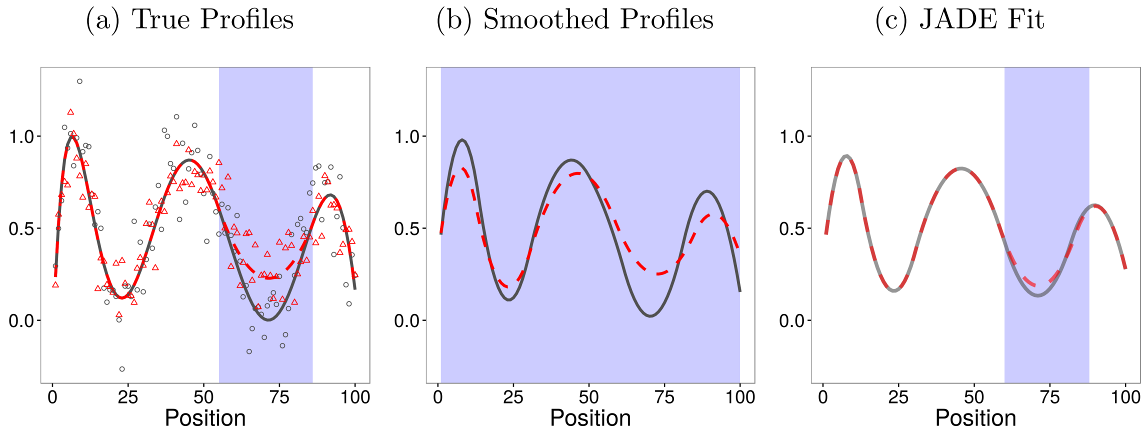
2.1.1 Example

We illustrate JADE with a simple toy example. In each of two groups, we simulate a quantitative genomic phenotype at a series of evenly spaced positions, $s_1, \ldots, s_p$. The data are generated as an overall group-specific mean curve, plus independent normal errors, as shown in Figure 2.1a. The two group-specific mean curves differ only for $s_j \in [55, 85]$.

We first consider estimating the mean curves by separately smoothing the data corresponding to each of the two groups. As is shown in Figure 2.1b, the two estimated profiles are somewhat different at nearly every location.

In contrast, the results from applying JADE to this data are shown in Figure 2.1c. JADE simultaneously smooths the data in each group, and penalizes the differences between the two estimated mean curves. Therefore, JADE can approximately recover the differential region shown in Figure 2.1a.

Of course, the data that we encounter in real biological problems, such as the application studied in Section 2.6, are more complicated than the toy example shown in Figure 2.1a. Real data are often characterized by unevenly spaced positions $s_1, \ldots, s_p$; sites for which a subset


(a) True Profiles

(b) Smoothed Profiles

(c) JADE Fit

Figure 2.1: An illustration of the toy example described in Section 2.1.1. In Figure 2.1a, red and black data points are generated as normal observations with mean given by the corresponding colored lines. Blue shading in 2.1a indicates the region in which the two true profiles are not identical. In Figure 2.1b, profile estimates are obtained by smoothing the two groups separately. These profiles are separated over the entire region. In Figure 2.1c, profile estimates are obtained from JADE. The small region in which the estimated profiles differ is shaded in blue. The detected region largely overlaps the true region of difference.

of groups are missing measurements; and non-constant variance of the genomic phenotype measurements. As we describe in the following sections, JADE is able to accommodate all of these characteristics.

2.2 Penalties to induce structure

JADE combines two tasks: (i) estimation of a smooth mean curve within each group; and (ii) fusion of the mean curves across groups. Here we use the term *fusion* to describe JADE's ability to provide mean curve estimates that are identical across multiple groups at a particular genomic position. That is, if our estimates of $f_m(s_j)$ and $f_{m'}(s_j)$ are identical for some $m \neq m'$, then we say that the estimated mean curves for the mth and m'th classes are *fused* at position s_j.

We briefly discuss the application of existing penalized regression methods to the two aforementioned tasks.

2.2.1 Smoothing a genomic phenotype

Consider the task of smoothing a single observation of a genomic phenotype, $\mathbf{y}_i \in \mathbb{R}^p$, measured at (potentially unevenly spaced) positions $s_1 < \ldots < s_p$. Given weights $a_1, \ldots, a_p$, we consider the optimization problem

$$\underset{f}{\text{minimize}} \left\{ \frac{1}{2} \sum_{j=1}^{p} a_j \left(y_{ij} - f(s_j) \right)^2 + \lambda P\left(f\right) \right\}. \tag{2.1}$$

The smoothed estimate, $\hat{f}$, minimizes the sum of two terms: a goodness-of-fit term between y_{ij} and $f(s_j)$, and a penalty term that discourages a rough or complex f. The penalty parameter λ controls the relative importance of these two terms. There are a number of options for $P(\cdot)$, such as a smoothing spline penalty [Reinsch, 1967] or an ℓ_1 trend filtering penalty [Kim et al., 2009, Tibshirani, 2014].

Trend filtering induces piecewise polynomial estimates for $\hat{f}$, of pre-specified order k, with adaptively chosen knots. The choice of k is guided by the characteristics of the data at hand:

for instance, trend filtering with $k = 0$ [also known as the fused lasso; see Tibshirani et al., 2005] is appropriate for the piecewise constant structure of copy number data [Tibshirani and Wang, 2008]; while $k = 2$ is appropriate for relatively smooth DNA methylation data. Trend filtering is *locally adaptive*, in the sense that it can be used to fit a curve that is very smooth in one region of the domain and very rough in another; this is discussed extensively in Tibshirani [2014]. This property is very attractive within the context of analyzing messy, heterogeneous, and heteroskedastic biological data. Consequently, in what follows, we take $P(\cdot)$ in (2.1) to be a trend filtering penalty.

For convenience, we now switch to using vector notation. The ℓ_1 trend filtering estimate, $\hat{\boldsymbol{\theta}}$, is the solution to the optimization problem

$$\underset{\boldsymbol{\theta} \in \mathbb{R}^p}{\text{minimize}} \left\{ \frac{1}{2} \sum_{j=1}^{p} a_j \left(y_{ij} - \theta_j \right)^2 + \lambda \left\| \mathbf{D}^{k+1,s} \boldsymbol{\theta} \right\|_1 \right\}, \tag{2.2}$$

where $\mathbf{D}^{k+1,s}$ is the $(p-k-1) \times p$ discrete $(k+1)$th derivative matrix, the entries of which depend on both k and the spacing of $s_1, \ldots, s_p$. The specific form of this matrix is detailed in Appendix A and in Tibshirani [2014].

The weights $a_1, \ldots, a_p$ in (2.1) and (2.2) can account for heterogeneity in the variance of y_{ij}. Setting $a_1 = \ldots = a_p$ gives equal weight to each position. Setting a_j proportional to the inverse of an estimate of the variance of y_{ij} gives less weight to positions with lower quality data.

2.2.2 Fusing genomic phenotypes

Now consider the task of fusing n observations of a genomic phenotype, $\mathbf{y}_1, \ldots, \mathbf{y}_n \in \mathbb{R}^p$ — that is, we seek to encourage the estimated means to be identical at a given site. The *convex clustering* estimates, $\hat{\boldsymbol{\theta}}_1, \ldots, \hat{\boldsymbol{\theta}}_n$, solve the optimization problem [Pelckmans et al., 2005, Hocking et al., 2011, Heinzl and Tutz, 2014]

$$\underset{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n \in \mathbb{R}^p}{\text{minimize}} \left\{ \sum_{i=1}^{n} \frac{1}{2} \sum_{j=1}^{p} a_{ij} \left(y_{ij} - \theta_{ij} \right)^2 + \gamma \sum_{i<i'} \left\| \boldsymbol{\theta}_i - \boldsymbol{\theta}_{i'} \right\|_q \right\}. \tag{2.3}$$

In (2.3), a_{ij} is a weight for the jth locus in the ith observation.

For γ sufficiently large, the ℓ_q penalty in (2.3) will encourage similarity between $\hat{\boldsymbol{\theta}}_i$ and $\hat{\boldsymbol{\theta}}_{i'}$. In particular, if $q = 2$, then when γ is large, the entire vectors $\hat{\boldsymbol{\theta}}_i$ and $\hat{\boldsymbol{\theta}}_{i'}$ will tend to be identical, or completely fused at all sites. In this case, the set of observations for which $\hat{\boldsymbol{\theta}}_i$ are identical can be interpreted as clusters. In contrast, if $q = 1$, then a large value of γ will encourage individual elements $\hat{\theta}_{ij}$ and $\hat{\theta}_{i'j}$ to be identical. This amounts to fusing the vectors $\hat{\theta}_i$ and $\hat{\theta}_{i'}$ at a subset of the sites.

2.3 Joint smoothing and comparison with JADE

Recall from the beginning of Section 2.1 the problem set-up: each observation belongs to one of M categories, N_m denotes the number of observations within the mth category, and $\bar{\mathbf{y}}_m \in \mathbb{R}^p$ denotes the mean of the observations of the genomic phenotype within the mth category.

Our goal is to estimate a mean genomic phenotype profile, $\hat{\boldsymbol{\theta}}_1, \ldots, \hat{\boldsymbol{\theta}}_M \in \mathbb{R}^p$, for each of the M categories. We want each mean profile to be smooth, and for the M mean profiles to be identically equal to each other for many of the loci $s_1, \ldots, s_p$. To do this, we combine the smoothing and fusion penalties seen in (2.2) and (2.3) into a single convex optimization problem.

The JADE estimator is defined as the solution to the convex optimization problem

$$\underset{\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_M \in \mathbb{R}^p}{\text{minimize}} \left\{ \sum_{m=1}^{M} \frac{N_m}{2} \left\| \mathbf{A}_m \left(\bar{\mathbf{y}}_m - \boldsymbol{\theta}_m \right) \right\|_2^2 + \lambda \sum_{m=1}^{M} \left\| \mathbf{D}^{k+1,s} \boldsymbol{\theta}_m \right\|_1 + \gamma \sum_{m<m'} \left\| \boldsymbol{\theta}_m - \boldsymbol{\theta}_{m'} \right\|_1 \right\}. \quad (2.4)$$

This minimization consists of three terms: a weighted sum of squared residuals, a sum of ℓ_1 trend filtering penalties, and a clustering penalty. When the non-negative tuning parameter λ is sufficiently large, the trend filtering penalty encourages each mean profile to be smooth. Equation (2.4) could be modified to allow each of the M groups to have its own smoothness tuning parameter, $\lambda_1, \ldots, \lambda_M$. For simplicity we use a single common parameter.

When the non-negative tuning parameter γ is sufficiently large, the clustering penalty encourages many of the p sites to have exactly the same value in the mth and m'th mean profiles, for $m \neq m'$. In fact, when γ is large enough, some of the p sites will have $\hat{\theta}_{1j} = \ldots = \hat{\theta}_{Mj}$; these can be interpreted as regions of the genome where the mean profile is constant across the M groups. Thus, JADE simultaneously identifies regions of the genome in which the genomic phenotype is associated with the categorical variable x, and estimates smooth average profiles for each group. It accomplishes this in an efficient way that borrows strength across nearby sites, without performing a separate test at each site in the genome. Selection of λ and γ in (2.4) is discussed in Section 2.4.2.

In (2.4), $\mathbf{A}_m$ are $p \times p$ diagonal weight matrices. These can be used to account for the fact that the elements of $\bar{\mathbf{y}}_m$ may have non-constant variance across the p sites, perhaps due to varying numbers of reads across the genome. Furthermore, if no data are available for the jth site in the mth group, then the jth diagonal element of $\mathbf{A}_m$ can be set to zero.

2.4 Solving the JADE optimization problem

2.4.1 An alternating direction method of multipliers algorithm for JADE

The JADE optimization problem (2.4) is convex, so in principle, it can be solved with general-purpose convex solvers, such as `SDPT3` [Tütüncü et al., 2003] or `SeDuMi` [Sturm, 1999]. However, these solvers do not scale well to genome-sized problems. Therefore, we have developed an efficient custom *alternating direction method of multipliers* [ADMM; Boyd et al., 2010] algorithm for solving (2.4).

Our algorithm relies on the key observation by Tibshirani [2014] that the trend filtering penalty matrix $\mathbf{D}^{k+1,s}$ can be decomposed as $\mathbf{D}^{k+1,s} = \mathbf{D}^1\tilde{\mathbf{D}}^{k,s}$, where $\mathbf{D}^1$ is the $(p-k-1) \times (p-k)$ first difference operator, and $\tilde{\mathbf{D}}^{k,s}$ is a $(p-k) \times p$ scaled kth-order difference operator. Details of these two matrices are provided in Section A.1 of Appendix A.

Using this decomposition, we can re-write (2.4) as

$$\underset{\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_M,\boldsymbol{\eta}_1,\ldots,\boldsymbol{\eta}_M,\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_M}{\text{minimize}} \left\{ \sum_{m=1}^{M} \frac{N_m}{2} \left\| \mathbf{A}_m \left(\bar{\mathbf{y}}_m - \boldsymbol{\theta}_m \right) \right\|_2^2 + \lambda \sum_{m=1}^{M} \left\| \mathbf{D}^1 \boldsymbol{\alpha}_m \right\|_1 + \gamma \sum_{m<m'} \left\| \boldsymbol{\eta}_m - \boldsymbol{\eta}_{m'} \right\|_1 \right\} \tag{2.5}$$

$$\text{subject to} \qquad \tilde{\mathbf{D}}^{k,s} \boldsymbol{\theta}_m = \boldsymbol{\alpha}_m, \quad \boldsymbol{\theta}_m = \boldsymbol{\eta}_m, \qquad m = 1, \ldots, M.$$

The scaled augmented Lagrangian for this problem is

$$\begin{aligned} L(\boldsymbol{\theta}, \boldsymbol{\alpha}, \boldsymbol{\eta}, \mathbf{u}) = & \sum_{m=1}^{M} \frac{N_m}{2} \left\| \mathbf{A}_m (\bar{\mathbf{y}}_m - \boldsymbol{\theta}_m) \right\|_2^2 + \lambda \sum_{m=1}^{M} \left\| \mathbf{D}^1 \boldsymbol{\alpha}_m \right\|_1 + \gamma \sum_{m<m'} \left\| \boldsymbol{\eta}_m - \boldsymbol{\eta}_{m'} \right\|_1 \\ & + \frac{1}{2} \sum_{m=1}^{M} \rho_{\alpha m} \left\| \tilde{\mathbf{D}}^{k,s} \boldsymbol{\theta}_m - \boldsymbol{\alpha}_m + \mathbf{u}_m^{(\alpha)} \right\|_2^2 + \frac{\rho_\eta}{2} \sum_{m=1}^{M} \left\| \boldsymbol{\theta}_m - \boldsymbol{\eta}_m + \mathbf{u}_m^{(\eta)} \right\|_2^2, \end{aligned} \tag{2.6}$$

where $\boldsymbol{\theta} \equiv (\boldsymbol{\theta}_1^\top, \ldots, \boldsymbol{\theta}_M^\top)^\top$, $\boldsymbol{\alpha} \equiv (\boldsymbol{\alpha}_1^\top, \ldots, \boldsymbol{\alpha}_M^\top)^\top$, and $\boldsymbol{\eta} \equiv (\boldsymbol{\eta}_1^\top, \ldots, \boldsymbol{\eta}_M^\top)^\top$. In (2.6), $\mathbf{u} \equiv (\mathbf{u}_1^\top, \ldots, \mathbf{u}_M^\top)^\top$ is a vector of dual variables, where $\mathbf{u}_m \equiv \left(\left(\mathbf{u}_m^{(\alpha)} \right)^\top, \left(\mathbf{u}_m^{(\eta)} \right)^\top \right)^\top$ for $\mathbf{u}_m^{(\alpha)} \in \mathbb{R}^{p-k}$ and $\mathbf{u}_m^{(\eta)} \in \mathbb{R}^p$. The dual variables are broken into multiple components in order to allow for different step sizes, $\rho_{\alpha 1}, \ldots, \rho_{\alpha M}$ and ρ_η, as this leads to faster convergence. In our implementation, we adjust the step sizes adaptively; details are in Section A.2.2 of Appendix A.

The ADMM algorithm corresponding to the scaled augmented Lagrangian (2.6) is given in Algorithm 1. The initialization in Step 1 simply amounts to solving a separate ℓ_1 trend filtering problem for each $\boldsymbol{\eta}_m$, $m = 1, \ldots, M$. The update in Step 3(b) involves solving a fused lasso problem; this can be done using the algorithm of Johnson [2013]. The update in Step 3(c) has an explicit form in the case of $M = 2$ groups (see Section A.2 of Appendix A). For $M \geq 3$ groups, we make use of the solution of Hocking et al. [2011].

If the output of Algorithm 1 has the property that $\eta_{1j} = \ldots = \eta_{Mj}$ for some j, $1 \leq j \leq p$, then we conclude that at the jth locus, the M mean genomic phenotype profiles are identical. Due to numerical issues, however, we may not observe exact equality between η_{mj} and $\eta_{m'j}$ for $m \neq m'$. Therefore, in practice, we set a threshold ε, and conclude that η_{mj} and $\eta_{m'j}$

Algorithm 1 ADMM Algorithm For Solving the JADE Optimization Problem (2.4)

```
1. Initialize $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_M$ as solutions to (2.4) with $\gamma = 0$.

2. For $m = 1, \ldots, M$, initialize $\mathbf{u}_m = 0$ and $\boldsymbol{\alpha}_m = \tilde{\mathbf{D}}^{k,s}\boldsymbol{\eta}_m$.

3. Iterate until the convergence criteria described in Section A.2.2 of Appendix A are satisfied:

    (a) For $m = 1, \ldots, M$, update

        $\boldsymbol{\theta}_m \longleftarrow \left( N_m \mathbf{A}_m^\top \mathbf{A}_m + \rho_{\alpha m} \left(\tilde{\mathbf{D}}^{k,s}\right)^\top \tilde{\mathbf{D}}^{k,s} + \rho_\eta \mathbf{I} \right)^{-1} \cdot \left( N_m \mathbf{A}_m^\top \mathbf{A}_m \bar{\mathbf{y}}_m + \rho_{\alpha m} \left(\tilde{\mathbf{D}}^{k,s}\right)^\top \left(\boldsymbol{\alpha}_m - \mathbf{u}_m^{(\alpha)}\right) + \rho_\eta \left(\boldsymbol{\eta}_m - \mathbf{u}_m^{(\eta)}\right) \right).$

    (b) For $m = 1, \ldots, M$, update

        $\boldsymbol{\alpha}_m \leftarrow \underset{\boldsymbol{\alpha}_m}{\operatorname{argmin}} \left\{ \frac{1}{2} \left\| \boldsymbol{\alpha}_m - \left(\tilde{\mathbf{D}}^{k,s}\boldsymbol{\theta}_m + \mathbf{u}_m^{(\alpha)}\right) \right\|_2^2 + \frac{\lambda}{\rho_{\alpha m}} \left\| \mathbf{D}^1 \boldsymbol{\alpha}_m \right\|_1 \right\}.$

    (c) For $m = 1, \ldots, M$, update

        $\boldsymbol{\eta}_m \leftarrow \underset{\boldsymbol{\eta}_m}{\operatorname{argmin}} \left\{ \sum_{m=1}^{M} \frac{1}{2} \left\| \boldsymbol{\eta}_m - \left(\boldsymbol{\theta}_m + \mathbf{u}_m^{(\eta)}\right) \right\|_2^2 + \frac{\gamma}{\rho_\eta} \sum_{m<m'} \left\| \boldsymbol{\eta}_m - \boldsymbol{\eta}_{m'} \right\|_1 \right\}.$

    (d) For $m = 1, \ldots, M$, update the dual variables by setting

        $\mathbf{u}_m^{(\alpha)} \leftarrow \mathbf{u}_m^{(\alpha)} + \tilde{\mathbf{D}}^{k,s}\boldsymbol{\theta}_m - \boldsymbol{\alpha}_m, \qquad \mathbf{u}_m^{(\eta)} \leftarrow \mathbf{u}_m^{(\eta)} + \boldsymbol{\theta}_m - \boldsymbol{\eta}_m.$

    (e) Update the step sizes $\rho_{\alpha 1}, \ldots, \rho_{\alpha M}$ and $\rho_\eta$ as described in Section A.2.2 of Appendix A, and rescale the dual variables by setting

        $\mathbf{u}_m^{(\alpha)} \leftarrow \mathbf{u}_m^{(\alpha)} \cdot \rho_{\alpha m}^{old} / \rho_{\alpha m} \qquad \mathbf{u}_m^{(\eta)} \leftarrow \mathbf{u}_m^{(\eta)} \cdot \rho_\eta^{old} / \rho_\eta.$
```

are equal if the absolute difference between them is below ε. The mean genomic phenotype profile for the mth group can be obtained from $\boldsymbol{\theta}_m$ in the output of Algorithm 1.

In practice, it is computationally prohibitive to solve the JADE optimization problem (2.4) on genome-sized data. Therefore, we take a pragmatic approach: we segment the genome, and apply JADE to each segment in parallel. In the methylation data application presented in Section 5, there is a natural segmentation that respects the biology of the problem. Other situations might require more arbitrary segmentation. Provided that the regions to be detected by JADE are short relative to the segmentation that we impose, we expect the segmentation to have little effect on the results.

2.4.2 Tuning parameter selection

The JADE optimization problem in (2.4) involves two non-negative tuning parameters. The parameter λ controls the smoothness of the mean genomic phenotype profiles while γ controls the amount of fusion between pairs of profiles. We take a two-stage approach to select λ and γ rather than performing a grid search over all combinations of values.

In both stages, cross-validation is performed by dividing the pM data points $\bar{y}_{mj}$, $m \in \{1 \dots M\}$, $j \in \{1 \dots p\}$, into l folds. For a given value of m, each fold contains a data point at every lth position, and the folds are staggered so that all m data points at a single position are not in the same fold. For example, if $M = 2$, $p = 10$, and $l = 5$, then the first fold could contain $\bar{y}_{1,1}, \bar{y}_{1,6}, \bar{y}_{2,2}$, and $\bar{y}_{2,7}$.

In the first stage, we set $\gamma = \infty$ in (2.4); this amounts to combining all of the data into a single trend filtering problem. We then perform cross-validation in order to select λ.

In the second stage of tuning parameter selection, we hold λ fixed at the value selected in the first stage, and select the tuning parameter γ using cross-validation. Additional details are provided in Section A.3 of Appendix A.

In both stages of cross-validation, we apply the one-standard-error rule, selecting the largest tuning parameter value that has cross-validation error within one standard deviation of the minimum [Hastie et al., 2009].

2.5 *Simulations*

In Section 2.5.1, we consider a setting in which the genomic phenotype is continuous-valued. In Section 2.5.2, we consider a setting that is modeled after methylation sequence data.

2.5.1 Normal simulations

Simulation set-up

We simulate $n = 20$ observations, 10 in each of $M = 2$ groups, at $p = 300$ evenly spaced sites, $s_1, \ldots, s_p$. (We explore other values of the sample size n in Section A.5.1 of Appendix A.) The data for the ith observation in the mth group at the jth site is generated as

$$y_{imj} = f_m(s_j) + \epsilon_{imj}, \tag{2.7}$$

where the functions f_1 and f_2 represent the mean genomic phenotype profiles for the two groups, and are displayed in Figure 2.2. The error terms ϵ_{imj} are generated in one of two ways:

1. *Auto-regressive model.* For $m = 1, 2$ and $i = 1, \ldots, 10$,

$$\begin{aligned} z_{imj} \sim & N(0, \sigma^2) \qquad \text{for } \; j = 1 \ldots 300, \\ \epsilon_{imj} = & \begin{cases} z_{imj} & \text{if } \; j = 1 \\ z_{imj} + \rho z_{im(j-1)} & \text{if } \; j > 1 \end{cases}. \end{aligned}$$

   We consider values of $\sigma \in \{0.5, 1, 2\}$ and $\rho \in \{0, 0.2, 0.4\}$.

2. *Random effects model.* For $m = 1, 2$, $i = 1, \ldots, 10$, and $j = 1, \ldots, 300$,

$$b_{im} \sim N(0, \sigma_{\text{re}}^2), \qquad z_{imj} \sim N(0, \sigma^2), \qquad \epsilon_{imj} = b_{im} + z_{imj}. \tag{2.8}$$

   In this set-up, b_{im} represents a mean shift for the ith observation in the mth group, such as one might expect as a result of a batch effect. We choose σ and σ_{re} such that $\sigma^2 + \sigma_{\text{re}}^2 = 5$, and the proportion of variance due to random effects, $\sigma_{\text{re}}^2 / (\sigma^2 + \sigma_{\text{re}}^2)$, takes on values of 0.05, 0.1, 0.15, and 0.2.

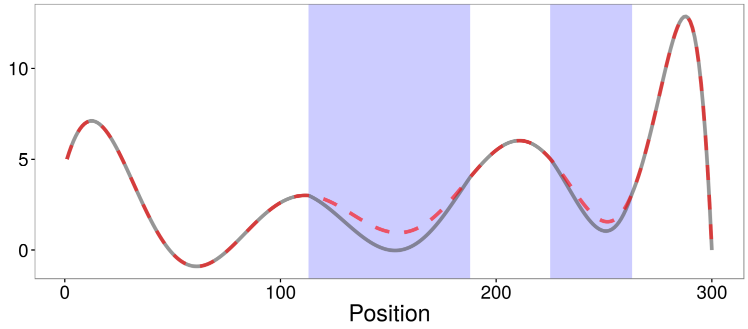


Figure 2.2: Average group profiles for simulated data in Section 2.5. The two profiles are separated in two regions highlighted in blue. In the white regions the two groups have the same mean. For binomial data simulations in Section 2.5.2 these mean curves are scaled to range between 0 and 1.

Methods for comparison

In this section, we compare JADE to three t-test based methods. These methods decouple the tasks of estimating the mean genomic phenotype profiles for each of the M groups, and testing for differences between the M mean genomic phenotype profiles. These approaches assume that $M = 2$.

1. A two-sample t-statistic is calculated at each site, without first smoothing the data. This approach is used by methylKit [Akalin et al., 2012].

2. Each observation is smoothed using local likelihood, with the bandwidth chosen by generalized cross-validation. Then a two sample t-statistic is computed at each site, using the smoothed observations. BSmooth [Hansen et al., 2012] uses this strategy with a fixed bandwidth optimized for methylation data.

3. Each observation is smoothed using a quadratic smoothing spline, with the tuning parameter chosen by generalized cross-validation. Then a two sample t-statistic is computed at each site, using the smoothed observations.

The third method is included in order to understand the impact of different smoothing strategies. For all three methods, a threshold is chosen, and any site with a test statistic exceeding that threshold in absolute value is declared to have a different mean value between the M groups.

We use our own implementation in `R` of all three strategies, because methylKit and BSmooth are both implemented specifically for methylation count data, whereas the genomic phenotypes in this simulation study are continuous. We do not include the WaveQTL method of Shim and Stephens [2015] in our comparisons, as it requires the user to pass in pre-specified genomic regions, and does not provide a per-site assessment of the association between genomic phenotype and category. In our application of JADE, we set the weight matrices A_1 and A_2 in (2.4) to the identity. Tuning parameters were selected according to the procedure in Section 2.4.2.

Results

We now compare the performances of JADE and the three t-test-based methods described in Section 2.5.1. In this section we compare JADE with the alternative methods based on pointwise true and false positive rates. In Section A.5.2 of Appendix A we make comparisons of these same simulations using region-level metrics and present a few additional simulations to further understand the region-wise accuracy of each method. Before presenting these results, we briefly discuss the calculation of false and true positives for each method.

For a given value of γ in the JADE optimization problem (2.4), we declare a false positive if $\hat{\theta}_{1j} \neq \hat{\theta}_{2j}$ and $f_1(s_j) = f_2(s_j)$, and a true positive if $\hat{\theta}_{1j} \neq \hat{\theta}_{2j}$ and $f_1(s_j) \neq f_2(s_j)$. We fit JADE at around 100 values of γ, as described in Section A.3 of Appendix A. For each value of γ considered, we calculate a true positive rate and a false positive rate.

For a given t-statistic method and a given choice of threshold, we declare a false positive if the absolute value of the t-statistic for the jth site exceeds the threshold and $f_1(s_j) = f_2(s_j)$. We declare a true positive if the absolute value of the t-statistic for the jth site exceeds the threshold and $f_1(s_j) \neq f_2(s_j)$. For each method, we calculate true positive and false positive rates for a sequence of threshold values.

Figures 2.3 and 2.4 display the average true positive rate (TPR) as a function of the false positive rate (FPR) for JADE and the three t-test-based methods, for the two error structures described in Section 2.5.1, averaged over 100 simulations. Colored points indicate the average TPR and FPR achieved with tuning parameters selected via cross-validation for JADE, or using a false discovery rate (FDR) of 10% for the t-statistic-based methods, as calculated using SLIM [Wang et al., 2011]. Details of the calculation of these curves are given in Section A.4 of Appendix A.

In all settings, JADE results in a higher TPR for any fixed FPR than the competing methods. We expect JADE to perform well in the random effects setting because it pools observations within each group before smoothing, thereby averaging out individual-level random effects. The JADE framework does not, however, account for the dependence between errors seen in the auto-regressive simulations. These results show that JADE is robust, at least in this setting, to dependence between errors.

The t-statistic-based methods with an FDR cutoff of 10% tend to be more conservative than JADE with γ chosen by cross-validation: that is, they yield fewer false positives and fewer true positives. The average FPR for JADE with γ chosen by cross-validation increases for larger values of ρ in the auto-regressive settings and σ_{re} in the random effects settings.

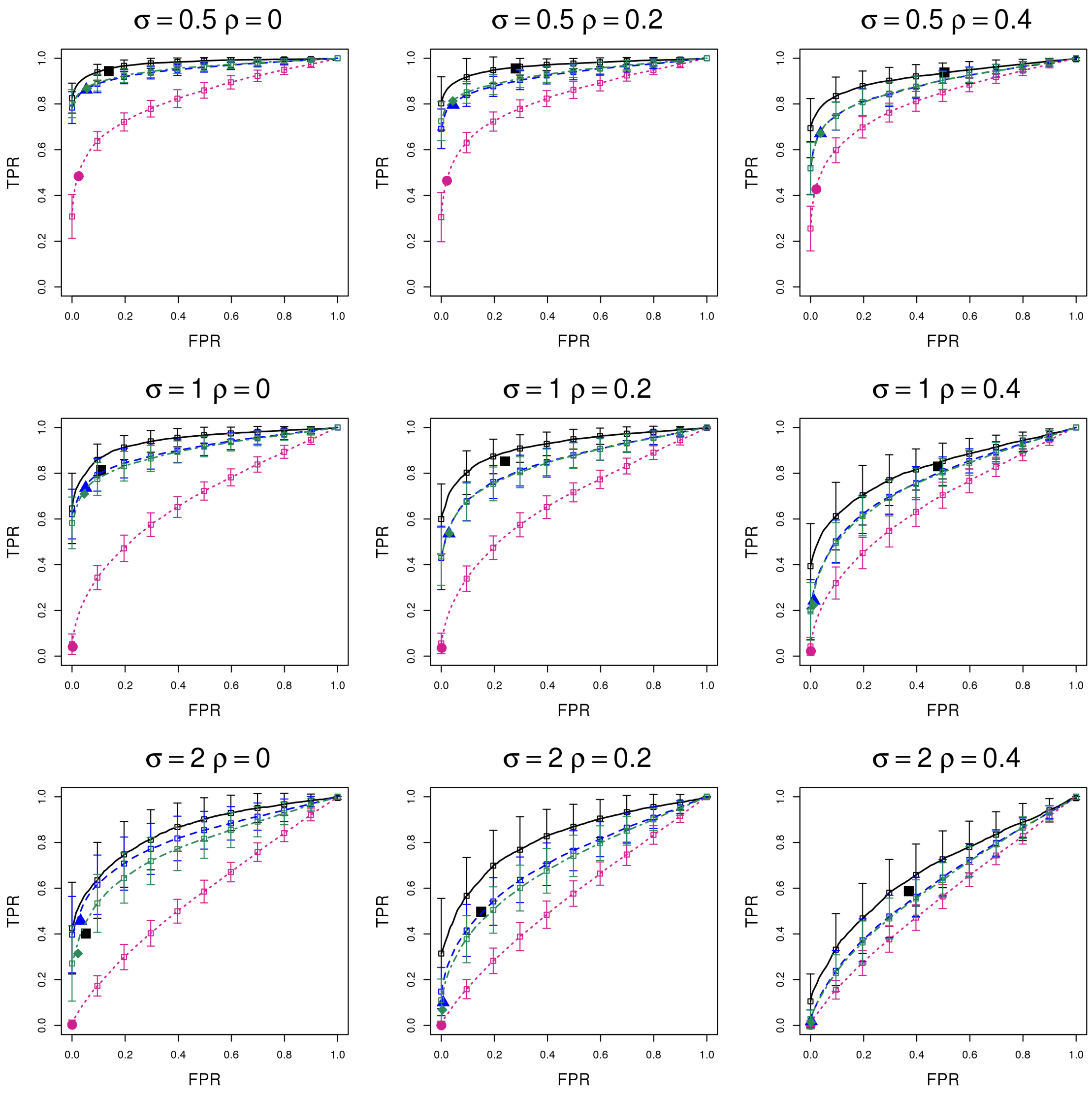


Figure 2.3: Performance of JADE and competing methods in the normal auto-regressive model described in Section 2.5.1. Each panel displays results for a distinct value of $\sigma \in \{0.5, 1, 2\}$, and a value of $\rho \in \{0, 0.2, 0.4\}$. Lines show the average TPR for a fixed FPR, averaged over 100 simulations. The lengths of the vertical bars on either side of the curves equal one sample standard deviation of the TPR. Points indicate average TPR and FPR achieved for JADE with the tuning parameter selected by cross-validation, and for the t-test approaches with an FDR threshold of 10%. Methods shown are JADE (—,■), per-site t-tests applied to the raw data (- - -, ●), and per-site t-tests after smoothing the raw data using splines (– –, ▲) and local likelihood (- - -, ◆). Results for the t-test with spline and local-likelihood smoothing are often nearly identical.

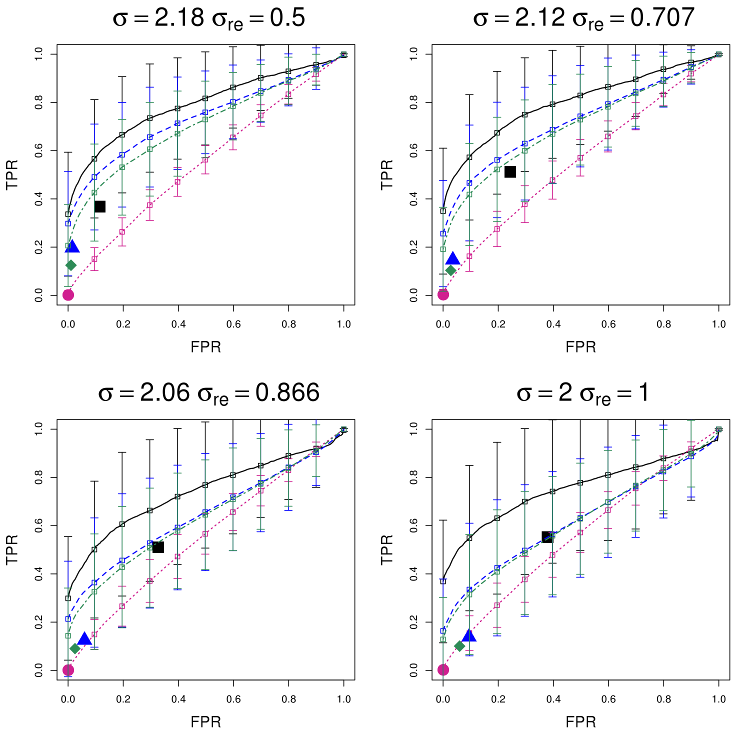


Figure 2.4: Performance of JADE and competing methods in the normal random effects model described in Section 2.5.1. Each panel represents a different proportion of variation due to random effects. Additional details are as in Figure 2.3.

2.5.2 Binomial simulations

Simulation set-up and methods for comparison

In this section, we use methylation sequence data to motivate our simulation set-up. DNA methylation is a chemical modification that can affect cytosine residues directly followed by guanine residues (CpGs). In methylation sequencing experiments, DNA is fragmented, amplified, and bisulfite converted, a process in which non-methylated cytosines in CpGs are converted to uracil. These fragments are then sequenced, and the uracils and cytosines at each CpG are counted. Thus, at each CpG site we obtain two numbers: the number of sequenced fragments (reads) and the number of observed uracils (counts). We analyze methylation data in Section 2.6. In this section we consider a simple simulation mimicking the binomial character of methylation data.

As in Section 2.5.1, we simulate $n = 20$ observations, 10 in each of $M = 2$ groups at $p = 300$ evenly spaced sites. We generate the observed number of counts for the ith individual in the mth group at the jth site as

$$c_{imj} \sim \text{Binom}(n_{imj}, p_{imj}).$$

Section A.6 of Appendix A describes the way in which n_{imj}, the total number of reads, is generated. No sites were permitted to have zero reads, as neither BSmooth [Hansen et al., 2012] nor methylKit [Akalin et al., 2012] can accommodate this.

In order to generate the binomial probability p_{imj}, we first scaled and translated f_1 and f_2, the two mean genomic phenotype profiles displayed in Figure 2.2, to take on values between 0 and 1. We then generated p_{imj} according to a random effects model, as follows:

$$b_{im} \sim N(0, \sigma^2_{\text{re}}), \qquad p_{imj} = \begin{cases} 0 & \text{if } f_m(s_j) + b_{im} < 0 \\ 1 & \text{if } f_m(s_j) + b_{im} > 1 \\ f_m(s_j) + b_{im} & \quad \text{otherwise} \end{cases} \cdot$$

We consider values of $\sigma_{\text{re}} \in \{0, 0.02, 0.05, 0.07\}$.

We fit JADE using the observed proportions $y_{imj} = c_{imj}/n_{imj}$. Due to the binomial mean-variance relationship as well as the variable read depth, the variance of y_{imj} is not constant across sites or observations. We can estimate the variance of y_{imj} as

$$\hat{\sigma}^2_{imj} = \frac{y^*_{imj}(1 - y^*_{imj})}{n_{imj}}, \qquad \text{where } y^*_{imj} = \frac{c_{imj} + 0.5}{n_{imj} + 1}. \tag{2.9}$$

Here, y^*_{imj} differs from y_{imj} in the inclusion of pseudo-counts to prevent estimates of zero variance. To accommodate these variance estimates in JADE, the diagonal elements of the matrix A_m in (2.4) were set to $1/\hat{\sigma}_{imj}$.

In what follows, we compare JADE to two existing methods for analyzing methylation data, methylKit [Akalin et al., 2012] and BSmooth [Hansen et al., 2012]. These are methylation specific implementations of methods 1 and 2 in Section 2.5.1.

The local likelihood smoothing bandwidth is fixed in BSmooth, making its performance dependent on the spacing of the measurement sites. For this comparison, we used a separation between sites of five units ($s_1 = 0, s_2 = 5, s_3 = 10, \ldots$). This spacing is close to what might be expected of bisulfite sequencing data in which measurements are closely spaced but not made at every base-pair.

Results

Both BSmooth and methylKit produce a score for each site, quantifying the evidence that the mean profiles differ at that location. TPRs and FPRs for JADE, BSmooth, and methylKit were computed as described in Section 2.5.1. The results, averaged over 100 simulated data sets, are displayed in Figure 2.5. We find that JADE gives a higher TPR than BSmooth and methylKit for any fixed FPR in all settings.

2.6 Application to methylation data

In this section, we apply JADE to DNA methylation patterns during three stages of skeletal muscle cell development (myoblast, myotube, and adult muscle cells), using reduced

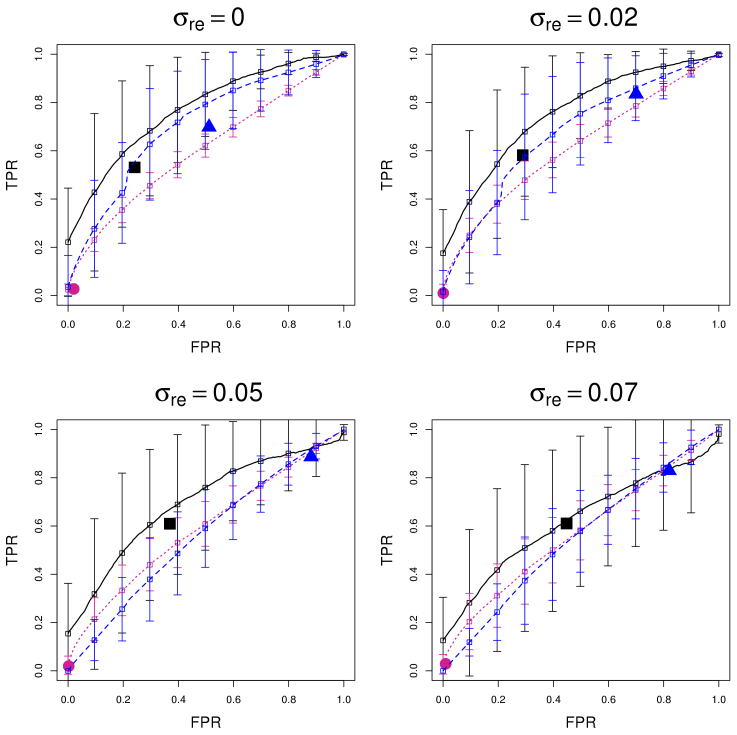


Figure 2.5: Performance of JADE and competing methods in the binomial simulations described in Section 2.5.2. Lines show average TPR for a fixed FPR over 100 simulations. The lengths of the vertical bars on either side of the curves equal one sample standard deviation of the TPR. Points indicate average TPR and FPR for JADE with the tuning parameter selected by cross-validation, and for the methylKit and BSmooth with an FDR threshold of 10%. Methods shown are JADE (—, ■), methylKit (···, ●), and BSmooth (– –, ▲).

representation bisulfite sequencing data from the ENCODE project [The Encode Project Consortium, 2012]. Methylation data were described at the beginning of Section 2.5.2.

These cell lines have been studied extensively: in particular, ChIP-seq peaks, DNaseI peaks, and H3K27ac marks are also available. Therefore, we are able to compare the set of differentially methylated regions (DMRs) detected by JADE with previous findings, and we can assess co-localization with other functional annotations in order to validate our results. In what follows, we will make use of the fact that there is a developmental ordering to the three cell types in our data: myoblasts precede myotubes, which precede mature muscle cells.

2.6.1 Analysis

We compared DNA methylation in myoblasts, myotubes, and mature skeletal muscle cells. Three technical replicates from a single cell line are available for both myoblasts and myotubes, and two technical replicates are available for mature skeletal muscle. We pooled technical replicates, and set the diagonal elements of the A_m weight matrices in (2.4) equal to the inverse of the standard deviation estimates in (2.9). In this analysis, we only examined chromosome 22.

The locations at which DNA methylation can occur, CpG sites, are irregularly distributed throughout the genome. Since there is no biological reason to smooth across very long distances containing no CpG sites, this irregular spacing provides a natural way to segment the genome. We divided the chromosome into segments such that neighboring CpG sites within a segment are separated by less than 2 kb, and the first and last CpG of each segment is measured in all three cell types. Segments with fewer than 20 CpGs were removed. This resulted in 477 segments with an average segment length of 3.0 kb and an average of 64 CpG sites per segment. Running JADE on each of the 477 segments in parallel, with 5-fold cross-validation, required computing efforts equivalent to running 120 cores for two days.

Neither methylKit nor BSmooth can be directly applied to this data, since both methods are intended for a two-group comparison, and in this data set we have three groups.

2.6.2 Results

DMRs identified by JADE

We applied JADE to each of the 477 segments on chromosome 22, with λ and γ selected using 5-fold cross-validation as described in Section 2.4.2 and Section A.3 of Appendix A, and with $\varepsilon = 0.005$ as described in Section 2.4.1.

We declared a DMR as any contiguous set of CpGs at which two or more profiles are separated in the JADE output. Adjacent DMRs separated by a single CpG are combined to form a single DMR. We removed 48 regions containing 536 base-pairs (0.3% of base-pairs in DMRs) in which two pairs of profiles are fused (separation $< \varepsilon$) while the third pair is un-fused (separation $> \varepsilon$), since such a pattern does not give a valid partition of the three profiles.

JADE identified 220 DMRs in 127 segments, with an average length of 826 base-pairs. An example JADE fit is shown in Figure 2.6. In this segment, two DMRs have been identified (shaded in blue). In the DMR on the left, all three profiles are separated, while on the right, the myotube and myoblast profiles are fused.

A single DMR may contain multiple partitions of the profiles, though those in Figure 2.6 each contain only one partition. The 220 DMRs identified by JADE can be divided into 380 sub-regions, each of which contains only a single partition of the profiles.

2.6.3 Loss-of-methylation over the course of muscle cell development

It is well-established in the literature that as myoblasts develop into mature skeletal muscles, a loss of methylation tends to occur [Hupkes et al., 2011, Segalés et al., 2014, Palacios and Puri, 2006, Carrió et al., 2015]. Here we assess whether the DMRs detected by JADE in Section 2.6 are consistent with this expectation.

Each DMR detected by JADE induces some ordering in the estimated mean profiles. For instance, the differential region shown on the left-hand side of Figure 2.6 has the ordering (Mature<Myoblast<Myotube), and the differential region shown on the right-hand side of

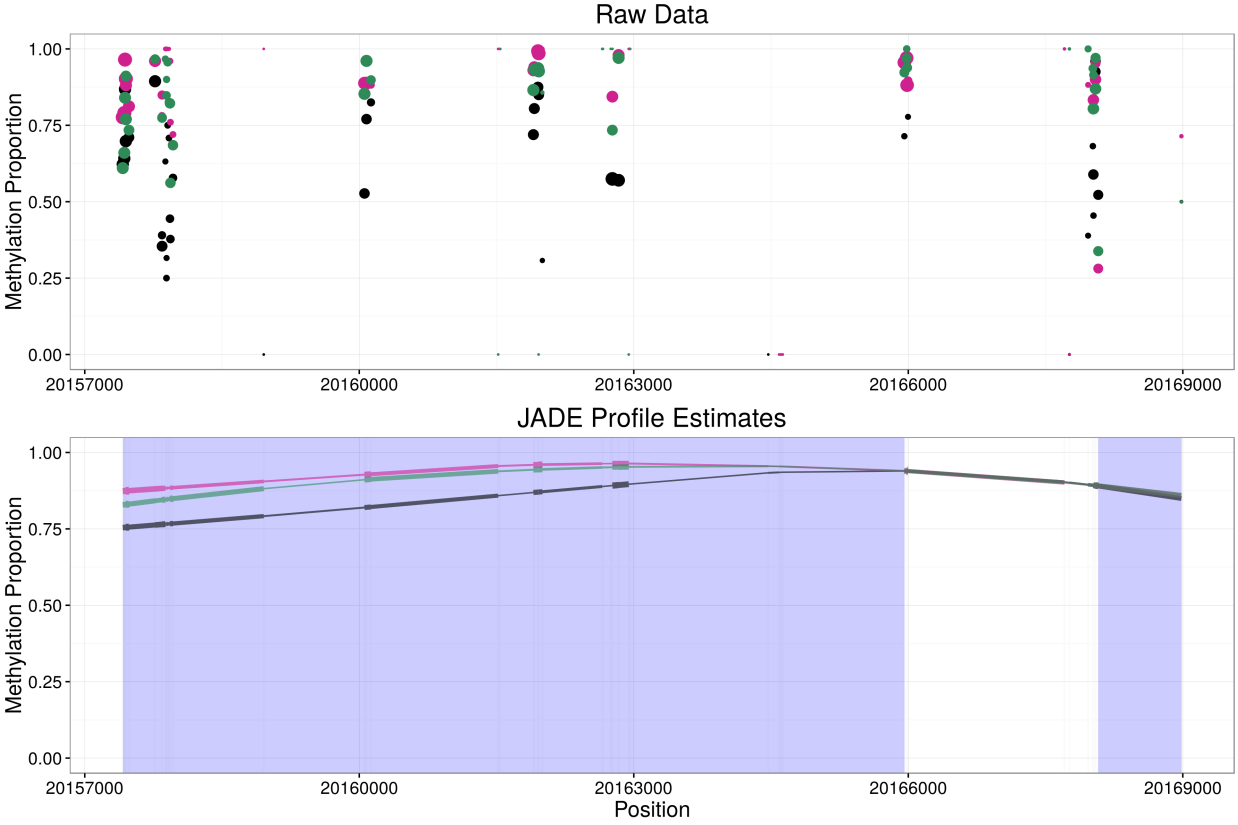


Figure 2.6: Results from one segment of the methylation data analysis, described in detail in Section 2.6.2. Top panel: Observed methylation proportions for myoblasts (●), myotubes (●) and mature skeletal muscle (●). Point size is proportional to the number of reads at each site. Bottom panel: The three profiles estimated by JADE. Colors correspond to the top panel. Blue shading indicates DMRs detected by JADE. Line width is proportional to the number of reads in a 200 base-pair window.

Figure 2.6 has the ordering (Mature<Myoblast=Myotube).

We will refer to a DMR with the induced ordering (Mature≤Myotube≤Myoblast) as a "loss-of-methylation" DMR, as such a DMR displays a monotone decrease in methylation over the course of development. We will refer to a DMR with the induced ordering (Mature≥Myotube≥Myoblast) as a "gain-of-methylation" DMR. Some DMRs display neither loss-of-methylation nor gain-of-methylation over the course of development. For instance, the DMR shown on the left-hand side of Figure 2.6 is disordered with respect to developmental stage.

The orderings induced by the DMRs detected by JADE are summarized in Supplementary Figure 2.7. Of the base-pairs that belong to a DMR, 35.9% fall within a loss-of-methylation DMR, 21.9% fall within a gain-of-methylation DMR, and the remaining 42.2% are disordered with respect to developmental stage. Of the DMRs that are disordered with respect to developmental stage, 94% have the ordering (Mature<Myoblast<Myotube) or (Myotube<Myoblast<Mature), and many of these display very small differences between the myoblast and myotube profiles, and much larger differences between the mature cell profiles and the other two. This makes them very similar to the classes (Mature<Myoblast=Myotube) and (Myotube=Myoblast<Mature), which are consistent with loss-of-methylation and gain-of-methylation, respectively.

Co-localization of DMRs with genomic features

In order to assess the quality of the DMRs detected by JADE, we evaluate their overlap with epigenetic annotations and sequence-based landmarks. We expect the DMRs detected by JADE to be enriched for some of these features.

We consider epigenetic annotations obtained in myotubes and mature skeletal muscle, available from ENCODE. (Annotations obtained in myoblasts were not available.) These annotations include (i) transcription factor binding sites identified through ChIP-seq; (ii) active enhancer regions indicated by H3K27ac histone methylation; and (iii) DNase I hypersensitive sites, which mark open chromatin.

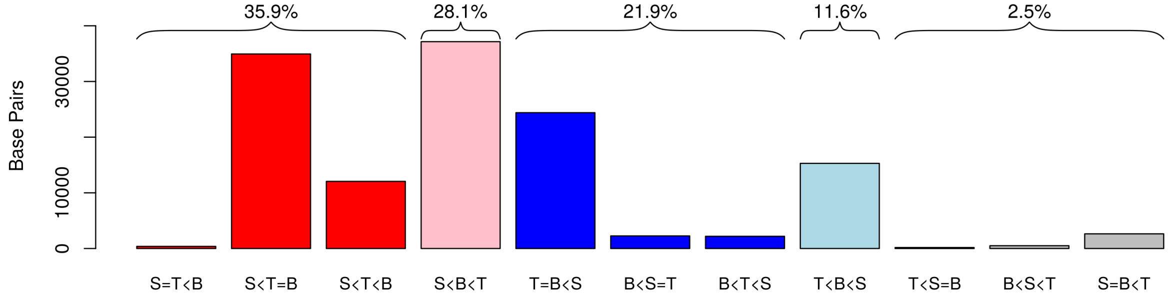


Figure 2.7: Methylation patterns in differential regions as described in Section 2.6.3. Each x-axis label indicates an ordering of the mean methylation profiles. For instance, S=T<B indicates the set of DMR sub-regions for which the estimated mean mature skeletal muscle profile (S) equals the estimated mean myotube profile (T), and is less than the estimated mean myoblast profile (B). For DMR sub-regions in which the estimated mean profiles for two tissue types intersect, the order is determined based on the average difference between profiles. The y-axis represents the total base-pairs in the DMR sub-regions with the specified ordering. Red bars are consistent with a decrease in methylation over the course of development, and bright blue bars are consistent with an increase in methylation over development. The proportion of total base-pairs in DMRs accounted for by each category is indicated.

We also consider three types of sequence-based landmarks: (i) CpG islands, annotated in the UCSC genome browser. Evidence suggests that methylation in these regions affects gene expression [Bell et al., 2011, Illingworth and Bird, 2009]. (ii) CpG island shores, defined as the 2 kb flanking regions of islands. Using BSmooth, Irizarry et al. [2009] found that DMRs between colon cancer and healthy colon cells tend to be located in CpG island shores. (iii) The 2 kb flanking regions of transcription start sites (TSSs), annotated in the Gencode project [Harrow et al., 2012] and available from ENCODE annotations. These 2 kb flanking regions serve as proxies for promoter regions, which are typically located immediately upstream of the TSS.

We tested whether the number of detected DMRs overlapping each genomic feature differed from what would be expected by chance. To do this, we compare the overlap observed between our set of DMRs and each genomic feature with the overlap that we might expect if none of the DMRs contain true associations. Determining this expected "null" overlap is a bit subtle — We cannot simply assume that false discoveries are uniformly distributed over the genome. JADE output may depend on aspects of the data such as measurement density, read depth, or average methylation, which might also be correlated with genomic features of interest. For example, we might be more likely to make false discoveries in areas with moderate methylation levels than in areas with methylation proportions close to 0 or 100%.

Instead, we conducted three "null" analyses, one for each cell type. In each of these analyses, we applied JADE to the data from a single cell type, treating each technical replicate as a separate group. The segments detected in these null analyses can be used to estimate the amount of overlap with a genomic feature that one might expect due to chance. We combined the three sets of "null" DMRs detected across the three cell types, and used a Fisher's exact test to compare the proportion of detected DMRs overlapping each genomic feature to the proportion observed in the null analyses. These results are shown in Table 2.1. We found that the DMRs identified by JADE are enriched in TSS flanking regions but did not find significant enrichment in the other annotation classes.

Next, we restricted our analysis to the sub-regions of DMRs detected by JADE that are

Table 2.1: Overlap between detected DMRs and genomic features, for the methylation data analysis discussed in Section 2.6.3. 'Total DMRs' is the number of DMRs that overlap each genomic feature. 'Fold' is the ratio of the observed number of overlapping DMRs to the number that would be expected by chance. 'P-value' is the p-value based on a Fisher's exact test comparing the proportion of DMRs overlapping the genomic feature to the proportion expected to occur by chance.

Genomic Feature	Total (N=220)	Fold	P-Value
Cpg Islands	119 (54.1%)	1.13	0.25
CpG Island Shores	70 (31.8%)	0.99	1
Transcription Start Sites	112 (50.9%)	1.32	0.011
TF Binding Sites	25 (11.4%)	1.03	1
DNase I HS Sites	95 (43.2%)	1.04	0.77
H3K27ac Modifications	36 (16.4%)	0.77	0.22

consistent with increasing methylation over the course of development (Myoblast≤Myotube≤Mature) or decreasing methylation over the course of development (Myoblast≥Myotube≥Mature). These two groups account for approximately 58% of all base-pairs in detected DMRs, as discussed in Section 2.6.3. The results can be found in Table 2.2. We found that the rate of overlap with CpG islands is higher in loss-of-methylation than in gain-of-methylation DMR sub-regions. This pattern is consistent with prior suggestions that demethylation plays a major role in up-regulating cell type specific gene expression over the course of development [Segalés et al., 2014, Hupkes et al., 2011]. We also found that the rates of overlap with both DNase-I hypersensitive sites and H3K27ac modifications are higher in gain-of-methylation than in loss-of-methylation DMR sub-regions.

2.7 Discussion

We have described JADE, a flexible method for the analysis of genomic phenotypes measured in two or more conditions. JADE combines smoothing and group comparison into a single optimization problem, resulting in improved power over competing methods.

In addition to gains in power for simple comparisons of two groups, JADE offers a novel approach for analyzing data with respect to a categorical outcome. Although the BSmooth and methylKit frameworks could be extended to categorical outcomes by using a one-way ANOVA, categorical outcomes are typically analyzed by performing pairwise comparisons, as in Carrió et al. [2015], or one-versus-all comparisons. For example, tissue-specific DMRs have been identified by finding regions for which one cell type differs from the average over all other cell types [Irizarry et al., 2009, The Encode Project Consortium, 2012]. JADE is able to identify DMRs across categorical outcomes, and determine the order and grouping of profiles within DMRs.

JADE is implemented as an `R` package `jadeTF`, currently available at the author's website https://github.com/jean997/jadeTF. This work also appears in Morrison et al. [2016].

Table 2.2: Overlap between gain-of-methylation and loss-of-methylation DMR sub-regions and genomic features, for the methylation data analysis described in Section 2.6.3. 'Total', 'Loss', and 'Gain' are the number of DMR sub-regions, loss-of-methylation DMR sub-regions, and gain-of-methylation DMR sub-regions that overlap each genomic feature. 'P-value' is the *p*-value based on a Fisher's exact test comparing whether the proportion of loss-of-methylation DMRs overlapping a genomic feature equals the proportion of gain-of-methylation DMRs overlapping the genomic feature. Note that the counts in the 'Total' column differ from those in Table 2.1 because here we are considering sub-regions rather than full DMRs. Furthermore, the 'Loss' and 'Gain' columns do not sum to equal the 'Total' column because only 58% of DMR sub-regions can be characterized as either loss-of-methylation or gain-of-methylation.

Genomic Feature	Total (N=380)	Loss (N=176)	Gain (N=46)	P-Value
CpG Islands	201 (52.9%)	100 (56.8%)	12 (26.1%)	$2.3 \cdot 10^{-4}$
CpG Island Shores	89 (23.4%)	36 (20.5%)	14 (30.4%)	0.17
Transcription Start Sites	186 (48.9%)	82 (46.6%)	23 (50%)	0.74
TF Binding Sites	24 (6.3%)	13 (7.4%)	6 (13%)	0.24
DNase I HS Sites	120 (31.6%)	52 (29.5%)	22 (47.8%)	0.022
H3K27ac Modifications	43 (11.3%)	18 (10.2%)	10 (21.7%)	0.046

Chapter 3

AN EXCURSION PROCEDURE FOR GENOMIC PHENOTYPES

3.1 Overview

In this chapter we describe a new proposal for identifying differential regions with genomic phenotypes, which we refer to as a *flexible robust excursion test* (FRET). Like JADE, FRET learns region boundaries adaptively, but unlike JADE, permits control of the region-wise false discovery rate.

Recall from Section 1.2 that, for each of n samples, we observe genomic phenotype $\mathbf{y}_i = (y_{i1}, \dots, y_{ip})^\top \in \mathbb{R}^p$ and a trait $\mathbf{x}_i \in \mathbb{R}^q$ where $i \in 1, \dots, n$ indexes the sample. We model the relationship between y_{ij} and $\mathbf{x}_i$ as

$$y_{ij} = \alpha_j + \boldsymbol{\beta}_j^\top \mathbf{x}_i + \epsilon_{ij}. \tag{3.1}$$

where $E[\epsilon_{ij}] = 0$. We are interested in using these data to identify *differential regions*, or blocks of contiguous positions for which $\boldsymbol{\beta}_j \neq 0$. In our discussion of JADE in Chapter 2, we considered only categorical traits. For this discussion, we do not need this limitation. The trait, $\mathbf{x}_i$, might be binary, categorical, or continuous. The idea behind FRET is to calculate a statistic at each position testing the null hypothesis that $\boldsymbol{\beta}_j = 0$ and scan for regions with large test statistics.

FRET is an adaptation of the procedures discussed by Siegmund et al. [2011] and Chouldechova [2014], which we refer to as excursion procedures. The simplest variation of an excursion procedure is described in the context of our problem in Algorithm 2 and can be briefly stated in three steps: 1) Calculate a test statistic $T(s_j)$ at each position; 2) Smooth the statistics — let $\tilde{T}(s_j)$ denote the smoothed statistic at s_j; 3) Using a threshold,

z, define differential regions as follows: Define the excursion set at threshold z as

$$\mathcal{E}_z = \left\{ s_j : |\tilde{T}(s_j)| \geq z \right\}. \tag{3.2}$$

Group the elements of $\mathcal{E}_z$ into sets of contiguous elements or *regions.* The set of regions, denoted $\mathcal{R}_z$, is the set of discoveries made by the procedure.

This procedure is illustrated in Figure 3.1. Here we have randomly generated normally distributed "statistics" at 250 locations. In the white areas, the statistics are sampled from a $N(0, 1)$ distribution, while in the blue "signal" regions they were sampled from a $N(1.5, 1)$ distribution. In the middle panel, we smooth the statistics using a 10 unit moving average. The only requirement of the smoothing method is that, for any s, $\tilde{T}(s)$ only depends on $T(s_j)$ for s_j in a small window around s. In Section 3.2.2 we will discuss this requirement and the choice of smoother in greater detail. Note that we need not use a test statistic that is mean 0 under the null. Any statistic that has a larger expectation under the alternative than under the null is acceptable.

In the right hand panel, orange shading shows the places where $\tilde{T}$ exceeds the threshold, z, which is marked by horizontal dashed lines (in some places orange shading overlaps blue shading). In this example, the usual point-wise false discovery proportion — the fraction of points shaded orange that are not contained in true signal regions — is 5/36 or about 14%. Since we are interested in signal regions rather than individual points, it is more relevant to measure the proportion of discovered regions that fail to contain true signal; in this case 1/4. In Section 3.2.1, we will formally define the notion of a *false discovery* for regions and the *region-wise false discovery rate* (rFDR).

The region-wise false discovery rate of an excursion procedure is controlled by the threshold. In Section 3.2.2, we review the key theorem of Siegmund et al. [2011] that provides a method for choosing the threshold that controls the rFDR for excursion procedures. In Section 3.2.3, we review the proposal of Chouldechova [2014] for merging closely spaced elements of $\mathcal{R}_z$.

Excursion procedures are very similar to the BSmooth method proposed by Hansen et al.

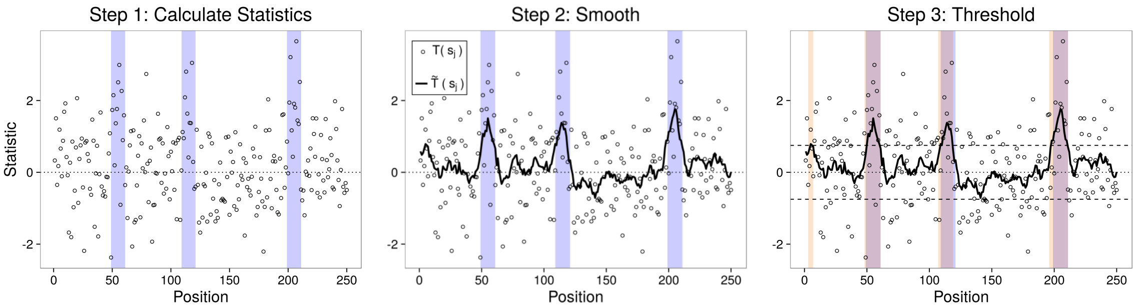


Figure 3.1: Illustration of an excursion procedure. *Left:* Normal statistics at 250 locations. Blue shading marks three regions where the mean of the statistics is not 0. *Middle:* Statistics are smoothed with a 10 unit moving average. *Right:* Positions in the excursion set defined in (3.2) are identified and shaded orange. These are clumped into four sets of contiguous positions. These clumps are the differential regions discovered by the procedure. Here the threshold used is $z = 0.75$ — horizontal lines mark $\pm z$. Three of the regions overlap true signal and one region is a false discovery.

Algorithm 2 Excursion Procedure: Adapted from Chouldechova [2014], Siegmund et al. [2011]

Given a threshold, z,

1. For j in $1, \ldots, p$, calculate a statistic $T(s_j)$ testing the hypothesis $\boldsymbol{\beta}_j = 0$ in (1.1).

2. Smooth the statistics. Let $\tilde{T}(s_j)$ denote the value of the smoothed statistic at s_j.

3. Retrieve the excursion set at threshold z,

$$\mathcal{E}_z = \left\{ s_j : |\tilde{T}(s_j)| \geq z \right\},$$

Group contiguous elements of $\mathcal{E}_z$ into regions. Denote this set of regions $\mathcal{R}_z$.

$\mathcal{R}_z$ is the set of differential regions at threshold z.

[2012]. BSmooth first smooths the phenotype for each sample and then calculates a test statistic at each position using the smoothed data while excursion procedures first test at each point and then smooth the test statistics. We find, in simulations that, if the same test statistic and smoothing method are used, the two strategies give very similar performance in many circumstances. The excursion procedure approach seems to work better when the variance of the phenotypes is much higher inside of differential regions (See Section B.1 of Appendix B). We found in Chapter 2, that JADE is more powerful than smoothing followed by point-wise testing in a variety of settings. There are two reasons why we might prefer an excursion procedure over JADE in the face of these results: 1) Using an excursion procedure, we are able to control the rFDR; and 2) An excursion procedure provides flexibility in the choice of test statistic while JADE is limited to using squared error loss. We discuss the choice of test statistic at greater length in Section 3.3.

Our proposal, FRET, is described in detail in Section 3.4. We make two contributions in adapting excursion procedures for use with genomic phenotypes. First, we propose the use of a robust test statistic obtained through Huber regression [Huber, 1973] in the first step of Algorithm 2. Biological data often show high levels of heterogeneity within levels of the trait of interest. This occurs because many other (potentially unmeasured) factors can affect the measurement. In Section 3.3 we review the robust test statistics introduced by Huber [1973] and motivate their use for biological data. We show that, for right skewed or highly variable data, the robust Huber statistic provides a dramatic power gain over the negative binomial or over-dispersed Poisson models that have previously been explored in the context of genomic phenotype analysis (e.g., in DEseq2 [Love et al., 2014] and edgeR [Robinson et al., 2009, McCarthy et al., 2012]).

Second, rather than using a single, constant threshold on the entire genome as in Algorithm 2, we permit different thresholds at different positions. In Section 3.4.3 we describe a method for choosing these thresholds that controls the region-wise false discovery rate. We find that this modification can improve power dramatically as well as simplifying computation for genome sized problems.

In Section 3.5 we explore the question of when it is better to use a *fixed-region test* — a strategy that aggregates and tests within pre-defined windows — than to use FRET. Recall from the discussion in Section 1.3 that, for several genomic phenotypes, the boundaries of functional genomic elements can be guessed by looking at the patterns in individual samples (e.g. peak calling) or can be defined using existing annotations. We ask how good the boundaries of pre-defined regions must be for it to be more advantageous to use a fixed-region test than to adaptively define region boundaries using FRET. Interestingly, we find that FRET performs as well as fixed-regions tests with good boundaries in many settings.

3.2 Background: Controlling the region-wise false discovery rate for excursion procedures

We now review the results of Siegmund et al. [2011] and Chouldechova [2014] that form the basis of FRET. In Section 3.2.1 we introduce notation and formally define the region-wise false discovery rate. In Section 3.2.2 we state the key theorem from Siegmund et al. [2011] that allows us to select the threshold value for an excursion procedure to control the region-wise false discovery rate. In Section 3.2.3 we describe the procedure for merging nearby discoveries proposed by Chouldechova [2014].

3.2.1 Region-wise false discovery rate

Here we formally describe the general spatial inference problem considered by Siegmund et al. [2011] and Chouldechova [2014]. Our problem of identifying differential regions in which a genomic phenotype is associated with a trait is a special case of this problem.

Let $D = \{s_1, \dots, s_p\}$ be a subset of $\mathbb{R}$. Let $H_0(s_j)$ be a null hypothesis that is either true or false at each point in D. In the genomic phenotype problem, $H_0(s_j)$ is that $\boldsymbol{\beta}_j = 0$ in the model in (3.1). We partition D into two subsets: the subset of null points, $D_0 = \{s_j \in D : H_0(s_j) \text{ is true}\}$ and the subset of non-null points $D_1 = D \setminus D_0$.

We assume that D_1 contains clumps of contiguous elements, or *regions*, and that these regions are the unit of scientific interest. We therefore focus on statistical procedures that return a set of regions in D on which the null hypothesis is rejected rather than a set of individual points. For example, in Figure 3.1, D_1 is the set of positions shaded blue, which make up three signal regions. In the right hand panel, the excursion procedure has rejected the null hypothesis at positions shaded in orange. In the context of this problem, it is more sensible to view these discoveries as four regions than as 36 individual points. To measure the accuracy of procedures that identify regions rather than points we introduce several definitions.

A region $C \subset D$ is defined to be a *false discovery* if $C \cap D_1 = \emptyset$ and a *true discovery* otherwise. Assume that we have some procedure that rejects the null hypothesis on a set of regions in D. We will refer to regions on which the null hypothesis is rejected as *dscoveries.* Let R be the total number of subsets of D on which the null hypothesis is rejected. Let V be the number of these that are false discoveries and S the number of true discoveries $(V + S = R)$. We define the *region-wise false discovery proportion* as V/R if $R > 0$ and 0 otherwise. We define the *region-wise false discovery rate* (rFDR) as $E[V/R; R > 0]$. This definition is used by Siegmund et al. [2011] and Chouldechova [2014] as well as other authors including Benjamini and Heller [2007] and Perone Pacifico et al. [2004]. For excursion procedures, we will use the notation V_z, S_z, and R_z to denote the number of false, true, and total discoveries made at threshold z.

When the scientific interest lies in regions rather than points, the rFDR is the most appropriate way to measure the false discovery rate. However, we note that, unlike the point-wise false discovery rate, the rFDR can allow a procedure to “cheat” — that is, it is possible to produce an undesirable result with a misleadingly low rFDR. For example, if a procedure selects a single discovery containing all of D, the rFDR is 0 if there is any signal at all in D. Excursion procedures avoid this pitfall by placing a fairly high lower bound on the thresholds considered, preventing large regions from being selected. This lower bound is discussed in further in Section 3.2.2.

A procedure could also cheat by breaking true signal regions into many small fragments. This type of bad behavior is easy to avoid since it requires systematically treating true discoveries differently from false discoveries. We employ the strategy of merging nearby discoveries described in Section 3.2.3 which reduces the risk of achieving a misleadingly low rFDR due to over-fragmentation of true discoveries. The fact that this modification improves the power of excursion procedures suggests that, without merging, excursion procedures tend to fragment false discoveries to a greater degree than true discoveries.

3.2.2 Threshold selection: Results of Siegmund et al. [2011]

Siegmund et al. [2011] establish results that provide a method for selecting the threshold for an excursion procedure that controls the false discovery rate. Here we review these results.

Recall that a *region* is defined as a set of contiguous elements of D. Let $\mathcal{P}$ be a class of statistical procedures each of which rejects the null hypothesis on a set of regions in D. Suppose that, for a particular data generating mechanism, the elements of $\mathcal{P}$ can be uniquely indexed by λ, the expected number of false discoveries, and that λ is bounded above in $\mathcal{P}$ by some finite value λ^*. We will denote the elements of $\mathcal{P}$ as P_λ for $\lambda \in [0, \lambda^*]$. Let R_λ be the total number of regions returned by P_λ applied to a particular set of input data, V_λ the number of these that are false discoveries and S_λ the number that are true discoveries.

The following result of Siegmund et al. [2011] allows us to select an element of $\mathcal{P}$ while guaranteeing control of the false discovery rate:

Theorem 3.2.1 (Siegmund et al. [2011] Theorem 2) *Define*

$$\Lambda = \max \left\{ \lambda : \lambda \leq \lambda^*, \lambda / R_\lambda \leq \alpha \right\}. \tag{3.3}$$

If

1. *V_λ is Poisson with expectation λ and*

2. *V_λ and S_λ are independent for all $\lambda \in [0, \lambda^*]$,*

then $E[V_\Lambda / R_\Lambda; R_\Lambda > 0] \leq \alpha$.

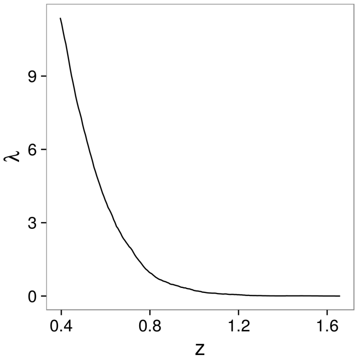


Figure 3.2: Relationship between the threshold, z, and $\lambda(z)$ for the example in Figure 3.1. For each value of z, we calculate $\lambda(z)$ by repeatedly sampling statistics from the data generating mechanism, applying the excursion procedure, and counting the number of false discoveries.

The class of excursion procedures described in Section 3.1 is naturally indexed by the threshold, z, rather than λ. Figure 3.2, shows the relationship between z, and the expected number of false discoveries, $\lambda(z) = E[V_z]$, for the data generating mechanism used to produce the example in Figure 3.1. This relationship is the connection between excursion procedures and the result of Theorem 3.2.1.

If the relationship between z and $\lambda(z) = E[V_z]$, is known and one-to-one, we can re-index the class of excursion procedures by λ, select Λ as in (3.3) and apply an inverse transformation to obtain the threshold $\tilde{z} = \lambda^{-1}(\Lambda)$. If $\lambda(z)$ is monotonic (as it must be if it is one-to-one), we can re-write (3.3) in terms of the threshold as

$$\tilde{z} = \min\{z : z \geq z^*, \lambda(z)/R_z \leq \alpha\} \tag{3.4}$$

where z^* is a fixed lower bound. If assumptions 1 and 2 hold, then Theorem 3.2.1 allows us to conclude that, using this procedure, $E[V_{\tilde{z}}/R_{\tilde{z}}; R_\Lambda > 0] \leq \alpha$.

One way to guarantee that the correspondence between z and $\lambda(z)$ is one-to-one and monotonic is to ensure that the condition that $R_z < R_{z'}$ for all $z < z'$ is met. This effect is achieved by the merging procedure suggested by Chouldechova [2014] discussed Section 3.2.3.

In practice, for a given value of z, $\lambda(z)$ is unknown and must be estimated. For the genomic phenotype problem, we can estimate $\lambda(z)$ using the permutation procedure described in Section 3.4.2. This estimate will be unbiased if $D_1 = \emptyset$. If $D_1 \neq \emptyset$, the estimate given by the permutation procedure is an over-estimate of $\lambda(z)$. Thus, if we replace $\lambda(z)$ by its estimate in (3.4), the selected threshold will be conservative.

We now address the two assumptions of Theorem 3.2.1. The assumption that $V_\lambda \sim$ Poisson(λ) is approximately met for excursion procedures when λ is small and z is large. This is a consequence of the established result that, for most common statistics, the number of excursions of $\tilde{T}$ above a fixed, large threshold on D_0 is approximately Poisson [Aldous, 1989, Arratia et al., 1989, 1990]. This result holds in a wide variety of circumstances including those in which $\{T(s_j)\}_{s_j \in D_0}$ are asymptotically Gaussian or χ^2 distributed. This approximation continues to hold even if there is local dependence between test statistics. For more details on this approximation, we refer the interested reader to Aldous [1989] for many examples as well as results on the size and distribution of clumps of excursions. Meeting this assumption requires that z^* is selected to be large enough that the approximation holds. In practice we find that using a large percentile (e.g., the 90th) of $\left\{\tilde{T}(s_j)\right\}$ works well.

Independence between V_z and S_z requires that the elements of $\left\{\tilde{T}(s_j)\right\}_{s_j \in D_0}$ and $\left\{\tilde{T}(s_j)\right\}_{s_j \in D_1}$ be independent or nearly independent. Any smoother will induce some dependence between these sets at the boundaries between D_0 and D_1. Let $\tilde{D}_1 \supset D_1$ be the *smoothed signal set* — the subset of D on which $|E[\tilde{T}(s_j)]| \geq |E[T(s_j)|H_0(s_j)]|$. In order to achieve independence between S_z and V_z, the chance of false discoveries occurring on $\tilde{D}_1 \cap D_0$ must be very low. This is achieved if D_1 is sparse, there is no long range dependence in the data, and the smoothed statistics only depend locally on on the un-smoothed statistics . Smoothing methods that meet this criterion include kernel smoothers with short bandwidths. Siegmund et al. [2011] point out that, in practice, nearly overlapping or very long discoveries are warning signs that these assumptions are violated.

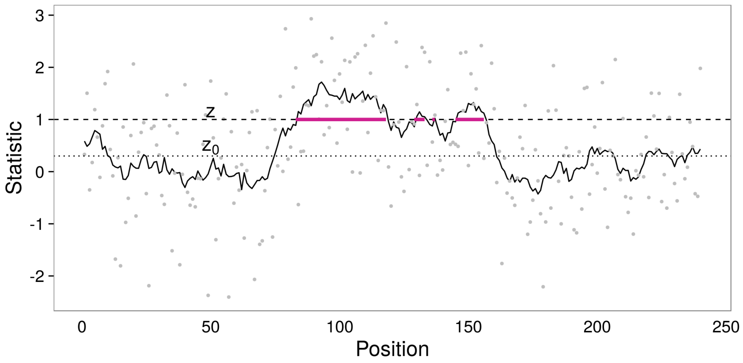


Figure 3.3: Motivation for the merging procedure described in Section 3.2.3. Normally distributed statistics (points) are generated at 250 positions. Statistics are smoothed with a 10 unit moving average (solid line). At the threshold, z, marked by the dashed line, there are four closely spaced discoveries. These can be combined into one discovery by merging with respect to z_0.

3.2.3 Merging closely spaced discoveries: Proposal of Chouldechova [2014]

Chouldechova [2014] proposes adding an additional post-processing step to Algorithm 2. In this step, we merge very close elements of $\mathcal{R}_z$. Their proposed merging method is motivated by the observation that $\tilde{T}(s_j)$ sometimes crosses the threshold z several times in short succession leading to short, closely spaced discoveries. An example of this behavior can be seen in Figure 3.3. If this occurs in a true signal region, merging these discoveries produces more desirable results. In a null region, merging reduces the number of false discoveries, allowing us to choose a lower threshold while keeping the false discovery rate below a fixed level. Thus, adding an additional merging step to Algorithm 2 can improve the power of an excursion procedure.

Chouldechova [2014] proposes selecting a “reference level” z_0, less than the smallest threshold considered, z^*. Let $\mathcal{E}_{z_0}$ be the excursion set at z_0 and let $\mathcal{R}_{z_0}$ be the set of regions in $\mathcal{E}_{z_0}$. For a given threshold, z, let $\mathcal{R}_z = \{C_1, \dots, C_{R_z}\}$ be the set of regions in $\mathcal{E}_z$. We merge C_i and C_j if there is an element $B \in \mathcal{R}_{z_0}$ such that $C_i \cap B \neq \emptyset$ and $C_j \cap B \neq \emptyset$. We denote the resulting set of merged discoveries $\mathcal{R}_z^{(z_0)}$. These steps are shown in Algorithm 3.

Merging tends to improve power because true discoveries tend to have higher test statistics than false discoveries. Excursions marking false discoveries will be closer to the threshold resulting in more short range crossings. Thus, most of our merging will prevent over-counting of closely spaced false discoveries. Chouldechova [2014] find that performance of this procedure is insensitive to the choice of z_0 but proposes $z_0 = 0.3 \cdot z^*$ based on simulation results.

Merging has the added advantage of conferring a highly desirable property on our procedure: If a region is detected at threshold z, it will not be fragmented into multiple regions at some higher threshold $z' > z$. This guarantees that $\lambda(z)$ is a monotonically decreasing function of z.

Chouldechova [2014] does not formally show that the threshold selection procedure (3.4) implied by Theorem 3.2.1 remains valid when merging is used. However, they note that the procedure is related to *semi-local maxima*, described by [Aldous, 1989, §A7, §C26]. In each merged region, the highest point is a semi-local maximum above z in the terminology of Aldous [1989]. Aldous [1989] claims that the number of semi-local maxima above a given threshold is well approximated by a Poisson distribution. This is the necessary requirement for the procedure in (3.4) to control the rFDR.

Algorithm 3 Merging procedure of Chouldechova [2014]

```
Given a threshold, $z$, and a merging reference level $z_0$,

  1. Using Algorithm 2, obtain $\mathcal{R}_{z_0}$ and $\mathcal{R}_z$.

  2. For each $B \in \mathcal{R}_{z_0}$, merge all pairs $C, C' \in \mathcal{R}_z$ with $C \cap B \neq \emptyset$ and $C' \cap B \neq \emptyset$.

Return the set of merged discoveries, denoted $\mathcal{R}_z^{(z_0)}$.
```

3.3 Background: Robust test statistics

The first step of an excursion procedure is to calculate a statistic testing the null hypothesis at each position. In this section, we provide background for the choice of test statistic used in FRET. Here, we focus only the problem of testing $\boldsymbol{\beta}_j = 0$ at a single position, disregarding, for the moment, data at surrounding sites.

One of the biggest challenges in testing for associations with genomic phenotypes (with either a fixed-region test or an adaptive test) is handling high levels of variability. Samples with the same or similar values of $\mathbf{x}$ may display very different patterns in their genomic phenotypes as a result of underlying biological heterogeneity. Since most genomic phenotypes are count data or functions of count data, the distribution of phenotype values at a single locus or within a region can have a very heavy right tail and a very high variance. Although this variance can be estimated at every locus or for every region, doing so can reduce power and, with small sample sizes, make the resulting estimator highly variable.

Many authors have proposed tests based on generalized linear models. For example, DESeq2 and edgeR, which are both fixed region tests, use negative binomial models and use empirical-Bayes methods to estimate the dispersion parameter for each region. One danger of this approach is that the variance model may not be flexible enough to accommodate the patterns seen in the data. We discuss this issue at greater length in Chapter 4. We propose that a robust Huber statistic [Huber, 1973], a simple test that essentially down-weights very large observations, is a good alternative.

In Section 3.3.1, we review robust regression estimates proposed by Huber [1973] and their associated test statistics. In Section 3.3.2 we briefly describe methods for computing these statistics. In Section 3.3.3 we present several numerical experiments comparing the Huber statistic to statistics based on linear regression or generalized linear models for a range of scenarios with similar qualities to what we expect from biological data. In Section 3.3.4, we briefly describe a modification of the statistics in Section 3.3.1, suggested by Tusher et al. [2001], that prevents very large test statistics at positions or regions with very low variance.

3.3.1 Robust regression estimates

The usual procedure for testing $\boldsymbol{\beta}_j = 0$ in (3.1) is to first estimate α_j and $\boldsymbol{\beta}_j$ by minimizing the squared error between the observed and fitted values:

$$\left(\hat{\alpha}_j, \hat{\boldsymbol{\beta}}_j\right) = \underset{\alpha_j, \boldsymbol{\beta}_j}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_{ij} - \alpha_j - \boldsymbol{\beta}_j^\top \mathbf{x}_i\right)^2. \tag{3.5}$$

Let $\mathbf{X}$ be the $n \times q + 1$ design matrix with ith row equal to $(1, \mathbf{x}_i^\top)$ and $\mathbf{Y}_j = (y_{1j}, \ldots, y_{nj})^\top$. Solving (3.5) gives the familiar estimator

$$\begin{pmatrix} \hat{\alpha}_j \\ \hat{\boldsymbol{\beta}}_j \end{pmatrix} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y}_j.$$

We then estimate the variance of the coefficient estimates, for example using a Huber-White estimator [Huber, 1967, White, 1980]. Let $\hat{\mathbf{V}}_j$ be an estimate of the variance of $\hat{\boldsymbol{\beta}}_j$ in (3.5). In the case of $q = 1$, we can construct a test statistic $T^{LS}(s_j) = \hat{\beta}_j / \sqrt{\hat{V}_j}$. Here we use the super-script LS to indicate that the test statistic is based on least-squares estimates. This statistic is asymptotically normal with variance 1 and, under the null, mean 0. When $q > 1$ the joint hypothesis $\boldsymbol{\beta}_j = 0$ can be tested with a quadratic form test, for example, a Wald style test

$$\hat{\boldsymbol{\beta}}_j^\top \hat{\mathbf{V}}_j^{-1} \hat{\boldsymbol{\beta}}_j,$$

which asymptotically has a χ_q^2 distribution.

Hampel et al. [1986] as well as many others in the field of robust statistics point out several flaws with least squares regression estimates. When ϵ_{ij} in (3.1) are normal, the estimators in (3.5) are maximum likelihood estimates and have optimal efficiency among all other estimators. Under departures from normality, the least-squares estimates remain consistent but lose efficiency relative to other choices. Additionally, least-squares estimates are highly sensitive to outliers and leverage points — i.e. a single large observation can move the estimates a great deal.

These problems were first considered in the context of robustness to outlying observations and erroneous measurements. In biology, heterogeneity within strata of the trait of interest can give the same effect as sporadic outlying observations. Consider a simple example: Observations of a genomic variable are made in two groups at a single locus. These observations are distributed as $y_i \sim \text{Poisson}(\gamma_i)$ where $\gamma_i = 3 + x_i$ with probability $1 - \varepsilon$ and 15 with probability ε. Here $x_i \in \{0, 1\}$ encodes the group membership of sample i. If $\varepsilon = 0$, the least squares estimator has 74% power at a significance threshold of 0.05 with 50 observations in each group. When $\varepsilon = 0.05$ however, that power drops to 31%.

Huber [1964] introduced a class of estimators for a location parameter called M-estimators. These were extended to linear regression by Huber [1973] and can be expressed in the context of our problem as

$$\left(\hat{\alpha}_j, \hat{\boldsymbol{\beta}}_j\right) = \underset{\alpha_j, \boldsymbol{\beta}_j}{\operatorname{argmin}} \sum_{i=1}^{n} \rho\left(\frac{y_{ij} - \alpha_j + \boldsymbol{\beta}_j \mathbf{x}_i}{\sigma_j}\right). \tag{3.6}$$

Here σ_j is a scale parameter and ρ is a non-negative loss function with a global minimum at $\rho(0) = 0$. This class of estimators contains the least squares estimate as well as other likelihood based estimators by taking $\rho(u) = -\log f(u)$ for some density function f.

Modifying ρ results in estimators with different robustness properties. Informally, robustness can be thought of as the amount of influence a single observation can exert on the estimates. The choice of the L_1 loss function, $\rho_{L_1}(u) = |u|$, gives the median regression estimate and, like the median, is highly robust but not very efficient. Huber [1964] proposes a compromise between the median and the least squares estimates:

$$\rho_c(u) = \begin{cases} \frac{1}{2}u^2 & |u| < c \\ c\left(|u| - \frac{1}{2}c\right) & \text{otherwise} \end{cases}. \tag{3.7}$$

Figure 3.4 compares the usual squared error loss, the L_1 median loss, and the Huber loss ρ_c with c taken to be 1. The Huber loss behaves like the squared error loss for small residual values and like the median loss for large values. This permits the estimators obtained to be both efficient and robust.

We will denote estimators found by solving (3.6) with the Huber loss in (3.7) as $\hat{\beta}_j^H$ and

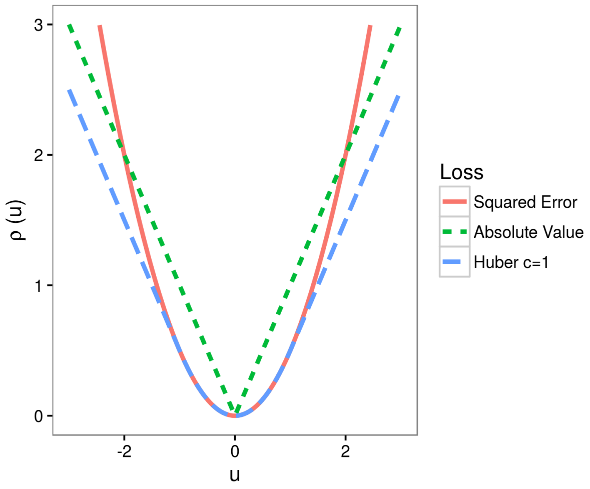


Figure 3.4: Squared error, L_1, and Huber Loss functions

their variance estimates $\hat{V}_j^H$. We can obtain a robust version of $T^{LS}(s_j)$ as

$$T^H(s_j) = \frac{\hat{\beta}_j^H}{\sqrt{\hat{V}_j^H}} \tag{3.8}$$

when $q = 1$ or, when $q > 1$, using a robust analog of the Wald test

$$T^H(s_j) = \hat{\boldsymbol{\beta}}_j^{H,\top} \hat{\mathbf{V}}_j^{H,-1} \hat{\boldsymbol{\beta}}_j^H \tag{3.9}$$

3.3.2 Computation of the Huber Statistic

In order to solve (3.6) with ρ as in (3.7), we must choose c and calculate the scale parameter σ_j. For the constant c, we take the suggestion of Huber [1964] who argues that $c = 1.345$ is a good choice. This constant gives an estimator that is 95% as efficient as the least squares estimator if the error terms are normally distributed.

Estimation of the scale parameter is more challenging. For the least-squares estimators, $\rho(u) = \frac{1}{2}u^2$. In this case the scale parameter is unnecessary for solving (3.6), allowing us to omit it in (3.5). Unfortunately, this is not the case for a general choice of ρ. Huber [1981, Chap. 7.8] describe a method for iterating estimation of σ_j and the regression coefficients, at each step solving a weighted least squares problem. There are several existing implementations of solutions to this problem. We use the `rlm` function available in the `R` package `MASS` [Venables and Ripley, 2002]. This function also provides an estimate of the variance of the coefficient estimates consistent with suggestions made by Huber [1981].

3.3.3 Comparison of test statistics for heterogeneous count data

The statistic that gives the best power for a fixed level of type 1 error in any given situation depends on the underlying data distribution. Here, we use a few numerical experiments to compare several alternatives — over-dispersed Poisson (quasi-Poisson), negative binomial, least squares, and Huber statistics — and describe the settings in which each is preferable. We find that the Huber statistic is the most powerful of the four tests in settings that are similar to what we expect in biological data. In these experiments we compare the type 1 error and power of each method for detecting an association at a single locus. More complex simulations examining region level detections are presented in Section 3.5.

Consider a model in which the number of counts observed for individual i (at a single locus) is Poisson

$$y_i \sim \text{Poisson}(\gamma_i) \tag{3.10}$$

and γ_i depends on a binary trait $x_i \in \{0, 1\}$. We consider four scenarios for γ_i:

1. No heterogeneity within groups: $\gamma_i = 3 + x_i$.

2. Rare outliers in both groups:

$$\begin{aligned} w_i \sim& \text{Bernouli}(0.05) \\ \gamma_i =& (1 - w_i)(3 + x_i) + w_i \cdot 15 \end{aligned}$$

3. The mean for the bulk of observations is the same in both groups, but outliers are

more common in one group:

$$\begin{aligned} w_i \sim& \text{Bernouli}(0.05 + 0.05 \cdot x_i) \\ \gamma_i =& (1 - w_i) \cdot 3 + w_i \cdot 12 \end{aligned}$$

.

4. Multinomial: γ_i takes the values 1.5, 3, 7, or 15 with probabilities:

	$x_i = 0$	$x_i = 1$
1.5	0.55	0.2
3	0.3	0.45
7	0.1	0.25
15	0.05	0.1

We consider four test statistics for comparing the means in two groups:

1. The least squares statistic in Section 3.3.1;
2. The Huber statistic in (3.8);
3. The Wald test statistic obtained from quasi-Poisson regression;
4. The Wald test statistic obtained from negative binomial regression.

We include the two statistics based on general linear models (statistics 3 and 4 above) because similar approaches are often used for count valued genomic phenotypes. In these experiments we fit the quasi-Poisson and negative binomial models using implementations in `R`. The quasi-Poisson model is fit using the `glm` function and the negative binomial model is fit using the `glm.nb` function in the package `MASS` [Venables and Ripley, 2002].

For each of the four data generating mechanisms, we conducted two experiments: one to measure the type 1 error and the second to measure power. To measure the type 1 error,

Table 3.1: Numerical experiments comparing test statistics under four hierarchical Poisson models described in Section 3.3.3. Results are averaged of 500 simulations. Type 1 error at a threshold of 0.05 is estimated by simulating data for 100 samples with a trait value of 0 and assigning these samples to two groups. Power at a threshold of 0.05, is estimated by comparing 50 samples with a trait value of 0 and 50 samples with a trait value of 1.

Setting		Quasi-Poisson	Neg. Binomial	Least Squares	Huber
1	T1E	0.052	0.046	0.052	0.062
	Power	0.772	0.764	0.772	0.786
2	T1E	0.068	0.118	0.060	0.074
	Power	0.306	0.386	0.306	0.658
3	T1E	0.038	0.078	0.034	0.060
	Power	0.108	0.180	0.096	0.044
4	T1E	0.050	0.102	0.042	0.054
	Power	0.566	0.630	0.570	0.706

we simulate count data in each setting for 100 individuals with $x_i = 0$ for all $i \in 1, \ldots, 100$. We then assign samples $51, \ldots, 100$ to be labeled as belonging to group 1 and compare the groups using the four tests described above. To measure power, we simulate count data in each setting for 100 samples with $x_i = 0$ for $i \in 1, \ldots 50$ and $x_i = 1$ for $i \in 51, \ldots, 100$ and compare the two groups using each testing method. Results averaged over 500 simulations are shown in Table 3.1.

The Huber estimator has a large power advantage in settings 2 and 4. In these settings, observations farther from the bulk of the data (those for which $\gamma_i = 15$) are less informative about the difference between groups than observations closer to the bulk of the data. The Huber estimator gains power by giving more weight to moderate observations than to the outlying observations.

All four tests perform similarly in setting 1 which has no heterogeneity within levels of x. This suggests that, when the Huber estimator is unnecessary, we lose little by using it. Setting 3 is a worst case scenario for the Huber estimator. In this setting one group has a higher rate of rare large outliers but no association can be detected by looking only at the moderate observations. The Huber statistic limits the influence of the large observations and therefore loses power. Note that the negative binomial statistic has inflated type 1 error under settings 3 and 4 giving it a misleadingly large power.

All test statistics have disadvantages under some data generating mechanisms. The Huber test statistic performs similarly to other methods when data are homogeneous within strata of x but also maintains power under most types of heterogeneity. Therefore, we believe that it is a good choice for genomic phenotype data.

3.3.4 Variance inflation

At some nucleotides we may observe almost no variability. This can lead to extremely small variance estimates and large test statistics. To prevent this behavior, we take the approach of Tusher et al. [2001], adding a small constant σ_0 to the denominator of (3.8) giving the modified statistic

$$T^{H,\sigma_0}(s_j) = \frac{\hat{\beta}_j^H}{\sqrt{\hat{V}_j^H} + \sigma_0}. \tag{3.11}$$

For $q > 1$, we can achieve the same effect by adding σ_0^2 to the diagonal elements of the covariance matrix, $\hat{\mathbf{V}}_j^H$ in(3.9), giving the statistic

$$T^{H,\sigma_0}(s_j) = \hat{\boldsymbol{\beta}}_j^{H,\top} \left(\hat{\mathbf{V}}_j^H + \sigma_0^2 \mathbf{I}_{q\times q} \right)^{-1} \hat{\boldsymbol{\beta}}_j^H, \tag{3.12}$$

where $\mathbf{I}_{q\times q}$ is the $q \times q$ identity matrix. We find that performance of the statistic is fairly insensitive to the choice of σ_0. Tusher et al. [2001] choose σ_0 to minimize the coefficient of variation of the set of median absolute deviations of $\{T^{H,\sigma_0}(s_j)\}$ in bins defined by quantiles of $\left\{\sqrt{\hat{V}_j^H}\right\}$. To reduce computational effort, in our implementation of FRET, we choose σ_0 using only a subset of the p locations.

3.4 FRET: The proposal

In this section we present FRET, our adaptation of excursion procedures for association analysis with genomic phenotypes. The steps of FRET are given in Algorithm 4.

In the first step we calculate the robust Huber statistic described in Section 3.3 at each location. We then smooth the statistics using a moving average. The next step departs from the basic excursion procedure in Algorithm 2. In that procedure, there is a single constant threshold on all of D. In FRET, we divide D into K intervals and allow different thresholds on each interval.

For the basic excursion procedure, the threshold can be chosen to control the false discovery rate using (3.4) in Section 3.2.2. For FRET, we must choose one threshold for each interval. Our procedure for choosing these thresholds is given in Algorithm 6 described in Section 3.4.3.

Once we have chosen a threshold for each interval, we obtain the excursion set on all of D, which is simply the union of the excursion sets in each interval. We group the elements of the excursion set into regions, or subsets of contiguous elements, and apply the merging procedure described in Section 3.2.3 to combine nearby regions. This gives the final set of discoveries. These steps are given explicitly in Algorithm 4.

The most challenging step in Algorithm 4 is step 3, selecting thresholds for each interval. The rest of this section is devoted to the details of this step.

In Section 3.4.1, we provide motivation for using different thresholds in different intervals. In Section 3.4.2 and Algorithm 5, we describe a permutation procedure for estimating $\lambda(z) = E[V_z]$ for the basic excursion procedure with a single threshold. In Section 3.4.3 and Algorithm 6, we make use of Algorithm 5 to construct a method for selecting a different threshold on each of K intervals that controls the rFDR.

Algorithm 4 Flexible Robust Excursion Test

Given a fixed partition of D into intervals $D = \bigcup_{k=1}^{K} E_k$,

1. Calculate $T^{H,\sigma_0}(s_j)$ for $j \in 1, \ldots p$ according to (3.11) if $q = 1$ or (3.12) if $q > 1$, with the variance inflation constant, σ_0, obtained using the procedure of Tusher et al. [2001] described in Section 3.3.4.

2. Smooth the statistics using a moving average with bandwidth equal to the expected size of signal regions to produce $\tilde{T}^{H,\sigma_0}(s_j)$, $j \in 1, \ldots, p$. Set the lower bound z^* to be the 90th percentile of $\left\{\tilde{T}(s_j)\right\}$. Set the merging reference level to be $z_0 = 0.3 \cdot z^*$.

3. Use Algorithm 6 to select a K-vector of thresholds, $\tilde{\mathbf{z}} = (\tilde{z}_1, \ldots, \tilde{z}_K)$. The threshold on E_k is $\tilde{z}_k$.

4. Define the excursion set with respect to the K-vector of thresholds $\tilde{\mathbf{z}}$ as

$$\mathcal{E}_{\tilde{\mathbf{z}}} = \bigcup_{k=1}^{K} \left\{ s_j \in E_k : |\tilde{T}(s_j)| \geq \tilde{z}_k \right\}.$$

   Combine contiguous elements of $\mathcal{E}_{\tilde{\mathbf{z}}}$ into regions. Denote the set of regions $\mathcal{R}_{\tilde{\mathbf{z}}}$.

5. Merge the elements of $\mathcal{R}_{\tilde{\mathbf{z}}}$ relative to z_0 using the procedure in Algorithm 3 to produce the set of merged discoveries $\mathcal{R}_{\tilde{\mathbf{z}}}^{(z_0)}$.

Return $\mathcal{R}_{\tilde{\mathbf{z}}}^{(z_0)}$.

3.4.1 Why have different thresholds?

Allowing the threshold to vary across the genome can improve power in some circumstances. In particular, flexibility is helpful when the rate of null excursions above a fixed threshold, z, is not constant across the genome and signal regions are distributed somewhat uniformly between areas with high and low excursion rates.

In Figure 3.5, we show a toy example to motivate this idea. Normally distributed statistics have been generated at 400 points. The variance of the statistics on the left is much smaller than the variance of the statistics on the right and each half contains one true signal region (shaded in blue) where the sampling distribution for the statistics has a mean larger than 0. In order to detect the signal on the left we must set the threshold lower than many of the "noise" peaks on the right. If we are forced to use a constant threshold over the entire region, we cannot detect signal on the left without incurring many false detections from the noise on the right.

3.4.2 Estimating $\lambda(z)$ using a constant threshold

Here we describe a permutation based method for estimating $\lambda(z)$ for a basic excursion procedure. The steps of this procedure are shown in Algorithm 5. In each repetition, we permute the trait values $\mathbf{x}_1, \dots, \mathbf{x}_n$. We then repeat the excursion procedure, calculating test statistics with the permuted trait values and merging discoveries according to Algorithm 3. We denote the number of discoveries made in repetition l at threshold z as $R_{l,z}$. After a large number of iterations, L, we can estimate $\lambda(z)$ as

$$\hat{\lambda}(z) = \frac{1}{L}\sum_{l=1}^{L} R_{l,z}. \tag{3.13}$$

In order to select a single, constant threshold $\tilde{z}$, that controls the rFDR at level α, we can use (3.4) replacing $\lambda(z)$ with its estimate:

$$\tilde{z} = \min\left\{ z \geq z^* : \hat{\lambda}(z)/R_z \leq \alpha \right\}. \tag{3.14}$$

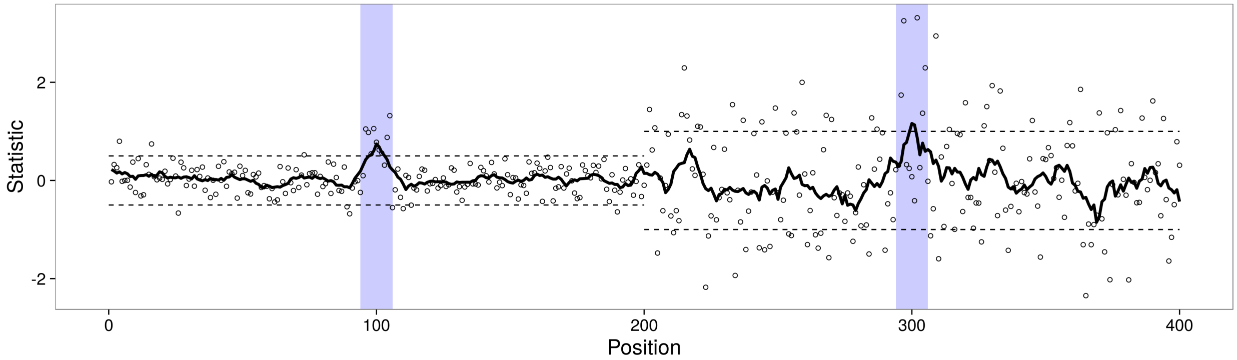


Figure 3.5: An example in which a variable threshold is helpful. The statistics (marked by points) on the left are much less noisy than the statistics on the right. Blue shading marks two regions of true signal. The solid line shows the smoothed statistics using a 10 unit moving average. Horizontal dashed lines show how different thresholds can be applied to each half to detect the two signals. Without using a variable threshold, we would not be able to detect the signal on the left without making many false detections.

Algorithm 5 Permutation procedure for estimating $\lambda(z)$

Given a threshold, z, and a merging reference level z_0

For $l \in 1, \dots, L$ (L large):

1. Permute the trait values $\mathbf{x}_1, \dots, \mathbf{x}_n$.

2. Perform steps 1-3 of Algorithm 2, calculating test statistics with the permuted trait values. Denote the resulting set of discoveries $\mathcal{R}_{l,z}$.

3. Apply the merging procedure in Algorithm 3 to obtain $\mathcal{R}_{l,z}^{(z_0)}$. Let $R_{l,z} = |\mathcal{R}_{l,z}^{(z_0)}|$ be the total number of merged discoveries.

Estimate $\lambda(z)$ as

$$\hat{\lambda}(z) = \frac{1}{L} \sum_{l=1}^{L} R_{l,z}.$$

3.4.3 Step 4 of Algorithm 4: Selecting thresholds for each interval

Here we describe a method for selecting a different threshold on each interval that controls the rFDR. The details of this method are given in Algorithm 6. We first partition D into K intervals $E_1, \ldots, E_K$. We assume that discoveries span multiple intervals with very low probability. This requires that the intervals be large relative the size of a discovery. We find that, if intervals are long, FRET is fairly insensitive to precisely where interval endpoints are.

Notation

Before describing the procedure in Algorithm 6, we introduce some notation. We denote the threshold on interval E_k as z_k and define the K-vector $\mathbf{z} = (z_1, \ldots, z_K)$. We define the excursion set with respect to $\mathbf{z}$ as

$$\mathcal{E}_{\mathbf{z}} = \bigcup_{k=1}^{K} \left\{ s_j \in E_k : |\tilde{T}(s_j)| \geq z_k \right\}.$$

We let $\mathcal{R}_{\mathbf{z}}$ denote the set of groups of contiguous elements of $\mathcal{E}_{\mathbf{z}}$. We denote the total number of discoveries $R_{\mathbf{z}}$, and the number of true and false discoveries, $S_{\mathbf{z}}$ and $V_{\mathbf{z}}$ respectively. Note that the sub-script is a bold-faced $\mathbf{z}$ indicating a K-vector rather than a scalar.

Let $\lambda_k(z_k)$ be the expected number of false discoveries made on E_k when the threshold on E_k is z_k. Assuming that discoveries overlap multiple intervals very rarely, the expected number of false discoveries made on the entire genome is

$$\lambda(\mathbf{z}) = \sum_{k=1}^{K} \lambda_k(z_k).$$

Algorithm 6: Selecting a combination of thresholds to control the rFDR

Attempting to choose K different thresholds with no constraints is unfeasible. In order to apply the results of Theorem 3.2.1, we must have a one-to-one mapping from combinations of thresholds, $\mathbf{z}$, to the expected number of false discoveries, $\lambda(\mathbf{z})$. Without constraints, we

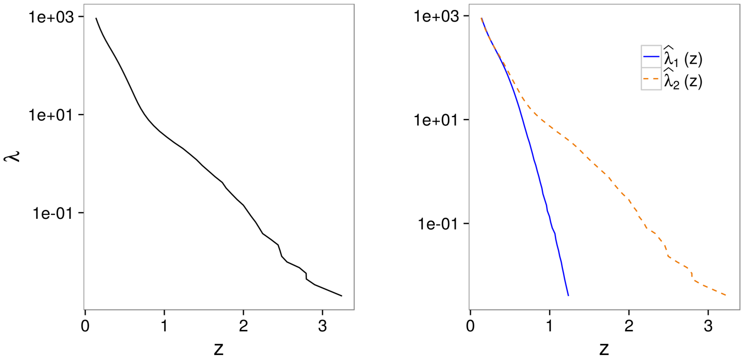


Figure 3.6: *Left:* Estimates of $\lambda(z)$ made using Algorithm 5 with a single threshold for simulated data. *Right:* The simulated data are divided into two equally sized intervals. We estimate $\lambda_1(z)$ and $\lambda_2(z)$ separately on the two intervals each interval.

will have many combinations of thresholds that result in the same expected number of false discoveries. Therefore, we propose choosing combinations of the K thresholds that give each interval approximately the same estimated rate of null excursions. This strategy provides the best power when signal enrichment is roughly equal across segments.

For this procedure, we require a lower bound, z^*, discussed in Section 3.2.2, and a merging reference, z_0, discussed in Section 3.2.3. The lower bound applies to each element of $\mathbf{z}$. We first separately estimate the K univariate functions $\lambda_k(z_k)$ for $k \in 1, \ldots, K$, on a discrete grid of values of z_k, using the permutation procedure in Algorithm 5 (we use the same grid for all K functions). Figure 3.6 shows estimates made for simulated data using only one interval (left) and dividing the region into two equal intervals (right). As in the example in Figure 3.5, these data are much noisier in one half of the simulated region than in the other. Thus, for any threshold value, more false discoveries are made in the noisier half.

For any K-vector of thresholds, $\mathbf{z}$, we can use the estimates $\hat{\lambda}_1(z_1), \ldots, \hat{\lambda}_K(z_K)$ to obtain the total expected number of false discoveries on all of D:

$$\hat{\lambda}(\mathbf{z}) = \sum_{k=1}^{K} \hat{\lambda}_k(z_k).$$

In steps 3 and 4 of Algorithm 6, we define an inverse mapping giving one K-vector of thresholds for each value of λ. For each $k \in 1, \ldots, K$, $\hat{\lambda}_k(z_k)$ is one-to-one, so the inverse, $\hat{\lambda}_k^{-1}(\lambda)$, is well defined. For $k \in 1, \ldots, K$, we calculate $\hat{\lambda}_k^{-1}(\lambda_k)$ on a grid of values of λ_k via linear interpolation.

In order to achieve a total expected number of false discoveries of λ, we choose the threshold on E_k to be

$$\hat{z}_k(\lambda) = \hat{\lambda}_k^{-1}\left(\lambda \cdot \frac{|E_k|}{|D|}\right) \tag{3.15}$$

and define $\hat{\mathbf{z}}(\lambda) = (\hat{z}_1(\lambda), \ldots, \hat{z}_K(\lambda))$.

For any λ, we can now estimate R_λ in (3.3) as the number of discoveries made using the K-vector of thresholds $\hat{\mathbf{z}}(\lambda)$. This allows us to select Λ, the expected number of false discoveries that controls the rFDR, using (3.3). We then find the corresponding K-vector of thresholds $\tilde{\mathbf{z}} = \mathbf{z}(\Lambda)$. This is the K-vector of thresholds returned by Algorithm 6.

Chouldechova [2014] proposes that, if some areas are a priori expected to have more signal enrichment than others, these areas could be permitted more false discoveries. They propose weighting the number of expected false discoveries allowed in each segment proportionally to its expected enrichment. More enriched regions are permitted to have more false discoveries, potentially boosting power. In many experiments there is little information about where signal enrichment is expected so, by default, we assume equal weighting. However, weights reflecting prior information could be inserted into (3.15) as

$$\hat{z}_k(\lambda) = \hat{\lambda}_k^{-1}\left(w_k\lambda \cdot \frac{|E_k|}{|D|}\right) \tag{3.16}$$

on E_k with weights chosen so that $\sum_{k=1}^{K} w_k = 1$.

3.5 Simulations

In this section, we use simulated data to compare the performance of FRET with that of fixed-region testing methods. Our simulations are modeled on DNase 1 sensitivity data which take the form of counts of cleavages at each genomic location. These data are illustrated

Algorithm 6 Selecting $\tilde{\mathbf{z}}$ to control the false discovery rate

Given a partition of D into intervals $D = \bigcup_{k=1}^{K} E_k$, smoothed statistics $\tilde{T}(s_j)$ for $j \in 1, \ldots, p$, a lower bound, z^*, and a merging threshold z_0,

1. For k in $1, \ldots, K$ and a grid of values $z_1 < \cdots < z_B$, use only data corresponding to positions in E_k to estimate $\lambda_k(z_b)$ using Algorithm 5, merging relative to z_0.

2. Select a grid of values $\lambda_1 < \cdots < \lambda_{B'}$ between $\sum_{k=1}^{K} \hat{\lambda}_k(z_B)$ and $\sum_{k=1}^{K} \hat{\lambda}_k(z_1)$.

3. For k in $1, \ldots K$ and b' in $1, \ldots, B'$ estimate $\lambda_k^{-1}\left(\lambda_{b'} \cdot \frac{|E_k|}{|D|}\right)$ via linear interpolation of $(z_1, \ldots, z_B)$ and $(\hat{\lambda}_k(z_1), \ldots, \hat{\lambda}_k(z_B))$ at $\lambda_{b'} \cdot \frac{|E_k|}{|D|}$.

   Define $\hat{\mathbf{z}}(\lambda_{b'})$ element-wise as

$$\hat{z}_k(\lambda_{b'}) = \hat{\lambda}_k^{-1}\left(\lambda_{b'} \cdot \frac{|E_k|}{|D|}\right).$$

4. For $b' \in 1, \ldots, B'$, calculate $R_{\hat{\mathbf{z}}(\lambda_{b'})}$ — the number of sets of contiguous elements of $\mathcal{E}_{\hat{\mathbf{z}}(\lambda_{b'})}$.

5. Select Λ as

$$\Lambda = \max\left\{\lambda : \lambda \in \lambda_1, \ldots, \lambda_{B'}, \lambda / R_{\hat{\mathbf{z}}(\lambda)} \leq \alpha\right\}$$

Return $\tilde{\mathbf{z}} = \hat{\mathbf{z}}(\Lambda)$.

in Figure 1.1a. It is reasonable to expect differential regions to overlap or contain peaks in DNase 1 sensitivity, since peaks tend to occur at regulatory features. Thus, fixed-region testing methods for DNase 1 sensitivity typically first identify peaks and then test within each peak. Here we explore the question of how "good" the region boundaries must be for fixed-region testing to be preferable to FRET.

3.5.1 Data generation

We now simulate data that approximates true DNase-1 data. For each sample we simulate count data at locations $s_1 = 1, \dots, s_p = p$ as

$$y_{ij} \sim \text{Poisson}(\gamma_{ij})$$

where γ_{ij} is the sample specific mean, or profile value, at location s_j. Each sample profile has peaks at various locations. A short example profile is shown in Figure 3.7. Within an experimental condition, peaks occur in the same places for each sample, however, peak heights can differ across samples due to sample specific random effects, trait effects, or both. There are no random effects or trait effects outside of peaks. In this setup, differential regions are sets of locations spanned by peaks with trait dependent heights.

Each simulated sample profile is 12,000 base-pairs long and contains 60 evenly spaced peaks. Most of the peaks have a simple uni-modal shape with a base-width of 20 base-pairs, like the first three peaks shown in Figure 3.7. In one experiment, some of the peaks are 40 base-pairs wide and have a bi-modal shape like the last peak in Figure 3.7.

We denote the height of the profile in the mth peak for sample i as p_{im}. If the mth peak is uni-modal, p_{im} is a scalar. If the mth peak is bi-modal, p_{im} is a vector of length 2 giving the heights of both modes. There are six ways that peak heights are generated. We refer to these as *peak types*. The first two give uni-modal peaks that are independent of the trait. These are

1. $p_{im} = 5$ (no random effects);

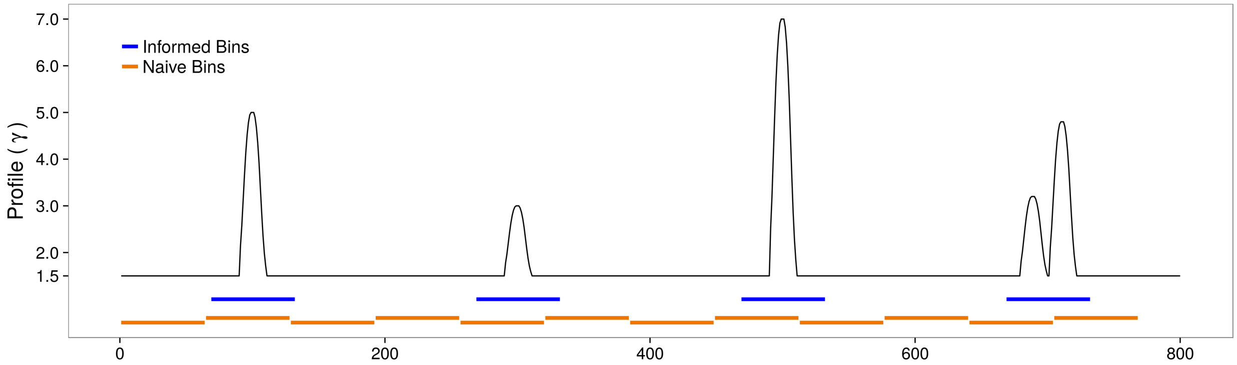


Figure 3.7: An example of a profile for simulated data described in Section 3.5.1. This profile contains four peaks. The first three peaks are uni-modal while the last peak is bi-modal. Blue and orange lines illustrate the two binning methods for fixed-region testing described in Section 3.5.2.

2. $p_{im} \sim \text{Exp}(5)$.

Peak types 3 and 4 have heights that depend on the value of a binary trait $x_i \in \{0, 1\}$. Peak type 3 is uni-modal while peak-type 4 is bi-modal. These are

3. p_{im} is distributed multinomially with values 1.5, 3, 7, and 10, and with probabilities given by the table

	$x_i = 0$	$x_i = 1$
1.5	0.55	0.2
3	0.3	0.45
7	0.1	0.25
10	0.05	0.1

.

4. Peaks are bi-modal like the one farthest to the right in Figure 3.7. The height of the first mode, p_{im1}, has the same distribution as the height of type 3 peaks. In the second mode, the direction of association with the trait is reversed. The height of the

second mode, p_{im2}, is distributed multinomially with values 1.5, 3, 7, and 10, and with probabilities given by the table

	$x_i = 0$	$x_i = 1$
1.5	0.2	0.55
3	0.45	0.3
7	0.25	0.1
10	0.1	0.05

.

Peak types 5 and 6 are both uni-modal and depend on the value of a continuous trait $x_i \in [0, 1]$. These are

5. $p_{im} = 3 + 4x_i$;

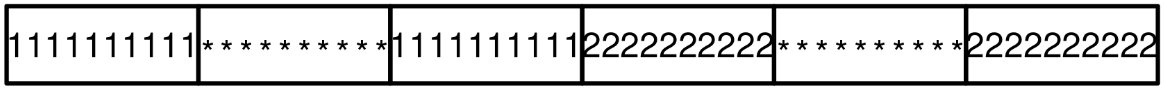
6. $p_{im} = 4x_i + w_i$ where $w_i \sim \text{Exp}(4)$.

Peak types 1 and 2 are non-differential regions while types 3-6 are differential and we hope to detect them. We run four sets of simulated experiments, described in Sections 3.5.3 and 3.5.4. For each of these experiments, the 60 peak heights for each sample are chosen according to the schematic in Figure 3.8. These simulated profiles contain three peak types: type 1, type 2, and one of the differential types 3-6. In this schematic, there are six adjacent regions. Within each region, 10 peaks of a single type are generated. The first and third regions have type 1 peaks, the fourth and sixth have type 2 peaks, while the second and fifth regions contain the differential type peaks.

3.5.2 Methods for Comparison

We will compare FRET with three fixed-region testing methods:

1. The Huber regression based test described in Section 3.3.1 calculated using the total number of counts in each region;



1111111111	**********	1111111111	2222222222	**********	2222222222

Figure 3.8: Schematic for constructing profiles described in Section 3.5.1. Each sample profile contains 60 evenly spaced peaks. Each peak is of one of the six types described in Section 3.5.1. The order of peak types is indicated by numbers and symbols in the schematic. Numbers 1 and 2 indicate peaks of type 1 or 2. The $*$ symbol indicates one of the peak types, 3-6, in which the trait is associated with the genomic phenotype.

2. The DESeq2 negative binomial test of Love et al. [2014];

3. The WaveQTL wavelet regression test of Shim and Stephens [2015].

Recall from Section 1.3 that WaveQTL performs a hierarchical Bayesian regression on wavelet transformed data. This allows it to model more complex associations than the simple strategy of summing counts within bins used by DESeq2 and the Huber test in bins. False discovery rates for each method are calculated using the Benjamini-Hochberg correction [Benjamini and Hochberg, 1995].

These tests require regions (or bins) to be pre-specified. We do so using two different strategies illustrated in Figure 3.7: *Naive binning* in which the entire length of simulated data is divided into bins of 64 positions, and *informed binning*, in which bins of width 64 are centered at peaks and data not falling into these bins is discarded. We choose windows of size 64 to accommodate WaveQTL which requires windows to have width equal to a power of two. We found that results for the Huber test and for DESeq2 were similar at other bin sizes. These two binning methods are illustrated in Figure 3.7.

We apply FRET using three choices of the partition described in Section 3.4.3: one, two, and six equal length intervals. The variance inflation constant, σ_0, described in Section 3.3.4 is set to 0.05 for all simulations. This constant was chosen by applying the method of Tusher et al. [2001] to one simulated data set.

3.5.3 Binary trait simulations

In this set of simulations, we search for regions associated with a binary trait. We simulate profiles for 30 samples with a binary trait: $x_i = 0$ for $i \in 1, \dots, 15$ and $x_i = 1$ for $i \in 16, \dots, 30$ where i indexes the sample.

In Figure 3.9 we show results using peak type 3 (top) and peak type 4 (bottom) in place of the $*$ in the schematic in Figure 3.8. These figures show the average false discovery proportion and detection rate over 100 simulations for FRET and the three window-based methods described in Section 3.5.2.

All methods except for DESeq2 control the false discovery rate in both settings. Using more than one interval for selecting the threshold for FRET makes an impact on its performance. For a fixed target false discovery rate threshold, using two or six intervals gives higher power than using only one. This is because the type 2 peaks, which have a sample specific random effect but no association with the trait, all occur in the right half of the profiles. In the left half of the profiles, all of the “null” peaks are type 1 peaks which have no random effects. Thus, the left half of each profile is less variable than the right half so using different thresholds in each half improves the power of FRET. This is similar to the phenomenon illustrated in Figure 3.5.

Using six intervals with this data is slightly less powerful than using two. With two intervals, both intervals contain the same number of signal regions. With six intervals, only two of the six contain any signal. Thus our assumption that signal is equally enriched in all intervals is no longer correct and it would be better to include weights as in (3.16).

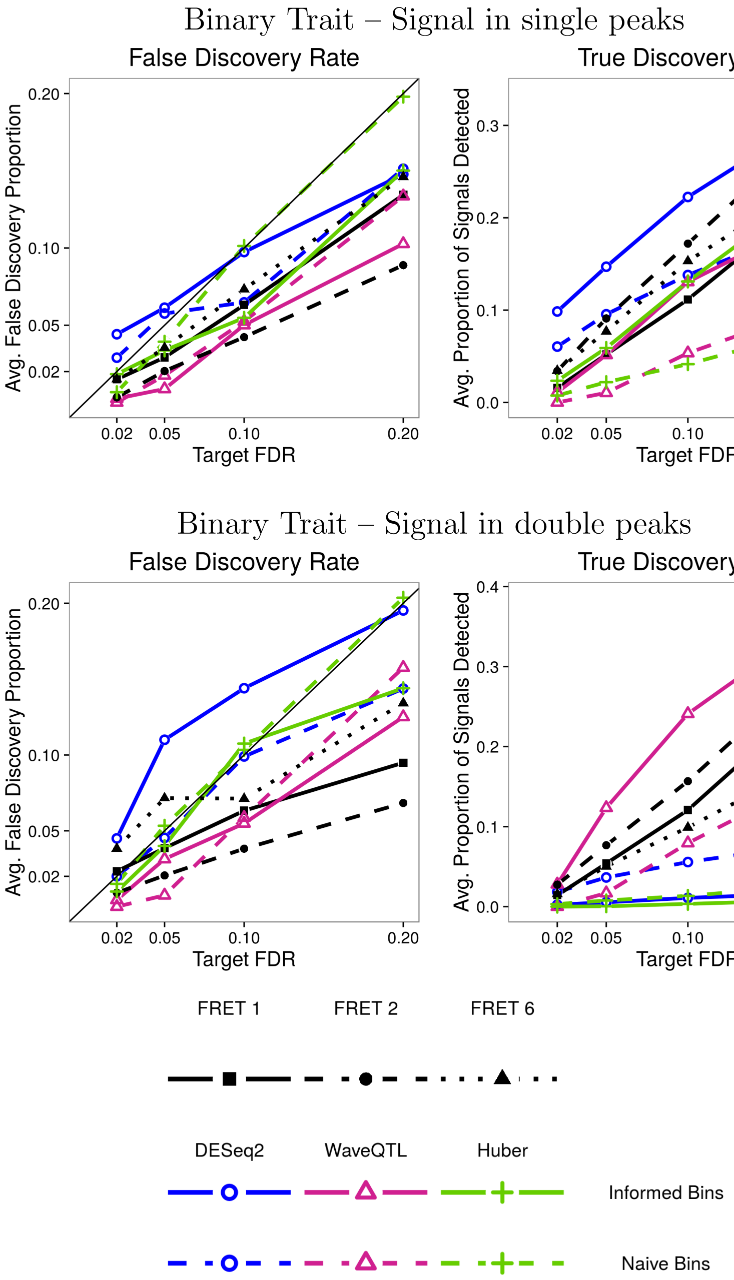


Figure 3.9: Comparison of FRET and fixed-region methods for the binary trait simulations described in Section 3.5.3. *Top:* Profiles include type 3 peaks, which contain a uni-modal peak, in place of the ∗ in Figure 3.8. *Bottom:* Profiles include type 4 peaks, which contain a bi-modal peak, in place of the ∗ in Figure 3.8. The two modes of type 4 peaks have opposite directions of effect.

In the setting with type 3 peaks (Figure 3.9, top), FRET with two intervals performs similarly to, or better than, most of the fixed-region methods with informed bins. This is notable because testing with informed bins has the advantage performing many fewer tests in null regions. DESeq2 has somewhat higher power than FRET.

All of the fixed-region methods except WaveQTL fail in the setting with type 4 peaks (Figure 3.9, bottom). This is not surprising. The two modes of type 4 peaks have association with the trait in opposite directions. The informed bins contain both modes so the signal is lost by simply summing over bins. Both FRET and WaveQTL with informed binning are able to detect the complex association. Interestingly, although WaveQTL had moderate power for the single peak experiments, it does very well for the double peak experiments.

3.5.4 Continuous Trait

Next we consider identifying associations with a continuous trait. We again simulated 30 samples. For each sample, and in each simulation, x_i is drawn from a Beta$(2, 2)$ distribution.

Figure 3.10 shows the results using type 5 peaks (Figure 3.10, top) and type 6 peaks(Figure 3.10, bottom) in place of the $*$ symbols in Figure 3.8. The height of type 5 peaks is dependent on the trait but does not include a sample specific random effect.

FRET struggles in the setting with type 5 peaks (Figure 3.10, top). This is essentially an extreme case of the situation in Fiure 3.5. Type 2 peaks, used in the right half of each profile, have no trait association but do have random effects and so are highly variable. Using FRET, it is very difficult to discover signal in the type 5 peaks, which are less variable, when they are included in the same interval used for threshold selection as type 2 peaks. This is because the threshold must be set very high to avoid false discoveries in the type 2 peaks. Thus, using a single threshold, we fail to detect any of the associated regions. Using two intervals, the power is close to 50%: We detect associations in the left half of the region but fail to detect associations on the right. With six intervals we are able to detect nearly all of the signal because no differential regions share an interval with type 2 peaks.

The variability of the heights of type 6 peaks include a random effect and are more variable than the heights of type 5 peaks. In this setting (Figure 3.10, bottom), FRET performs well.

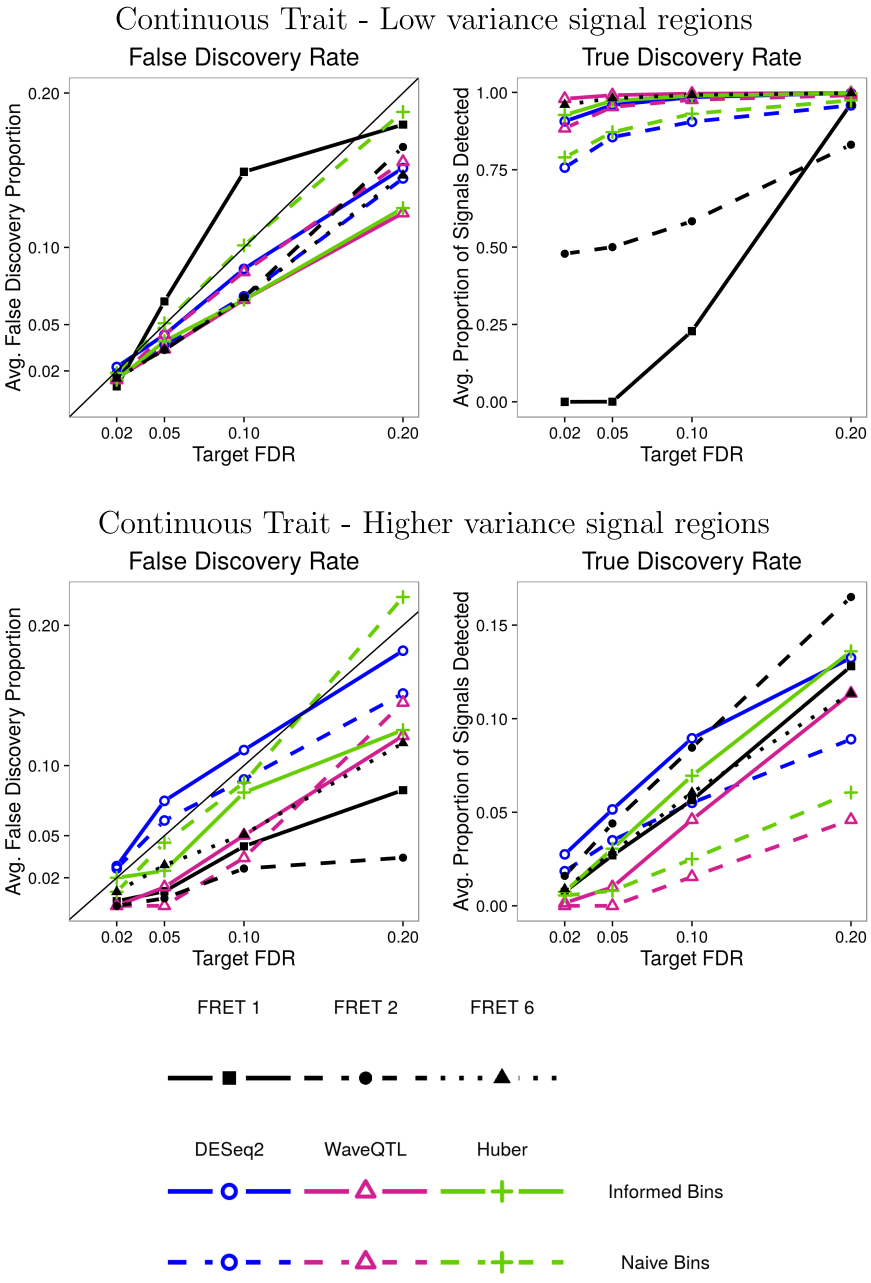


Figure 3.10: Comparison of FRET and fixed-region methods for the continuous trait simulations described in Section 3.5.4. *Top:* Profiles are generated with type 5 peaks, with no random effects, in place of the $*$ in Figure 3.8. *Bottom:* Profiles are generated with type 6 peaks, with both trait and random effects, in place of the $*$ in Figure 3.8.

3.6 Discussion

In this chapter we have presented a flexible, robust alternative to fixed-region testing for identifying differential regions with genomic phenotypes. We find that FRET controls the rFDR effectively in a variety of situations and often performs as well as fixed-region tests with very good pre-specified regions. FRET's flexibility provides an advantage for detecting complex associations. It is also valuable when it is difficult to pre-define regions for testing. In almost every case, FRET is a much better alternative to fixed-region testing with naive bins.

We find that extending the excursion procedure to allow variable thresholds is an important development. One weakness of the excursion procedure is that the presence of one very noisy region can force the threshold required to control the rFDR upward, reducing power over the entire space. Allowing the threshold to vary protects us from losing signal due to this phenomenon. As seen in simulations, increasing the number of intervals used to determine the threshold does not guarantee an increase in power but it does limit the influence of noisy regions on the power in other locations.

Chapter 4

ANALYSIS OF DNASE 1 SENSITIVITY IN DRUG SUSCEPTIBLE AND RESISTANT CANCER CELLS

In this chapter we compare the behavior of FRET described in Chapter 3 with several alternatives for analyzing DNase 1 sensitivity data.

4.1 Description of the problem

We worked with our collaborators in the lab of John Stamatoyannopoulos/Altius Institute to analyze DNase 1 sensitivity data for several small cell lung cancer (SCLC) cell lines with the goal of identifying regions in which DNase 1 sensitivity is associated with susceptibility to an anti-cancer agent.

A total of 42 cell lines were classified as resistant, susceptible, or having indeterminate drug response. We limit this analysis to a comparison of 25 cell lines with determined drug response — 9 cell lines are susceptible and 16 are resistant. SCLC cell lines were grown following distributor (ATCC) protocols. DNase-seq experiments and library construction, follows the protocol outlined in John et al. [2013].

Recall the discussion of DNase-seq assays from Section 1.1. At each position we obtain a count of sequence fragment ends. We normalized these counts using a simple "library size correction". The *library size* for individual i is the total number of read ends over the entire genome. This normalization strategy simply scales the phenotype at each position so that each sample has the same effective library size in the normalized data. The count of fragment ends for individual i at position j is multiplied by the weight w_0/w_i where w_i is library size for individual i, and w_0 is the geometric mean of library sizes for all 25 cell lines.

4.2 Methods

4.2.1 Analysis with FRET

We analyzed these data using FRET following Algorithm 6 in Section 3.4.3. We partitioned the genome into 10kb windows, allowing each widow to have its own threshold. To smooth the statistics in step 2, we used a moving average with a bandwidth of 50 base-pairs. In step 3(b), the estimation of $\lambda_k(z)$, we used 500 permutations.

We chose σ_0 to be 0.05 and the lower bound z^* to be 0.9. Details of how these values were chosen are given in Section C.1 of Appendix C. We use the merging reference threshold suggested by Chouldechova [2014] of $z_0 = 0.3 \cdot z^*$.

4.2.2 Comparison methods

We obtained two sets of region boundaries learned from the data from our collaborators:

Hotspots are regions with elevated rates of cleavage relative to the local background. In these data, hotspots were called for each sample individually by our collaborators using the method proposed John et al. [2011]. We provide an overview of this method in Section C.2 of Appendix C. We then merged hotspots across all 25 samples to obtain a list of about 1.4 million hotspots in autosomes with an average width of 637 base-pairs.

Peaks are short regions with very high DNase 1 sensitivity such as those shown in Figure 1.1. Our collaborators defined peaks for individual samples as the highest density 150 base-pair sub-region of a hotspot. We combined the list of peaks for all 25 samples into a single "master" peak list using a strategy in devised by our collaborators. Details are provided in Section C.2 of Appendix C. This results in about 2.6 million peaks in autosomes with an average width of 164 base-pairs. About 10% of the merged peaks are larger than 200 base-pairs and fewer than 1% are larger than 300 base-pairs.

In addition to FRET, we analyzed these data using four alternatives:

1. The Huber regression based test in (3.11) calculated using the total normalized counts in each peak;

2. The DESeq2 negative binomial test of Anders and Huber [2010] and Love et al. [2014] applied to peaks;

3. The WaveQTL wavelet regression test of Shim and Stephens [2015] applied to the central 128 base-pairs of each peak;

4. The Wellington-Bootstrap method of Piper et al. [2015].

The first three methods are *fixed-region tests.* They aggregate data within peaks and perform one test per peak. For WaveQTL only, each peak is reduced to its central 128 base-pairs. This accommodates the requirement that regions analyzed with WaveQTL have a width equal to a power of 2. WaveQTL uses permutations to estimate p-values. In this analysis we set the maximum number of permutation to 10^8. For the Huber test in 1, we set the value of σ_0 to 0.05. For the rest of this discussion, we refer to this method as the "Huber fixed-region test" to distinguish it from the Huber statistic used as part of FRET. False discovery rates (q-values) for the three fixed-region tests are calculated using the Benjamini-Hochberg correction [Benjamini and Hochberg, 1995].

We note that, although DESeq2 was developed for RNA-seq data, we consider it here because it is the method our collaborators have found the most promising for analysis of DNase-seq data.

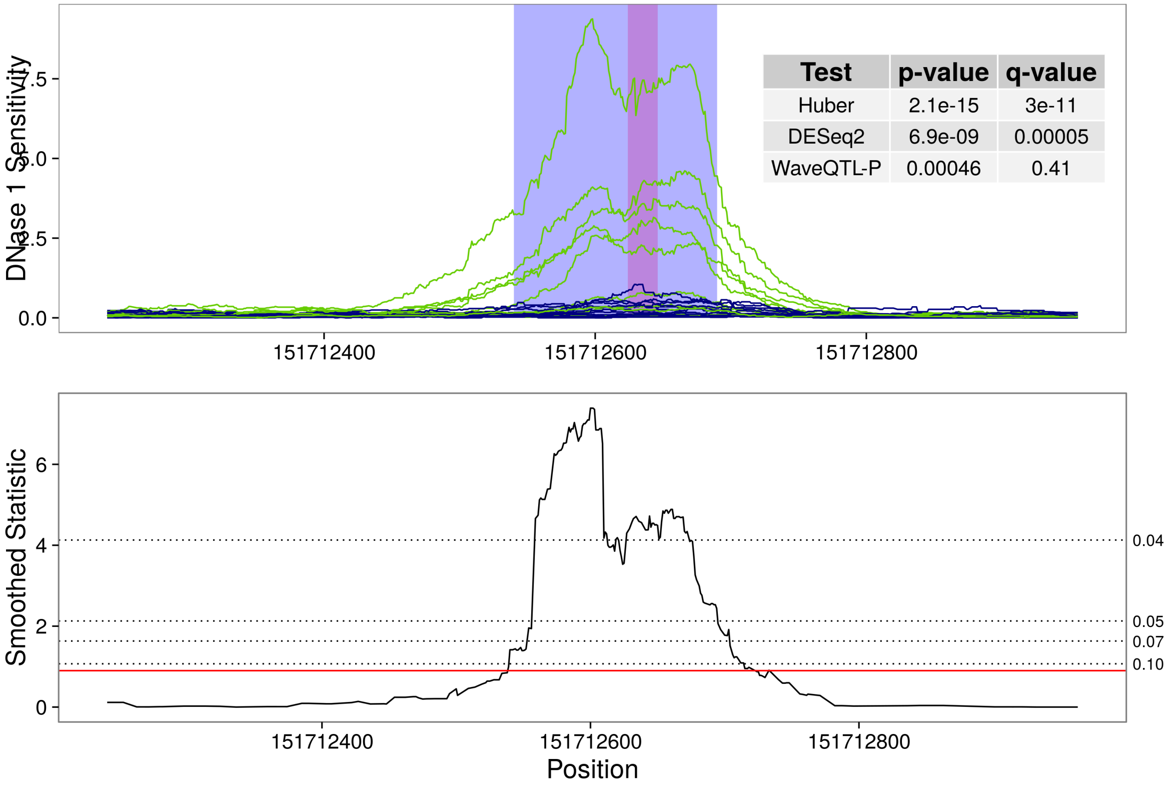
The fourth method, Wellington-Bootstrap, is not a fixed-region method. Instead, it searches over all hotspots and attempts to detect *footprints* — regions with a cleavage pattern characteristic of transcription factor binding — that are present in the resistant group but not in the susceptible group or vice-versa. Wellington-Bootstrap first identifies footprints in each group (pooling sequences for samples in the same group) using the Wellington footprinting method [Piper et al., 2013]. Each footprint is then reassessed in the combined data. Footprints that are detected in one trait group but not in the full data are considered to be differential. Each footprint is assigned a score between 0 and 100 quantifying the difference in evidence for a footprint provided by the two groups. These scores can be used to rank footprints but no false discovery rates are calculated. In the original publication, Piper et al. [2015] use a score threshold of 10 and recommend selecting a threshold to give a desired number of results.

In Section C.3 of Appendix C, we describe two additional fixed-region tests: one using a quasi-Poisson regression statistic and one using a t-test applied to the normalized sum of counts aggregated within peaks.

4.3 Results

Figure 4.1 shows of results from FRET and the four comparator methods in a small region of chromosome 1. This region contains a peak shaded in blue and a footprint called by Wellington-Bootstrap (shaded pink). All three fixed-region tests assign a very low p-value to this peak, though the WaveQTL result does not remain significant after transforming to q-values. Wellington-Bootstrap identifies a small footprint inside the peak, but the score assigned to it is only 1.1. The smoothed test statistics used by FRET are shown in the bottom panel along with horizontal lines marking the thresholds associated with several region-wise false discovery rates. Note that, because the merging procedure in Algorithm 3 is used, FRET identifies only one region at an rFDR level of 0.04.

In the rest of the section, we describe the results of all five analysis methods.



Test	p-value	q-value
Huber	2.1e-15	3e-11
DESeq2	6.9e-09	0.00005
WaveQTL-P	0.00046	0.41

Figure 4.1: A region identified as differential by multiple methods: *Top:* DNase 1 sensitivity for all 25 samples is shown as green (susceptible) and blue (resistant) lines. DNase 1 sensitivity has been smoothed using a 50 base-pair moving average in order to visualize patterns. The peak in which the fixed-region methods test is shaded blue. Pink shading marks a footprint called by Wellington-Bootstrap. This footprint received a score of 1.1. *Bottom:* The smoothed test statistic used by FRET. Horizontal lines show thresholds corresponding to several rFDR values. The red line is at the lower bound, $z^* = 0.9$.

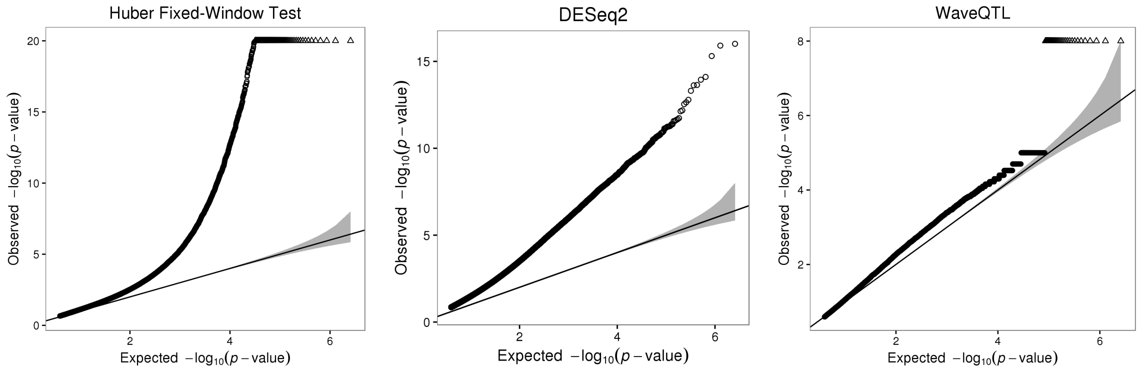


Figure 4.2: Quanitle-quantile plots for $-\log$ transformed p-values produced by the three fixed-region tests described in Section 4.2.2. Triangles indicate peaks with $-\log$ p-values higher than the limit of the y-axis. For WaveQTL, triangles correspond to peaks assigned a p-value of 0. The solid line and grey shading show the expectation and 95% confidence interval for quantiles of the Uniform(0, 1) distribution.

4.3.1 Distribution of p-values for fixed-region tests

Figure 4.2 shows quantile-quantile plots for $-\log$ transformed p-values produced by the Huber fixed-region test, DESeq2, and WaveQTL. The behavior of the WaveQTL p-values is somewhat unexpected — WaveQTL uses permutations. We allowed a maximum of 10^8 permutations. However, our results contained 30 peaks assigned a p-value of exactly 0 with the next smallest p-value being 10^{-5}. In Figure 4.2, we display the p-values reported as 0 as triangles at the level of 10^{-8}. Besides this anomaly, we found that WaveQTL p-values are only slightly inflated compared to the expectation under the null. After transforming to q-values using the Benjamini-Hochberg correction, the smallest q-value for a peak with a non-zero p-value was 0.29.

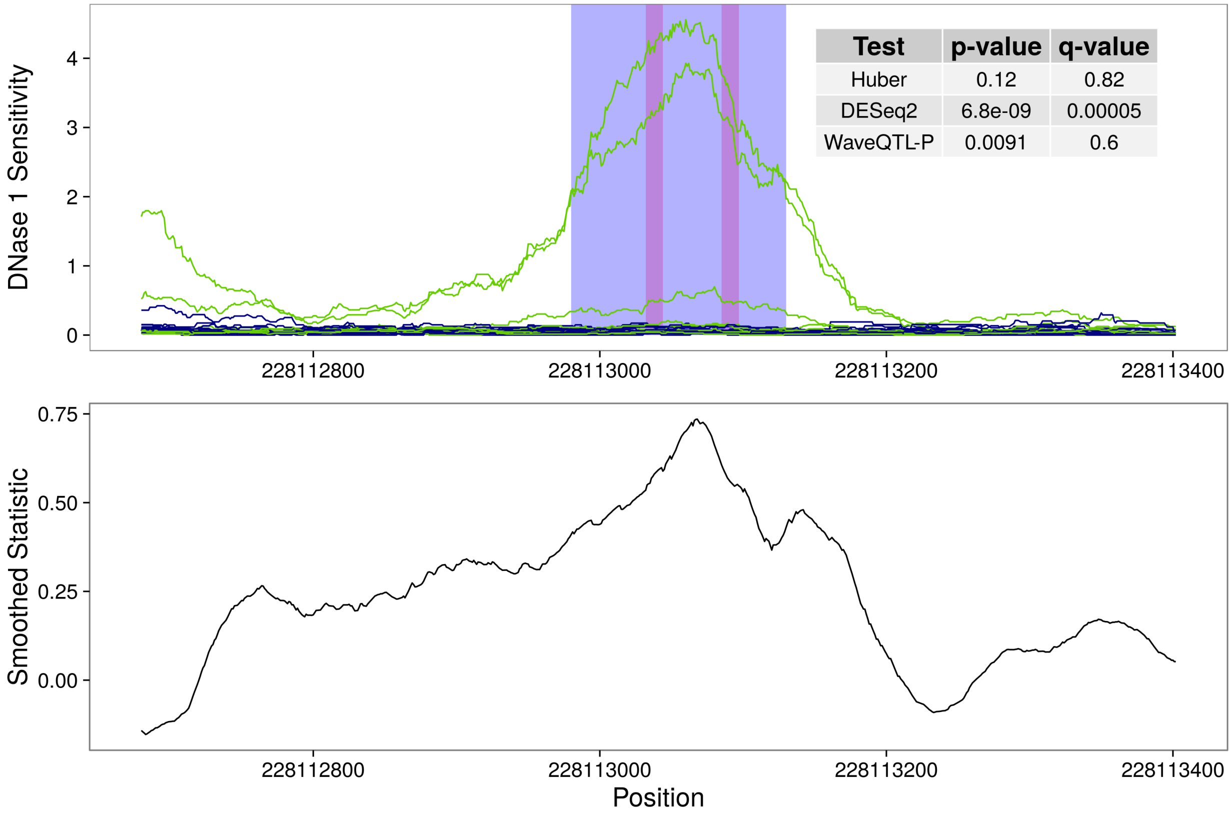
DESeq2 and the Huber fixed-region test both produce dramatically inflated p-values. The DESeq2 p-values depart from the expectation for uniformly distributed p-values very early. This pattern suggests that DESeq2 is poorly calibrated in this problem since we would expect a large proportion of peaks to have no association with the drug resistance trait. Our simulations show that DESeq2 does not always control the false discovery rate at the target level (see results displayed in Figure 3.9). Furthermore, we find that DESeq2 often assigns high significance to regions in which only one or two cell lines have high levels of DNase 1 sensitivity. Figure 4.3 shows an example of such a region. In this region two samples have a peak in DNase 1 sensitivity while the rest have almost no sensitivity. Only DESeq2 finds a significant result here. We suspect that, because DESeq2 combines information across peaks to estimate the dispersion parameter, it is under-estimating the variance of peaks with outliers.

We expect the Huber fixed-region test to be less sensitive to outlying samples as a result of using a robust loss function. Although the Huber fixed-region p-values appear to be better calibrated in that the larger p-values are more consistent with a uniform distribution, the sample size here is small. The Huber p-value is based on the asymptotic distribution of the test statistic. We may not have a large enough sample size for this approximation to hold.

4.3.2 Total number of discoveries

Figure 4.4 shows, for each method, the relationship between the number of differential regions detected and the significance threshold applied. For FRET, this threshold is the region-wise false discovery rate. For the Huber fixed-region test and DESeq2, the threshold is the (peak-wise) false discovery rate and for Wellington-Bootstrap, the threshold is the score. WaveQTL is omitted from this comparison because it does not produce any q-values smaller than 0.29 for peaks that were not assigned a p-value of 0. For Wellington-Bootstrap, we display results of score thresholds greater than 50. This value was chosen because it gives a similar number of regions as produced by FRET at an rFDR threshold of 0.2.

The smallest rFDR value possible using FRET in this analysis is 0.038. This bound is governed by the number of permutations. A larger number of permutations can allow FRET to make discoveries at lower thresholds. We note that most of the regions discovered by FRET and by Wellington-Bootstrap overlap peaks despite the fact that both consider much larger portions of the genome. This suggests that the merged peak set contains most of the



Test	p-value	q-value
Huber	0.12	0.82
DESeq2	6.8e-09	0.00005
WaveQTL-P	0.0091	0.6

Figure 4.3: A region identified as differential only by DESeq2 and Wellington-Bootstrap. Colors and shading are as in Figure 4.1. Wellington Bootstrap identifies two footprints shaded in pink. The left and right footprints are given scores of 97.9 and 31.8 respectively. In this region two of the susceptible cell lines have a peak while the other samples have almost no sensitivity. The smoothed statistics for FRET never exceed the minimum threshold of $z^* = 0.9$.

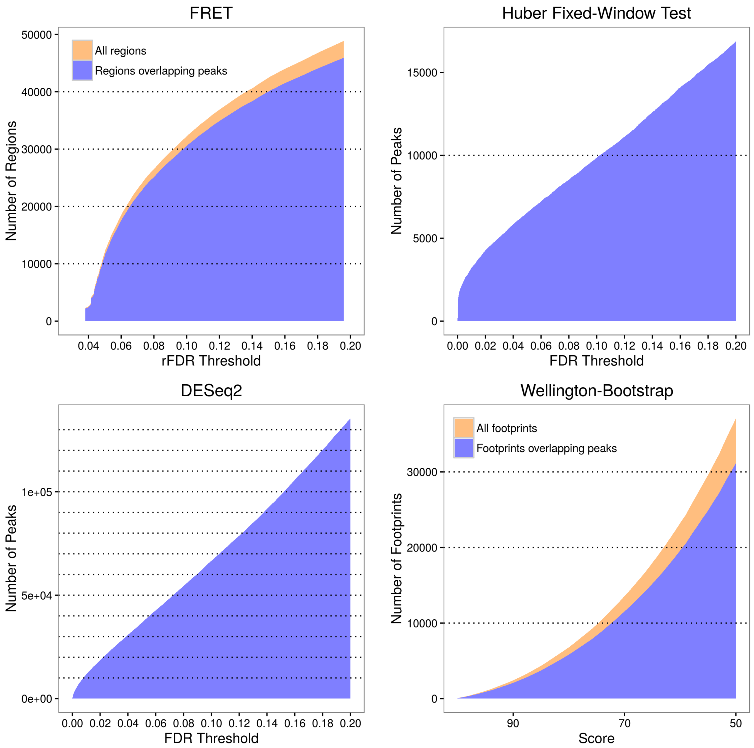


Figure 4.4: Relationship between thresholds and the number regions discovered by FRET and comparator methods described in Section 4.2.2. Dotted horizontal lines show every 10,000 units on the y-axis.

true signal discoverable using these methods.

4.3.3 Rank correlation between methods

In order to compare the results of FRET with the fixed-region methods, we associate peaks with the lowest rFDR threshold assigned by FRET anywhere in the peak. We refer to this value as the FRET mimimum rFDR value of a peak. Only 66,726 peaks of the total 2.6 million contain regions in which the smoothed statistics exceeded the FRET lower bound ($z^* = 0.9$) in absolute value. No minimum rFDR values are associated with peaks in which the smoothed statistics did not reach z^*.

Similarly, for Wellington-Bootstrap, each peak is associated with the largest score for a footprint overlapping the peak. We refer to this as the Wellington-Bootstrap maximum footprint score. This score can be assigned to 608,589 peaks. The rest of the peaks contain no Wellington footprints.

Table 4.1 shows the rank-based correlation between all five methods within the set of 66k peaks assigned a FRET minimum rFDR value. Wellington-Bootstrap has the lowest correlation with any other method. The highest rank correlation is found between WaveQTL and the Huber fixed-region test. FRET has similar rank-based correlation with each of the three fixed-region tests.

Table 4.1: Rank correlation between methods described in Section 4.2.2 in the 66k peaks assigned a FRET minimum rFDR value.

	Huber	DESeq2	WaveQTL	Wellington-Bootstrap
FRET	0.55	0.46	0.56	0.24
Huber		0.62	0.73	-0.02
DESeq2			0.69	0.10
WaveQTL				0.03

4.3.4 DNase 1 sensitivity patterns inside detections

In this section we explore the characteristics of regions detected by each method. Figure 4.5 shows the distributions of summary statistics for top peaks as ranked by the three fixed-region tests and for peaks overlapping top differential regions detected by FRET and Wellington-Bootstrap. For FRET and Wellington-Bootstrap we choose to compare summary statistics in overlapping peaks for comparability with the other methods.

For the Huber fixed-region test and DEseq2, we chose peaks significant at an FDR threshold of 0.05 (6,534 and 36,683 peaks respectively). For FRET we chose peaks that overlap differential regions significant at an rFDR threshold of 0.05 (11,822 peaks). The thresholds for Wellington-Bootstrap and WaveQTL were chosen arbitrarily: For Wellington-Bootstrap we evaluated peaks overlapping footprints with scores higher than 80 (5,797 peaks) and for WaveQTL we chose peaks with p-values less than 0.001 (5,759 peaks).

The left panels of Figure 4.5 show the mean to median ratio of the top discoveries for each method. A high mean to median ratio is indicative of a peak with outliers like the example shown in Figure 4.3. Consistent with some of the patterns noted in the previous sections, DESeq2 detects more peaks with a high mean to median ratio than the other methods with the exception of Wellington-Bootstrap. Wellington-Bootstrap pools data within trait groups so it cannot distinguish between areas where all the samples in one group show a consistent trend and areas where all of the observed cleavages are made in a single sample. It is therefore not surprising that Wellington-Bootstrap strongly favors peaks with skewed DNase 1 sensitivity distributions. The peaks detected by WaveQTL and the Huber fixed-region tend to have lower mean-median ratios than those detected by FRET.

The middle panels of Figure 4.5 show the distributions of the fold change between resistant and susceptible cell lines. These figures show a striking difference between the methods. Only DESeq2 and Wellington-Bootstrap identify many peaks with higher DNase 1 sensitivity in the resistant cell lines (indicated by a fold change greater than 1). At an FDR threshold of 0.05, 42% of peaks identified by DESeq2 have more sensitivity in the resistant group. By contrast, at the same threshold, fewer than 1% of the peaks identified by FRET and about 5% of the peaks identified by the Huber fixed-region test have this quality. Wellington-Bootstrap strongly favors peaks with higher sensitivity in the resistant group – These are about 81% of peaks with scores more than 80.

This pattern may be explainable by the difference in distributions of DNase 1 sensitivity between resistant and susceptible cell lines: Resistant cell lines appear to be much more heterogeneous than the susceptible cell lines. They have higher variance and more often have one or two outliers. These patterns are supported by Figure 4.6 where we show the mean, variance and mean to median ratio for all 2.6 million peaks separated by treatment

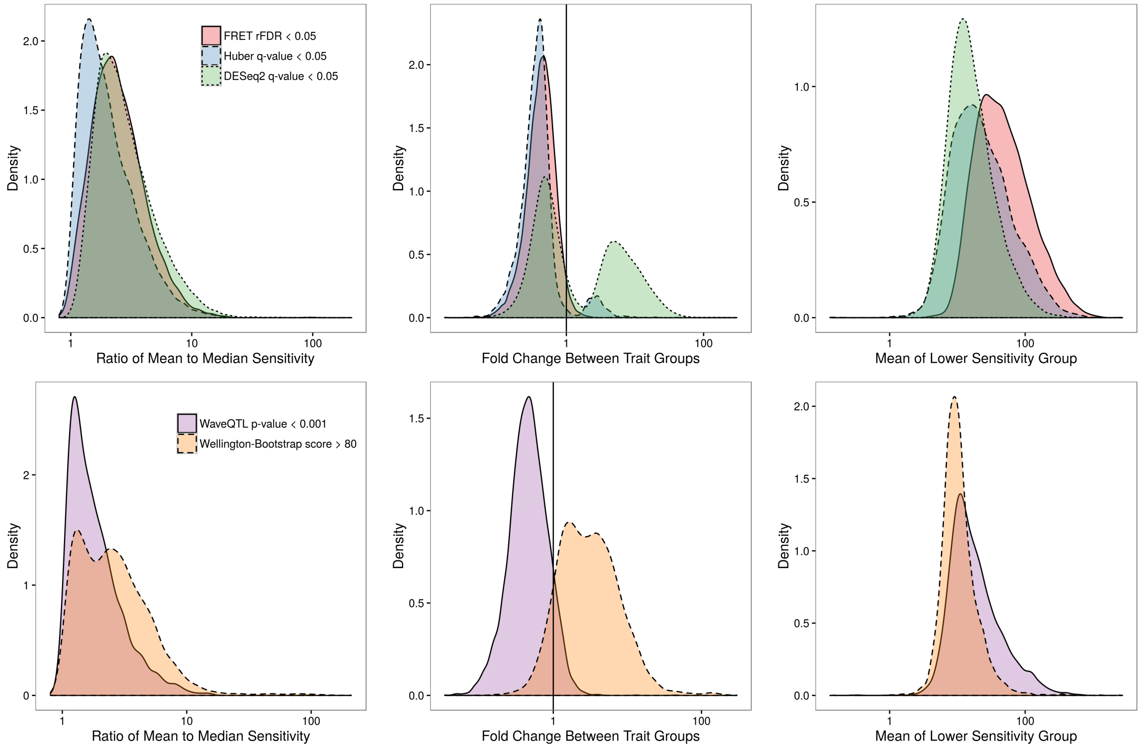


Figure 4.5: Summary statistics for top peaks ranked by FRETand the four methods described in Section 4.2.2. *Left:* The distribution of the ratio of the median to the mean in top peaks for FRET, DESeq2 and the Huber fixed-region test (top) and by WaveQTL and Wellington-Bootstrap (bottom). *Middle:* The distributions of the fold change between the resistant and susceptible groups. *Right:* The distributions of the mean in the lower sensitivity trait group. Note that the horizontal axes are on a log scale. In the middle panels, the vertical line is at 1.

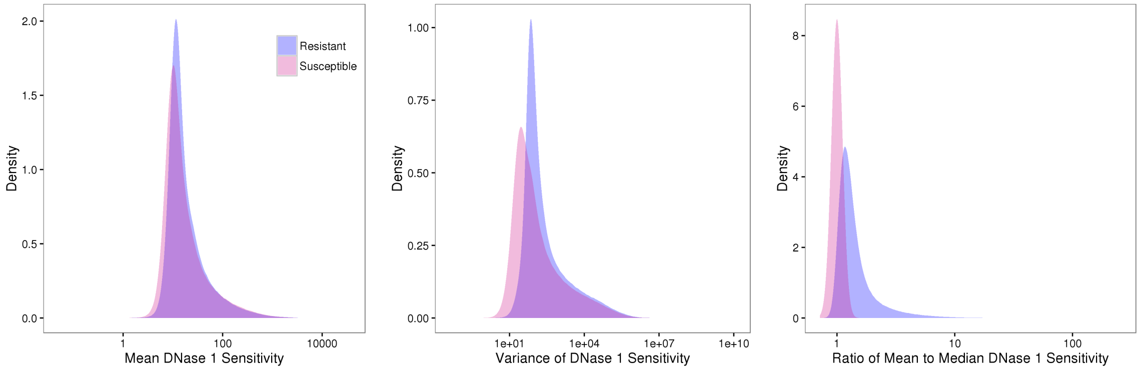


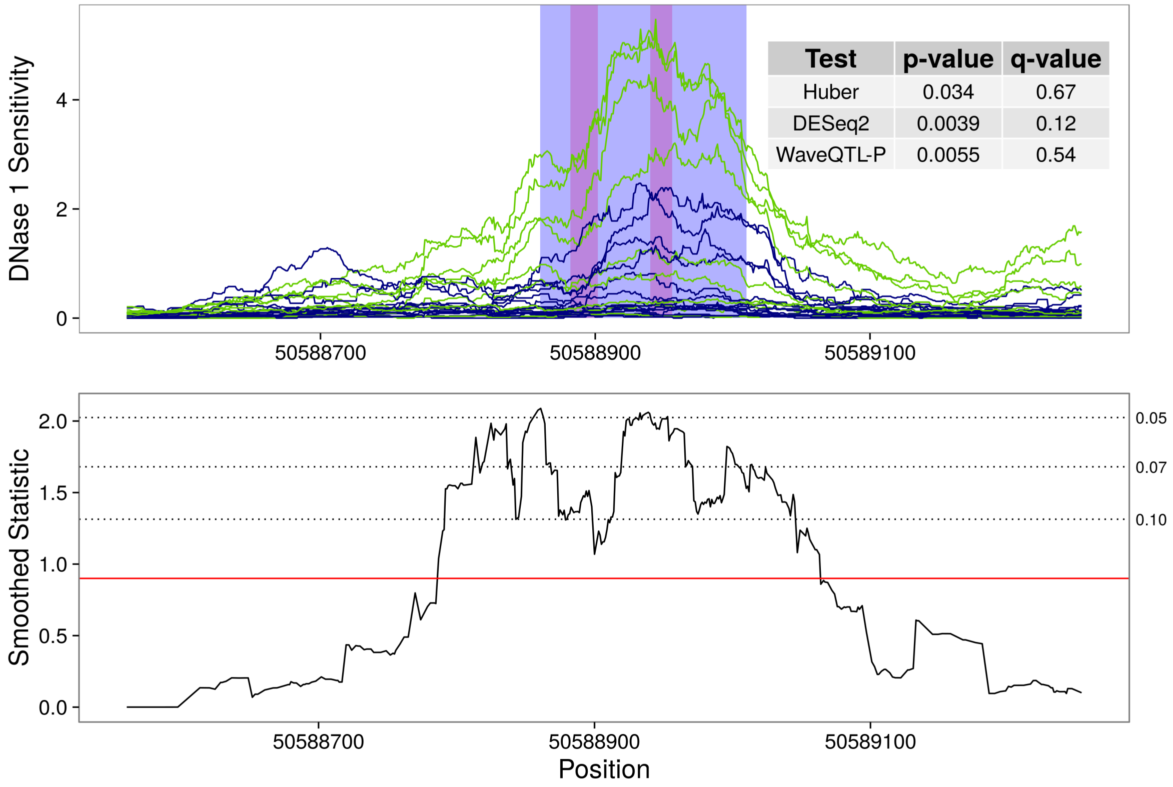
Figure 4.6: Summary statistics for all 2.6 million peaks separated by treatment group. Note that the horizontal axes are on a log scale.

group. Although the two groups have similar distributions of mean DNase 1 sensitivity, the resistant group has much higher variance and and a higher mean to median ratio. Our collaborators do not find this pattern surprising – They believe there is likely to be more biological similarity between susceptible cell lines than between resistant cell lines.

This pattern suggests that, in peaks with a higher mean in the resistant group, the difference is often driven by one or two outliers. Because FRET, the Huber fixed-region test, and WaveQTL tend not to give high significance to regions with this pattern, these methods identify many fewer differential regions with higher sensitivity in the resistant group.

Finally, the right-hand panels of Figure 4.5 show an interesting difference between FRET and the other methods. In these plots we show the mean DNase 1 sensitivity only in the less sensitive group. DESeq2 and Wellington-Bootstrap both strongly favor peaks in which the less sensitive group has almost no DNase 1 sensitivity. These might be regions in which a functional element is "off" entirely in one group. FRET and, to a lesser degree, WaveQTL and the Huber fixed-region test, detect more regions that have some DNase 1 sensitivity in both groups.

Figure 4.7 shows an example of a region with positive DNase 1 sensitivity in both groups but higher sensitivity in the susceptible group. This region is detected only by FRET. This



Test	p-value	q-value
Huber	0.034	0.67
DESeq2	0.0039	0.12
WaveQTL-P	0.0055	0.54

Figure 4.7: A region identified as differential only by FRET with non-zero DNase 1 sensitivity in both groups. Colors and shading are as in Figure 4.1. Wellington Bootstrap identifies two footprints shaded in peaks. The left and right footprints are given scores of 1.1 and 0.9 respectively.

type of pattern may indicate a difference in the degree of activity of a functional element.

4.4 Summary

In this analysis we have compared FRET with four alternative methods. We find the following patterns:

- DESeq2 is likely poorly calibrated in this problem. We find evidence that its p-values are too significant, possibly as a result of underestimating the variance in noisy peaks.

- DESeq2 and Wellington-Bootstrap favor peaks in which the difference between groups is driven by outliers.

- WaveQTL makes almost no discoveries at low FDR thresholds.

- FRET and the Huber fixed-region test identify very few peaks with higher sensitivity in the resistant group.

- FRET favors regions in which both groups have some DNase 1 sensitivity. DESeq2 and Wellington-Bootstrap favor peaks in which one group has almost no DNase 1 sensitivity.

- FRET makes more discoveries at a given false discovery rate than either the Huber fixed-region test or WaveQTL.

- The rankings obtained based on WaveQTL are inconsistent with rankings obtained using all other methods.

Our results suggest that FRET is a powerful alternative for analyzing data with highly variable DNase 1 sensitivity patterns between samples. Interestingly, it appears to be more powerful than the Huber fixed-region test despite the fact that almost all of the detectable signal falls inside of peaks.

The primary disagreement between these methods is over the significance of peaks in which one or two resistant cell lines have high DNase 1 sensitivity while the rest have almost no sensitivity. DESeq2 and Wellington-Bootstrap are highly optimistic about these regions while FRET, the Huber fixed-region test and WaveQTL are more cautious.

It is possible that these regions are biologically interesting. Perhaps they are regions in which the susceptible cell lines are more tightly regulated than the resistant cell lines. However, we believe that few researchers would be comfortable making claims for significance of a differential region based on a difference observed in only one sample. We therefore prefer the behavior of FRET.

Chapter 5

CONCLUSIONS AND FUTURE DIRECTIONS

We have presented two approaches to the problem of identifying differential regions in genomic phenotype data. These two methods employ very different strategies and are best suited to somewhat different situations.

JADE is a tool for constrained estimation of phenotypic profiles. Constraining the profiles to be identical over most of the genome allows us to recover associated regions from the estimates. JADE can provide interpretable data summaries and can rank regions by their evidence for an association. In its current state of development, JADE cannot provide false discovery rate control.

This method may be best suited to analyses that are descriptive or exploratory, especially those in which the interest is in describing complex relationships between levels of a categorical trait. JADE is also useful for describing patterns in data with no or very few replicates. Unlike methods that rely on testing, replicates are unnecessary for JADE.

There are (at least) two features that would widen the utility of JADE as an analysis tool:

- False discovery rate control or significance estimation for discoveries
- Incorporation of other loss functions

These are both computationally challenging improvements. Permutation testing may be a reasonable strategy for significance testing but this would require re-running JADE (which is already computationally expensive) many times.

We learned in our analysis of DNase 1 sensitivity data that tests based on squared error loss can have dramatically reduced power when the data includes outliers — a problem that can be improved by using the Huber loss instead. We therefore suspect that allowing JADE to use the Huber loss might improve its performance for skewed data. This modification could be explored using off-the-shelf convex solvers before attempting to modify the specialized ADMM algorithm that solves the JADE optimization problem.

Our second proposal, FRET, is appealing for many of the applications to which JADE is ill-suited. FRET allows control of the region-wise false discovery rate, is robust to outliers and is computationally more practical (though with the need to perform permutations, it still requires a substantial amount of computing time). FRET doesn't provide profile estimates but it is applicable to more types of analysis, including those with continuous traits or a large number of trait levels (which would be intractable for JADE). Our exploration of DNase 1 sensitivity data suggests that FRET is a powerful alternative to fixed-region testing even when we can define good region boundaries a priori. Our results also suggest that FRET can do a better job of controlling the rFDR than tests that assume a distributional form for the data.

In this work we have focused on genomic phenotype analysis, as this is the biological problem that spurred our interest in the methods presented here. However, both JADE and FRET could easily apply to any problem in which the goal is to identify regions of signal from dense data. These might include time-course data or data associated with physical locations (in one dimension).

Extending FRET to function in two (or more) dimensions is an interesting and potentially useful direction. Making this extension should be fairly straight-forward. None of the foundational results limit the dimensionality of the test space. Allowing variable thresholds would simply require dividing the space, D, into cubes rather than intervals.

One immediate application of two-dimensional FRET is the identification of regulatory elements that interact using genomic phenotypes. We would look for pairs of regions in which the phenotypic value in one region is associated with the phenotypic value in the other. This could be done within one phenotype or even as a way of connecting multiple different genomic phenotypes. Practically, this type of analysis involves far too many tests to be done genome-wide but could provide useful insights on a more limited scale.

Association analysis with genomic phenotypes is a promising avenue for gaining greater insight into regulatory mechanisms that control organismal level variation. We have attempted to contribute practical methods that facilitate this endeavour.

BIBLIOGRAPHY

Altuna Akalin, Matthias Kormaksson, Sheng Li, Francine E Garrett-Bakelman, Maria E Figueroa, Ari Melnick, and Christopher E Mason. methylKit: a comprehensive R package for the analysis of genome-wide DNA methylation profiles. *Genome Biology*, 13(10):R87, oct 2012.

David Aldous. *Probability Approximations via the Poisson Clumping Heuristic : An Update*. Springer, 1989. ISBN 978-1-4757-6283-9.

Manuel Allhoff, Kristin Sere, Heike Chauvistre, Qiong Lin, Martin Zenke, and Ivan G. Costa. Detecting differential peaks in ChIP-seq signals with ODIN. *Bioinformatics*, 30(24):3467–3475, dec 2014.

Simon Anders and Wolfgang Huber. Differential expression analysis for sequence count data. *Genome Biology*, 11(10):1–12, jan 2010. ISSN 1465-6914.

Richard Arratia, Larry Goldstein, and Louis Gordon. Two Moments Suffice for Poisson Approximations: The Chen-Stein Method. *The Annals of Probability*, 17(1):9–25, jan 1989.

Richard Arratia, Larry Goldstein, and Louis Gordon. Poisson Approximation and the Chen-Stein Method. *Statistical Science*, 5(4):403–424, nov 1990.

Jordana T Bell, Athma a Pai, Joseph K Pickrell, Daniel J Gaffney, Roger Pique-Regi, Jacob F Degner, Yoav Gilad, and Jonathan K Pritchard. DNA methylation patterns associate with genetic and gene expression variation in HapMap cell lines. *Genome Biology*, 12(1):R10, jan 2011.

Y Benjamini and Y Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society B*, 57(1):289–300, 1995.

Yoav Benjamini and Ruth Heller. False Discovery Rates for Spatial Signals. *Journal of the American Statistical Association*, 102(480):1272–1281, dec 2007.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathon Eckstein. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2010.

Elvira Carrió, Anna Díez-Villanueva, Sergi Lois, Izaskun Mallona, Ildefonso Cases, Marta Forn, Miguel A. Peinado, and Mònica Suelves. Deconstruction of DNA Methylation Patterns During Myogenesis Reveals Specific Epigenetic Events in the Establishment of the Skeletal Muscle Lineage. *Stem Cells*, 33(6):2025–2036, 2015.

Alexandra Chouldechova. *False Discovery Rate Control for Spatial Data.* PhD thesis, Stanford University, 2014.

Leonardo Collado Torres, Abhinav Nellore, Alyssa C Frazee, Christopher Wilks, Michael I Love, Ben Langmead, Rafael A Irizarry, Jeffrey Leek, and Andrew E Jaffe. Flexible expressed region analysis for rna-seq with derfinder. *bioRxiv*, 2016.

Patrick Danaher, Pei Wang, and Daniela M. Witten. The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 76(2):373–397, 2014.

Alyssa C. Frazee, Sarven Sabunciyan, Kasper D. Hansen, Rafael A. Irizarry, and Jeffrey T. Leek. Differential expression analysis of RNA-seq data at single-base resolution. *Biostatistics*, 15(3):413–426, jul 2014.

Frank R Hampel, Elvezio M Ronchetti, Peter J Rousseeuw, and Werner A Stahel. *Robust statistics: The approach based on influence functions.* John Wiley and Sons, 1986. ISBN 0-471-82921-8.

Kasper D Hansen, Benjamin Langmead, and Rafael A Irizarry. BSmooth: from whole genome bisulfite sequencing reads to differentially methylated regions. *Genome Biology*, 13(10): 1–10, oct 2012.

Jennifer Harrow, Adam Frankish, Jose M. Gonzalez, Electra Tapanari, Mark Diekhans, Felix Kokocinski, Bronwen L. Aken, Daniel Barrell, Amondia Zadissa, Stephen Searle, If Barnes, Alexandra Bignell, Veronika Boychenko, Toby Hunt, Mike Kay, Gaurab Mukherjee, Jeena Rajan, Gloria Despacio-Reyes, Gary Saunders, Charles Steward, Rachel Harte, Michael Lin, Cedric Howald, Andrea Tanzer, Thomas Derrien, Jacqueline Chrast, Nathalie Walters, Suganthi Balasubramanian, Baikang Pei, Michael Tress, Jose Manuel Rodriguez, Iakes Ezkurdia, Jeltje van Baren, Michael Brent, David Haussler, Manolis Kellis, Alfonso Valencia, AAlexandre Reymond, Mark Gerstein, Roderic Guigo, and Tim J. Hubbard. GENCODE: The reference human genome annotation for The ENCODE Project. *Genome Research*, 22(9):1760–1774, sep 2012. ISSN 1088-9051.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, second edition, 2009. ISBN 9780387848570.

Felix Heinzl and Gerhard Tutz. Clustering in linear-mixed models with a group fused lasso penalty. *Biometrical Journal*, 56(1):44–68, jan 2014.

Toby Hocking, Jean-Philippe Vert, Francis Bach, and Armand Joulin. Clusterpath: an algorithm for clustering using convex fusion penalties. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pages 745–752, New York, NY, USA, June 2011. ACM. ISBN 978-1-4503-0619-5.

Peter J. Huber. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.

Peter J Huber. The behavior of maximum likelihood estimates under nonstandard conditions. In Lucien M. Le Cam and Jerzy Neyman, editors, *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 221–233, Berkeley, CA, USA, 1967. University of California Press.

Peter J. Huber. Robust Regression: Asymptotics, Conjectures and Monte Carlo. *The Annals of Statistics*, 1(5):799–821, 1973.

Peter J. Huber. *Robust Statistics*. John Wiley and Sons, 1981.

M. Hupkes, M. K. B. Jonsson, W. J. Scheenen, W. van Rotterdam, a. M. Sotoca, E. P. van Someren, M. a. G. van der Heyden, T. a. van Veen, R. I. van Ravestein-van Os, S. Bauerschmidt, E. Piek, D. L. Ypey, E. J. van Zoelen, and K. J. Dechering. Epigenetics: DNA demethylation promotes skeletal myotube maturation. *The FASEB Journal*, 25(11): 3861–3872, nov 2011.

Robert S. Illingworth and Adrian P. Bird. CpG islands - 'A rough guide'. *FEBS Letters*, 583(11):1713–1720, jun 2009.

Rafael A. Irizarry, Christine Ladd-Acosta, Bo Wen, Zhijin Wu, Carolina Montano, Patrick Onyango, Hengmi Cui, Kevin Gabo, Michael Rongione, Maree Webster, Hong Ji, James B Potash, Sarven Sabunciyan, and Andrew P Feinberg. The human colon cancer methylome shows similar hypo- and hypermethylation at conserved tissue-specific CpG island shores. *Nature Genetics*, 41(2):178–186, feb 2009.

Sam John, Peter J. Sabo, Robert E. Thurman, Myong-Hee Sung, Simon C. Biddie, Thomas A. Johnson, Gordon L. Hager, and John A. Stamatoyannopoulos. Chromatin accessibility pre-determines glucocorticoid receptor binding patterns. *Nature Genetics*, 43 (3):264–8, 2011.

Sam John, Peter J. Sabo, Theresa K. Canfield, Kristen Lee, Shinny Vong, Molly Weaver, Hao Wang, Jeff Vierstra, Alex P. Reynolds, Robert E. Thurman, and John A. Stamatoyannopoulos. Genome-scale Mapping of DNase1 Hypersensitivity. *Current Protocols in Molecular Biology*, pages 1–27, 2013.

Nicholas Johnson. A Dynamic Programming Algorithm for the Fused Lasso and L_0-Segmentation. *Journal of Computational and Graphical Statistics*, 22(2):246–260, apr 2013.

Seung-Jean Kim, Kwangmoo Koh, Stephen Boyd, and Dimitry Gorinevsky. ℓ_1 Trend Filtering. *SIAM Review*, 51(2):339–360, 2009.

Michael I Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, 15(12):1–21, dec 2014.

Pedro Madrigal and Paweł Krajewski. Current bioinformatic approaches to identify DNase I hypersensitive sites and genomic footprints from DNase-seq data. *Frontiers in Genetics*, 3:1–3, jan 2012.

Davis J. McCarthy, Yunshun Chen, and Gordon K. Smyth. Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation. *Nucleic Acids Research*, 40(10):4288–4297, 2012.

Jean Morrison, Daniela Witten, and Noah Simon. Joint Adaptive Differential Estimation: A tool for comparative analysis of smooth genomic data types. *Biostatistics*, In press, 2016.

Daniela Palacios and Pier Lorenzo Puri. The epigenetic network regulating muscle development and regeneration. *Journal of Cellular Physiology*, 207(1):1–11, apr 2006.

K. Pelckmans, J. De Brabanter, J. A. K. Suykens, and B. De Moor. Convex Clustering Shrinkage. In *Workshop on Statistics and Optimization of Clustering Workshop (PASCAL)*, 2005.

M Perone Pacifico, C Genovese, I Verdinelli, and L Wasserman. False Discovery Control for Random Fields. *Journal of the American Statistical Association*, 99(468):1002–1014, 2004. ISSN 0162-1459.

Jason Piper, Markus C. Elze, Pierre Cauchy, Peter N. Cockerill, Constanze Bonifer, and Sascha Ott. Wellington: A novel method for the accurate identification of digital genomic footprints from DNase-seq data. *Nucleic Acids Research*, 41(21):e201, nov 2013.

Jason Piper, Salam A Assi, Pierre Cauchy, Christophe Ladroue, Peter N Cockerill, Constanze Bonifer, and Sascha Ott. Wellington-bootstrap: differential DNase-seq footprinting identifies cell-type determining transcription factors. *BMC Genomics*, 16(1):1–8, jan 2015.

Aaditya Ramdas and Ryan J. Tibshirani. Fast and Flexible ADMM Algorithms for Trend Filtering. *Journal of Computational and Graphical Statistics*, jun 2015.

Christian H. Reinsch. Smoothing by spline functions. *Numerische Mathematik*, 10(3):177–183, 1967.

Mark D. Robinson, Davis J. McCarthy, and Gordon K. Smyth. edgeR: A Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140, jan 2009.

Jessica Segalés, Eusebio Perdiguero, and Pura Muñoz-Cánoves. Epigenetic control of adult skeletal muscle stem cell functions. *FEBS Journal*, 282(9):1571–1588, sep 2014.

Heejung Shim and Matthew Stephens. Wavelet-based genetic association analysis of functional phenotypes arising from high-throughput sequencing assays. *Annals of Applied Statistics*, 9(2):665–686, jul 2015.

D. O. Siegmund, N. R. Zhang, and B. Yakir. False discovery rate for scanning statistics. *Biometrika*, 98(4):979–985, 2011.

Jos F. Sturm. Using SeDuMi 1.02, A Matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11(1-4):625–653, 1999.

The Encode Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74, sep 2012.

Robert Tibshirani. Regression Shrinkage and Selection via the Lasso Robert Tibshirani. *Journal of the Royal Statistical Society B*, 58(1):267–288, 1996.

Robert Tibshirani and Pei Wang. Spatial smoothing and hot spot detection for CGH data using the fused lasso. *Biostatistics*, 9(1):18–29, jan 2008.

Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 67(1):91–108, feb 2005.

Ryan J. Tibshirani. Adaptive piecewise polynomial estimation via trend filtering. *The Annals of Statistics*, 42(1):285–323, feb 2014.

Virginia Goss Tusher, Robert Tibshirani, and Gilbert Chu. Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences*, 98(9):5116–21, 2001.

R. H. Tütüncü, K. C. Toh, and M. J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming, Series B*, 95(2):189–217, 2003.

W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002.

Hong Qiang Wang, Lindsey K. Tuominen, and Chung Jui Tsai. SLIM: A sliding linear model for estimating the proportion of true null hypotheses in datasets with dependence structures. *Bioinformatics*, 27(2):225–231, jan 2011.

Halbert White. A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity. *Econometrica*, 48(4):817–838, 1980.

Han Xu, Chia-Lin Wei, Feng Lin, and Wing-Kin Sung. An HMM approach to genome-wide identification of differential histone modification sites from ChIP-seq data. *Bioinformatics*, 24(20):2344–9, oct 2008.

Appendix A

TECHNICAL DETAILS AND ADDITIONAL SIMULATION RESULTS FOR CHAPTER 2

Here we include several supporting details for the discussion of JADE in Chapter 2.

A.1 Trend filtering

$\mathbf{D}^{k+1,s}$ in (2.2) is the discrete $(k+1)$st derivative operator for sites $\mathbf{s} = (s_1, \ldots, s_p)$. This matrix can be defined recursively. For $k = 0$,

$$\mathbf{D}^{1,s} \equiv \mathbf{D}^1 = \begin{pmatrix} -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{pmatrix}, \tag{A.1}$$

a $(p-1) \times p$ matrix that does not actually depend on $\mathbf{s}$. This corresponds exactly to the fused lasso [Tibshirani et al., 2005]. For $k \geq 1$,

$$\mathbf{D}^{k+1,s} = \mathbf{D}^1 \cdot \text{diag}\left(\frac{k}{s_{k+1} - s_1} \cdots \frac{k}{s_p - s_{p-k}}\right) \cdot \mathbf{D}^{k,s} \equiv \mathbf{D}^1 \tilde{\mathbf{D}}^{k,s} \tag{A.2}$$

as described in Ramdas and Tibshirani [2015] and Tibshirani [2014]. Equation A.2 admits a slight abuse of notation: $\mathbf{D}^1$ in (A.2) is the $(p-k-1) \times (p-k)$-dimensional version of (A.1).

A.2 Details of algorithm 1

A.2.1 Step 3(c)

When $M = 2$, the $\boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_2$ updates have a simple closed form, given in Danaher et al. [2014]:

$$\begin{aligned}\boldsymbol{\eta}_1 =& \text{sign}\left(\mathbf{z}_1 - \mathbf{z}_2\right) \cdot \max\left(\left|\frac{\mathbf{z}_1 - \mathbf{z}_2}{2}\right| - \frac{\gamma}{\rho_\eta}, 0\right), \qquad \boldsymbol{\eta}_2 = \frac{\mathbf{z}_1 + \mathbf{z}_2}{2}, \\ \boldsymbol{\eta}_1 =& \boldsymbol{\eta}_1 + \boldsymbol{\eta}_2, \qquad \boldsymbol{\eta}_2 = \boldsymbol{\eta}_2 - \boldsymbol{\eta}_1,\end{aligned}$$

where $\mathbf{z}_m = \boldsymbol{\theta}_m + \mathbf{u}_m^{(\eta)}$. Here, the sign and max operators are applied element-wise.

A.2.2 Step Size and Stopping Criteria

Step size update rules and stopping criteria are taken from Boyd et al. [2010]. The rules are based on primal and dual residuals, defined as

$$\begin{aligned}\mathbf{r}_{\text{primal}}^{(\alpha m)} &= \tilde{\mathbf{D}}^{k,s}\boldsymbol{\theta}_m - \boldsymbol{\alpha}_m, & \mathbf{r}_{\text{primal}}^{(\eta)} &= \boldsymbol{\eta} - \boldsymbol{\theta}, \\ \mathbf{r}_{\text{dual}}^{(\alpha m)} &= \rho_{\alpha m}^{old}\left[\tilde{\mathbf{D}}^{k,s}\right]^\top\left(\boldsymbol{\alpha}_m - \boldsymbol{\alpha}_m^{old}\right), & \mathbf{r}_{\text{dual}}^{(\eta)} &= \rho_\eta^{old}\left(\boldsymbol{\eta} - \boldsymbol{\eta}^{old}\right),\end{aligned}$$

where $\boldsymbol{\alpha}^{old}$, $\boldsymbol{\eta}^{old}$, and ρ^{old} are the $\boldsymbol{\alpha}$, $\boldsymbol{\eta}$, and ρ values from the previous iteration of the algorithm.

For the first 500 iterations of the ADMM algorithm, we update the step sizes as

$$\rho_* = \begin{cases} \tau_{\text{incr}}\rho_*^{old} & \left\|\mathbf{r}_{\text{primal}}^*\right\|_2 > \mu\left\|\mathbf{r}_{\text{dual}}^*\right\|_2 \\ \rho_*^{old}/\tau_{\text{decr}} & \left\|\mathbf{r}_{\text{dual}}^*\right\|_2 > \mu\left\|\mathbf{r}_{\text{primal}}^*\right\|_2 \\ \rho_*^{old} & \text{otherwise} \end{cases},$$

where $*$ indicates the $m + 1$ indices $\alpha 1, \dots, \alpha M$ and η; ρ_*^{old} indicates the step size at the previous iteration; and τ_{incr}, τ_{decr} and μ are parameters which we set to 2, 2 and 10 as suggested by Boyd et al. [2010]. We initialize $\rho_\eta = 1$ and $\rho_{\alpha m} = \lambda\left(\frac{\max(\{s_j\}) - \min(\{s_j\})}{p}\right)^{k-1}$ based on a suggestion in Ramdas and Tibshirani [2015].

We use the stopping criteria discussed in Section 3.3.1 of Boyd et al. [2010], terminating when

$$\begin{aligned}\|\mathbf{r}_{\text{primal}}\|_2 &\leq \epsilon^{\text{abs}}\sqrt{M(2p-k)} + \epsilon^{\text{rel}} \max\left(\sqrt{\sum_m \left[\left\|\tilde{\mathbf{D}}^{k,s}\boldsymbol{\theta}_m\right\|_2^2 + \|\boldsymbol{\theta}_m\|_2^2\right]}, \sqrt{\sum_m \left[\|\boldsymbol{\alpha}_m\|_2^2 + \|\boldsymbol{\eta}_m\|_2^2\right]}\right), \\ \|\mathbf{r}_{\text{dual}}\|_2 &\leq \epsilon^{\text{abs}}\sqrt{M(2p-k+1)} + \epsilon^{\text{rel}}\left(\sqrt{\left\|\tilde{\mathbf{D}}^{k,s}\mathbf{u}_m^{(\alpha)}\right\|_2^2 + \left\|\mathbf{u}_m^{(\eta)}\right\|_2^2}\right),\end{aligned}$$

where $\mathbf{r}_{\text{primal}} = \left(\mathbf{r}_{\text{primal}}^{(\alpha 1)}, \ldots, \mathbf{r}_{\text{primal}}^{(\alpha M)}, \mathbf{r}_{\text{primal}}^{(\eta)}\right)$, $\mathbf{r}_{\text{dual}} = \left(\mathbf{r}_{\text{dual}}^{(\alpha 1)}, \ldots, \mathbf{r}_{\text{dual}}^{(\alpha M)}, \mathbf{r}_{\text{dual}}^{(\eta)}\right)$, and ϵ^{abs} and ϵ^{rel} are parameters which, by default, we set to 10^{-4} and 10^{-8}.

A.3 Cross-validation of γ

The JADE optimization problem given in (2.4) has an equivalent constrained form,

$$\begin{aligned}\underset{\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_M}{\text{minimize}} \quad & \sum_{m=1}^{M} \frac{N_m}{2}\left\|\mathbf{A}_m(\bar{\mathbf{y}}_m - \boldsymbol{\theta}_m)\right\|_2^2 + \lambda \sum_{m=1}^{M}\left\|\mathbf{D}^{k+1,s}\boldsymbol{\theta}_m\right\|_1 \\ \text{subject to} \quad & \sum_{m<m'} \|\boldsymbol{\theta}_m - \boldsymbol{\theta}_{m'}\|_1 \leq C_\gamma.\end{aligned} \tag{A.3}$$

For each γ, there is a corresponding C_γ such that (2.4) and (A.3) have identical solutions. The mapping from γ to C_γ is quite complicated, and depends on the data. In practice, we have seen that solutions to (A.3) for a fixed value of C_γ are more similar across cross-validation folds than solutions to (2.4) for a fixed γ, so we choose to cross-validate based on C_γ rather than γ.

Unfortunately, it is difficult to solve (A.3) for a specified value of C_γ. Instead, we choose a grid of C_γ values, and in each fold of our cross-validation we find a grid of γ values that approximately covers those C_γ values. We then linearly interpolate to estimate test error for our specified grid of C_γ values.

A.4 Details of figures 2.3, 2.4, and 2.5 in Section 2.5

A.4.1 Calculation of curves

For the t-statistic methods, we allowed the significance threshold (the absolute value threshold at which a statistic is declared significant) to vary between 0 and the value of the largest statistic observed. For a given threshold value, and for a given simulated data set, we computed the true and false positive rates (TPR and FPR) using information about whether each site is part of a differential region. For a given simulated data set, we then linearly interpolated the corresponding TPR and FPR values. Finally, for each FPR value along a fine grid, we averaged the corresponding TPR values across the simulated data sets, in order to obtain the curves displayed in the figure.

For JADE, we varied the value of γ in (2.4). For each value of γ, and for a given simulated data set, we computed the TPR and the FPR of the corresponding JADE fit. For a given simulated data set, we then linearly interpolated the corresponding TPR and FPR values. Finally, for each FPR value along a fine grid, we averaged the corresponding TPR values across the simulated data sets, in order to obtain the curve displayed in the figure.

A.4.2 Calculation of colored points

For the t-statistic methods, for each simulated data set, we calculated the value of the significance threshold that resulted in an estimated false discovery rate of 10%, and calculated the TPR and FPR corresponding to this threshold. We then averaged these TPRs and FPRs over the simulated data sets, and displayed the resulting average FPR and average TPR using a colored point.

For JADE, for each simulated data set, we used cross-validation to select a value for γ, and calculated the corresponding TPR and FPR. We then averaged these TPRs and FPRs over the simulated data sets, and displayed the resulting average TPR and average FPR using a colored point.

A.5 Additional simulation results

A.5.1 Variable Sample Size

In Section 2.5, we present simulations using $M = 2$ groups of size $n_1 = n_2 = 10$. In practice, due to time and cost constraints, experiments tend to have small sample sizes. For example, the ENCODE project provides DNA methylation and more for a large number of cell types, with only one biological replicate for most cell types.

In this section, we explore the effect of sample size in the context of the normal simulations described in Section 2.5.1.

We considered two simulation settings:

Setting (i): We generated data as in (2.7), with $\epsilon_{imj} \sim N(0, 0.4 \cdot n_m)$.

Setting (ii): We generated data according to (2.7) and (2.8), with σ^2 and σ^2_{re} chosen so that $\sigma^2 + \sigma^2_{\text{re}} = 0.5 \cdot n_m$ and $\sigma^2_{\text{re}}/(\sigma^2 + \sigma^2_{\text{re}}) = 0.2$.

In each simulation setting, we generated data with $n_1 = n_2$ equal to 3, 5, 10, 20, and 50. Results are shown in Figure A.1. Using a larger sample size has little effect on JADE, but results in slightly higher power for the t-test methods.

A.5.2 Comparisons of Region-Level Performance Metrics

In Section 2.5, we investigate the site-level accuracy of JADE and the other methods. In this section, we instead perform a region-level analysis.

We treat consecutive sites within a differential region detected by JADE as a single discovery. For the t-statistic approaches, for a given threshold value, we define a discovery to be any string of three or more consecutive sites for which the t-statistic exceeds that threshold in absolute value. Discoveries separated by only one non-significant site are merged.

We define a true positive to be any discovery that overlaps a signal region, and a false positive to be any discovery that does not overlap a signal region. The TPR is defined to

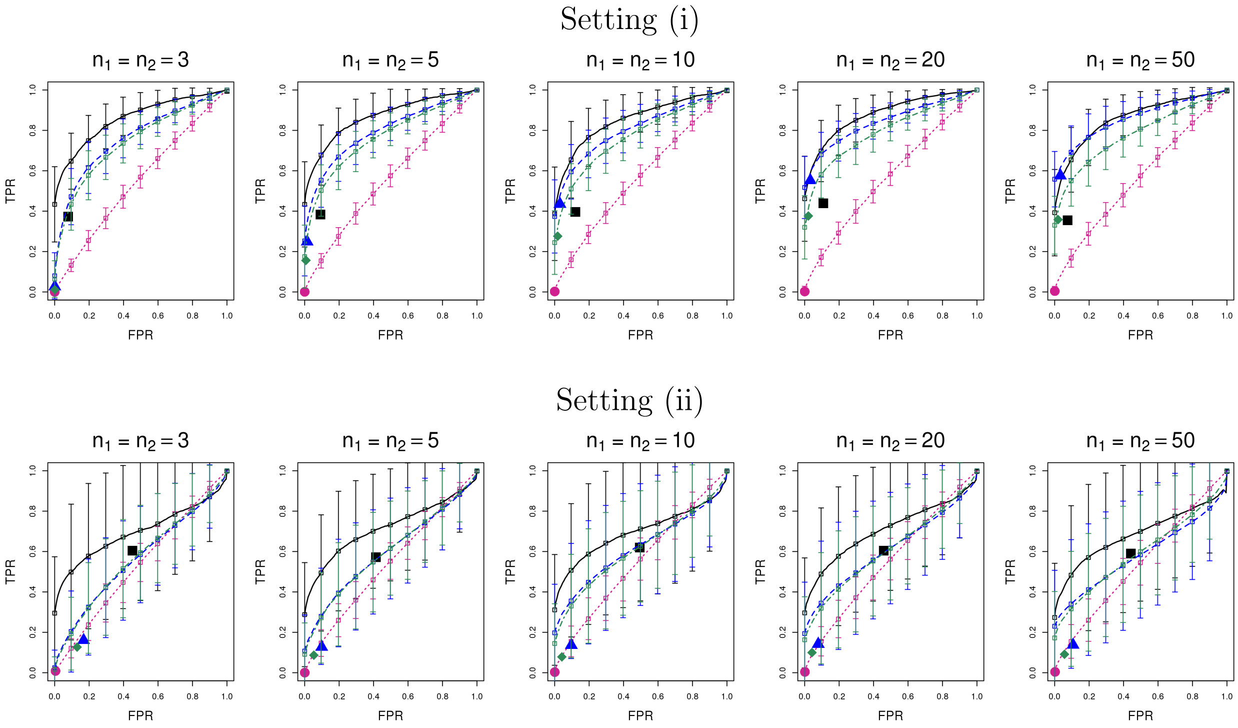


Figure A.1: Simulation study described in Section A.5.1. Curves display the average TPR for a fixed FPR, averaged over 100 simulations. The vertical bars indicate one sample standard deviation. Points indicate average TPR and FPR achieved for JADE with the tuning parameter selected by cross-validation, and for the *t*-test approaches with an FDR threshold of 10%. Methods shown are JADE (—,■), per-site *t*-tests applied to the raw data (···, ●), and per-site *t*-tests after smoothing the raw data using splines (– –, ▲) and local likelihood (- - -, ◆).

be the proportion of signal regions that overlap a discovery. This means that a method that makes multiple disjoint discoveries within one large signal region will be assigned the same TPR as a method that make a single discovery that exactly overlaps the signal region.

In this region-level analysis, it is hard to define the FPR, since there is no natural partition of non-signal sites into regions. Therefore, instead of considering the FPR, we simply consider the number of false positives.

These definitions of TPR and false positives are sensible when the discoveries span small regions. If, however, a method produces a few discoveries that span long regions containing most of the sites, then the method will have a high TPR and few false positives but qualitatively undesirable results. To avoid this problem, we limit our analysis to a range of thresholds for the t-statistic methods and γ values for JADE that result in discoveries that span less than 50% of the total region. Furthermore, we note that both site-level and region-level results should be considered when summarizing a method's accuracy.

Region-level summaries for the simulations presented in Figures 2.3, 2.4, 2.5 of the main text are shown in Figures A.2, A.3, and A.4. Additional details of how these figures were generated are provided in Section A.4, with FPR replaced with the number of false positives. The figures indicate that all methods perform very well in terms of region-level metrics, with the exception of methylKit in the binomial simulations shown in Figure A.4.

A.6 Read tiling in binomial simulations

We now describe the strategy used to generate n_{imj} in Section 2.5.2. In order to mimic the variable read depth observed in methylation sequencing data, reads at each position are assigned by layering contiguous tiles. A schematic of 30 tiles is shown in Figure A.5. A tile is placed by sampling a length from an Exponential(30) distribution, and a start point from a Uniform(-30, 300) distribution. Portions of tiles extending above 300 or below 1 are discarded. For each observation, 110 tiles are placed initially, so that in expectation there are 10 reads per site. For each site that has zero reads at the end of this procedure, one additional tile is drawn using the sampling scheme above, conditional on covering the zero-

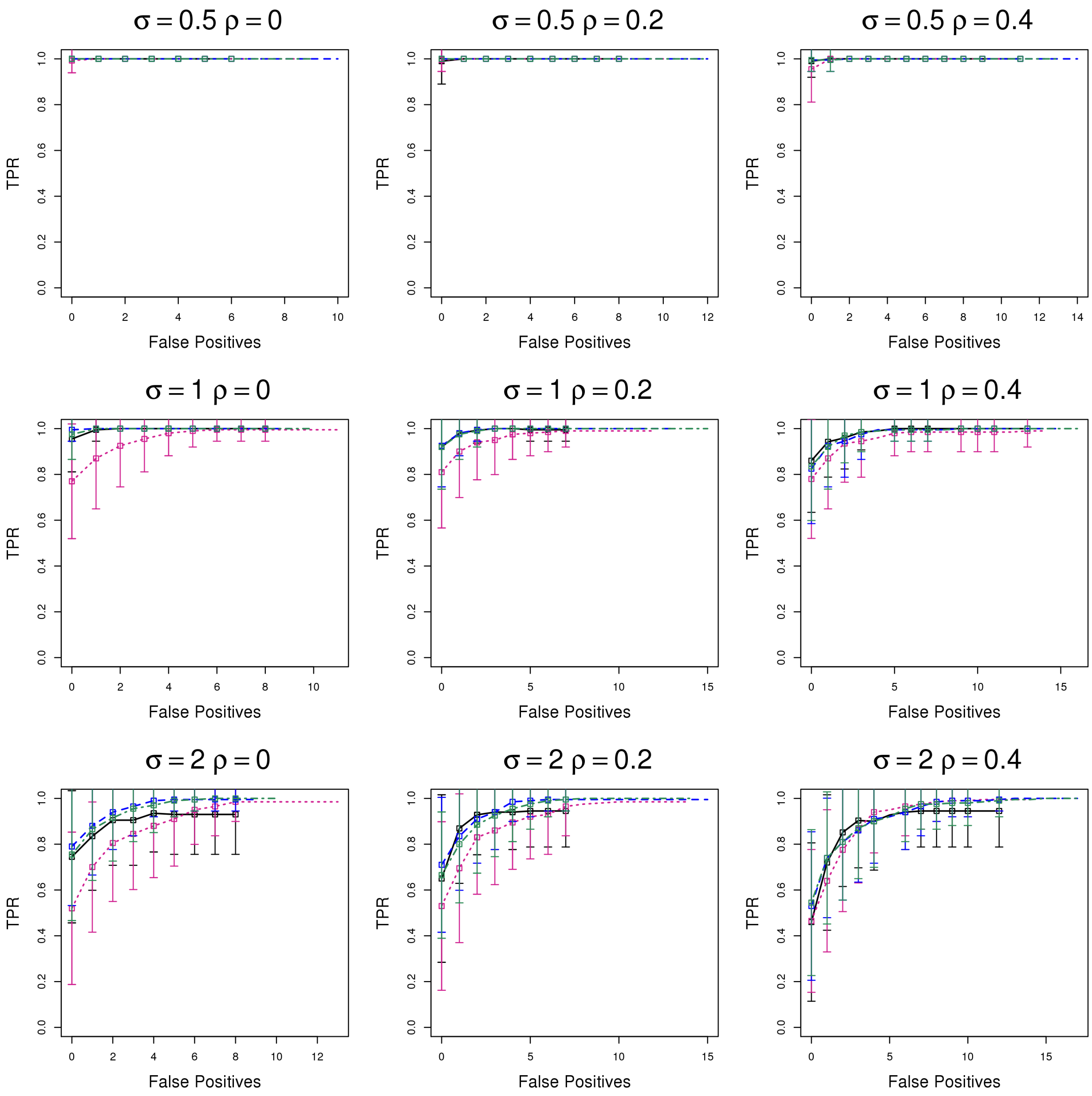


Figure A.2: Performance of JADE and competing methods in the normal auto-regressive simulations presented in Figure 2.3 of the main text. Here, performance is quantified using a region-level analysis, described in Section A.5.2. Curves display the average region-level TPR, for a fixed number of false positives. The vertical bars indicate one sample standard deviation. Methods shown are JADE (—), per-site t-tests applied to the raw data (⋯), and per-site t-tests after smoothing the raw data using splines (– –) and local likelihood (– · –).

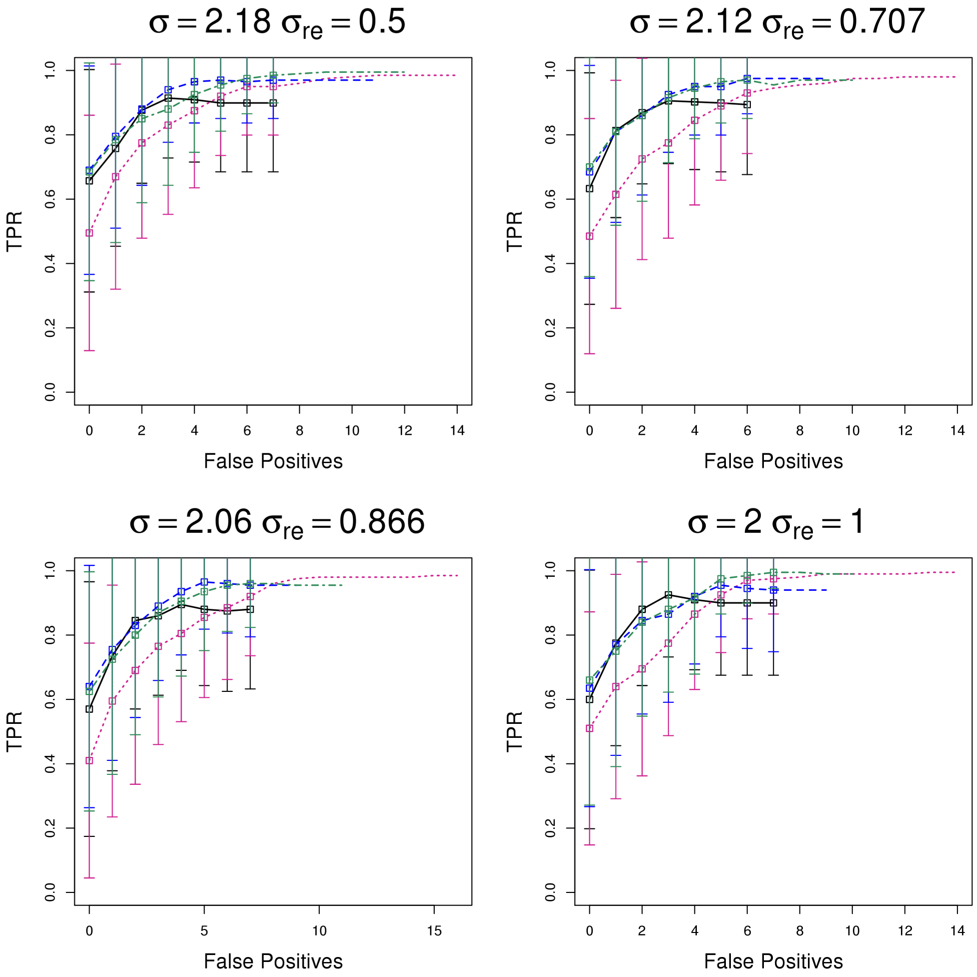


Figure A.3: Performance of JADE and competing methods in the normal random effects simulations presented in Figure 2.4 of the main text. Here, performance is quantified using a region-level analysis, described in Section A.5.2. Details are as in Figure A.2.

read site. This procedure guarantees that every site is covered, while keeping the expected coverage of each site close to 10 reads.

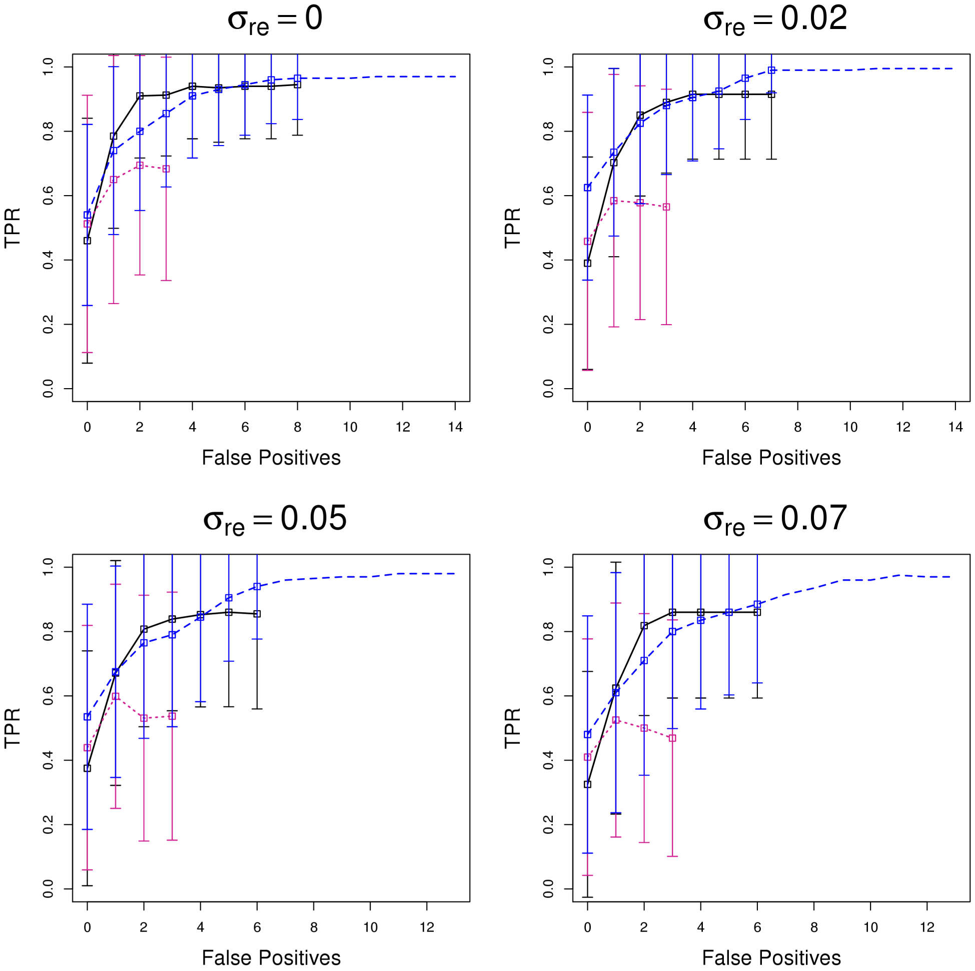


Figure A.4: Performance of JADE and competing methods in the binomial simulations presented in Figure 2.5. Here, performance is quantified using a region-level analysis, described in Section A.5.2. Methods shown are JADE (—), methylKit (- - -), and BSmooth (– –). Additional details are as in Figure A.2.

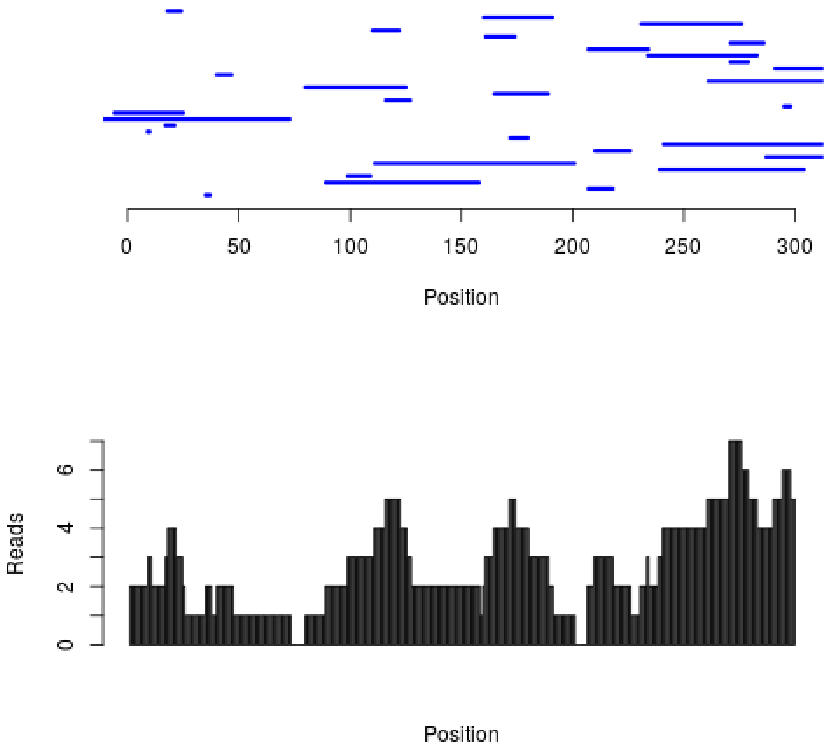


Figure A.5: Read tiling for binomial simulations in Section 2.5.2. The tiling procedure is described in Section A.6. *Top panel:* Read tiles. *Bottom Panel:* Total reads.

Appendix B
ADDITIONAL DISCUSSION FOR CHAPTER 3

B.1 Smooth then test or test then smooth?

In this section we explore the differences between two similar point-wise testing strategies through simulations. These two strategies are:

Smooth then test (ST): This is the BSmooth strategy. First, the phenotype for each sample is smoothed. Next, a test statistic at each position is calculated using the smoothed phenotype values. The null hypothesis is rejected at positions where the test statistics exceed some threshold in absolute value.

Test then smooth (TS): This is the excursion procedure strategy. First, a test statistic is calculated at each position and then these statistics are smoothed. The null hypothesis is rejected at positions where the smoothed test statistics exceed some threshold in absolute value.

B.1.1 Normal Data

We first consider the simulation set-up used in Section 2.5.1 of Chapter 2. In these simulations, we generate phenotypes at 300 sites for 10 samples in each of two groups. The data for the ith sample in the mth group at the jth site is generated as

$$y_{imj} = f_m(s_j) + \epsilon_{imj}, \tag{B.1}$$

where the functions f_1 and f_2 represent the mean genomic phenotype profiles for the two groups, and are displayed in Figure 2.2. Our goal is to identify the two regions over which the mean profiles for the two groups differ (these have length 74 and 37 positions respectively). Here, we consider only the setting when $\epsilon_{imj} \sim N(0, 4)$. For both the ST and TS methods, the smoother used is a moving average with width 5, 20, or 50 positions and the test statistic is a two sample t-test.

Figure B.1 shows the point-wise and region-wise performance of the ST and TS methods. The curves on the left describe point-wise performance and are calculated as described in Appendix Section A.4. These curves show the average true positive rate for a fixed false positive rate averaged over 100 simulations. The curves on the right show the region-wise metrics for the two methods, where regions are defined as blocks of contiguous points at which the null hypothesis is rejected (without using the merging procedure in Section 3.2.3). Recall that a region is defined as a false discovery if it does not contain any truly differential sites. A true differential region is considered to be "detected" if it is overlapped by a discovery. The curves on the right of Figure B.1 show the average proportion of differential regions detected for a fixed number of false discoveries, averaged over 100 simulations.

For each smoothing bandwidth, the ST and TS methods have nearly identical performance. In fact, we find that in any simulation, the statistics for the two methods are nearly identical up to a scaling constant. We obtain the highest power at a bandwidth of 50 because both of the differential regions are longer than 20 positions wide. Using a longer bandwidth allows us to combine more information within a differential region.

In Section 2.5.1 we consider settings in which the error terms, ϵ_{imj}, include sample-specific random effects as well as settings in which they have an auto-regressive dependence structure across sites. In all of these settings, we find similar results to those shown in Figure B.1 — the ST and TS methods are almost identical.

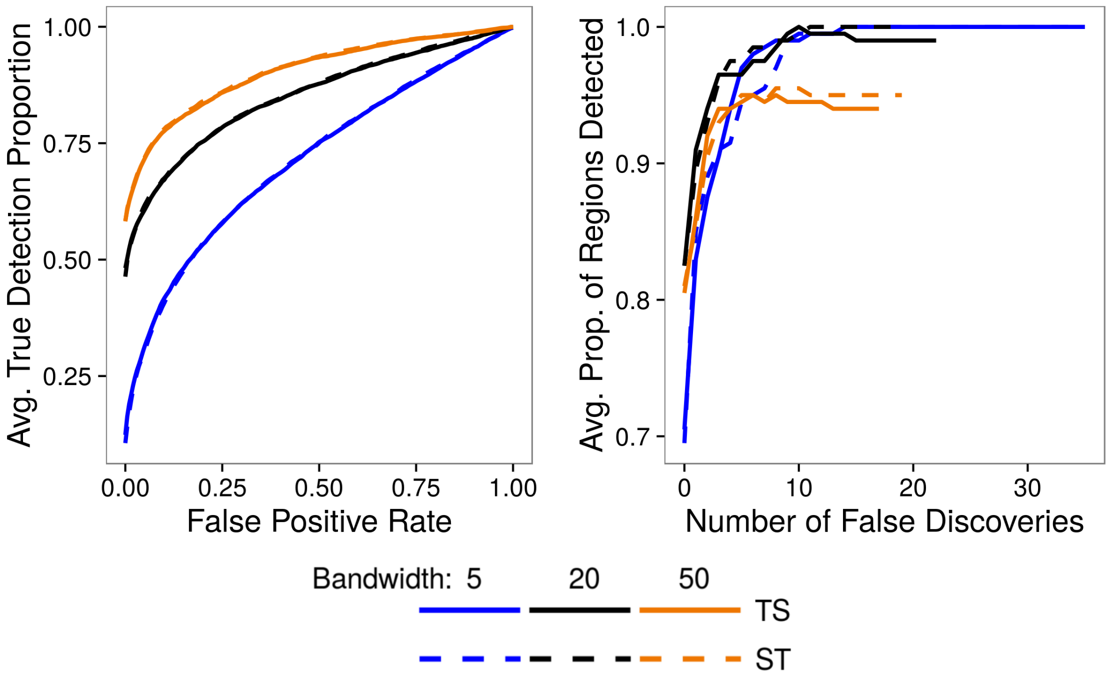


Figure B.1: Performance of ST and TS testing methods for auto-regressive simulations described in Section 2.5.1 with $\rho = 0$ and $\sigma = 2$. *Left:* The average proportion of differential sites detected is shown as a function of the point-wise false discovery rate. *Right:* The average number of differential regions overlapped by a discovery is shown as a function of the number of false discoveries.

B.1.2 Poisson Data

Next, we consider the simulations described in Section 3.5 of Chapter 3. In these simulations, the phenotype value for sample i at position j is generated as

$$y_{ij} \sim \text{Poisson}(\gamma_{ij})$$

where γ_{ij} is the value of a sample specific mean profile at site j. Each sample is also assigned a trait value, x_i. The sample specific profiles contain peaks as shown in Figure 3.7. The heights of some peaks depend on the trait value — these peaks are the differential regions we hope to detect. In Section 3.5.1 we describe how profiles are created according to the schematic in Figure 3.8. We describe six types of peaks: types 1 and 2 are "noise" peaks (heights are not dependent on the trait) while types 3-6 are "signal" peaks (heights do depend on the trait). Here we consider the setting in which x_i is binary and type 3 signal peaks are used in place of the $*$ in Figure 3.8. The heights of these peaks are distributed multinomially as described in Section 3.5.1.

As in Section B.1.1, we use a moving average with bandwidth 5, 20, and 50 positions wide. For both the ST and TS methods we use a two-sample t-test, though we obtain very similar results using the Huber test described in Section 3.3.

Figure B.2 shows point-wise and region-wise performance of the ST and TS methods. Curves are calculated as for Figure B.1. We find that the TS method has better performance as measured by both point-wise and region-wise metrics. We find the same patterns using the other signal peak types described in Section 3.5.1. All differential regions have width 20 positions, so this smoothing bandwidth gives the best performance.

We find that the ST method suffers in when the data in differential regions have much higher variance than the data outside of differential regions. In this setting, this occurs primarily because there is heterogeneity within trait levels inside of signal peaks, but no heterogeneity outside of peaks. We find that the ST and TS test statistics are nearly identical up to a scaling factor except in heterogeneous peaks.

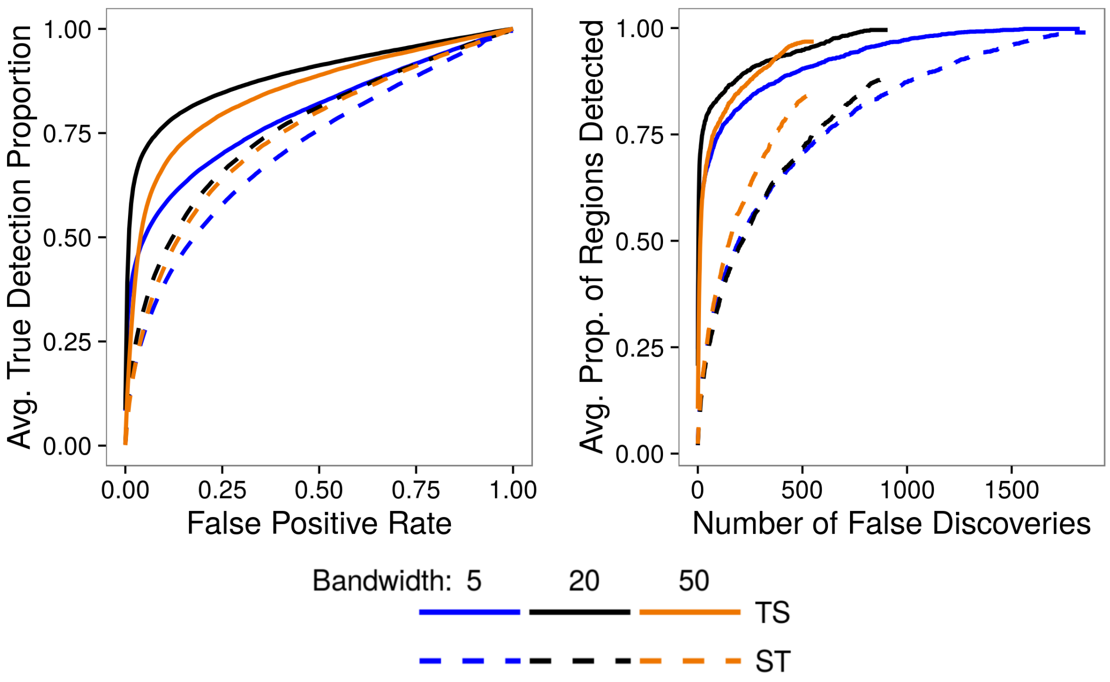


Figure B.2: Performance of ST and TS testing methods for simulations described in Section 3.5 using type 3 signal peaks. *Left:* The average proportion of differential sites detected is shown as a function of the point-wise false discovery rate. *Right:* The average number of differential regions overlapped by a discovery is shown as a function of the number of false discoveries.

Appendix C
ADDITIONAL DETAILS FOR CHAPTER 4

Here we include a few additional details of the analysis of DNase 1 data in Chapter 4.

C.1 Selection of σ_0 and z^*

We chose the values of σ_0 and z^* in Algorithm 6 using a randomly selected 500kb "test" segment of chromosome 1. First we choose σ_0: We calculate the Huber regression coefficient $\hat{\beta}_j^H$ and an estimate of its variance $\hat{V}_j^H$ as described in Section 3.3.1 at each base-pair in the test segment. We then select σ_0 to be 0.05 by following the procedure of Tusher et al. [2001] and calculate the statistics $T_j^{H,0.05}$ in (3.11).

Since many base-pairs have almost no DNase 1 sensitivity in any sample, to save computation we set a very permissive minimum sensitivity level of 5 total normalized cleavages summing over all samples. At base-pairs not meeting this criterion, the test statistic is always very close to zero. Therefore, here and throughout the analysis, to save computation, we set $T(s_j)$ to 0 for all positions not achieving this minimum.

Next we choose z^* and z_0: In the simulations presented in Section 3.5, we select z^* to be the 90th percentile of the smoothed test statistics. In this data, that approach cannot be used because most of the genome has very low DNase 1 sensitivity — Approximately 90% of the base-pairs in the test segment do not meet the minimum sensitivity requirement. In low sensitivity areas, test statistics will always be small. Since z^* must be large enough that the Poisson approximation holds even in high sensitivity regions where it is possible to achieve larger statistics, we select z^* based on the distribution smoothed test statistics in the the *higher sensitivity* portions of the test segment.

We first retrieve the smoothed test statistics at base-pairs meeting the minimum sensitivity requirement in the test segment (about 50k base-pairs). We then select the subset of these at which more than a third of the surrounding 50 base-pairs also meet this criterion (about 40% or 20k positions). We chose z^* to be the 90th percentile of the smoothed statistics for this subset of positions which we found to be 0.9.

The value of z^* could have been selected using other subsets of positions or quantiles of the smoothed test statistics. These choices have a minimal effect on the analysis because FRET is quite robust to the choice of z^* as long as it is large enough that excursions are rare though very large values can make the procedure more conservative.

C.2 Hotspot calling and peak master list construction

Hotspots were called by our collaborators using the algorithm developed by John et al. [2011]. These are regions of enriched cleavage density relative to the local background. The algorithm of John et al. [2011] first assigns a score to each position by comparing the number of cleavages in a small 200-300 base-pair window centered at the target position to the number of cleavages observed in the large 50 kb window also centred at the target position. This score is calculated based on a binomial model that treats the number of cleavages in the small window as a sample from the larger window.

Hotspots are defined as contiguous blocks of base-pairs with a large score. Once hotspots have been defined, each hotspot is itself assigned a score in a similar way to the individual position scores — By comparing the number of cleavages in the hotspot to the number of cleavages in the large surrounding window.

Each hotspot is then assigned an FDR value based on its score. The FDR value for a hotspot is calculated based on simulated "null" data. First, cleavages are randomly distributed over the genome. Next, hotspots are recalculated with the simulated data. Finally, the FDR value assigned to a hotspot with score T is given by

$$FDR(T) = \frac{\text{number of random hotspots with score } > T}{\text{number of observed hotspots with score } > T}$$

Peaks are defined as the 150 base-pair windows with highest cleavage density within hotspots. In order to combine peaks called in the 25 samples, we used code provided by our collaborators that attempts to identify "consensus peaks" with many samples. First all peaks are merged. Within each resulting interval, the peak that belongs to the hotspot with the highest score is retained for the master list. Any peaks that overlap this selected peak by more than 25% are discarded. This process is repeated until all the peaks have either been discarded or added to the master list. Because a threshold of 25% is used for discarding peaks, the resulting master list contains a small number of overlapping peaks. We therefore merged any overlapping peaks to achieve our final list of 2,570,268 peaks.

C.3 Additional fixed-regions tests

We considered two additional fixed-region tests not described in Section 4.2.2. These are

1. Quasi-Poisson regression statistic calculated using `glm` in `R` applied to sum of normalized counts in each peak

2. A t-test applied to the sum of normalized counts in each peak

Neither method detected any differential regions at low FDR thresholds. Quantile-quantile plots for the p-values produced by these tests are shown in Figure C.1.

The quasi-Poisson test differs from DESeq2 in that it allows a separate over-dispersion parameter for each peak rather than combining information across peaks (and uses a slightly different form for the variance). DESeq2 forces the dispersion parameter to be a smooth function of the normalized mean DNase 1 sensitivity. This suggests that DESeq2 finds more evidence for association as a result of making smaller variance estimates.

The primary difference between the t-test and the Huber fixed-region test is the treatment of outliers. The Huber test down-weights large observations and, as a result, makes lower variability estimates of $\boldsymbol{\beta}_j$ with noisy data. As seen in the numerical experiments in Section 3.3.3, this can give the Huber test dramatically better power.

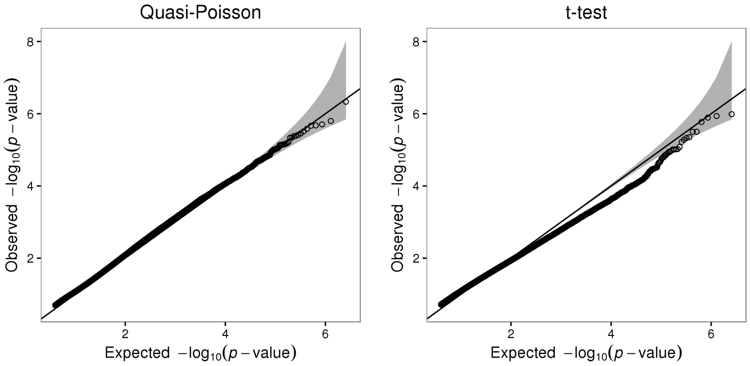


Figure C.1: Quanitle-quantile plots for $-\log$ transformed p-values produced by quasi-Poisson and t-test fixed-region tests. The solid line and grey shading show the expectation and 95% confidence interval for quantiles of the Uniform(0, 1) distribution.