The University of Melbourne

Hidden Markov Tree priors in multiscale models for genomic data

by

Brendan Kai Hong Law

Supervised by Dr. Heejung Shim

A thesis submitted in partial fulfillment for the
degree of Master of Science (Mathematics and Statistics)

in the
School of Mathematics and Statistics

October 2019

“If I have seen further, it is by standing on the shoulders of giants.”

– Isaac Newton

Abstract

Genome-wide association studies (GWAS) aim to identify potential genetic variants which are associated with variations of traits – or *phenotypes* – such as disease susceptibility in humans. While these studies focus on organismal-level phenotypic variation, identifying variants which influence behaviour at the molecular level can help us better understand the biological mechanisms behind such organismal-level variations. Measuring molecular-level behaviour has been made possible by the advent of high-throughput sequencing (HTS), which provides sequencing data at increased granularities, such as at the DNA base-pair level. However, few existing statistical methods have been designed to deal with the increased granularity, and potentially, noise, present in such data.

This project is centred around a multiscale approach to genetic association analysis of HTS data. The key ideas stem from using a Wavelet-based approach within a computationally efficient Bayesian framework to extract the smoothed underlying association signal function (Shim and Stephens (2015)). An alternative approach to this problem uses multiscale models for Poisson processes to directly model the count nature of the sequencing data (Shim et al. (in preparation), Xing (2016)).

We review key ideas from each of these models, then extend both by imposing a Hidden Markov Tree (HMT) prior on the association signals, in order to better capture the structure in the signal at different scales and locations in the multiscale space. This contrasts with the priors used in the previous methods, which assume independence between different scales and locations. We compare models on simulated data, and demonstrate where our method shows improvement compared to the existing methods.

Acknowledgements

Firstly, I'd like to thank my supervisor, Heejung Shim, for all your help with the standard supervisor things; giving me the chance to work on this project, helping chart its course, and for being generous with your time in responding to all the speed bumps I threw in along the way. More importantly, beyond the standard things, thanks for being a great person to talk to – beyond just research – and to work with, all throughout.

Secondly, none of this would have been possible without the unwavering support and flexibility that my employer has afforded me. Thanks, especially, to my managers throughout this time – Sam, Sumith and Noah – you have each played a significant part in quelling any uncertainties I initially had about how I would maintain this three-year juggling act.

Thirdly, thanks to all my friends, both within and outside the course. To those within, for being a great source of support throughout – it's always comforting to know that you are not alone in riding the ups and downs. To those outside, for putting up with my poor, infrequent attendance at catch-ups and brunches. I'll try and make it up to you all, I promise.

Lastly, thanks to Alice, my partner in crime (not literally), for being a supportive rock and for constantly reminding me that I need to calm down, and thanks to my family for once again supporting me all the way through my course, from start to finish, just like last time, and all the years before that. Your care and love is felt everyday.

In particular, to my Mum, I'm so glad we'll be able to get to the end together.

Contents

Abbreviations

SNP	**S**ingle **N**ucleotide **P**olymorphism
GWAS	**G**enome-**W**ide **A**ssociation **S**tudies
HTS	**H**igh **T**hroughput **S**equencing
HMT	**H**idden **M**arkov **T**ree
DWT	**D**iscrete **W**avelet **T**ransform
WC	**W**avelet **C**oefficient
IDWT	**I**nverse **D**iscrete **W**avelet **T**ransform
BF	**B**ayes **F**actor
EM	**E**xpectation **M**aximisation
EB	**E**mpirical **B**ayes
MLE	**M**aximum **L**ikelihood **E**stimate
auROC	**a**rea **u**nder the **R**eceiver **O**perating **C**haracteristic
PDF	**P**robability **D**istribution **F**unction
ABF	**A**pproximate **B**ayes **F**actor
DP	**D**ynamic **P**rogramming

Chapter 1

Introduction

Understanding the factors which influence differences between humans is a mammoth task. Common areas of study include the increased susceptibility of specific diseases among particular individuals, as well as the variation of observed physical traits (e.g. height, eye-colour) between individuals, both of which are known as organismal-level traits, or *phenotypes*. While environmental factors account for variation in some phenotypes, genetic variation also plays a significant role. For all the differences we observe between ourselves, human DNA contains surprisingly little variability, in terms of the acids found at each base of our DNA. Nevertheless, there are some loci along the DNA which do contain relatively common variations (present in at least 1% of individuals) which are termed *polymorphisms*, of which a common type is the Single Nucleotide Polymorphism (SNP) (Thomas et al. (2004)). Such variation in our DNA – described as an individual's *genotype* – has formed the basis by which researchers have been able to conduct Genome-wide association studies (GWAS); studies which aim to identify genetic variants which are associated with organismal-level phenotypes.

Although GWAS have helped to identify numerous candidate genetic loci to investigate further, the studies themselves provide little guidance on understanding the precise biological mechanisms behind such associations. Instead, identifying genetic variants which are associated with *molecular* phenotypes – behaviours on a molecular level, such as gene expression, chromatin accessibility and transcription factor binding – have emerged as an important tool for explaining such mechanisms, and therefore, driving our understanding of relationships between genetic and organismal-level phenotypic variation (Albert and Kruglyak (2015)). However, only in the past decade have we seen the proliferation of methods and technologies which have made it possible to measure molecular-level behaviour, and therefore, to be able to study the association of genetic and phenotypic variants at a molecular level. One of these advances is the advent of cheap high-throughput sequencing (HTS) technologies, which are characterised by their high sequencing speed and unprecedented level of output data granularity, such as at a base pair level – the units which form the building blocks of our DNA.

1.1 Association analysis from HTS assays

The accessibility of such detailed data allows molecular-level association analyses to focus on both identifying potential genetic variants associated with molecular-level phenotypic variants, as well as quantifying the size of the effect that particular genotypes have on molecular-level phenotypic variation at particular locations along the genome. However, typical analyses which use data from HTS fail to exploit the high-resolution measurements. Previous studies (e.g. Degner et al. (2012)) aggregate data at coarser resolutions, such as at the gene-level, or by using windows with fixed lengths. This approach is suitable if the effect of interest has some known, fixed length, but would not be well powered in identifying effects of differing lengths. In particular, effects which are far broader than a given window do not allow all the information from the signal to be used, effects which are far narrower risk not being identified due to their narrow length relative to the window, and effects where the effect size moves in opposite directions within a given window would fail to be captured as the quantities in opposite directions risk cancelling each other out (Shim and Stephens (2015)) – see Figure 1.1 for an illustration. A key motivator behind using multiscale methods is to design a method which is flexible and adaptive with regard to this.

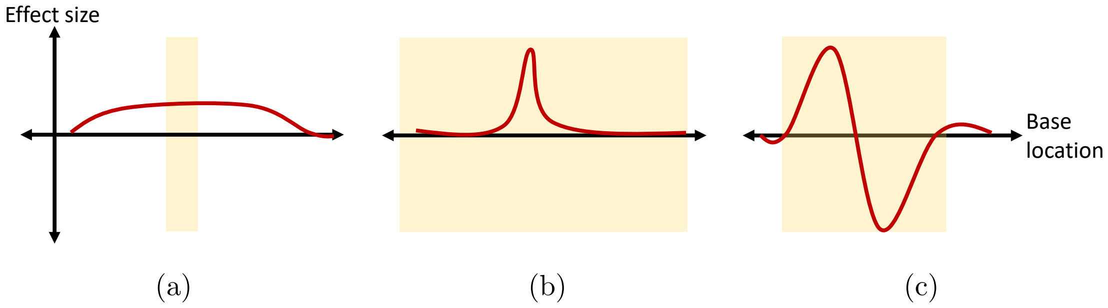


Figure 1.1: Three situations where window-based methods struggle (window length shown in yellow): (a) effect is broader than window, (b) effect is narrower than window, (c) effect size moves in opposite directions within the window.

1.1.1 Properties of the data

We use a sample of the data from Degner et al. (2012) (which was also analysed in Shim and Stephens (2015)) to provide an example of the type of data intended for use by our model. This data is a sequence of counts at each base along the human genome obtained from a process called *DNase-seq*, which measures a molecular-level behaviour that is a major determinant of variation in gene expression, and likely to be an important contributor to organismal-level phenotypic variation (Degner et al. (2012)). In summary, the molecular-level behaviour which is measured by the DNase-seq process is the accessibility, or openness, of chromatin – a substance found in cell nuclei – along the genome. This is measured by having an enzyme selectively cut the DNA at locations where the chromatin is more accessible, with a higher frequency of cuts generally associated with a higher level of chromatin accessibility.

We observe a sequence of the normalised count of cut points, d_b, at each base, b, on the genome. The data we use is DNase-seq data at region chr17.10160989.10162012, spanning 1024 bases. An example of this sequence for an individual is shown in Figure 1.2, and when averaged over 70 individuals, shown in Figure 1.3.

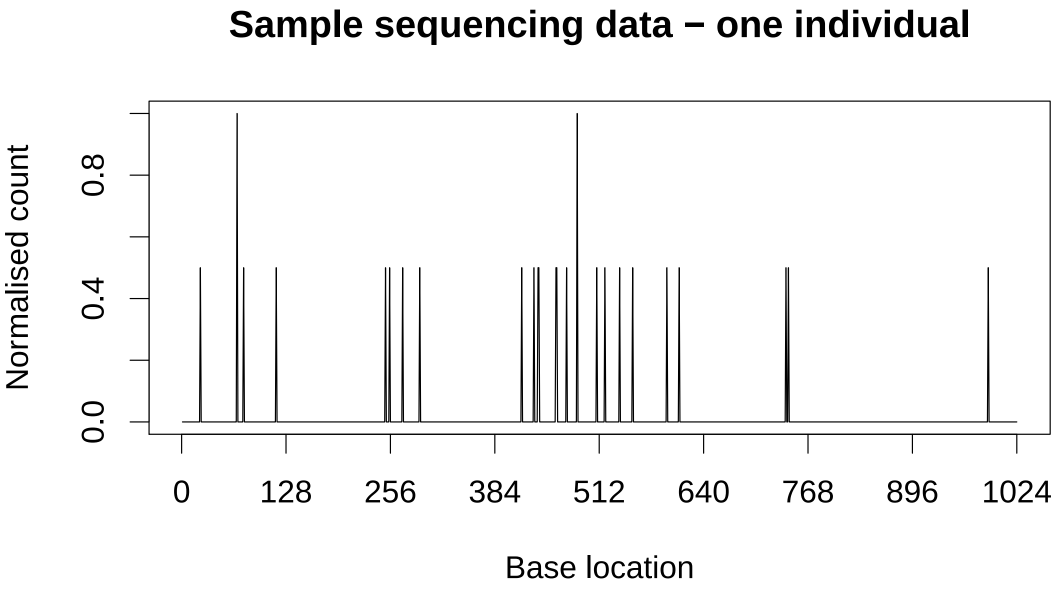


Figure 1.2: Sequencing data from one individual across all 1024 bases

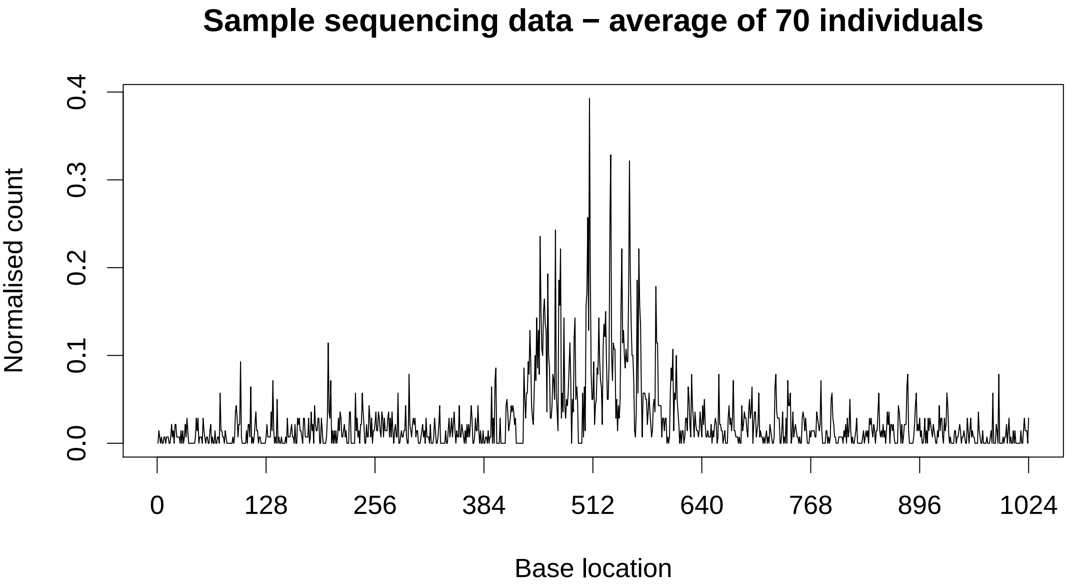


Figure 1.3: Averaged sequencing data across 70 individuals, across all 1024 bases

The covariate in this dataset is a representation of each individual's genotype data at each SNP. In this case, it is represented by a number – 0, 1 or 2 – which denotes the number of copies of the minor allele at a certain SNP of interest. Therefore, the statistical problem of interest for this dataset would be to identify which particular genetic variants are associated with chromatin accessibility (proxied by the sequencing count), as well as quantifying the strength of the association – the size of the effect that particular genetic variants have on the accessibility of chromatin – at a specific base pair. To help visualise such a relationship, we plot the average sequencing count, like in Figure 1.3, but grouped by each individual's genotype, yielding Figure 1.4. Although methods do exist which test for association with genotypes at multiple SNPs (Servin and Stephens (2007), Shim and Stephens (2015)), our work will focus on the case where the genotype at one SNP is tested at a time.

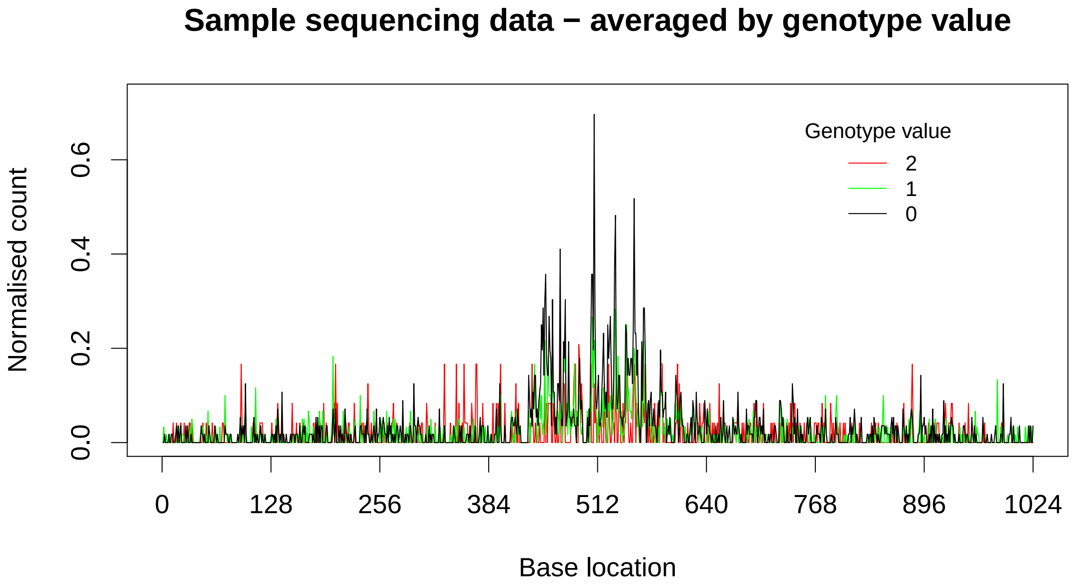


Figure 1.4: Averaged sequencing data across 70 individuals, by genotype value

We see that the data is somewhat noisy – many individuals have zero counts at many bases, yet the occasional non-zero count at a given base may not be reflective of any true signal at that base either. Furthermore, we can see potential differences between the average sequencing counts from different genotype values (represented by colours), but observe that the signals are local and contain spatial structure. Therefore, methods which tackle this problem must be able to effectively combine information from different bases to increase their overall power in detecting association signals.

Wavelet-based (Shim and Stephens (2015)) and Poisson-based (Xing (2016), Shim et al. (in preparation)) models have been developed which aim to utilise the granularity of the HTS data and to deal with the noisy and locally-structured nature of such data. Most importantly, all of these works emphasise computational efficiency, allowing them to be used at scale for genetic analyses spanning hundreds of thousands of tests. Our work builds upon both the Wavelet-based and Poisson-based frameworks, by addressing a key independence assumption upon which they are based upon. Specifically, as we will see, the nature of this data means that there is an underlying structure between the association signals in a multiscale space, and we address this by incorporating a Hidden Markov Tree (HMT) structure to the relevant priors in our model, in order to capture some of the dependencies between the signals being modelled.

1.2 Thesis structure

Chapter 2 will begin with a review of the Wavelet-based association analysis work in Shim and Stephens (2015). We then discuss limitations of this work, and introduce the HMT approach, as described in Crouse et al. (1998) and Ma and Soriano (2018), and how it can be incorporated into our model. Chapter 3 is dedicated to discussing simulation studies using the HMT-based model constructed in Chapter 2.

Chapter 4 discusses a variant of the model introduced in Chapter 2 – one which is more suitable for count data – a significant variant given that our model deals with sequencing counts. We review the ideas from Kolaczyk (1999), Timmerman and Nowak (1999) and subsequently, Xing (2016) and Shim et al. (in preparation), where such a model has been built with independent priors. As before, we show how we can adapt the HMT framework into this model.

We wrap up in Chapter 5 with some concluding remarks, and an outline of some discussion points arising from this thesis.

Chapter 2

Wavelet-based models with HMT priors

The HTS data we are working with can be thought as noisy measurements of an underlying function that has spatial structure along the genome with many local features. Shim and Stephens (2015) introduced a wavelet-based method, *WaveQTL,* that better uses the properties of the HTS data for identification of genetic variants associated with molecular-level phenotypes. They borrowed ideas from the 'functional mixed models' work of Morris and Carroll (2006), first transforming the HTS data using a wavelet transform, and then modelling associations in the (multiscale) wavelet space, rather than the original data space. However, a limitation of *WaveQTL* is that it ignores dependencies between associations in the wavelet space. Here, we propose methods to better model the dependencies by exploiting the tree structure in wavelets as described in Crouse et al. (1998) and Ma and Soriano (2018).

Section 2.1 will provide a brief overview of some key ideas regarding wavelets. Section 2.2 will review the wavelet-based method, *WaveQTL,* from Shim and Stephens (2015) and describe its limitations. In Section 2.3, we will introduce the proposed method that better models the dependencies in the wavelet space.

2.1 Wavelets

Wavelets are a commonly used tool in signal and image processing. They are a family of orthonormal basis functions that decompose the original function into different scales and locations. Here, we informally introduce the necessary concepts of Wavelets through the Haar Discrete Wavelet Transform (DWT) – often considered the simplest type of wavelet transform – as an illustrative example. More formal background on Wavelets are available in Mallat (1989) and Daubechies (1992).

Consider a vector of data, such as a vector containing normalised sequencing outputs, $\mathbf{d} = (d_1, \ldots, d_B)$, where B is the length of the region and is assumed to be a power of 2 ($B = 2^J,\ J \in \mathbb{N}$). The DWT decomposes $\mathbf{d}$ into a length B vector of wavelet coefficients (WCs), $\mathbf{y}$, where each coefficient, $y_{s,l}$, summarises information from $\mathbf{d}$ at a certain scale, s, and across different locations, l, at that scale.

Specifically, at the 'zero-th' scale, there is one location, and hence one WC, which sum all the elements of $\mathbf{d}$ together:

$$y_{0,1} := \sum_{b=1}^{B} d_b$$

This can be thought of as a summary of d's information at the 'coarsest' scale. At the next scale, we have:

$$y_{1,1} := \sum_{b \leq B/2} d_b - \sum_{b > B/2} d_b$$

This contrasts the counts from the region composed of the first half of $\mathbf{d}$ to that composed of the second. At the next scale, there are two coefficients:

$$\begin{aligned} y_{2,1} &:= \sum_{b \leq B/4} d_b - \sum_{B/4 < b \leq B/2} d_b \\ y_{2,2} &:= \sum_{B/2 < b \leq 3B/4} d_b - \sum_{3B/4 < b \leq B} d_b \end{aligned}$$

with each coefficient contrasting counts from adjacent quarters of $\mathbf{d}$. Each subsequent scale has double the number of coefficients as the previous scale, such that each scale has 2^{s-1} coefficients, until the 'finest' scale, J, which has 2^{J-1} coefficients. The finest scale starts with the coefficient $y_{J,1} = d_1 - d_2$, and contains coefficients which contrast adjacent pairs of the original data vector. Putting this together, we see that $\mathbf{y}$ is made of the two standalone coefficients at the zero-th and first scales respectively, two coefficients from the two locations at the second scale, four coefficients from the four locations at the third scale, and so on, until the finest scale, J. Hence, we have that $\mathbf{y} = (y_{0,1}, y_{1,1}, y_{2,1}, y_{2,2}, \ldots, y_{J,1}, \ldots, y_{J,2^{J-1}})$, a vector of length B. Figure 2.1 shows a graphical representation of the WCs at different scales and locations, with the blue dots representing the WCs, and red dots representing the original data.

Note that although $y_{0,1}$ is not precisely a WC, we will refer to it as a WC (at the zero-th scale) for convenience. Also, our representation of WCs are correct up to a scaling constant, which is also ignored here for simplicity.

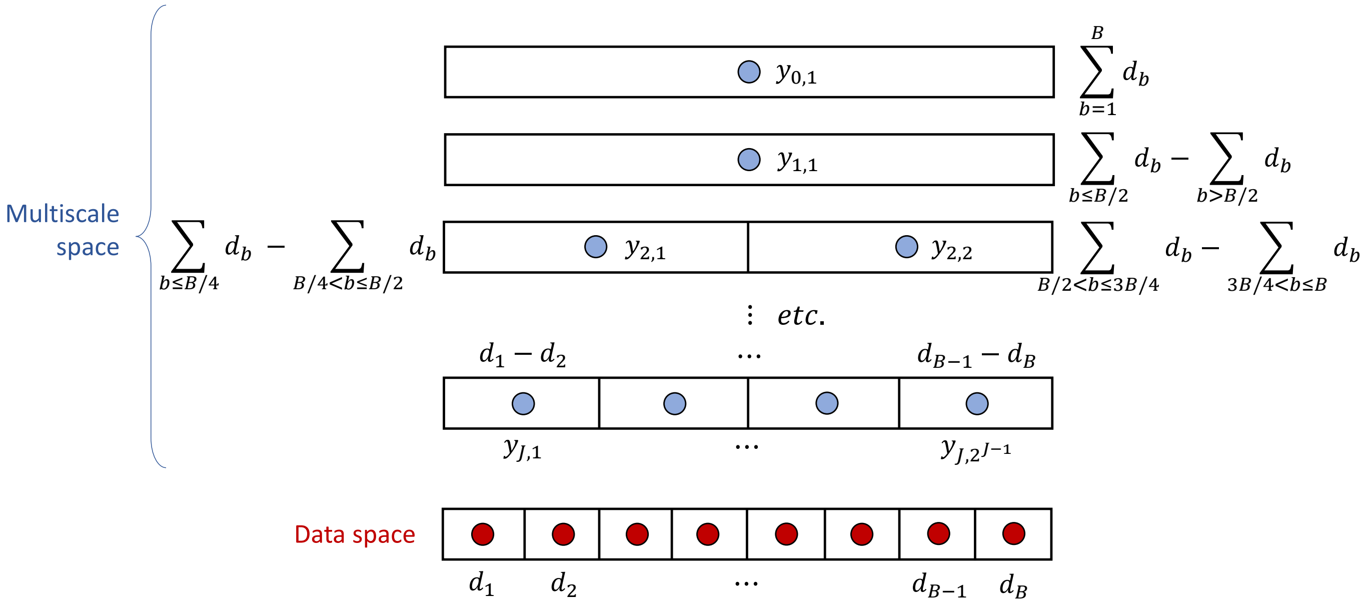


Figure 2.1: Representation of WCs at different scales and locations, and their relationship to original data.

For data containing spatially-structured signals, the nature of the decomposition transforms the data into a wavelet space where the WCs have a sparse structure; many WCs will be small, consisting almost entirely of noise, while the signal will be concentrated around a few large WCs (an example is shown in Figure 2.3). This property lends itself to a modelling strategy where the data is projected onto the wavelet space using a DWT, where the sparse structure in the WCs is relatively easy to model. The observed function is then denoised by filtering out or shrinking smaller WCs; this process is known as 'wavelet denoising'. This may be done, for example, by applying thresholding rules (Donoho and Johnstone (1995)), or placing shrinkage priors on the WCs (Abramovich et al. (1998)).

Figure 2.2 shows the normalised counts from an individual where signals are evident in the data space, as well as the WCs after a DWT is applied. Signals in the data space correspond with concentrations of non-zero WCs around specific locations in the multiscale space, propagated through different scales.

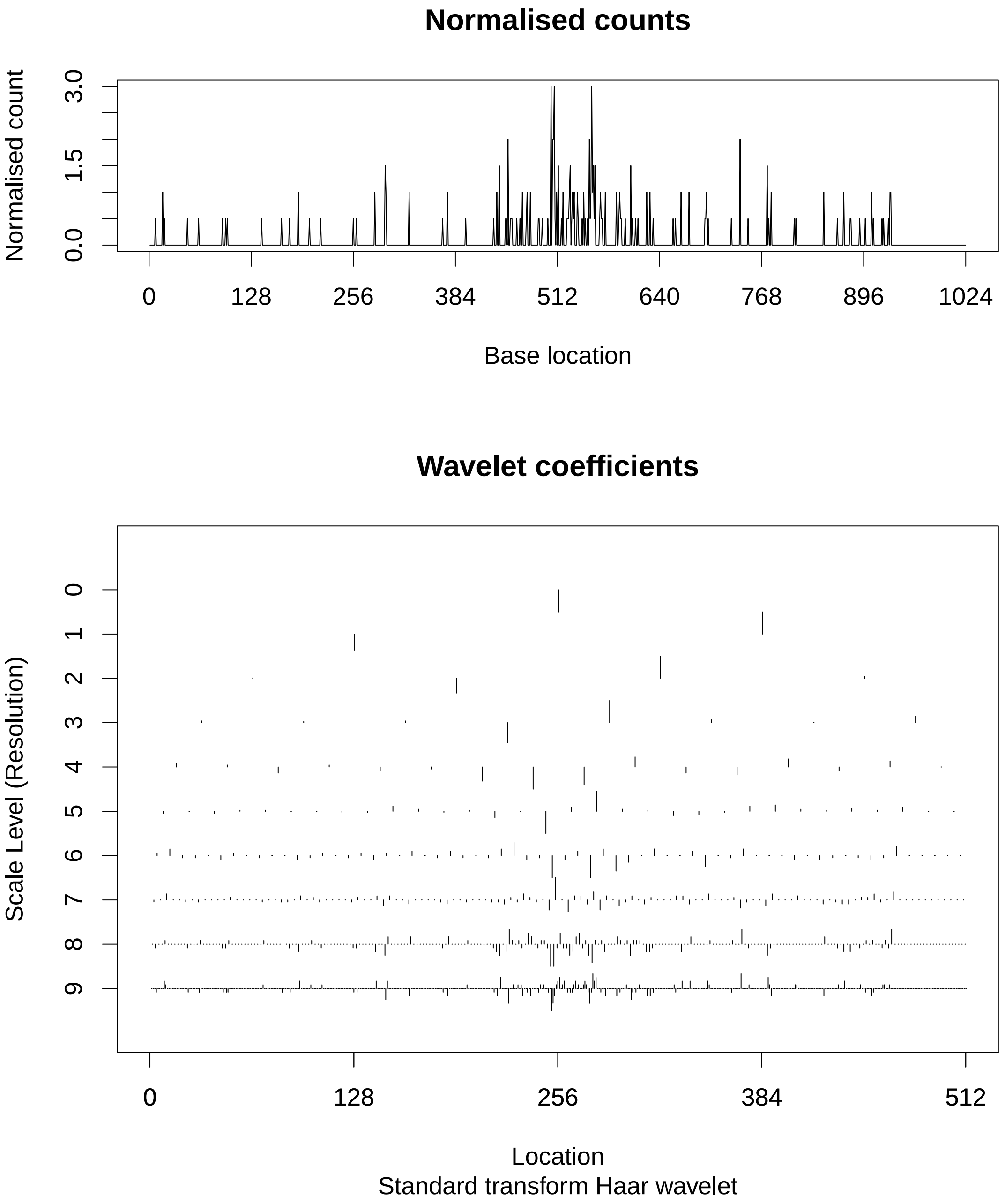


Figure 2.2: Sequencing data from an individual containing signals corresponds to non-zero WCs in the wavelet space.

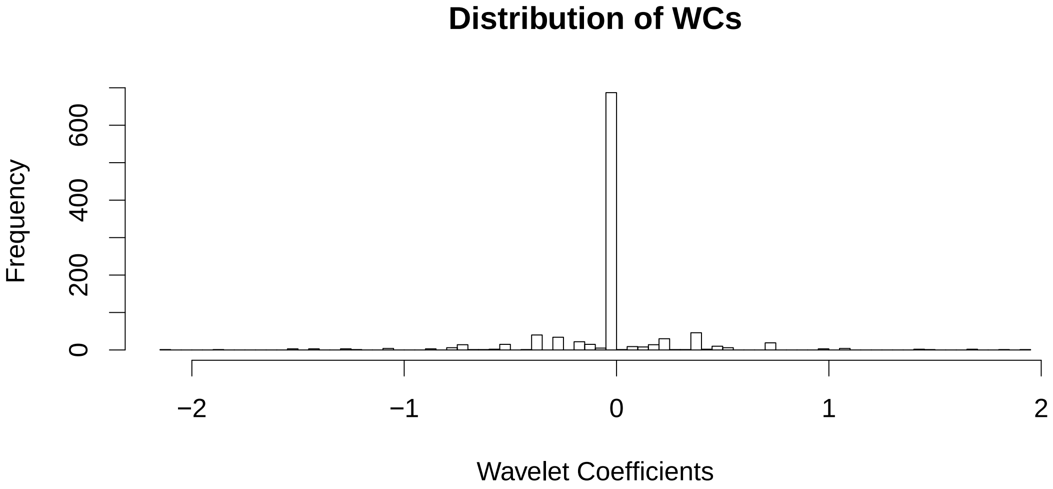


Figure 2.3: Histogram of WCs from the same data as Figure 2.2.

The last step involves moving from the wavelet space back to the data space. The DWT represents a linear transform which can be represented in matrix form, and as the transformation is one-to-one, the matrix is invertible, allowing the Inverse Discrete Wavelet Transform (IDWT) to also be represented in a matrix. This allows us to move easily between the data- and wavelet-space, and ultimately, to obtain estimates of the underlying signal function in terms of data-space quantities. Importantly, the one-to-one nature of the transform means that $\mathbf{y}$ contains exactly the same information as $\mathbf{d}$.

2.2 *WaveQTL*

Shim and Stephens (2015) use this wavelet technique to develop computationally efficient methods to identify genetic variants associated with molecular-level phenotypes. They also provide methods to estimate the effect of the genetic variant along the region of interest. We review their methods in this section.

2.2.1 Model

For individual $i = 1, \ldots, N$, and across each base, $b = 1, \ldots, B$ (where $B = 2^J$ for some integer, J), consider a vector of (normalised) sequencing data, $\mathbf{d}^i =$

$(d^i_1, \ldots, d^i_B)$. We denote the covariate for each individual as g^i.

Firstly, a DWT is used to transform the data into the wavelet space, resulting in the vector of WCs, $\mathbf{y}^i = (y^i_{0,1}, y^i_{1,1}, \ldots, y^i_{J,2^J-1})$. Borrowing ideas for the model and priors from Servin and Stephens (2007), for each scale, s and location, l:

$$y^i_{s,l} = \mu_{s,l} + \beta_{s,l} g^i + \varepsilon^i_{s,l} \tag{2.1}$$

$$\varepsilon^i_{s,l} \overset{d}{\sim} N(0, \sigma^2_{s,l}) \tag{2.2}$$

where $\mu_{s,l}$ is the mean WC of individuals with $g^i = 0$, $\beta_{s,l}$ is the effect size of g on the WC, and $\varepsilon^i_{s,l}$ is an error term. The priors are:

$$\sigma^2_{s,l} \overset{d}{\sim} \Gamma^{-1}(\kappa^a_{s,l}, \kappa^b_{s,l}) \tag{2.3}$$

$$\gamma_{s,l} \overset{d}{\sim} \text{Bernoulli}(\pi_s) \tag{2.4}$$

$$\mu_{s,l} \mid \sigma^2_{s,l} \overset{d}{\sim} N(0, \sigma^2_{\mu,s,l}\sigma^2_{s,l}) \tag{2.5}$$

$$\beta_{s,l} \mid \gamma_{s,l}, \sigma^2_{s,l} \overset{d}{\sim} \gamma_{s,l} N(0, \sigma^2_{\beta,s,l}\sigma^2_{,s,l}) + (1 - \gamma_{s,l})\delta_0 \tag{2.6}$$

where δ_0 is a point mass at 0 and $\gamma_{s,l}$ is a latent binary 'state' variable which indicates whether there is any association between $y_{s,l}$ and g. With these priors, we can obtain closed form posteriors with no Markov Chain Monte Carlo methods required.

2.2.2 Hyperparameter choice

Following on from Servin and Stephens (2007), Shim and Stephens (2015) consider limiting forms of the priors, with $\kappa^a_{s,l}, \kappa^b_{s,l} \to 0$ and $\sigma^2_{\mu,s,l} \to \infty$, and put a discrete uniform prior on $\sigma^2_{\beta,s,l}$ by averaging over several pre-set values for $\sigma^2_{\beta,s,l}$. Under these settings, the density of the posterior is proper (integrates to a finite value) – see Servin and Stephens (2007) for further discussion about the choice of hyperparameters and its impact on analysis.

The other hyperparameters, π_s for $s = 0, \ldots, J$, represent the proportion of WCs, at a specific scale, s, which are associated with g, controlling the sparsity of the model. Shim and Stephens (2015) estimated the vector, $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_J)$, from the data using Empirical Bayes (EB) methods.

2.2.3 Association testing

To test whether the genotype, g^i, is associated with a molecular-level phenotype, they ask whether the null hypothesis – H_0: d is independent of g – can be rejected. The null hypothesis in this case is represented by the model where $\gamma_{s,l} = 0$ for all s and l – no association between *any* WC and g. As their model is hierarchical, H_0 holds when $\pi_s = 0$ at all s ($\boldsymbol{\pi} \equiv 0$). Therefore, the likelihood ratio test statistic to test H_0, evaluated at the MLEs of $\boldsymbol{\pi}$, is:

$$\widehat{\Lambda}(\mathbf{y}, \mathbf{g}) := \frac{P(\mathbf{y} \mid \mathbf{g}, \widehat{\boldsymbol{\pi}})}{P(\mathbf{y} \mid \mathbf{g}, \boldsymbol{\pi} \equiv 0)} \tag{2.7}$$

$$= \prod_{s,l} \left[\pi_s \frac{P(y_{s,l} \mid \mathbf{g}, \gamma_{s,l} = 1)}{P(y_{s,l} \mid \mathbf{g}, \gamma_{s,l} = 0)} + (1 - \pi_s) \right] \tag{2.8}$$

where bold quantities represent the vector of all quantities across all respective scales, locations and individuals and $y_{s,l}$ represents the vector of all individuals' WCs at that scale-location, $(y^1_{s,l}, \ldots, y^N_{s,l})$. They then assess the significance of this likelihood ratio test statistic by obtaining a p-value through permutation. See Shim and Stephens (2015) for more details. Note that the fraction in (2.8) is the Bayes Factor (BF); a quantity which measures the support for $\gamma_{s,l} = 1$ at a scale s and location l, for the set of (y, g) observed across all individuals:

$$\mathrm{BF}_{s,l}(y, g) = \frac{P(y_{s,l} \mid \mathbf{g}, \gamma_{s,l} = 1)}{P(y_{s,l} \mid \mathbf{g}, \gamma_{s,l} = 0)} \tag{2.9}$$

This is available in closed form (Servin and Stephens (2007)). For more details about Bayes Factors, refer to Appendix A.

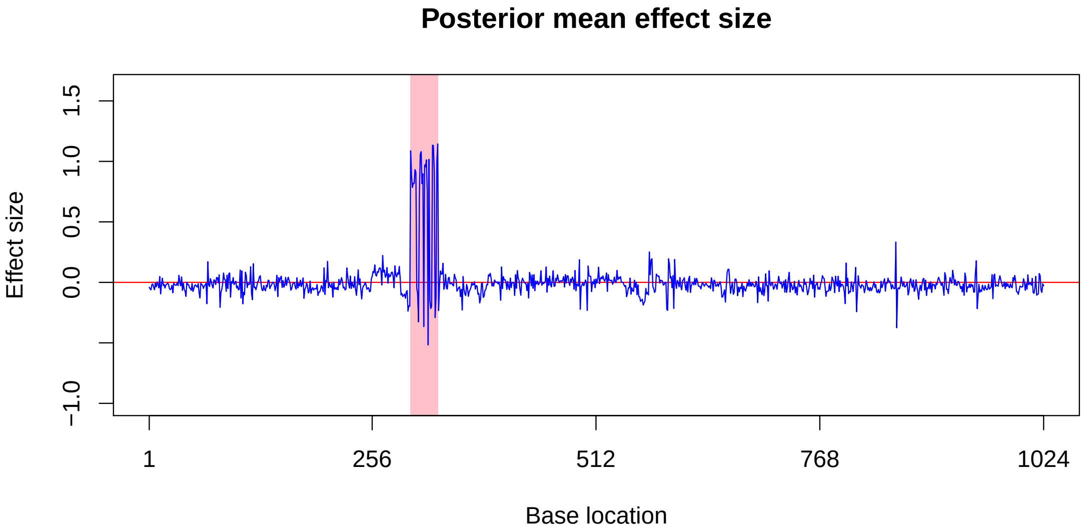
2.2.4 Effect size estimation

Shim and Stephens (2015) show that the posterior distribution of the effect size in the wavelet space, $P(\beta_{s,l} \mid \mathbf{y}, \mathbf{g})$ has a closed form, which is a mixture of a point mass at zero, and a three-parameter version of the t-distribution. However, the posterior distribution of the effect size in the data space, which is more interpretable, is not available in closed form. Despite this, closed forms for the posterior means and variances of the data space effect size at each base are available, by applying the IDWT matrix to the corresponding wavelet space quantities, owing to the linear relationship between the wavelet and data spaces. See Appendix A for details. Other posterior quantities, such as credible intervals, can be attained by simulation.

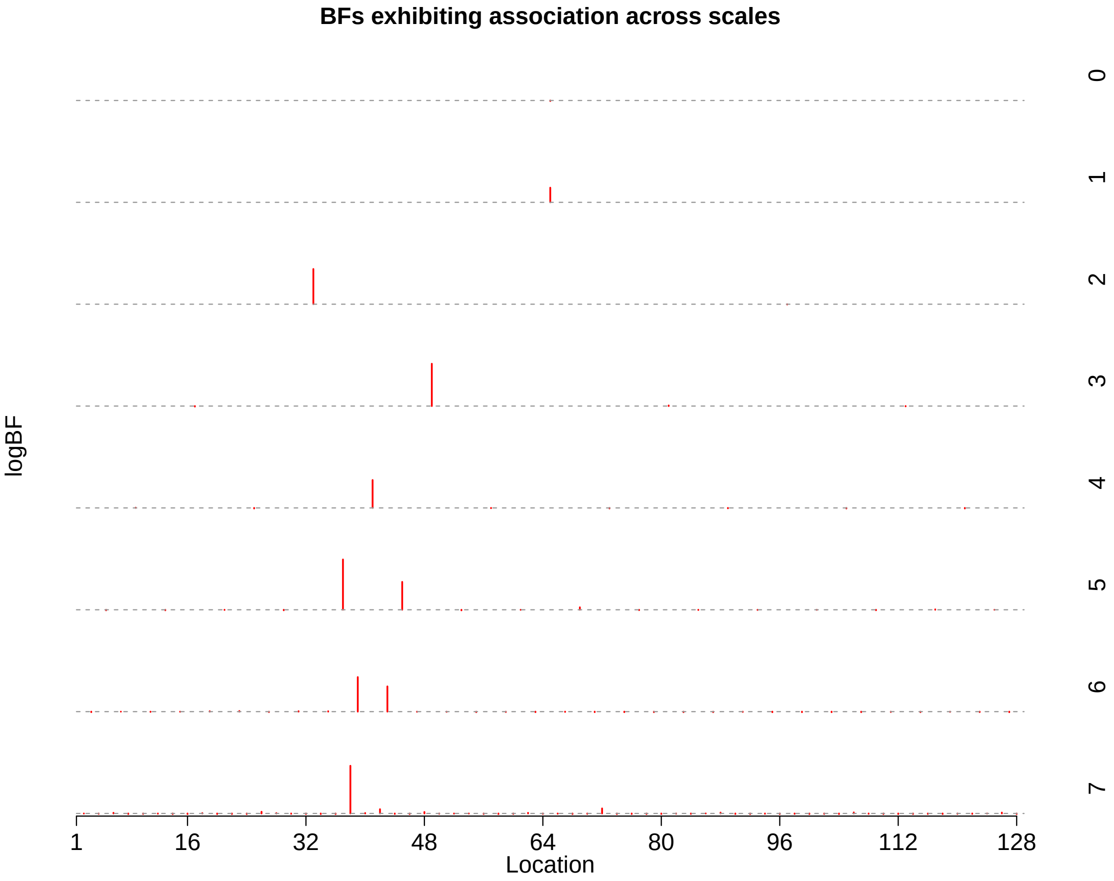
2.2.5 Limitations

WaveQTL assumes independence of $\gamma_{s,l}$'s, (and therefore, effect sizes, $\beta_{s,l}$), across different scales and locations. However, this independence assumption does not hold exactly in practice. From the signal in Figure 2.4a), Figure 2.4b) shows that, at the scales and locations where the BFs indicate WCs with strong association in the wavelet space, the children are also likely to show evidence of a strong association. That is, the association signal tends to *propagate* between scales which correspond to areas in the data space where an association is present.

Methods that exploit dependencies between $\gamma_{s,l}$'s (and therefore, effect sizes), should perform better by effectively combining information about association signals across different scales and locations. Motivated by this observation, and the framework described in Crouse et al. (1998) for applying a tree-shaped structure to WCs, we propose imposing a tree structure on $\gamma_{s,l}$ through the method in the next chapter.



(a) Plot of posterior mean effect size, showing a strong association signal in the pink region.



(b) BFs of the signal in multiscale space showing association across scales (only first 7 scales shown – scales shown on right).

Figure 2.4: Motivation for using tree-shaped dependencies.

2.3 ***WaveQTL*** with Hidden Markov Tree (HMT) prior

2.3.1 Hidden Markov Trees

Crouse et al. (1998) propose modelling the dependencies between WCs using a Hidden Markov Tree (HMT). An HMT is a tree-shaped model where the underlying states exhibit the Markov property – the probability of each state is only dependent on its parent's state.

In our model, we impose a tree-shaped prior on the hidden state parameter, the set of $\gamma_{s,l}$'s. This means that all the previous priors and likelihoods remain the same, except that the statement about the prior of $\gamma_{s,l}$ from (2.4) no longer holds. As our state parameters are binary, $\gamma_{s,l} \in \{0, 1\}$, the following hyperparameters are required for the HMT prior:

1. $\pi_{s,l} = P(\gamma_{s,l} = 1 \mid \boldsymbol{\theta})$, the probability of an effect at any given s, l
2. $\varepsilon^{kl}_{(s,l),p(s,l)} = P(\gamma_{s,l} = k \mid \gamma_{p(s,l)} = l, \boldsymbol{\theta})$, the transition probability between particular parent-child state combinations (such that $\sum_{k=0}^{1} \varepsilon^{kl}_{(s,l),p(s,l)} = 1$)

where we use $\boldsymbol{\theta}$ to denote this set of hyperparameters. Due to their dependencies, we now consider the joint prior distribution of the $\boldsymbol{\gamma} = (\gamma_{0,1}, \ldots, \gamma_{J,2^{J-1}})$ as a whole. We derive the joint distribution of $\boldsymbol{\gamma}$ using the HMT assumptions on a smaller example; a tree with 7 nodes, each representing a state variable, as per Figure 2.5. Indexing the nodes as $i = 1, \ldots, 7$ for simplicity, we repeatedly apply the conditional probability definition to obtain the joint distribution in terms of the parameterised quantities:

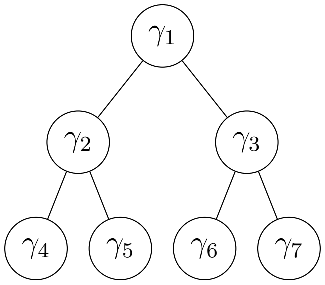


Figure 2.5: Example 7-node tree.

$$
\begin{aligned}
P(\boldsymbol{\gamma}) &= P(\gamma_1, \ldots, \gamma_7) \\
&= P(\gamma_1)P(\gamma_2, \ldots, \gamma_7 \mid \gamma_1) \\
&= P(\gamma_1)P(\gamma_2, \gamma_3 \mid \gamma_1)P(\gamma_4, \ldots, \gamma_7 \mid \gamma_1, \gamma_2, \gamma_3) \\
&= P(\gamma_1)P(\gamma_2 \mid \gamma_1)P(\gamma_3 \mid \gamma_1)P(\gamma_4, \ldots, \gamma_7 \mid \gamma_2, \gamma_3) \\
&= P(\gamma_1)P(\gamma_2 \mid \gamma_1)P(\gamma_3 \mid \gamma_1)P(\gamma_4 \mid \gamma_2)P(\gamma_5 \mid \gamma_2)P(\gamma_6 \mid \gamma_3)P(\gamma_7 \mid \gamma_3) \quad (2.10) \\
\therefore P(\boldsymbol{\gamma}) &= P(\gamma_1 = s_1, \ldots, \gamma_7 = s_7) = \pi_1^{s_1} \varepsilon_{2,1}^{s_2,s_1} \varepsilon_{3,1}^{s_3,s_1} \varepsilon_{4,2}^{s_4,s_2} \varepsilon_{5,2}^{s_5,s_2} \varepsilon_{6,3}^{s_6,s_3} \varepsilon_{7,3}^{s_7,s_3}
\end{aligned}
$$

where $\pi_1^{s_1}$ is used here to denote $P(\gamma_{1,1} = s_1 \mid \boldsymbol{\pi})$ in a general case where more than two states exist.

As the $\gamma_{s,l}$ state parameters underpin the $\beta_{s,l}$'s, placing a tree-shaped prior on the states results in a HMT prior on the effect size estimates, $\beta_{s,l}$. Whilst this means we do not *directly* model dependency between the $\beta_{s,l}$'s, the dependencies between the $\gamma_{s,l}$'s are sufficient to *indirectly* capture the structure between the $\beta_{s,l}$'s, whilst remaining tractable and computationally efficient. This relationship is illustrated in Figure 2.6.

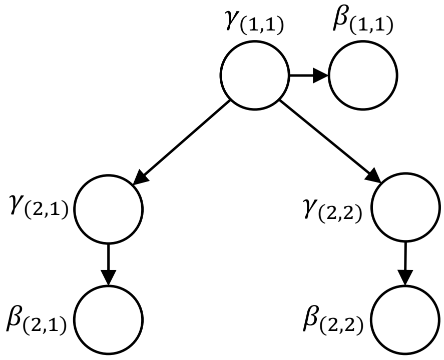


Figure 2.6: Illustration of how priors on γ impose structure on β.

Note that the WC at the zero-th scale sits at the top of each tree, and is assumed to be independent to the rest of the tree. As a result, its quantities will be estimated by the model without HMT, and we will omit it from most of our discussions regarding HMTs in the remainder of this chapter.

2.3.2 Hyperparameter choice

In our model, the κ and σ hyperparameters are the same as in Section 2.2.2, and therefore we focus on computing the MLEs for $\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{\varepsilon})$, which will be estimated from the data using EB methods. With latent variables $\gamma_{s,l}$, these can be computed using an Expectation-Maximisation (EM) algorithm adapted to HMTs. This adaptation was explored in Ronen et al. (1995), and extended in Crouse et al. (1998) and is known as the *Upward-Downward* algorithm. We describe the main steps and results of this algorithm here, and provide full details of the derivation in Appendix B.

2.3.2.1 Upward-Downward Algorithm

We denote $y_{s,l}$ as the vector of all individuals' WCs at a scale s and location l, $(y^1_{s,l}, \ldots, y^N_{s,l})$. In our model, the incomplete data refers to the WCs at all scale-locations, $\mathbf{y}$, whilst the complete data contains the WCs, augmented with the latent states, $(\mathbf{y}, \boldsymbol{\gamma})$. Although we omit modelling the WC at the zero-th scale here, there are simple extensions to handle this (see Crouse et al. (1998)), hence all quantities presented here follow from a tree rooted at (1,1), as per Figure 2.5. We begin by writing out the complete log-likelihood as follows:

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}; \mathbf{y}, \boldsymbol{\gamma}, \boldsymbol{g}) := P(\mathbf{y}, \boldsymbol{\gamma} \mid \boldsymbol{g}, \boldsymbol{\theta}) &= P(\mathbf{y} \mid \boldsymbol{\gamma}, \boldsymbol{g}) P(\boldsymbol{\gamma} \mid \boldsymbol{g}, \boldsymbol{\theta}) \\
&= \left[\prod_{s,l} P(y_{s,l} \mid \gamma_{s,l}, \boldsymbol{g})\right] P(\boldsymbol{\gamma} \mid \boldsymbol{g}, \boldsymbol{\theta}) \\
&= \left[\prod_{s,l} P(y_{s,l} \mid \gamma_{s,l}, \boldsymbol{g})\right] P(\gamma_{1,1} \mid \boldsymbol{g}, \boldsymbol{\theta}) \prod_{s=2}^{J} \prod_{l=1}^{2^{s-1}} P(\gamma_{s,l} \mid \gamma_{p(s,l)}, \boldsymbol{g}, \boldsymbol{\theta}) \\
&= \prod_{s,l} \prod_{m=0}^{1} P(y_{s,l} \mid \gamma_{s,l} = m, \boldsymbol{g})^{\mathbb{1}\{\gamma_{s,l}=m\}} \prod_{m=0}^{1} P(\gamma_{1,1} = m \mid \boldsymbol{g}, \boldsymbol{\theta})^{\mathbb{1}\{\gamma_{1,1}=m\}} \ldots \\
&\quad \times \prod_{s=2}^{J} \prod_{l=1}^{2^{s-1}} \prod_{m=0}^{1} \prod_{n=0}^{1} P(\gamma_{s,l} = m \mid \gamma_{p(s,l)} = n, \boldsymbol{g}, \boldsymbol{\theta})^{\mathbb{1}\{\gamma_{s,l}=m, \gamma_{p(s,l)}=n\}} \\
&= \prod_{s,l} \prod_{m=0}^{1} P(y_{s,l} \mid \gamma_{s,l} = m, \boldsymbol{g})^{\mathbb{1}\{\gamma_{s,l}=m\}} \prod_{m=0}^{1} (\pi_{11}^{m})^{\mathbb{1}\{\gamma_{1,1}=m\}} \ldots \\
&\quad \times \prod_{s=2}^{J} \prod_{l=1}^{2^{s-1}} \prod_{m=0}^{1} \prod_{n=0}^{1} (\varepsilon_{(s,l),p(s,l)}^{mn})^{\mathbb{1}\{\gamma_{s,l}=m, \gamma_{p(s,l)}=n\}}
\end{aligned}$$

where 2^{s-1} denotes the number of locations at each scale, s, and where $\mathbb{1}\{\gamma_{s,l} = m\}$ represents the indicator random variable:

$$\mathbb{1}\{\gamma_{s,l} = m\} = \begin{cases} 1 & \text{if } \gamma_{s,l} = m \\ 0 & \text{otherwise} \end{cases}$$

As we have the BF in closed form, it is easier to divide through by the constant term, $P(y_{s,l} \mid \gamma_{s,l} = 0, \boldsymbol{g})$, and obtain:

$$\begin{aligned}
P(\mathbf{y}, \boldsymbol{\gamma} \mid \boldsymbol{g}, \boldsymbol{\theta}) = \Big[\prod_{s,l} P(y_{s,l} \mid \gamma_{s,l} = 0, \boldsymbol{g})\Big] \prod_{s,l} \prod_{m=0}^{1} \frac{P(y_{s,l} \mid \gamma_{s,l} = m, \boldsymbol{g})^{\mathbb{1}\{\gamma_{s,l}=m\}}}{P(y_{s,l} \mid \gamma_{s,l} = 0, \boldsymbol{g})} \ldots \\
\times \prod_{m=0}^{1} (\pi_{11}^{m})^{\mathbb{1}\{\gamma_{1,1}=m\}} \prod_{s=2}^{J} \prod_{l=1}^{2^{s-1}} \prod_{m=0}^{1} \prod_{n=0}^{1} (\varepsilon_{(s,l),p(s,l)}^{mn})^{\mathbb{1}\{\gamma_{s,l}=m, \gamma_{p(s,l)}=n\}}
\end{aligned}$$

Therefore, the complete data log-likelihood:

$$\begin{aligned}\log \mathcal{L}(\boldsymbol{\theta};\mathbf{y},\boldsymbol{\gamma},\boldsymbol{g}) = \sum_{s,l} \log P(y_{s,l} \mid \gamma_{s,l}=0,\boldsymbol{g}) + \sum_{s,l} \mathbb{1}\{\gamma_{s,l}=1\} \log \mathrm{BF}_{s,l}(y,g)\ldots \\ + \mathbb{1}\{\gamma_{1,1}=1\}\log(\pi_{1,1}) + \mathbb{1}\{\gamma_{1,1}=0\}\log(1-\pi_{1,1})\ldots \\ + \sum_{s=2}^{J}\sum_{l=1}^{2^{s-1}}\sum_{m=0}^{1}\sum_{n=0}^{1} \mathbb{1}\{\gamma_{s,l}=m,\gamma_{p(s,l)}=n\}\log\big(\varepsilon^{mn}_{(s,l),p(s,l)}\big)\end{aligned} \tag{2.11}$$

2.3.2.2 E-step

For the Expectation step (E-step), given a current set of parameter estimates after the t-th iteration, $\boldsymbol{\theta}^{(t)}$, and the observed WCs, we evaluate the conditional expectation of the complete data log-likelihood from equation 2.11, denoted as $Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)})$:

$$\begin{aligned}Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)}) &:= \mathbb{E}_{\boldsymbol{\gamma}}[\log \mathcal{L}(\boldsymbol{\theta};\mathbf{y},\boldsymbol{\gamma},\mathbf{g}) \mid \mathbf{y},\mathbf{g},\boldsymbol{\theta}^{(t)})] \\ &= \log P(\mathbf{y} \mid \boldsymbol{\gamma} \equiv 0,\mathbf{g}) + \sum_{s,l} P(\gamma_{s,l}=1 \mid \mathbf{y},\boldsymbol{\theta}^{(t)}) \log \mathrm{BF}_{s,l}(y,g)\ldots \\ &\quad + P(\gamma_{s,l}=1 \mid \mathbf{y},\boldsymbol{\theta}^{(t)})\log(\pi_{s,l}) + P(\gamma_{s,l}=0 \mid \mathbf{y},\boldsymbol{\theta}^{(t)})\log(1-\pi_{s,l})\ldots \\ &\quad + \sum_{s=2}^{J}\sum_{l=1}^{2^{s-1}}\sum_{m=0}^{1}\sum_{n=0}^{1} P(\gamma_{s,l}=m,\gamma_{p(s,l)}=n \mid \mathbf{y},\boldsymbol{\theta}^{(t)})\log\big(\varepsilon^{mn}_{(s,l),p(s,l)}\big)\end{aligned}$$

using two properties of expectations of indicators:

$$\begin{aligned}\mathbb{E}[\mathbb{1}\{\gamma_{1,1}=1\} \mid \mathbf{y},\boldsymbol{\theta}^{(t)}] &= P(\gamma_{1,1}=1 \mid \mathbf{y},\boldsymbol{\theta}^{(t)}) \\ \mathbb{E}[\mathbb{1}\{\gamma_{s,l}=m,\gamma_{p(s,l)}=n\} \mid \mathbf{y},\boldsymbol{\theta}^{(t)}] &= P(\gamma_{s,l}=m,\gamma_{p(s,l)}=n \mid \mathbf{y},\boldsymbol{\theta}^{(t)})\end{aligned}$$

The resulting quantities to be solved for are:

$$P(\gamma_{s,l}=m \mid \mathbf{y},\boldsymbol{\theta}^{(t)}) = \frac{P(\gamma_{s,l}=m,\mathbf{y} \mid \boldsymbol{\theta}^{(t)})}{\sum_{k=0}^{1} P(\gamma_{s,l}=k,\mathbf{y} \mid \boldsymbol{\theta}^{(t)})} \tag{2.12}$$

$$P(\gamma_{s,l}=m,\gamma_{p(s,l)}=n \mid \mathbf{y},\boldsymbol{\theta}^{(t)}) = \frac{P(\gamma_{s,l}=m,\gamma_{p(s,l)}=n,\mathbf{y} \mid \boldsymbol{\theta}^{(t)})}{\sum_{k=0}^{1} P(\gamma_{s,l}=k,\mathbf{y} \mid \boldsymbol{\theta}^{(t)})} \tag{2.13}$$

for $m, n \in \{0, 1\}, s \in \{1, \ldots, J\}, l \in \{1, \ldots, 2^{s-1}\}$. These rely on the outputs from the Upward-Downward algorithm, but before we proceed, we introduce some concepts and notation about graphs and trees.

An undirected *graph* consists of a set of *nodes*, $\{v_1, v_2, \ldots, v_N\}$ and a set of *edges* which link the nodes together. A *path* is a set of connections between two nodes, and a rooted *tree* is an undirected graph with no cycles. The *root* node is a designated node; in our case, the node at the coarsest scale ('top') of the tree at (1,1). All nodes lying on the path from v_i to the *root* are *ancestors* of v_i, and similarly, all nodes on paths from v_i away from the root are called *descendants* of v_i. The *parent* of v_i is its immediate ancestor, denoted as $v_{p(i)}$, and similarly, the set of v_i's children are its immediate descendants, denoted by $\{v_j\}_{j \in c(i)}$. In the HMT in our model, each node represents a particular scale and location and has at most two children, hence being modelled as a binary tree. Nodes with no children are called *leaves* of the tree, and these are found at the finest scale, J, of our tree. All nodes can only have one parent.

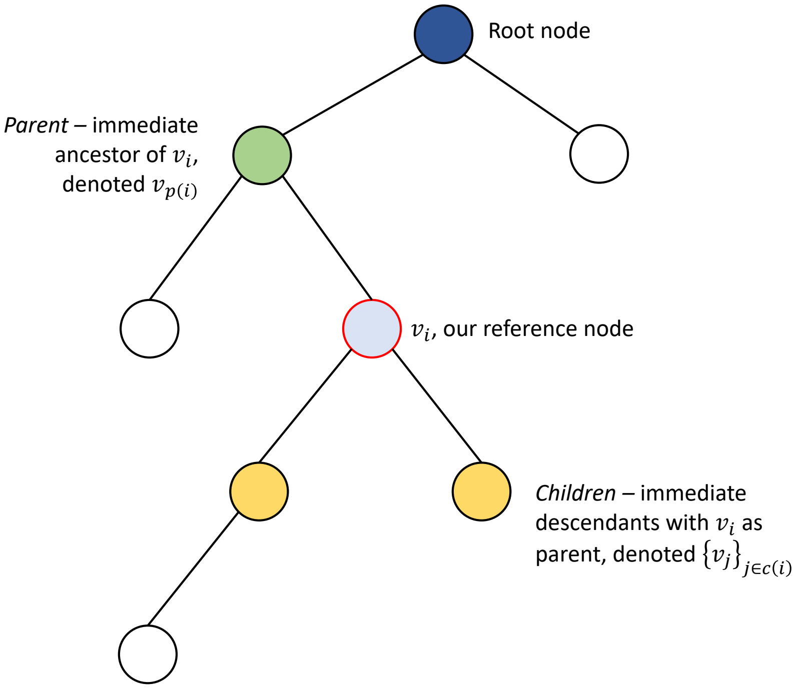

Root node
Parent – immediate ancestor of v_i, denoted $v_{p(i)}$
v_i, our reference node
Children – immediate descendants with v_i as parent, denoted $\{v_j\}_{j \in c(i)}$

Figure 2.7: Tree terminology

Denote:

- $\mathcal{T}_{s,l}$ as the vector of WCs in the subtree rooted at the node (s, l), with $\mathcal{T}_{1,1}$ therefore representing the vector of all WCs in the tree,
- $\mathcal{T}_{p(s,l)\backslash(s,l)}$ as the vector of WCs in the subtree obtained by removing subtree $\mathcal{T}_{s,l}$ from $\mathcal{T}_{p(s,l)}$,
- $\nu_{(s,l)}$ as the set of all *nodes* (as opposed to WCs) in the subtree rooted at (s, l).

For clarity, this notation is represented graphically in Figure 2.8.

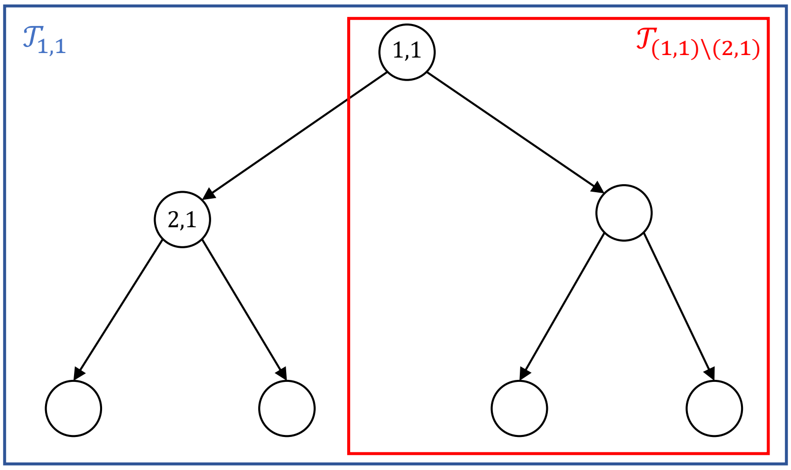


Figure 2.8: Graphical illustration of the tree and subtree notation.

The Upward-Downward algorithm has two parts. The first is the 'Up' step, where we propagate information about the association signal up the tree from the leaves toward the root node by conditioning on the state of a given node, its parent, and the transition parameters. The second is the 'Down' step where we propagate information regarding the states of the nodes down the tree, and obtain joint probabilities of states and associations by combining these quantities with the conditional associations we found in the 'Up' step. These components constitute the bulk of the E-step in the Upward-Downward algorithm. Note that as per Crouse et al. (1998), we use β to denote the outputs from the 'Up' step, and α to denote the outputs from the 'Down' step – these are not to be confused with the β effect size coefficients of our model. See Appendix B for full details behind this algorithm.

Up-step

0. At the **finest scale**, $s = J$, initialise: $\beta_{s,l}(1) = \mathrm{BF}_{s,l}(y,g)$ and $\beta_{s,l}(0) = 1$, for all locations $l = 1, \ldots, 2^{J-1}$

1. For all locations at scale s, for $m \in \{0,1\}$, calculate each of the following three quantities:

   (a) At a particular node, the joint association of all nodes in the subtree, given its parent's state:

$$\begin{aligned}\beta_{(s,l),p(s,l)}(m) &:= \frac{P(\mathcal{T}_{s,l} \mid \gamma_{p(s,l)} = m, \boldsymbol{\theta}^{(t)})}{\prod_{j\in\nu_{(s,l)}} P(y_j \mid \gamma_j = 0, \mathbf{g})} \\ &= \sum_{n=0}^{1} \beta_{s,l}(n)\varepsilon^{nm}_{(s,l),p(s,l)} \qquad (2.14)\end{aligned}$$

   (b) Joint association of a subtree rooted at a particular node given its own state, by incorporating its own association probability, and those of the subtrees rooted at both children:

$$\begin{aligned}\beta_{p(s,l)}(m) &:= \frac{P(\mathcal{T}_{p(s,l)} \mid \gamma_{p(s,l)} = m, \boldsymbol{\theta}^{(t)})}{\prod_{j\in\nu_{p(s,l)}} P(y_j \mid \gamma_j = 0, \mathbf{g})} \\ &= \Big[\prod_{j\in c(p(s,l))} \beta_{j,p(j)}(m)\Big]\mathrm{BF}_{p(s,l)}(y,g)(m) \qquad (2.15)\end{aligned}$$

   where $\mathrm{BF}_{p(s,l)}(y,g)(1) = \mathrm{BF}_{p(s,l)}(y,g)$ and $\mathrm{BF}_{p(s,l)}(y,g)(0) = 1$

   (c) Joint association of a particular node and all descendants from only one of its children, given its own state:

$$\begin{aligned}\beta_{p(s,l)\backslash(s,l)}(m) &:= \frac{P(\mathcal{T}_{p(s,l)\backslash(s,l)} \mid \gamma_{p(s,l)} = m, \boldsymbol{\theta}^{(t)})}{\prod_{j\in\nu_{p(s,l)\backslash(s,l)}} P(y_j \mid \gamma_j = 0, \mathbf{g})} \\ &= \frac{\beta_{p(s,l)}(m)}{\beta_{(s,l),p(s,l)}(m)} \qquad (2.16)\end{aligned}$$

2. $s := s - 1$

3. If $s = 1$ (coarsest scale), then stop, else return to step 1.

Down-step

0. Initialise state $\gamma_{1,1}$ at the **coarsest** scale level $s = 1$: $\alpha_{1,1}(1) = P(\gamma_{1,1} = 1 \mid \boldsymbol{\theta}^{(t)}) = \pi_{1,1}$ and $\alpha_{1,1}(0) = 1 - \pi_{1,1}$

1. $s := s + 1$

2. Calculate, for all locations at scale s, for $m \in \{0, 1\}$:

$$\begin{aligned}\alpha_{s,l}(m) &:= \frac{P(\gamma_{s,l} = m, \mathcal{T}_{(1,1)\backslash(s,l)} \mid \boldsymbol{\theta}^{(t)})}{\prod_{j \in \nu_{(1,1)\backslash(s,l)}} P(y_j \mid \gamma_j = 0, \mathbf{g})} \\ &= \sum_{n=0}^{1} \beta_{p(s,l)\backslash(s,l)}(n) \varepsilon^{mn}_{(s,l),p(s,l)} \alpha_{p(s,l)}(n) \qquad (2.17)\end{aligned}$$

   the joint probability of the state, and associations of all nodes in the tree except those in the subtree rooted at itself.

3. If $s = J$ (finest scale), then stop, else return to step 1.

Although the original implementation of the algorithm uses densities of individual WCs which we do not have in closed form, we replace them with BFs in our implementation, which we do have in closed form. As BFs represent the required WC densities divided by a constant, the final probabilities of interest are unaffected due to cancelling, despite the intermediate β and α quantities differing to the original implementation as per Crouse et al. (1998). A rough illustration demonstrating how these probabilities remain unchanged when using BFs can be found in Appendix B.

We use these quantities to calculate (2.12) and (2.13):

$$\begin{aligned}
P(\gamma_{s,l} = m \mid \mathbf{y}, \boldsymbol{\theta}^{(t)}) &= \frac{P(\gamma_{s,l} = m, \mathbf{y} \mid \boldsymbol{\theta}^{(t)})}{\displaystyle\sum_{k=0}^{1} P(\gamma_{s,l} = k, \mathbf{y} \mid \boldsymbol{\theta}^{(t)})} \\
&= \frac{P(\mathcal{T}_{s,l} \mid \gamma_{s,l} = m, \boldsymbol{\theta}^{(t)}) P(\mathcal{T}_{(1,1)\backslash(s,l)}, \gamma_{s,l} = m \mid \boldsymbol{\theta}^{(t)})}{\displaystyle\sum_{k=0}^{1} P(\mathcal{T}_{s,l} \mid \gamma_{s,l} = k, \boldsymbol{\theta}^{(t)}) P(\mathcal{T}_{(1,1)\backslash(s,l)}, \gamma_{s,l} = k \mid \boldsymbol{\theta}^{(t)})} \\
&= \frac{\beta_{s,l}(m)\alpha_{s,l}(m)}{\displaystyle\sum_{k=0}^{1} \beta_{s,l}(k)\alpha_{s,l}(k)} \qquad (2.18) \\
P(\gamma_{s,l} = m, \gamma_{p(s,l)} = n \mid \mathbf{y}, \boldsymbol{\theta}^{(t)}) &= \frac{P(\gamma_{s,l} = m, \gamma_{p(s,l)} = n, \mathbf{y} \mid \boldsymbol{\theta}^{(t)})}{\displaystyle\sum_{k=0}^{1} P(\gamma_{s,l} = k, \mathbf{y} \mid \boldsymbol{\theta}^{(t)})} \\
&= \frac{\beta_{s,l}(m)\varepsilon^{mn}_{s,l,p(s,l)}\beta_{p(s,l)\backslash(s,l)}(n)\alpha_{p(s,l)}(n)}{\displaystyle\sum_{k=0}^{1} \beta_{s,l}(k)\alpha_{s,l}(k)} \qquad (2.19)
\end{aligned}$$

2.3.2.3 M-step

We now have all the information required to evaluate $Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)})$, and can compute the quantities of $\boldsymbol{\theta}$ which maximise this expression (the maximisation, or M-step). Denote:

$$\begin{aligned}
A_{s,l}(m) &:= P(\gamma_{s,l} = m \mid \mathbf{y}, \boldsymbol{\theta}^{(t)}) \\
B_{(s,l),p(s,l)}(m,n) &:= P(\gamma_{s,l} = m, \gamma_{p(s,l)} = n \mid \mathbf{y}, \boldsymbol{\theta}^{(t)})
\end{aligned}$$

Then, for each s, l and states m, n:

$$\pi^{(t+1)}_{s,l} = A_{s,l}(1) \qquad (2.20)$$

$$(\varepsilon^{mn}_{(s,l),p(s,l)})^{(t+1)} = \frac{B_{(s,l),p(s,l)}(m,n)}{A_{s,l}(n)} \qquad (2.21)$$

However, with individual parameters calculated at a scale-location granularity, there is a risk that there is insufficient data to make robust estimates, leading to overfitting, and instability in the estimates. A practical solution to this is to increase the amount of data associated with each parameter. One way to achieve this is to use a common parameter to estimate quantities which have similar properties. This practice is known as *tying* within trees (Rabiner (1989)).

To implement tying, we augment the M-step slightly. For a set of nodes, $[i]$, which belong to a given tying group, parameters are now estimated by averaging over all the estimates of that corresponding parameter for each node in that group. Letting $|[i]|$ denote the number of nodes in that tying group, we have:

$$\pi_i^{(t+1)} = \frac{1}{|i|} \sum_{j \in [i]} A_j(m) \tag{2.22}$$

$$(\varepsilon_{i,p(i)}^{mn})^{(t+1)} = \frac{1}{|i|} \sum_{j \in [i]} \frac{B_{j,p(j)}(m,n)}{A_{p(j)}(n)} \tag{2.23}$$

An example of a group of nodes which may have similar properties would be all the nodes from a particular scale of the tree. That is, there would be a unique $\varepsilon_{s,p(s)}^{mn}$, for each of $s = 2, \ldots, J$, and this parameter would be the same for all locations in a given scale, allowing more data to be used to estimate this parameter.

2.3.3 Association testing

The only difference between association testing in our model and in *WaveQTL* is that the joint density of all $\gamma_{s,l}$'s no longer factorises into a product of individual distributions at each scale and location, due to dependencies between the states induced by the tree-shaped prior. Instead, we appeal to the recursive nature of our algorithm's outputs to calculate the likelihood evaluated at the MLEs of the hyperparameters, $\widehat{\boldsymbol{\theta}}$. Up to a constant (due to the use of BFs in our algorithm's calculations), we have that:

$$
\begin{aligned}
&\beta^{(EM)}_{1,1}(1)\alpha^{(EM)}_{1,1}(1) + \beta^{(EM)}_{1,1}(0)\alpha^{(EM)}_{1,1}(0) \\
&\qquad \propto P(\mathcal{T}_{1,1} \mid \gamma_{1,1} = 1)P(\gamma_{1,1} = 1 \mid \widehat{\boldsymbol{\theta}}) + P(\mathcal{T}_{1,1} \mid \gamma_{1,1} = 0)P(\gamma_{1,1} = 0 \mid \widehat{\boldsymbol{\theta}}) \qquad (2.24) \\
&\qquad = \sum_{m=0}^{1} P(\mathcal{T}_{1,1} \mid \gamma_{1,1} = m)P(\gamma_{1,1} = m \mid \widehat{\boldsymbol{\theta}}) \\
&\qquad = P(\mathbf{y} \mid \mathbf{g}, \widehat{\boldsymbol{\theta}})
\end{aligned}
$$

where the $^{(EM)}$ superscript is used to denote outputs from the algorithm, to prevent confusion with the β from our model.

To calculate the likelihood ratio test statistic, we also need to evaluate the maximum likelihood under the null model, enforcing all the parameters which induce $\gamma_{s,l} = 0$, for all s, l. That corresponds to a parameter set where the probability of the root state being 0 is 1, and all subsequent transitions from state 0 to 1 has probability 0:

$$(1 - \pi_{1,1}) = 1 \Rightarrow \pi_{1,1} = 0 \qquad (2.25)$$

$$\varepsilon^{10}_{(s,l),p(s,l)} = 0 \Rightarrow \varepsilon^{00}_{(s,l),p(s,l)} = 1 \qquad (2.26)$$

Here, we do not place any restriction on the remaining parameters ($\varepsilon^{01}_{(s,l),p(s,l)}, \varepsilon^{11}_{(s,l),p(s,l)}$ across all s, l), and it would be customary to use MLEs of these parameters to obtain the maximum likelihood under the null hypothesis. Denoting the above, null-hypothesis inducing parameter set as $\boldsymbol{\theta}_0$, and marginalising out $\boldsymbol{\gamma}$ from the log-likelihood expression from earlier:

$$P(\mathbf{y} \mid \boldsymbol{g}, \boldsymbol{\theta}_0) = \sum_{\boldsymbol{\gamma}} \left[\prod_{s,l} P(y_{s,l} \mid \gamma_{s,l}, \boldsymbol{g}) \right] P(\gamma_{1,1} \mid \boldsymbol{g}, \boldsymbol{\theta}_0) \prod_{s=2}^{J} \prod_{l=1}^{2^{s-1}} P(\gamma_{s,l} \mid \gamma_{p(s,l)}, \boldsymbol{g}, \boldsymbol{\theta}_0) \tag{2.27}$$

$$= \left[\prod_{s,l} P(y_{s,l} \mid \gamma_{s,l} = 0, \boldsymbol{g}) \right] P(\gamma_{1,1} = 0 \mid \boldsymbol{g}, \boldsymbol{\theta}_0) \prod_{s=2}^{J} \prod_{l=1}^{2^{s-1}} P(\gamma_{s,l} = 0 \mid \gamma_{p(s,l)} = 0, \boldsymbol{g}, \boldsymbol{\theta}_0) \tag{2.28}$$

$$= \left[\prod_{s,l} P(y_{s,l} \mid \gamma_{s,l} = 0, \boldsymbol{g}) \right] P(\boldsymbol{\gamma} \equiv 0 \mid \boldsymbol{g}, \boldsymbol{\theta}_0) \tag{2.29}$$

$$= \prod_{s,l} P(y_{s,l} \mid \gamma_{s,l} = 0, \boldsymbol{g}) \tag{2.30}$$

where we initially sum over all possible combinations of the entire vector of γ's. We get (2.28) as the only vector of states which contains any (all) of the probability under $\boldsymbol{\theta}_0$ corresponds to where $\boldsymbol{\gamma} \equiv 0$, and where $P(\boldsymbol{\gamma} \equiv 0 \mid \mathbf{g}, \boldsymbol{\theta}_0) = 1$. Therefore, we do not need to worry about MLEs for the remaining parameters as the expression in (2.28) is only a function of the parameters we defined in (2.25) – (2.26).

To perform association testing, the likelihood test statistic is:

$$\begin{aligned} \widehat{\Lambda}(\mathbf{y}, \mathbf{g}) &:= \frac{P(\mathbf{y} \mid \mathbf{g}, \widehat{\boldsymbol{\theta}})}{P(\mathbf{y} \mid \mathbf{g}, \boldsymbol{\theta}_0)} \\ &= \frac{P(\mathcal{T}_{1,1} \mid \gamma_{1,1} = 1, \mathbf{g}, \widehat{\boldsymbol{\theta}}) P(\gamma_{1,1} = 1 \mid \widehat{\boldsymbol{\theta}}) + P(\mathcal{T}_{1,1} \mid \gamma_{1,1} = 0, \mathbf{g}, \widehat{\boldsymbol{\theta}}) P(\gamma_{1,1} = 0 \mid \widehat{\boldsymbol{\theta}})}{\prod_{s,l} P(y_{s,l} \mid \gamma_{s,l} = 0, \boldsymbol{g})} \\ &= \frac{P(\mathcal{T}_{1,1} \mid \gamma_{1,1} = 1, \mathbf{g}, \widehat{\boldsymbol{\theta}}) \pi_{1,1} + P(\mathcal{T}_{1,1} \mid \gamma_{1,1} = 0, \mathbf{g}, \widehat{\boldsymbol{\theta}})(1 - \pi_{1,1})}{\prod_{s,l} P(y_{s,l} \mid \gamma_{s,l} = 0, \boldsymbol{g})} \end{aligned} \tag{2.31}$$

where the denominator is from (2.30). As we have used BFs throughout our algorithm, the full form of (2.24) yields the same result as (2.31), and therefore is equivalent to the desired likelihood test statistic in a HMT setting.

Up until now, we have omitted the WC at the zero-th scale from all the HMT-based analysis. To include the effect of this WC in the likelihood ratio test statistic, we note that our model assumes no dependency between this WC and $\widehat{\Lambda}(\mathbf{y}, \mathbf{g})$. We denote Λ_W as the likelihood test statistic for the whole tree, including the WC at the zero-th scale, and Λ_0 as the statistic for the WC at the zero-th scale, which was obtained from *WaveQTL*. Then, we have:

$$\widehat{\Lambda}_W(\mathbf{y}, \mathbf{g}) := \widehat{\Lambda}_0(\mathbf{y}, \mathbf{g})\widehat{\Lambda}(\mathbf{y}, \mathbf{g}) \tag{2.32}$$

2.3.4 Effect size estimation

The posterior distribution of the effect size, conditional on the state, is known, as mentioned in 2.2.4, and is unaffected by our prior tree structure. However, as $P(\gamma_{s,l} = 1 \mid \mathbf{y}, \mathbf{g})$ is no longer independent between scales and locations, the posterior distribution for $(\beta_{s,l} \mid \mathbf{y}, \mathbf{g})$ also contains dependencies between scales and locations. When converting this posterior into the data space, this lack of independence is not an issue for means, which are additive, but is an issue for calculating variances. Therefore, we have opted for a sampling approach to estimating the effect size in the data space.

We sample from the joint $\gamma_{s,l}$ distribution, incorporating the tree-shaped dependencies using the posterior transition probabilities output from the E-step of the Upward-Downward algorithm. Conditional on the $\gamma_{s,l}$'s drawn, we sample from the posterior effect size distribution in the wavelet space, and convert these into data space quantities using the IDWT. The result is a dataset which represents a draw from the posterior effect size distribution in the data space, across all scales and locations. We drew 1000 samples from the posterior distribution, and used the sample means and variances to estimate the posterior effect size means and variances in the data space. For *WaveQTL*, a similar procedure is performed, except that $\gamma_{s,l}$ are each drawn independently from a Bernoulli distribution. See Appendix A for details.

In order to verify that the sampling approach for the model with HMT priors works well, we compare the posterior means and variances against those from *WaveQTL*. Using the DNase-seq dataset and genotypes for the 70 individuals introduced in Section 1.1.1, we expect that effect sizes from the model with HMT priors should be similar to those obtained from *WaveQTL*. In the effect size plots below, the darker blue lines represent the posterior mean effect size, and the lighter blue lines represent the 0.5th and 99.5th percentiles, giving a 99% central credible region. The pink regions mark out locations along the genome where an effect is significantly different from 0, based on these credible regions. As expected, the effect size plots from the model with HMT priors and *WaveQTL* are almost identical, in terms of the means, intervals, and regions of significant effects identified.

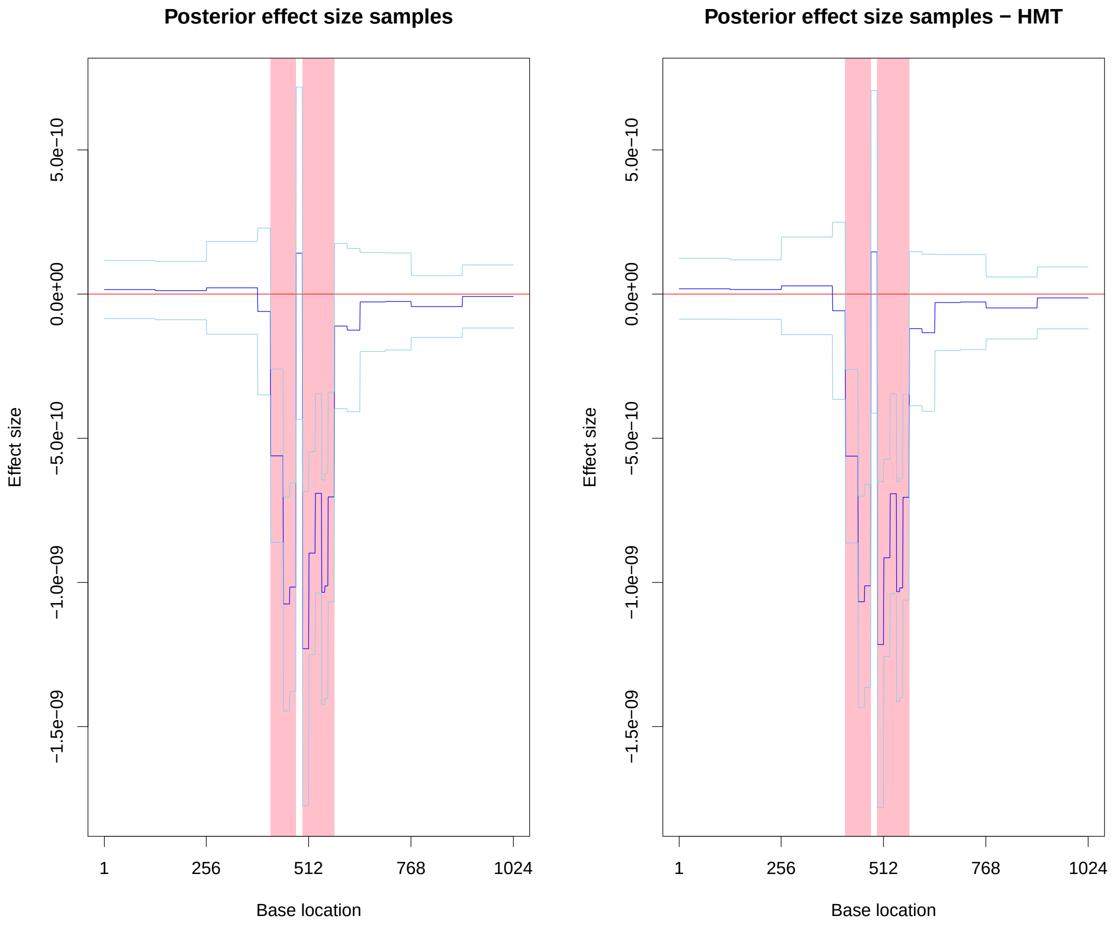


Figure 2.9: Posterior effect size plot.

Chapter 3

WaveQTL-HMT: implementation and analysis

3.1 Implementation

WaveQTL was written in C++, with some functions covering data cleaning and effect size interpretation written in R. We have added to the C++ codebase to add functionality for the model with HMT priors, as introduced in Section 2.3. All the analysis in this chapter runs data through either *WaveQTL,* or *WaveQTL* with HMT priors, which we will call *WaveQTL-HMT.* All model and analysis code can be found at the GitHub repositories listed in Appendix C.

3.2 Practicalities for data analysis

There were some practical considerations which were made when preparing data for use in these models – see Shim and Stephens (2015) for details. In summary, these are:

- *Filtering out low count WCs:* Some WCs may have very low, or zero values because they are computed based on very low counts in the data space. These WCs have high sampling error, which is not accounted for in the Wavelet-based model. In all subsequent analysis, we filter out WCs where the average count per individual was less than 2, and set their $\mathrm{BF}_{s,l} = 1$, corresponding to no information regarding association.

- *Confounding factors:* Association testing results may be affected by confounding factors. For this dataset, unobserved confounding factors were estimated using Principal Components Analysis (PCA) (Degner et al. (2012)), and PCA regression was performed to remove the effects of the first few PCs, leaving the residuals as the inputs to our model.
- *Quantile transformation:* Some WCs may not follow a normal distribution, and as our models assume the residuals are normally distributed, we quantile transform the WCs to the quantiles of a standard normal distribution, before computing BFs and performing association testing. However, this makes effect sizes hard to interpret, so effect sizes are calculated on WCs without a quantile transformation applied.

The steps are performed in the following order: filtering out low-counts, quantile transforming the WCs, regressing out confounding factors, then quantile transforming the WCs again.

3.3 Validation of *WaveQTL-HMT* implementation

3.3.1 Procedure

The focus here is to simulate data using pre-set hyperparameters, and to validate the implementation of our model by investigating whether our algorithm is able to retrieve the true hyperparameters. The hyperparameters which we pre-set are $\pi_{1,1}$, $\varepsilon^{11}_{(s,l),p(s,l)}$ and $\varepsilon^{10}_{(s,l),p(s,l)}$, but as we kept each ε the same across all s, l for this procedure, we will denote these parameters as ε^{11} ("Epsilon 11") and ε^{10} ("Epsilon 10") in this section.

Based off the true hyperparameters, we simulated a dataset of WCs representative of counts from $N = 70$ individuals, across $B = 1024$ base locations, in order to mimic the data presented in Section 1.1.1. We used *WaveQTL-HMT* to estimate the hyperparameters from each of the 200 datasets we simulated, instructing it to perform tying on estimated hyperparameters from scales 1–5, and tying by scale for each of scales 6 until 10. As a result, *WaveQTL-HMT* would output 12 distinct ε hyperparameters (6 ε^{11}'s and 6 ε^{10}'s). More details can be found in Appendix C.

3.3.2 Results

For $\varepsilon^{11} = 0.75, \varepsilon^{10} = 0.25$, Figure 3.1 shows the mean, 2.5th and 97.5th percentile from the 200 hyperparameter estimates. The x-axis represents each tying group, with the labels showing the scales included in each tying group.

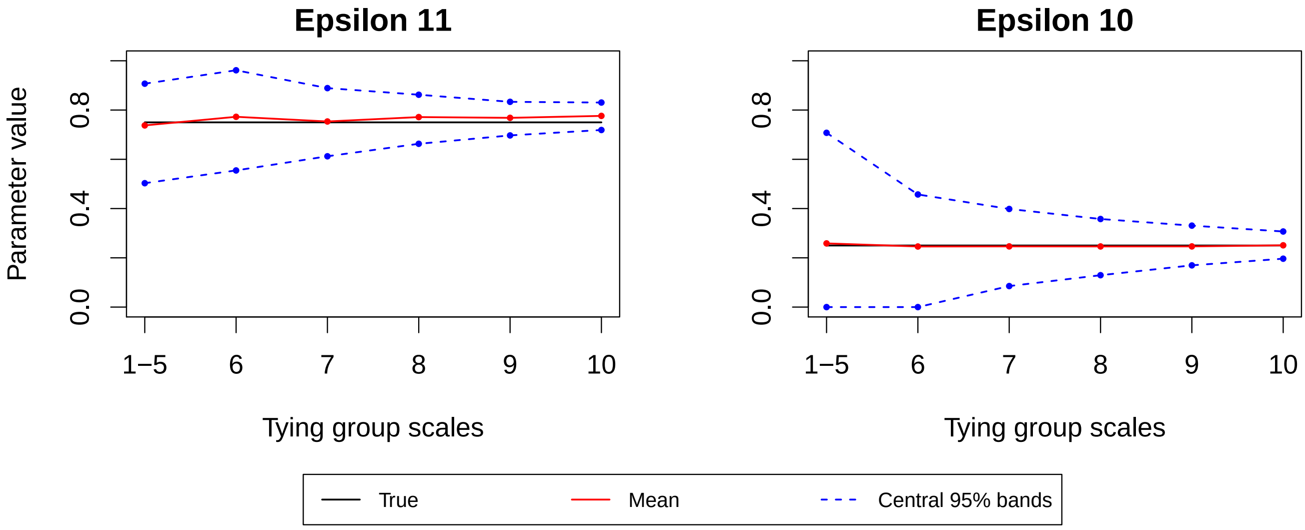


Figure 3.1: *WaveQTL-HMT* had no issues retrieving the simulated ε hyperparameters. The black line is the true hyperparameter, the red line is the mean, and the blue dotted lines represent 2.5th and 97.5th percentiles.

Parameter	2.5th percentile	Mean	97.5th percentile
$\pi_{1,1}$	1	1	1

In this case, the central 95% band included the true hyperparameters, with the same result holding for $\pi_{1,1}$, showing that our model had successfully retrieved the true hyperparameters. We repeated this procedure with different combinations of ε, and the results were similar across all cases, showing that the model had been implemented correctly and was functioning as desired. These results can be found in Appendix C.

3.4 Analysis of simulated data

This section aims to demonstrate where *WaveQTL-HMT* displays more strength in detecting signals over *WaveQTL*. The intuition is that *WaveQTL-HMT* has a better power to identify effects which occur over a narrower interval, but where the signal is not strong enough to be detected by *WaveQTL,* as *WaveQTL-HMT* effectively combines association signals across different scales and locations by modelling dependencies between $\gamma_{s,l}$. However, for a given effect length, if the signal is too strong, both methods will be able to detect the signal and likewise, if too weak, neither will detect the signal.

3.4.1 Procedure

To try to make our simulation more realistic, we based our simulations on estimated effect sizes from the DNase-seq data and genotype data of 70 individuals presented in Section 1.1.1. To assess the impact of the effect *length* as well as the effect *strength* on the performance of *WaveQTL* and *WaveQTL-HMT*, we consider different combinations of effect length and strength in our simulation studies. For each combination, we simulated counts at each base by 'thinning' the DNase-seq data by an amount determined by the effect strength and length from that combination. See Appendix C for more details on the simulation procedure.

First, we simulated a dataset with two groups of individuals with one particular combination of effect size length (16) and strength using the procedure described in Appendix C. Then we assessed the performance of each method by checking whether each method could identify the locations where a true effect was generated as containing a significant effect. Figure 3.2 shows effect sizes in the data space estimated by *WaveQTL* and *WaveQTL-HMT*, respectively. Figure 3.3 shows the same information, focusing only on the bases where a true effect was generated to clearly illustrate the difference between the two models. The pink regions correspond to bases where the posterior mean ± 3 standard deviations does not include zero (± 3 SDs was used to remain consistent with analysis in Shim and Stephens (2015)). We define these to be bases where a significant effect has been detected.

In all locations where true effects were generated, *WaveQTL-HMT* was able to identify these as locations which contained a significant effect, while *WaveQTL* failed to identify almost all of the locations where true effects were generated. This result illustrates that *WaveQTL-HMT* is better powered to detect effects that occur over a narrower region.

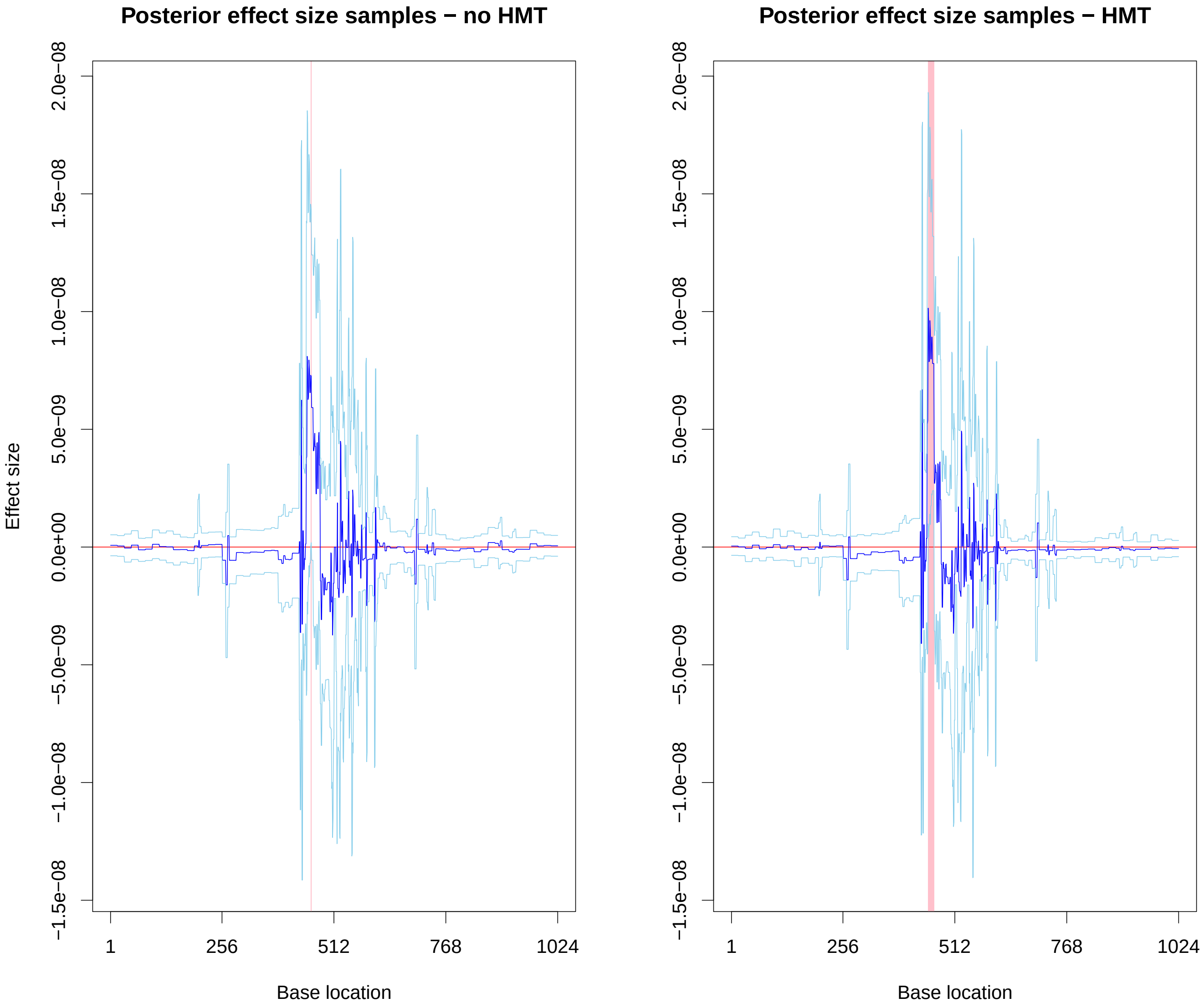


Figure 3.2: Posterior effect size mean, and ±3 SD bands. Pink regions denote bases where a significant effect has been detected – the region is wider under *WaveQTL-HMT* than *WaveQTL*, as *WaveQTL* has had difficulty detecting this effect.

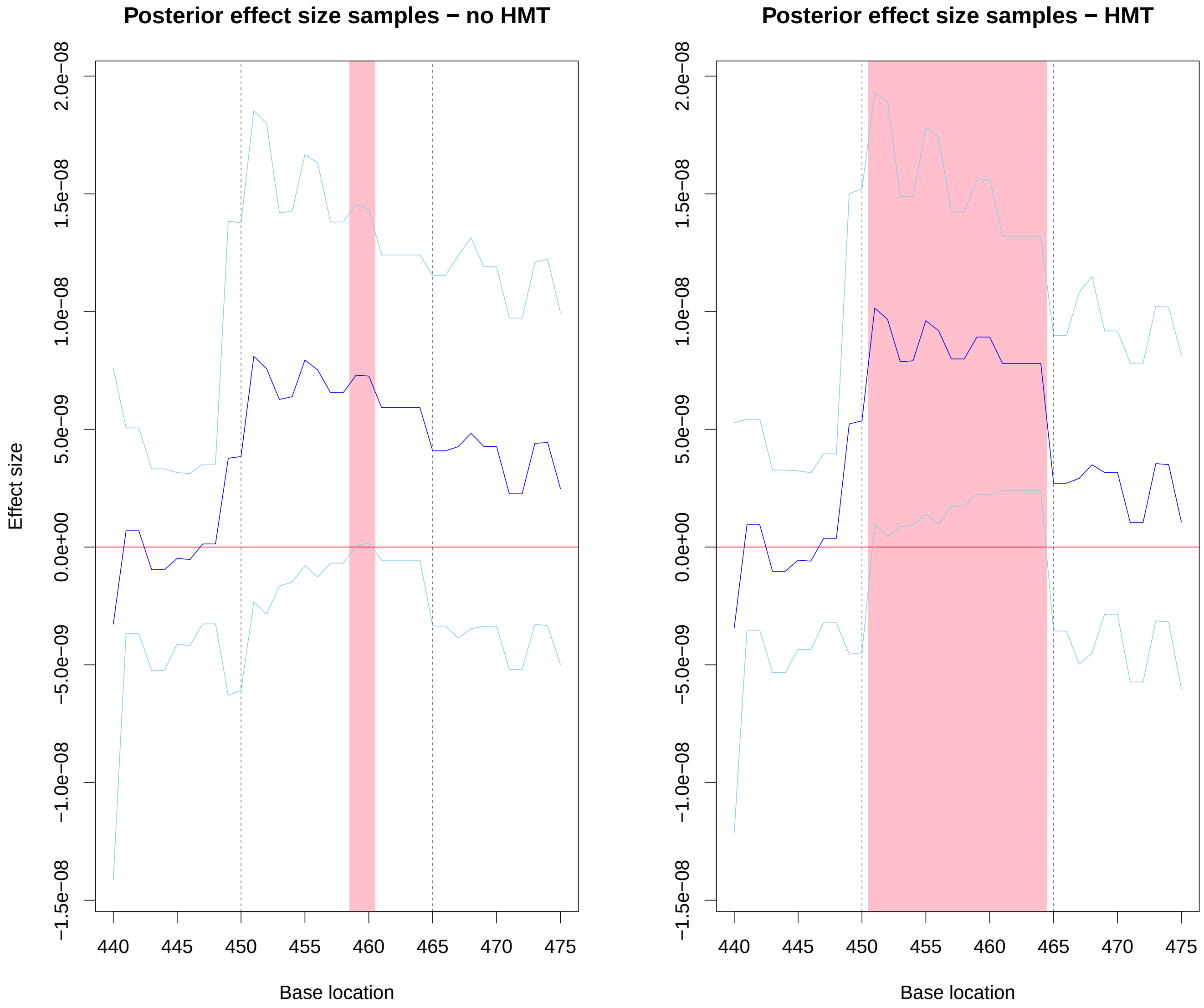
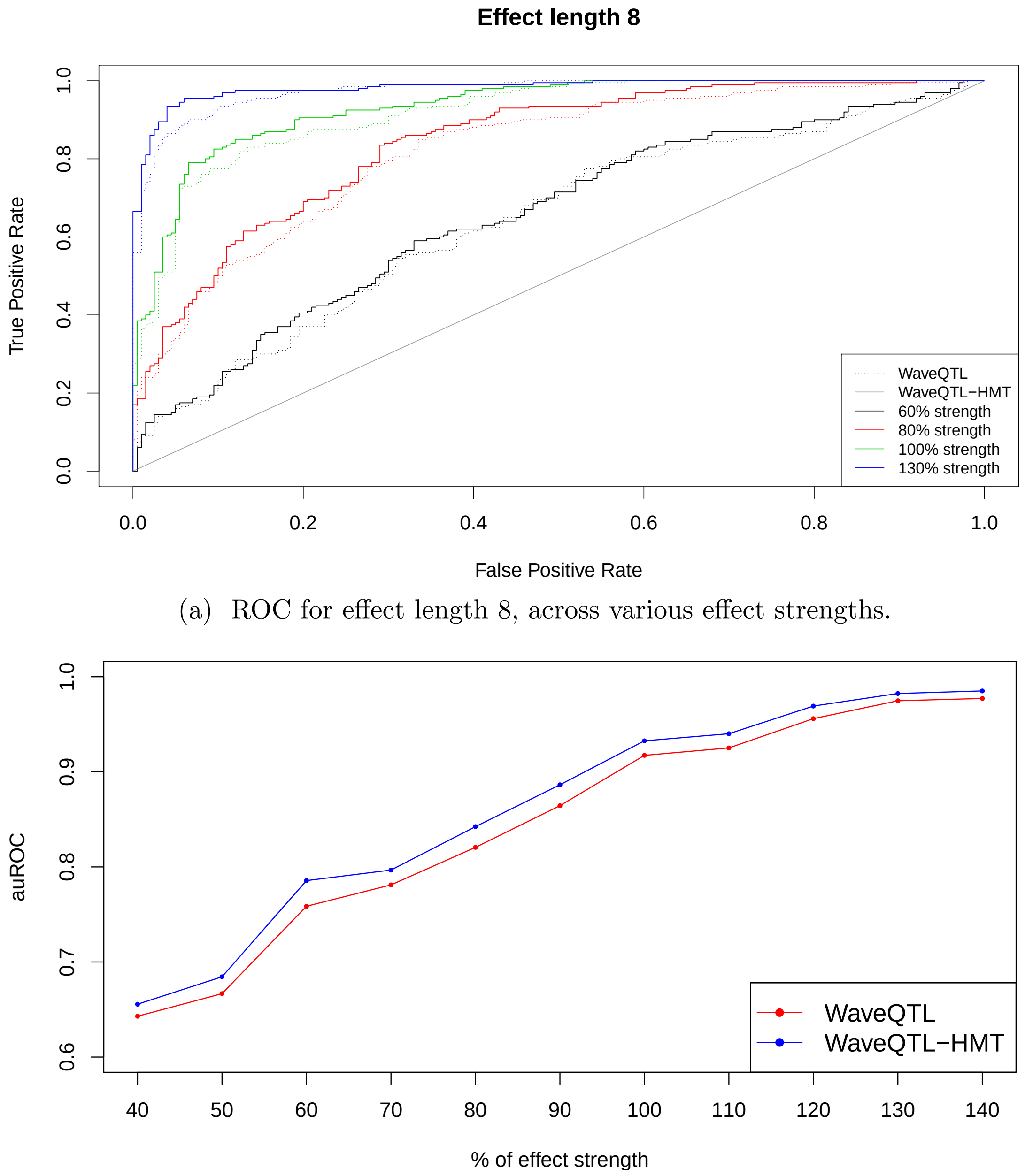


Figure 3.3: Same as above, but focusing on the simulated effect region of length 16. Dashed grey lines denote the boundaries of the simulated effect region, showing that *WaveQTL-HMT* detects an effect throughout the simulated effect region, whereas *WaveQTL* does not.

3.4.2 Results

Next, we compared the two methods for other different combinations of effect length (8, 16, 32 and 64) and strength (40% – 140% of the effect size). To better illustrate the difference in performance between the two methods, we omit combinations where the two methods performed almost identically. For each combination of effect length and strength, we simulated 200 datasets with two groups of individuals using the 'thinning' procedure described above in Section 3.4.1 – these constitute the *alt* datasets. We also simulated 200 datasets using the same procedure but with no effect between the two groups of individuals – the *null* datasets. We then applied both methods to each of the 400 datasets (200 alternative and 200 null), obtained the likelihood ratio test statistic from each method, and assessed the performance of each method by varying the likelihood ratio thresholds to plot out a Receiver-Operator Characteristic (ROC) curve. We define 'true positives' as cases where signals were detected in *alt* datasets, and 'false positives' as cases where signals were detected in *null* datasets, based on these thresholds.

Figure 3.4a) shows ROC curves from both methods for effect length 8 and with effect strengths varying from 60% (black line) to 130% (blue line). Figure 3.4b) shows the respective area under the ROC (auROC) statistics.



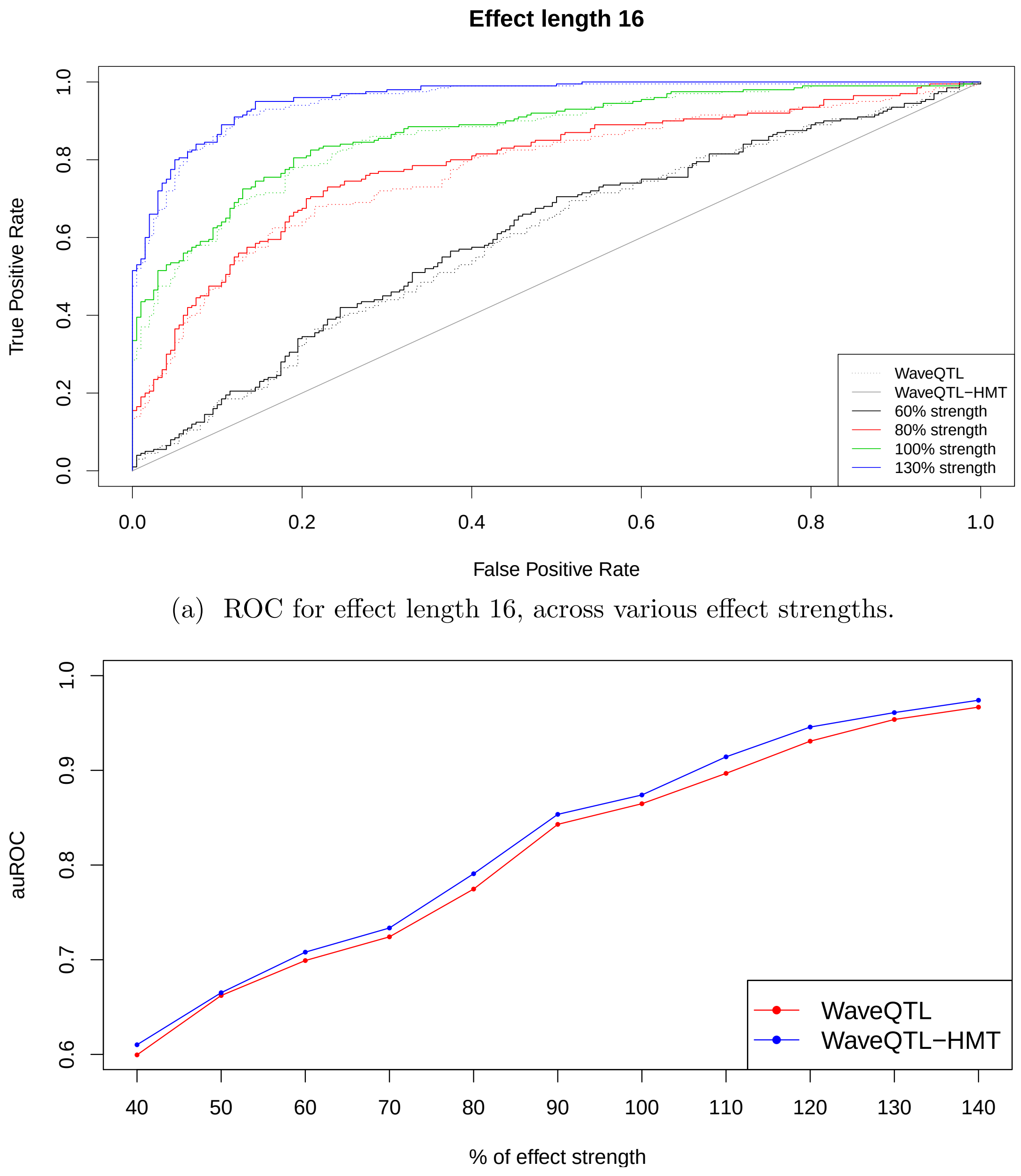
(a) ROC for effect length 8, across various effect strengths.

(b) auROC summaries for effect length 8, by effect strength.

Figure 3.4

We can see that, for a small effect strength, there is minimal difference between the two methods, as both struggle to identify the generated signals. *WaveQTL-HMT* begins to outperform *WaveQTL* for the increased effect strengths represented by the red, green and blue curves, as it is able to identify more of the true positives than

WaveQTL without significantly increasing the number of false positives it identified. We do the same for effect length 16:

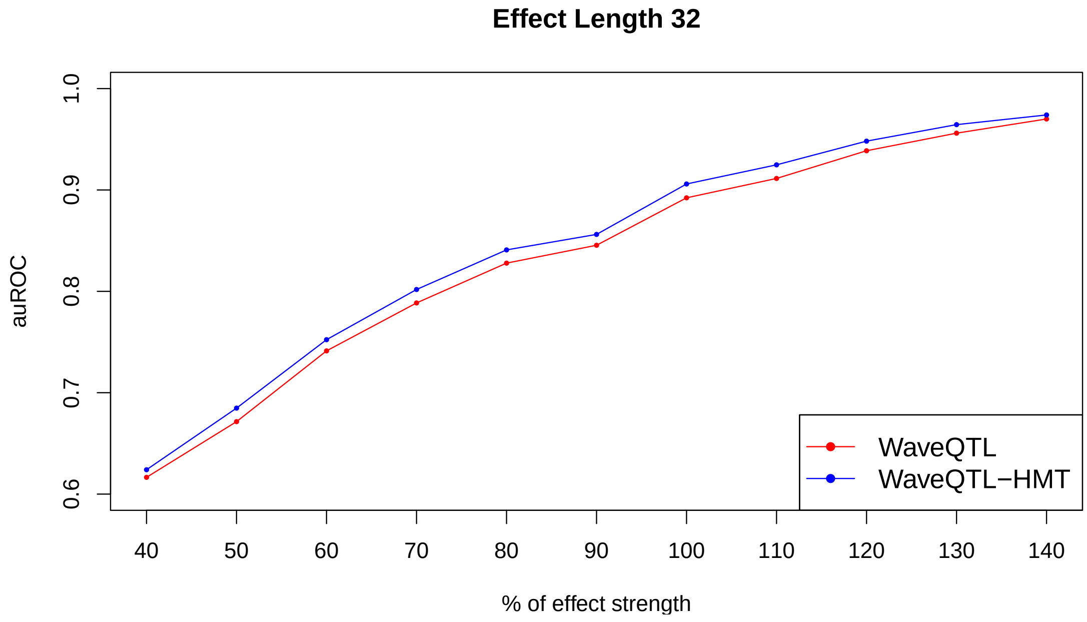


(a) ROC for effect length 16, across various effect strengths.

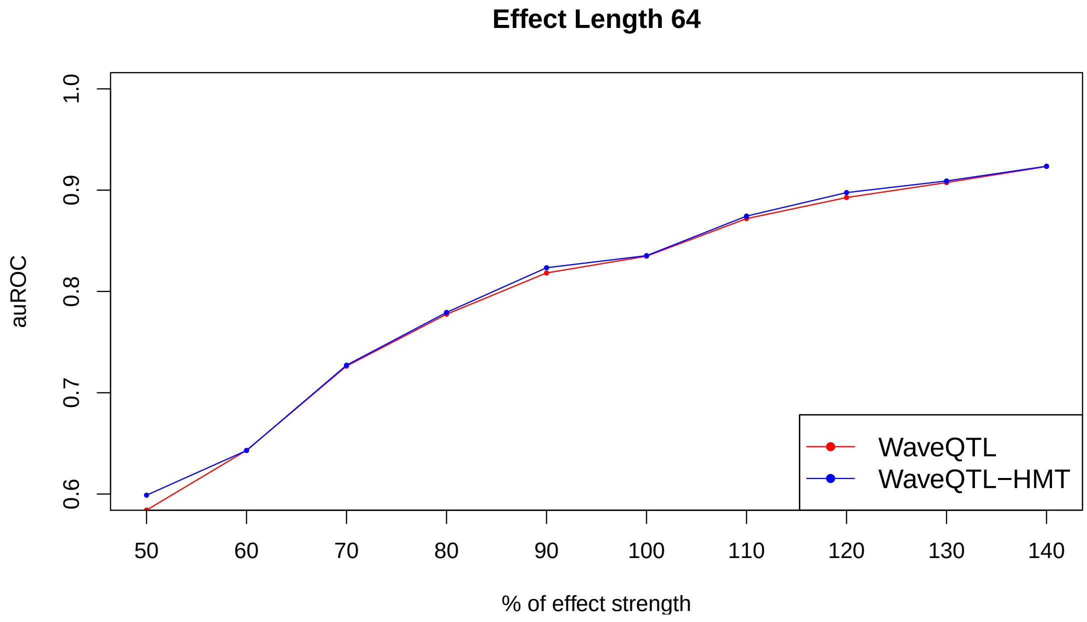
(b) auROC summaries for effect length 16, by effect strength.

Figure 3.5

We did the same for effects with length 32 and 64, with their auROCs as follows (respective ROCs can be found in Appendix C):



(a) auROC summaries for effect length 32, by effect strength.



(b) auROC summaries for effect length 64, by effect strength.

Figure 3.6

We can see the distinction between *WaveQTL-HMT* and *WaveQTL* decreases as the effect lengths increase, and has become almost indistinguishable for effect length 64.

Our findings appeared to confirm our initial intuition, which was that *WaveQTL-HMT* is better powered than *WaveQTL* for identifying signals which are narrow in length. For effects of length 8 and 16, for example, *WaveQTL-HMT* was able to do this consistently, over a wide range of simulated effect strengths. As expected, the difference between the two methods decreased as the simulated effect lengths increased. We note that all of the simulations were run on a sequence of length 1024. However, it is likely that the effect lengths by which *WaveQTL-HMT* would show improvement over *WaveQTL* is relative to the length of the sequence analysed. For example, we believe *WaveQTL-HMT* would be able to show greater improvements over *WaveQTL* in detecting effects of length 32 if run over a data sequence which is 2048 bases long.

Chapter 4

Multiscale Poisson models with HMT priors

A natural extension when dealing with count data, such as sequencing data, is to treat the observed counts as noisy observations of an inhomogeneous Poisson process. In this chapter, we describe a model where the data is projected onto a multiscale space using a natural analogue to a Wavelet transform which deals directly with the Poisson assumption. The motivation behind this is to address some key limitations to the Wavelet-based model which was presented in Chapter 2 surrounding normal distribution approximations for count data, and quantile transforms. Treating the underlying count data as observations from a Poisson distribution as the basis for our multiscale model allows us to reduce the potential for information loss in the face of low counts and small samples – see Shim and Stephens (2015) and Shim et al. (in preparation) for details.

In this chapter we present the multiscale Poisson transformation and then outline the structure of a multiscale Poisson model with HMT priors. The multiscale Poisson model with independent priors – *multiseq* – was presented in Shim et al. (in preparation) and Xing (2016), and form the basis for the key ideas in our model.

4.1 A multiscale decomposition for Poisson data

In this model, we assume that our data, d_b, are counts drawn from independent Poisson distributions, that is, $(d_b \mid \lambda_b) \overset{d}{\sim} \text{Poisson}(\lambda_b)$, for each base $b = 1, \dots, B$.

We can think of these data points as observations from an inhomogeneous Poisson process which is driven by some underlying intensity function, λ, which has spatial structure. Our goal is to use this underlying function to identify potential genetic variants associated with molecular-level phenotypes, as well as estimate the effect that particular genotypes have on molecular-level phenotypes. As was mentioned in Chapter 2, projecting the data onto a multiscale space results in a sparse structure where we are better able to adapt to effects which are not of a fixed length. For a multiscale decomposition in a Poisson setting, we turn to Kolaczyk (1999) and Timmerman and Nowak (1999), which use recursive dyadic partitions to factorise the likelihood into a binary tree representation, analogous to what the DWT achieved in Chapter 2.

Denote:

$$d_{1:B} := \sum_{b=1}^{B} d_b$$
$$\lambda_{1:B} := \sum_{b=1}^{B} \lambda_b$$

Recall that two independent Poisson random variables, X and Y, with respective parameters λ_X and λ_Y may have their joint distributions expressed as:

$$P(X, Y \mid \lambda_X, \lambda_Y) = P(X + Y \mid \lambda_X + \lambda_Y) P\Big(X \mid X + Y, \frac{\lambda_X}{\lambda_X + \lambda_Y}\Big) \tag{4.1}$$

where:

$$(X + Y \mid \lambda_X + \lambda_Y) \overset{d}{\sim} \text{Poisson}(\lambda_X + \lambda_Y)$$
$$\Big(X \mid X + Y, \frac{\lambda_X}{\lambda_X + \lambda_Y}\Big) \overset{d}{\sim} \text{Binomial}\Big(X + Y, \frac{\lambda_X}{\lambda_X + \lambda_Y}\Big)$$

As an example, if $B = 4$, considering each adjacent non-overlapping pair of d_j's yields:

$$\begin{aligned} &P(d_1, \ldots, d_4 \mid \lambda_1, \ldots, \lambda_4) \\ &\qquad = P\Big(d_1 \mid d_{1:2}, \frac{\lambda_1}{\lambda_{1:2}}\Big) P(d_{1:2} \mid \lambda_{1:2}) P\Big(d_3 \mid d_{3:4}, \frac{\lambda_3}{\lambda_{3:4}}\Big) P(d_{3:4} \mid \lambda_{3:4}) \\ &\qquad = P\Big(d_1 \mid d_{1:2}, \frac{\lambda_1}{\lambda_{1:2}}\Big) P\Big(d_3 \mid d_{3:4}, \frac{\lambda_3}{\lambda_{3:4}}\Big) \underbrace{P(d_{1:2}, d_{3:4} \mid \lambda_{1:2}, \lambda_{3:4})}_{(\star)} \end{aligned}$$

Recursively applying the same procedure to $(\star)$, we get:

$$= P\Big(d_1 \mid d_{1:2}, \frac{\lambda_1}{\lambda_{1:2}}\Big) P\Big(d_3 \mid d_{3:4}, \frac{\lambda_3}{\lambda_{3:4}}\Big) P\Big(d_{1:2} \mid d_{1:4}, \frac{\lambda_{1:2}}{\lambda_{1:4}}\Big) P(d_{1:4} \mid \lambda_{1:4})$$

where:

$$\Big(d_1 \mid d_{1:2}, \frac{\lambda_1}{\lambda_{1:2}}\Big) \overset{d}{\sim} \text{Binomial}\Big(d_{1:2}, \frac{\lambda_1}{\lambda_{1:2}}\Big) \tag{4.2}$$

$$\Big(d_3 \mid d_{3:4}, \frac{\lambda_3}{\lambda_{3:4}}\Big) \overset{d}{\sim} \text{Binomial}\Big(d_{3:4}, \frac{\lambda_3}{\lambda_{3:4}}\Big) \tag{4.3}$$

$$\Big(d_{1:2} \mid d_{1:4}, \frac{\lambda_{1:2}}{\lambda_{1:4}}\Big) \overset{d}{\sim} \text{Binomial}\Big(d_{1:4}, \frac{\lambda_{1:2}}{\lambda_{1:4}}\Big) \tag{4.4}$$

$$(d_{1:4} \mid \lambda_{1:4}) \overset{d}{\sim} \text{Poisson}(\lambda_{1:4}) \tag{4.5}$$

Together, (4.2) – (4.5) represents a multiscale decomposition of the joint distribution, $d_1, \ldots, d_4$, that is equivalent to $d_j \overset{d}{\sim} \text{Poisson}(\lambda_j)$. (4.5) is a summary of d's information at the coarsest scale, and (4.2) – (4.3) represent summaries at the finest level. Figure 4.1 shows this decomposition graphically:

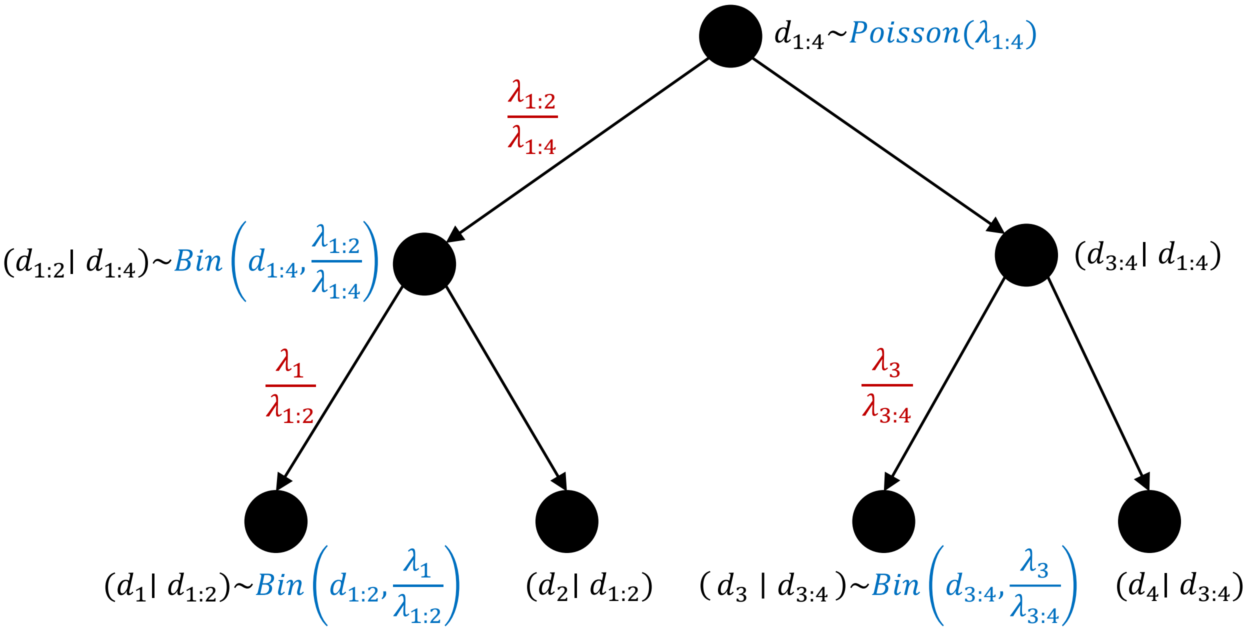

$d_{1:4} \sim Poisson(\lambda_{1:4})$
$\frac{\lambda_{1:2}}{\lambda_{1:4}}$
$(d_{1:2} | d_{1:4}) \sim Bin\left(d_{1:4}, \frac{\lambda_{1:2}}{\lambda_{1:4}}\right)$
$(d_{3:4} | d_{1:4})$
$\frac{\lambda_1}{\lambda_{1:2}}$
$\frac{\lambda_3}{\lambda_{3:4}}$
$(d_1 | d_{1:2}) \sim Bin\left(d_{1:2}, \frac{\lambda_1}{\lambda_{1:2}}\right)$
$(d_2 | d_{1:2})$
$(d_3 | d_{3:4}) \sim Bin\left(d_{3:4}, \frac{\lambda_3}{\lambda_{3:4}}\right)$
$(d_4 | d_{3:4})$

Figure 4.1: Representation of the decomposition with observed (conditional) quantities in black, distributions represented in blue, and the proportion of counts from one scale to the next in red.

Moreover, we can use the above to reparameterise the joint distribution while preserving all the information from before:

$$P(d_1, \ldots, d_4 \mid \lambda_1, \ldots, \lambda_4) = P\Big(d_1, \ldots, d_4 \mid \lambda_{1:4}, \frac{\lambda_{1:2}}{\lambda_{1:4}}, \frac{\lambda_1}{\lambda_{1:2}}, \frac{\lambda_3}{\lambda_{3:4}}\Big)$$

The parameters – which now span multiple resolutions – contain important information for capturing spatial structure in the underlying λ. For example, $\lambda_1/\lambda_{1:2}$ represents the contrast in intensity at the finest scale, as opposed to $\lambda_{1:2}/\lambda_{1:4}$, which represents contrast at a coarser scale.

Extending this out to a vector of length $B = 2^J, J \in \mathbb{N}$, the result is as follows:

$$\begin{aligned} P(d_1, \ldots, d_B \mid \lambda_1, \ldots, \lambda_B) = P\Big(d_1 \mid d_{1:2}, \frac{\lambda_1}{\lambda_{1:2}}\Big) \times \cdots \times P\Big(d_{B-1} \mid d_{(B-1):B}, \frac{\lambda_{B-1}}{\lambda_{(B-1):B}}\Big) \times \ldots \\ P\Big(d_{1:B/2} \mid d_{1:B}, \frac{\lambda_{1:B/2}}{\lambda_{1:B}}\Big) P(d_{1:B} \mid \lambda_{1:B}) \end{aligned} \tag{4.6}$$

Shim et al. (in preparation) and Xing (2016) then take a logit transformation (log-odds) of these quantities to make them more suitable for their model. The quantity at the ‘zero-th’ scale is defined slightly differently:

$$\alpha_{0,1} := \log(\lambda_{1:B}) \tag{4.7}$$

whilst the remaining quantities, much like those obtained from the DWT, represent (the log of) differences between adjacent locations at different scales, until the finest scale, J, which captures the differences between intensities at adjacent bases in the data space:

$$\alpha_{1,1} := \operatorname{logit}\Big(\frac{\lambda_{1:(B/2)}}{\lambda_{1:B}}\Big) = \log\big(\lambda_{1:(B/2)}\big) - \log\big(\lambda_{((B/2)+1):B}\big) \tag{4.8}$$

$$\vdots$$

$$\alpha_{J,1} := \operatorname{logit}\Big(\frac{\lambda_{1}}{\lambda_{1:2}}\Big) = \log(\lambda_{1}) - \log(\lambda_{2}) \tag{4.9}$$

$$\vdots$$

$$\alpha_{J,2^{J-1}} := \operatorname{logit}\Big(\frac{\lambda_{B-1}}{\lambda_{(B-1):B}}\Big) = \log(\lambda_{B-1}) - \log(\lambda_{B}) \tag{4.10}$$

From this, it is evident that this procedure represents an analogue of the Haar DWT of λ when applied to Poisson data. Note that this still represents a one-to-one transformation of the parameter space as $\boldsymbol{\alpha}$ still contains the same information as the original parameters, $\boldsymbol{\lambda}$. We can now express the joint likelihood as follows:

$$P(d_1, \ldots, d_B \mid \lambda_1, \ldots, \lambda_B) = P(d_1, \ldots, d_B \mid \alpha_{0,1}, \alpha_{1,1}, \ldots, \alpha_{J,1}, \ldots, \alpha_{J,2^{J-1}})$$

Due to the logit transformation, spatial structure in λ will correspond to the majority of α’s being close to 0, with any signal typified by a small subset of non-zero α parameters. This transformation is represented in Figure 4.2:

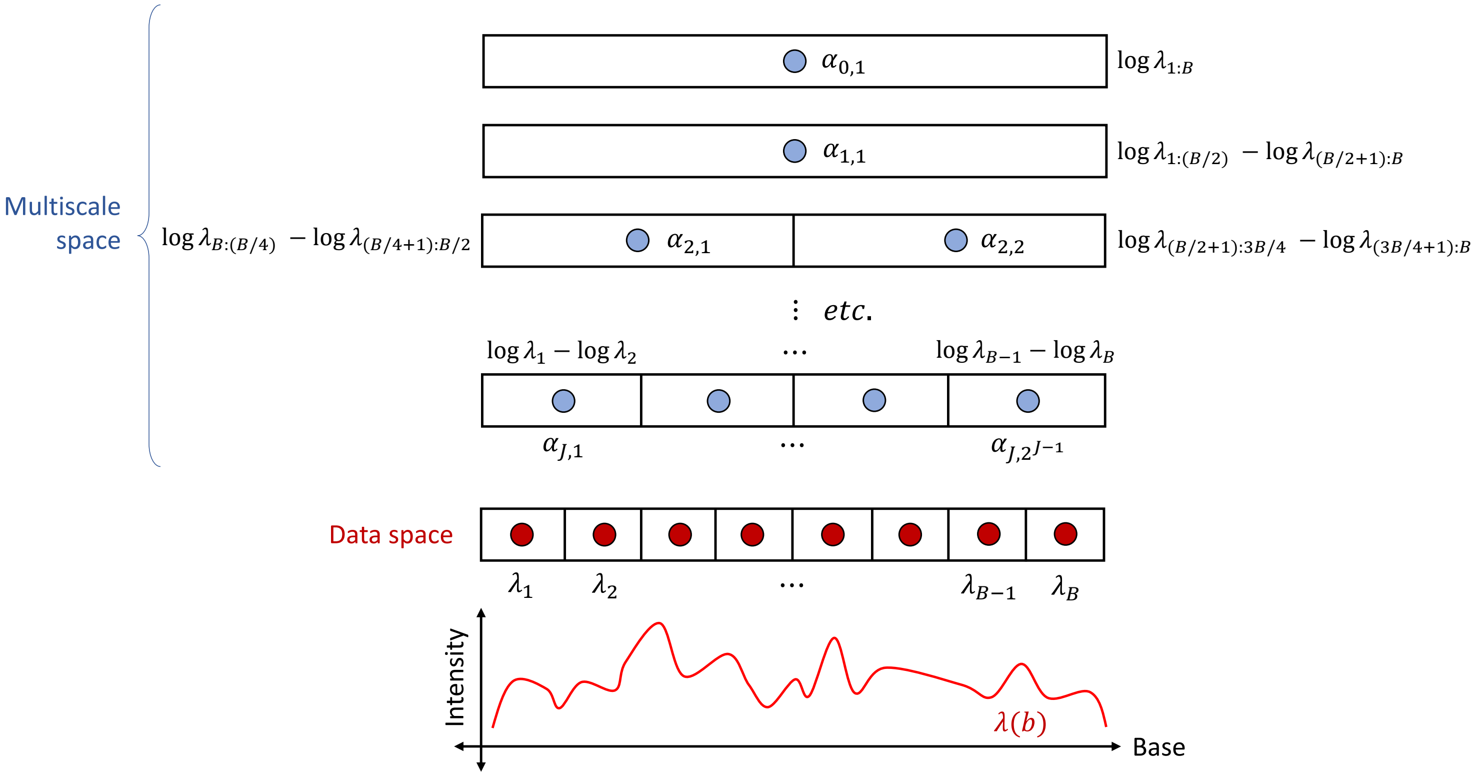

Multiscale space
$\alpha_{0,1}$
$\log \lambda_{1:B}$
$\alpha_{1,1}$
$\log \lambda_{1:(B/2)} - \log \lambda_{(B/2+1):B}$
$\log \lambda_{B:(B/4)} - \log \lambda_{(B/4+1):B/2}$
$\alpha_{2,1}$
$\alpha_{2,2}$
$\log \lambda_{(B/2+1):3B/4} - \log \lambda_{(3B/4+1):B}$
⋮ etc.
$\log \lambda_1 - \log \lambda_2$
…
$\log \lambda_{B-1} - \log \lambda_B$
$\alpha_{J,1}$
…
$\alpha_{J,2^{J-1}}$
Data space
λ_1
λ_2
…
λ_{B-1}
λ_B
Intensity
$\lambda(b)$
Base

Figure 4.2: Representation of Poisson multiscale decomposition, and the relationship between multiscale parameters and the original data. The graph at the bottom represents the underlying intensity function (red).

4.2 Model with HMT priors

4.2.1 *multiseq*

We start by outlining the areas of the *multiseq* model as described in Shim et al. (in preparation) and Xing (2016), which are relevant to our model with HMT priors. We observe a vector of normalised counts, $\mathbf{d}^i = (d_1^i, \ldots, d_B^i)$, across each base, b, and individual i, as well as some covariate of interest, g^i. The relationship between the underlying intensity function and the covariate of interest is modelled as follows:

$$\log\left(\lambda_b^i\right) = \mu_b^{(d)} + \beta_b^{(d)} g^i + v_b^i \tag{4.11}$$

The quantity of interest is the coefficient, $\beta_b^{(d)}$, which measures the effect of a genotype on a molecular-level phenotype. The superscript $^{(d)}$ is used to denote a data space quantity, which they assume has spatial structure. $\mu_b^{(d)}$ denotes the mean $\log(\lambda_b)$ of counts where $g^i = 0$, and v_b^i is a zero-mean random effect term used to capture additional variation across individuals (over-dispersion).

From the multiscale Poisson decomposition, they have, for each individual:

$$(d_{1:B}^i \mid \lambda_{1:B}^i) \overset{d}{\sim} \text{Poisson}(\lambda_{1:B}^i) \tag{4.12}$$

$$(d_{1:(B/2)}^i \mid d_{1:B}^i, \alpha_{1,1}^i) \overset{d}{\sim} \text{Binomial}(d_{1:B}^i, \text{logit}^{-1}(\alpha_{1,1}^i)) \tag{4.13}$$

$$\vdots$$

$$(d_{B-1}^i \mid d_{(B-1):B}^i, \alpha_{J,2^{J-1}}^i) \overset{d}{\sim} \text{Binomial}(d_{(B-1):B}^i, \text{logit}^{-1}(\alpha_{J,2^{J-1}}^i)) \tag{4.14}$$

Starting with the quantity at the 'zero-th' scale, they model the multiscale quantities as follows:

$$\log\left(\lambda_{1:B}^i\right) = \mu_{0,1} + \beta_{0,1} g^i + u_{0,1}^i \tag{4.15}$$

and in general, for the remaining multiscale quantities at all other s, l:

$$\alpha_{s,l}^i = \mu_{s,l} + \beta_{s,l} g^i + u_{s,l}^i \tag{4.16}$$

where $\mu_{s,l}$ is the mean of $\alpha_{s,l}^i$ (or $\log(\lambda_{1:B}^i)$ for the 'zero-th' scale) when $g^i = 0$ and $\beta_{s,l}$ captures the relationship between the covariate and intensity function in the multiscale space. $u_{s,l}^i$ is a zero-mean random-effect to capture overdispersion in the multiscale space.

The probability distribution function (PDF) of the likelihood of $(\mu_{s,l}, \beta_{s,l})$ can be approximated by the product of the PDF of two univariate normal distributions – one of which contains $\beta_{s,l}$, and the other of which contains $\mu_{s,l}^*$, a re-parameterised version of $\mu_{s,l}$ which contains $\beta_{s,l}$. This is after applying a Laplace approximation which involves the MLEs of both the parameters – see Shim et al. (in preparation) and Xing (2016) for more details.

The priors used in Shim et al. (in preparation) and Xing (2016) are similar to those in the Wavelet-based model, with a key difference being that no priors are placed on

$\sigma^2_{s,l}$. Instead, they represent the prior for $\beta_{s,l}$ by a mixture of a point mass at zero and M normal distributions with zero mean and *fixed* $\sigma^2_{s,l}$ – a *grid* of $\sigma^2_{s,l}$, chosen to span a range of small and large effects. Therefore, priors on $\beta_{s,l}$ can be written as:

$$\beta_{s,l} \overset{d}{\sim} \mathbb{1}\{\gamma_{s,l} = 0\}\delta_0 + \sum_{m=1}^{M} \mathbb{1}\{\gamma_{s,l} = m\}N(0, \sigma^2{}_m) \tag{4.17}$$

where $\gamma_{s,l} \in \{0, \dots, M\}$, δ_0 is a point mass at 0, and $(\sigma_1^2, \dots, \sigma_M^2)$ are the grid of known variances. Furthermore, in their model, the $\gamma_{s,l}$'s have independent priors across scales and locations, similar to *WaveQTL*. Therefore, the model's hyperparameters consist of the mixing proportions for each scale, s, across each value of $m = 0, \dots, M$:

$$P(\gamma_{s,l} = m \mid \boldsymbol{\pi}) = \pi_s^{(m)} \tag{4.18}$$

where $\boldsymbol{\pi}$ is the vector containing all the above hyperparameters. The mixing proportions are estimated from the data using an EB approach and an EM algorithm, implemented in *ash* (Stephens (2016)).

As mentioned previously, $\mu^*_{s,l}$ contains elements of $\beta_{s,l}$ and therefore its prior must have a structure which reflects that of the mixture prior of $\beta_{s,l}$. They enforce similar priors on $\mu^*_{s,l}$ with its own set of parameters and hyperparameters:

$$\mu^*_{s,l} \overset{d}{\sim} \mathbb{1}\{\gamma_{s,l}^{(\mu)} = 0\}\delta_0 + \sum_{m=1}^{M} \mathbb{1}\{\gamma_{s,l}^{(\mu)} = m\}N(0, \sigma^{2(\mu)}_{m}) \tag{4.19}$$

Note that the priors placed on $\beta_{s,l}$ and $\mu^*_{s,l}$ are independent, due to the reparameterisation. In their model, they also provide an (approximate) Bayes Factor (ABF) in closed form, a framework to test for association between a genotype and molecular-level phenotype through likelihood ratio test statistics, and the closed form of the posterior on $\beta_{s,l}$ in the multi-scale space, which is a mixture of a point mass at 0 and M normal distributions.

4.2.2 *multiseq* with HMT prior

In our model, both $\beta_{s,l}$ and $\mu^*_{s,l}$ are modelled with a HMT prior by imposing a tree-shaped structure on their respective $\gamma_{s,l}$ and $\gamma^{(\mu)}_{s,l}$ parameters, each governed by their own set of hyperparameters, $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^{(\mu)}$. For simplicity, in the remaining part of this section and in the next section we only discuss details as they apply to $\gamma_{s,l}$ and $\boldsymbol{\theta}$, noting that the same applies to $\gamma^{(\mu)}_{s,l}$ and $\boldsymbol{\theta}^{(\mu)}$.

As seen previously in (2.10), a HMT prior means that the joint distribution of all the $\gamma_{s,l}$'s are now a product of the probability of the root state, and the probability of transitions to each subsequent child in the tree. Similarly, the two sets of hyperparameters of interest are $\pi^{(k)}_{s,l}$, the mixing probability for state k at s, l ($\gamma_{s,l} = k$), and $\varepsilon^{kl}_{(s,l),(p(s,l)}$, the transition probability from state l to k, now expanded to cover $k, l \in \{0, \dots, M\}$.

4.2.3 Hyperparmameter choice

The only hyperparameters in our model are those contained in $\boldsymbol{\theta}$, relating to the mixing weights of the $\beta_{s,l}$ prior. The MLEs of these hyperparameters are estimated using the Upward-Downward algorithm, as described in 2.3.2.1, run on the ABFs which apply to this model, and expanded to accommodate the M mixtures of normal distributions. Note that in practice, we must run two Upward-Downward algorithms to estimate the hyperparameters to both the $\beta_{s,l}$ and $\mu^*_{s,l}$ trees, which are independent of each other.

4.2.4 Association testing

The setup here is similar to Section 2.3.3. The null hypothesis, H_0, holds when we have a set of parameters where the probability of the root state being 1 is 0, and where all subsequent transitions out of a 0 state has probability 0:

$$\pi^{(0)}_{1,1} = 1 \Rightarrow \pi^{(k)}_{1,1} = 0, \qquad k = 1, \dots, M \tag{4.20}$$

$$\varepsilon^{00}_{(s,l),p(s,l)} = 1 \Rightarrow \varepsilon^{k0}_{(s,l),p(s,l)} = 0, \qquad k = 1, \dots, M \tag{4.21}$$

which is equivalent to having $\gamma_{s,l} = 0$ at all s, l. Denoting the above parameter set by $\boldsymbol{\theta}_0$, the likelihood ratio is as follows:

$$\Lambda(\boldsymbol{\theta}; \mathbf{x}, \mathbf{g}) := \frac{P(\mathbf{x} \mid \mathbf{g}, \boldsymbol{\theta}, \boldsymbol{\theta}^{(\mu)})}{P(\mathbf{x} \mid \mathbf{g}, \boldsymbol{\theta}_0, \boldsymbol{\theta}^{(\mu)})} \tag{4.22}$$

It is worth noting that we are only placing restrictions on the set of hyperparameters related to $\beta_{s,l}$ as we are testing against a hypothesis where $\gamma_{s,l} = 0$, regardless of the value of $\gamma_{s,l}^{(\mu)}$, or its respective hyperparameters. Therefore, there is no restriction placed on $\boldsymbol{\theta}^{(\mu)}$. The likelihood ratio is calculated at the MLEs of the hyperparameter sets, and similar to (2.24), can be evaluated by recursive quantities from our Upward-Downward algorithm:

$$\therefore \widehat{\Lambda}(\mathbf{x}, \mathbf{g}) = \sum_{k=0}^{M} \beta_{1,1}^{(EM)} \alpha_{1,1}^{(EM)} \tag{4.23}$$

See Appendix D for the full derivation. Furthermore, we can account for the coefficient at the zero-th scale in our likelihood ratio test statistic in a similar fashion to the Wavelet-based model. That is, our new likelihood ratio test statistic for the whole tree, $\widehat{\Lambda}_W(\mathbf{x}, \mathbf{g})$ becomes:

$$\widehat{\Lambda}_W(\mathbf{x}, \mathbf{g}) := \widehat{\Lambda}_0(\mathbf{x}, \mathbf{g}) \widehat{\Lambda}(\mathbf{x}, \mathbf{g}) \tag{4.24}$$

where $\widehat{\Lambda}_0(\mathbf{x}, \mathbf{g})$ is the statistic for the coefficient at the zero-th scale, which is obtained from *multiseq*.

4.2.5 Effect size estimation

The goal in this section is to approximate the pointwise means and variances of the posterior effect size in the data space, noting that a closed form of the posterior distribution in the data space is not available. Other types of posterior inference, such as credible intervals, can be attained by sampling from the posterior of $\beta_{s,l}^{(d)}$. The sampling method is similar to the one described in 2.3.4, and so we only describe the method to approximate the pointwise quantities here.

To obtain the posterior mean and variances, there are two key steps: (i) converting the effect size in the multiscale space into a data space quantity, $\beta_{s,l}^{(d)}$, by exploiting the relationship between α and λ, and then, (ii) finding means of this expression using Taylor series approximations, and for the variance, a Dynamic Programming (DP) algorithm to handle the tree-structured dependencies, based on work by Shim and Zuk (unpublished). In this section, we illustrate the key ideas behind (i), as explained in Shim et al. (in preparation) and Xing (2016), in order to explain why the DP algorithm is necessary, and then proceed to give a high-level overview about how the DP algorithm works. We make some simplifying assumptions:

- g^i is a categorical group indicator, admitting only two values; either 0 or 1
- There is no random effect in either space, i.e. $v_b^i = u_{s,l}^i = 0$

Such analysis can easily be extended to the case of a quantitative covariate, g^i with a random effect, see Shim et al. (in preparation) and Xing (2016) for details. With these assumptions, notation can be simplified by considering the model at a group level using the superscripts ^{m} (i.e. g^m, α^m, λ^m) for groups $m \in \{0, 1\}$ rather than rather than at an individual level with superscript ^{i}.

The transformations described in (4.7) – (4.10) describe the relationship between λ^m and α^m:

$$\lambda_{1:B}^m = \exp\bigl(\alpha_{0,1}^m\bigr) \tag{4.25}$$

$$p_{s,l}^m = \frac{\exp\bigl(\alpha_{s,l}^m\bigr)}{1 + \exp\bigl(\alpha_{s,l}^m\bigr)} \tag{4.26}$$

$$q_{s,l}^m := 1 - p_{s,l}^m = \frac{1}{1 + \exp\bigl(\alpha_{s,l}^m\bigr)} \tag{4.27}$$

where, for example, $p_{1,1}^m = \lambda_{1:(B/2)}/\lambda_{1:B}$ – the only location at the coarsest scale of the multiscale transform – represented the proportion of total counts from bases 1 to B which were propagated down the left branch of the tree (as per Figure 4.1). Similarly, $q_{1,1}^m$ represents the proportions of counts propagated down the right branch of the tree. By thinking of p's and q's as movements left and right down the tree, sequences of p's and q's are multiplied together to obtain the data space intensity at the desired base, λ_b. Using a small example where $B = 8$ ($J = 3$), to obtain the intensity at the leftmost base:

$$\lambda_1^m = \lambda_{1:8}^m \frac{\lambda_{1:4}^m}{\lambda_{1:8}^m} \frac{\lambda_{1:2}^m}{\lambda_{1:4}^m} \frac{\lambda_1^m}{\lambda_{1:2}^m} \tag{4.28}$$

$$= \exp\left(\alpha_{0,1}^m\right) p_{1,1}^m p_{2,1}^m p_{3,1}^m \tag{4.29}$$

Graphically, this can be seen as:

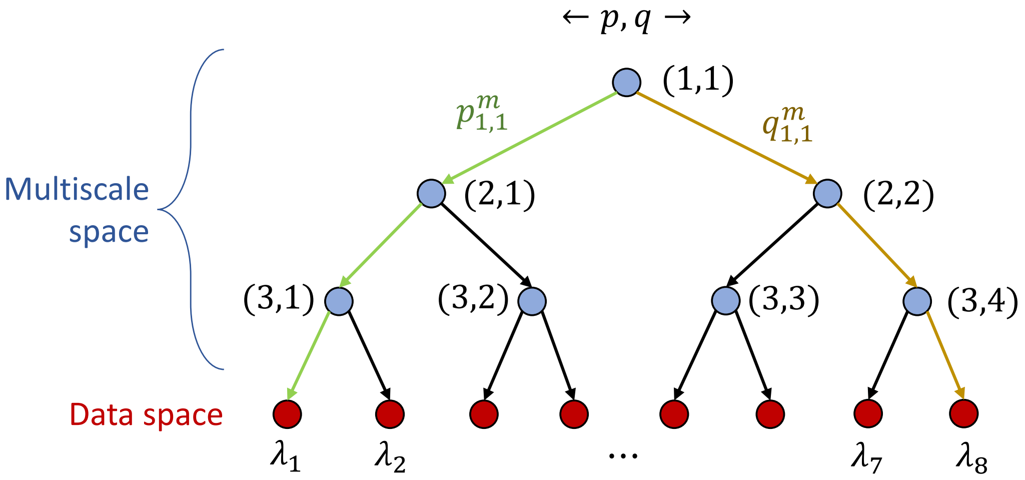


Figure 4.3: Multiscale representation of effect size calculation. Nodes are labelled by scale and location, and the direction down the tree which $p_{1,1}^m$ and $q_{1,1}^m$ moves us in are shown for illustration. The quantities required to calculate the leftmost and rightmost intensities are highlighted in green and gold respectively.

All other intensities can be found using the same principle, but require a product of a mix of p and q terms. Continuing with the example of the leftmost base, λ_b^m can be related to the effect size in the data space, $\beta_b^{(d)}$:

$$\beta_1^{(d)} = \log \lambda_1^1 - \log \lambda_1^0 \tag{4.30}$$

$$= \log\left(\exp\left(\alpha_{0,1}^1\right) \prod_{s=1}^{J} p_{s,1}^1\right) - \log\left(\exp\left(\alpha_{0,1}^0\right) \prod_{s=1}^{J} p_{s,1}^0\right) \tag{4.31}$$

$$= (\alpha_{0,1}^1 - \alpha_{0,1}^0) + \sum_{s=1}^{J} [\log p_{s,1}^1 - \log p_{s,1}^0] \tag{4.32}$$

where all the above quantities can be expressed in terms of $\alpha_{s,l}$, and therefore, the coefficients, $\mu_{s,l}^*$ and $\beta_{s,l}$, from the multiscale model. See Shim et al. (in preparation) for details.

For simplicity, we discuss the following in the case where there is only one set of parameters, $\beta_{s,l}$ with its corresponding $\gamma_{s,l}$ and only one HMT. Extending the following to include both trees can be handled easily due to their independence – see Appendix D for details. Using $\mathbf{D}$ to represent the data (in this case, MLE of parameters obtained in the likelihood approximation), to calculate the (pointwise) posterior mean for $\beta_1^{(d)}$:

$$\mathbb{E}[\beta_b^{(d)} \mid \mathbf{D}, \mathbf{g}] = \mathbb{E}[\beta_{0,1} \mid \mathbf{D}, \mathbf{g}] + \sum_{s=1}^{J} \mathbb{E}\Big[\log p_{s,1}^1 - \log p_{s,1}^0 \mid \mathbf{D}, \mathbf{g}\Big] \tag{4.33}$$

$$= \mathbb{E}[\mathbb{E}[\beta_{0,1} \mid \mathbf{D}, \mathbf{g}, \gamma_{0,1}]] + \sum_{s=1}^{J} \mathbb{E}\Big[\mathbb{E}\Big[\log p_{s,1}^1 - \log p_{s,1}^0 \mid \mathbf{D}, \mathbf{g}, \gamma_{s,1}\Big]\Big] \tag{4.34}$$

by additivity of expectations and law of iterated expectations. Approximating the posterior expectation, $\mathbb{E}\Big[\log p_{s,1}^1 - \log p_{s,1}^0 \mid \mathbf{D}, \mathbf{g}\Big]$, can be performed using a second-order Taylor expansion and the Delta method, and details can be found in the Appendix of Xing (2016). Note that the dependencies between $\gamma_{s,l}$ do not impact this calculation of means.

For the posterior variance at the leftmost base, as the coefficient at $(0, 1)$ is independent to all other quantities of the tree, we can split the variance into two sums:

$$V(\beta_1^{(d)} \mid \mathbf{D}, \mathbf{g}) = V(\beta_{0,1} \mid \mathbf{D}, \mathbf{g}) + V\Big(\sum_{s=1}^{J} \log p_{s,1}^1 - \log p_{s,1}^0 \mid \mathbf{D}, \mathbf{g}\Big) \tag{4.35}$$

For the term on the left:

$$V(\beta_{0,1} \mid \mathbf{D}, \mathbf{g}) = \mathbb{E}[\beta_{0,1}^2 \mid \mathbf{D}, \mathbf{g}] - \mathbb{E}[\beta_{0,1}^2 \mid \mathbf{D}, \mathbf{g}] \tag{4.36}$$

$$= \mathbb{E}[\mathbb{E}[\beta_{0,1}^2 \mid \mathbf{D}, \mathbf{g}, \gamma_{0,1}]] - \mathbb{E}[\mathbb{E}[\beta_{0,1}^2 \mid \mathbf{D}, \mathbf{g}, \gamma_{0,1}]] \tag{4.37}$$

$$= \sum_{m=0}^{M} \pi_m(\mu_m^2 + \sigma_m^2) - (\sum_{m=0}^{M} \pi_m \mu_m)^2 \tag{4.38}$$

where $\pi_m = P(\gamma_{0,1} = m \mid \mathbf{D}, \mathbf{g})$ are the mixing probabilities and $\mu_m = \mathbb{E}[\beta_{0,1} \mid \mathbf{D}, \mathbf{g}, \gamma_{0,1} = m], \sigma_m^2 = V(\beta_{0,1} \mid \gamma_{0,1} = m)$ are the posterior mean and variances for the coefficient at the zero-th scale. (4.38) is a standard result – see Appendix D.

For the term on the right, there are dependencies between the elements from different scales inside the sum, due to the HMT prior. The strategy here is to run a DP algorithm where we sum all the desired $\log p^1_{s,l} - \log p^0_{s,l}$ terms from the bottom of the tree to the top by conditioning on their states and considering their transition probabilities each time we move to a coarser scale.

To provide a high-level overview of the algorithm, consider the intensity at the leftmost base. For convenience, we re-index the sequence of nodes where $\log p^1_{s,1} - \log p^0_{s,1}$ are required, $\{(1,1),(2,1),\dots,(J,1)\}$, by just their scale, $(1,2,\dots,J)$, as well as dropping the $\mathbf{D}$ and $\mathbf{g}$ from the notation. Define:

$$\eta_s := \log p^1_s - \log p^0_s$$

and therefore the objective is to calculate:

$$V\Big(\sum_{s=1}^{J}\eta_s\Big) = \sum_{m=0}^{M}\Big[V\big(\sum_{s=1}^{J}\eta_s \mid \gamma_J = m\big)P(\gamma_J = m)\Big]$$

We start by initialising, for all states, the quantity at the bottom of the tree:

$$V(\eta_J \mid \gamma_J = m)$$

which can also be calculated using the method from the Appendix of Xing (2016). We then use the transition probabilities output from the Upward-Downward algorithm to move up the tree, and iteratively calculate, for each scale $s = J,\dots,2$:

$$V\Big(\eta_{j-1} + \sum_{s=j}^{J}\eta_s \mid \gamma_{j-1} = m, \gamma_j = n\Big) = V\Big(\eta_{j-1} \mid \gamma_{j-1} = m\Big) + V\Big(\sum_{s=j}^{J}\eta_s \mid \gamma_j = n\Big)$$

In order to run this algorithm, a similar algorithm must be applied to the expected values first, which provides an alternative method for approximating the posterior mean in the data space, $\beta^{(d)}_b$. We provide more detail on how both of the algorithms operate on a specific example in Appendix D, and informally mention how it would be generalised to calculate quantities across all bases.

Chapter 5

Conclusion

In Chapter 1, we began by discussing the importance of association analysis for molecular-level phenotypes, how high-throughput sequencing data is used to measure molecular-level behaviour and then outlining some obstacles in analysing high-throughput sequencing data. In particular, we discussed the need for methods which can account for the jagged, noisy nature of sequencing data at a high-resolution, and which are capable of adaptively detecting effects which may vary in length along the genome.

In Chapter 2, we reviewed a Wavelet-based multiscale model (Shim and Stephens (2015)) which aimed to address these obstacles, but also pointed out how one of its assumptions regarding independence was a limitation. We adopted the idea of HMTs as applied to WCs (Crouse et al. (1998)) as a way of accounting for dependencies between scales and locations in a tractable manner, and described how a variation of the EM algorithm applied to trees (Upward-Downward algorithm) could be used to choose the hyperparameters for this model in an EB setting. Furthermore, we described how we would evaluate the likelihood ratio test statistic which would help us test whether a particular genotype is associated with a molecular-level phenotype, as well as how to determine the posterior means and variances of effect sizes in the data space under the HMT framework.

In Chapter 3, we gave a brief overview of some of the practical considerations in implementing our method and described how we validated the method implementation through simulation. We then assessed the performance of *WaveQTL* and

WaveQTL-HMT, and illustrated that *WaveQTL-HMT* was better powered to identify effects which occur over a narrower interval.

Chapter 4 was dedicated to explaining an alternative multiscale model, better suited for data of a count nature (Shim et al. (in preparation), Xing (2016)), and showing how a HMT prior structure could be imposed on its effect size coefficients to address its independence assumptions. Once again, we ran through hyperparameter choice, likelihood test statistic calculation, and estimating posterior means and variances in the data space under the HMT framework.

In amongst all of this, the key to both of these models is that they are able to perform to deal with locally structured signals within a Bayesian framework in a computationally efficient manner. Imposing HMT priors on each of the models has not changed this – *WaveQTL-HMT* runs in a couple of seconds for a dataset with 70 individuals across 1024 bases, a similar runtime to *WaveQTL* – and we have shown that the key quantities for inference and estimation can still be evaluated under both models. Therefore, we conclude that the key contribution here is a model which accounts for dependencies in the multiscale space, whilst still remaining fit for purpose of being computationally tractable for genetic studies of high-throughput sequencing data involving hundreds of thousands of tests.

5.1 Future work

Due to time limitations, we did not get a chance to implement the multiscale Poisson model with HMT priors, or to evaluate its performance against *multiseq*, but is something we aim to do soon.

At present, we have implemented this algorithm to handle 1D-data – a sequence of counts – but the aim is to extend the implementation of both of these algorithms to be able to handle 2-D data. This will allow us to analyse signals and intensities in image data, as was the case in each of the papers where the respective HMT ideas were presented (Crouse et al. (1998), Timmerman and Nowak (1999)). It was suggested in both of these works that the natural binary tree structure in the 1-D case would be easily extensible to a 2-D case by using a ‘quadtree’ structure, where each parent node would be connected to four ‘child’ wavelets beneath it. We believe that *WaveQTL-HMT* would be effective in detecting signals in image data as the signals in image data are likely to occur in narrower regions in the 2D space. This would allow these models to become a viable proposition for analysing large datasets of both sequence and image data in a tractable manner.

Appendix A

WaveQTL

A.1 Bayes Factors

In this model, a quantity which is key to calculations involving the posterior is the BF. Although the BF is classically seen in Bayesian approaches to hypothesis testing, it is used in this model as a conduit to constructing more frequentest-style hypothesis tests regarding association, as well as for easily deriving posterior quantities regarding effect size, due to its availability in closed form.

As defined in Kass and Raftery (1995), for some observed data, $\mathbf{D}$, which may have arisen only from two hypotheses, H_0 and H_1, the BF to measure the support for the model under H_1 is:

$$\text{BF} = \frac{P(\mathbf{D} \mid H_1)}{P(\mathbf{D} \mid H_0)} \tag{A.1}$$

after integrating over the parameter space.

In our case, the support for $\gamma_{s,l} = 1$ at a specific s, l can be measured by the BF, after integrating out the remaining parameters $\mu_{s,l}, \beta_{s,l}$ and $\sigma^2_{s,l}$, leading to the expression in (2.9). The full derivation for the closed form expression of this can be found in Supplementary Material Protocol S1 of Servin and Stephens (2007).

A.2 Effect size in data space

For the conversion of the effect size from the wavelet space back to the original data space, it is helpful to express the system of equations in (2.1) in matrix form:

$$\mathbf{Y} = \mathbf{M} + \boldsymbol{\beta}\mathbf{g} + \mathbf{E}$$

where the $\mathbf{Y}, \mathbf{M}, \mathbf{E}$ are all $B \times N$ matrices, and $\boldsymbol{\beta}\mathbf{g}$ is a $B \times N$ matrix. To represent the linear transform from data to wavelet space in matrix form, we can write $y = Wd$ for WCs y, and data d, with W being a $B \times B$ matrix representing the DWT. As previously mentioned, the one-one nature of the transformation means that we can express the IDWT as $d = W^{-1}y$. Applying this to our system of equations:

$$\begin{aligned} \mathbf{D} &= \mathbf{W}^{-1}\mathbf{Y} \\ \Rightarrow \mathbf{D} &= \mathbf{W}^{-1}\mathbf{M} + \mathbf{W}^{-1}\boldsymbol{\beta}\mathbf{g} + \mathbf{W}^{-1}\mathbf{E} \end{aligned}$$

where $\mathbf{D}$ is a $B \times N$ matrix representing the original data. Furthermore, as $\mathbf{W}$ is orthogonal, $\mathbf{W}^{-1} = \mathbf{W}^T$, and the effect sizes in the data space can be written as $\boldsymbol{\alpha} := \mathbf{W}^T\boldsymbol{\beta}$.

A.2.1 Effect size simulation methodology

Our approach uses the output parameters of the EM-algorithm to draw samples from the posterior of β. We start by simulating instances of the $\boldsymbol{\gamma}$ vector in a recursive manner, starting from the root node, working down to the leaves, conditional on the states assigned to each parent.

Starting at the root node, we have the quantity $A_{1,1}(m) := P(\gamma_{1,1} = 1 \mid \mathbf{y}, \widehat{\boldsymbol{\theta}})$ from our EM algorithm, therefore we assign a state to $\gamma_{1,1}$ as follows (w.p. stands for 'with probability'):

$$\gamma_{1,1} = \begin{cases} 1 & \text{w.p. } A_{1,1}(m) \\ 0 & \text{w.p. } 1 - A_{1,1}(m) \end{cases} \tag{A.2}$$

Conditioning on this, we can assign states to each of the two children. Denoting $B_{(s,l),p(s,l)}(m,n) := P(\gamma_{s,l} = m, \gamma_{p(s,l)} = n \mid \mathbf{y}, \widehat{\boldsymbol{\theta}})$, and noting that $P(\gamma_{s,l} = m \mid$

$\gamma_{p(s,l)} = n, \mathbf{y}, \widehat{\boldsymbol{\theta}}) = \frac{B_{(s,l),p(s,l)}(m,n)}{A_{p(s,l)}(n)}$, we get, for all remaining s, l:

$$\gamma_{s,l} = \begin{cases} 1 & \text{w.p. } P(\gamma_{s,l} = 1 \mid \gamma_{p(s,l)} = n, \mathbf{y}, \widehat{\boldsymbol{\theta}}) \\ 0 & \text{w.p. } 1 - P(\gamma_{s,l} = 1 \mid \gamma_{p(s,l)} = n, \mathbf{y}, \widehat{\boldsymbol{\theta}}) \end{cases} \tag{A.3}$$

knowing that n was the parent state which we simulated previously. We treat the state and value of the scaling coefficient, $\gamma_{0,0}$, the same as per the no-HMT treatment, as its state is not modelled as dependent to any other state in the tree, and therefore is not affected by the tree structure.

Now that each $\gamma_{s,l}$ has been assigned a state (0 or 1), we simulate wavelet-space effect sizes from the respective posterior β distribution, conditional on the assigned state of that node. That is:

$$\beta_{s,l} = \begin{cases} t(\nu_{s,l}, a_{s,l}, b_{s,l}) & \text{if } \gamma_{s,l} = 1 \\ 0 & \text{if } \gamma_{s,l} = 0 \end{cases} \tag{A.4}$$

where $t(\nu_{s,l}, a_{s,l}, b_{s,l})$ is a random draw from a 3-parameter t-distribution with scale-location specific parameters $\nu_{s,l}, a_{s,l}, b_{s,l}$, and is output from our program during the process of calculating BFs.

The result is a simulated state vector, $\boldsymbol{\gamma}$, and corresponding draw from the posterior distribution of the effect size, $(\boldsymbol{\beta} \mid \mathbf{y}, \mathbf{g}, \widehat{\boldsymbol{\theta}})$, which we then transform into the data space using the IDWT. We can then take sample means, standard deviations, or quantiles of the simulated data-space effect sizes at each scale-location, yielding simulated estimates of these quantities.

Appendix B

WaveQTL with HMT

B.1 Upward-Downward algorithm E-step details

Here are extra details, beyond that given in the Appendix to Crouse et al. (1998), regarding derivations behind the E-step quantities in the EM algorithm for HMTs. Note that for convenience, in this section, we drop the (s, l) notation in favour of the single-index $i = 1, \ldots, n$ notation to denote the n nodes of the tree. $i = 1$ represents the root of the tree, and the indices increase from top to bottom, left to right, as per Figure 2.5. In addition, this section uses the original implementation of the algorithm, and therefore all derivations are only proportional to the quantities we obtain using our algorithm, due to the absence of the dividing factors. We denote outputs from this algorithm with the superscript * to contrast from those obtained in our algorithm.

B.1.1 Upward-downward algorithm – further derivations

Notation, as per Crouse et al. (1998), is as follows:

- $\mathcal{T}_i$ represents the vector of WCs in the subtree rooted at node i of the tree
- $\mathcal{T}_1 = (y_1, \ldots, y_n)$, i.e. all the WCs in the tree
- $\mathcal{T}_{i \backslash j}$ represents the vector of WCs obtained by removing subtree $\mathcal{T}_j$ from $\mathcal{T}_i$

The quantities we evaluate in the algorithm are:

$$\beta_i^*(k) := P(\mathcal{T}_i \mid \gamma_i = k, \boldsymbol{\theta}) \tag{B.1}$$

$$\beta_{i,p(i)}^*(k) := P(\mathcal{T}_i \mid \gamma_{p(i)} = k, \boldsymbol{\theta}) \tag{B.2}$$

$$\beta_{p(i)\backslash i}^*(k) := P(\mathcal{T}_{p(i)\backslash i} \mid \gamma_{p(i)} = k, \boldsymbol{\theta}) \tag{B.3}$$

$$\alpha_i^*(k) := P(\gamma_i = k, \mathcal{T}_{1\backslash i} \mid \boldsymbol{\theta}) \tag{B.4}$$

where (B.1) – (B.3) are joint conditional likelihoods, and (B.4) represents a joint probability of WCs and a particular node's state. (B.1) – (B.3) are derived as follows:

$$\begin{aligned}
\beta_{i,p(i)}^*(k) &= P(\mathcal{T}_i \mid \gamma_{p(i)} = k, \boldsymbol{\theta}) \\
&= \sum_{l=1}^{m} P(\mathcal{T}_i, \gamma_i = l \mid \gamma_{p(i)} = k, \boldsymbol{\theta}) && \text{(law of total probability)} \\
&= \sum_{l=1}^{m} P(\mathcal{T}_i \mid \gamma_i = l, \gamma_{p(i)} = k, \boldsymbol{\theta}) P(\gamma_i = l \mid \gamma_{p(i)} = k, \boldsymbol{\theta}) \\
&= \sum_{l=1}^{m} P(\mathcal{T}_i \mid \gamma_i = l, \boldsymbol{\theta}) P(\gamma_i = l \mid \gamma_{p(i)} = k, \boldsymbol{\theta})
\end{aligned}$$

as the subtree rooted at i is conditionally independent of its parent's state given its own state. This equals:

$$= \sum_{l=1}^{m} \beta_i^*(l) \varepsilon_{i,p(i)}^{lk} \qquad \text{(equation (21) in Crouse et al. (1998))}$$

For (B.2):

$$\begin{aligned}
\beta_i^*(k) &= P(\mathcal{T}_i \mid \gamma_i = k, \boldsymbol{\theta}) \\
&= P(y_i, \mathcal{T}_{c(i)_1}, \mathcal{T}_{c(i)_2} \mid \gamma_i = k, \boldsymbol{\theta}) \\
&= \Big[\prod_{j \in c(i)} P(\mathcal{T}_j \mid \gamma_i = k, \boldsymbol{\theta}) \Big] P(y_i \mid \gamma_i = k, \boldsymbol{\theta}) \\
& \qquad \text{(equation (22) in Crouse et al. (1998))}
\end{aligned}$$

where $c(i)_j$ represents the jth child of i. The last step is due to the conditional independence between the trees rooted at $c(i)_j$, and y_i due to conditioning on γ_i. For (B.3):

$$\begin{aligned}\beta^*_{p(i)\backslash i}(k) &= P(\mathcal{T}_{p(i)\backslash i} \mid \gamma_{p(i)} = k, \boldsymbol{\theta}) \\ &= \frac{\beta^*_{p(i)}(k)}{P(\mathcal{T}_i \mid S_{p(i)} = k, \boldsymbol{\theta})} \\ &= \frac{\beta^*_{p(i)}(k)}{\beta^*_{i,p(i)}(k)} \qquad \text{(equation (23) in Crouse et al. (1998))}\end{aligned}$$

For (B.4):

$$\begin{aligned}\alpha^*_i(k) &= P(\gamma_i = k, \mathcal{T}_{1\backslash i} \mid \boldsymbol{\theta}) \\ &= \sum_{l=1}^{m} P(\gamma_i = k, \gamma_{p(i)} = l, \mathcal{T}_{1\backslash i} \mid \boldsymbol{\theta}) \\ &= \sum_{l=1}^{m} P(\gamma_i = k, \gamma_{p(i)} = l, \mathcal{T}_{1\backslash p(i)}, \mathcal{T}_{p(i)\backslash i} \mid \boldsymbol{\theta}) \\ &= \sum_{l=1}^{m} P(\mathcal{T}_{p(i)\backslash i}, \gamma_i = k \mid \mathcal{T}_{1\backslash p(i)}, \gamma_{p(i)} = l, \boldsymbol{\theta}) P(\mathcal{T}_{1\backslash p(i)}, \gamma_{p(i)} = l \mid \boldsymbol{\theta}) \\ &= \sum_{l=1}^{m} P(\mathcal{T}_{p(i)\backslash i}, \gamma_i = k \mid \gamma_{p(i)} = l, \boldsymbol{\theta}) \alpha^*_{p(i)}(l)\end{aligned}$$

due, again, to conditional independence on the two sets of WCs due to conditioning on parent state. Then:

$$\begin{aligned}&= \sum_{l=1}^{m} P(\mathcal{T}_{p(i)\backslash i} \mid \gamma_i = k, \gamma_{p(i)} = l, \boldsymbol{\theta}) P(\gamma_i = k \mid \gamma_{p(i)} = l, \boldsymbol{\theta}) \alpha_{p(i)}(l) \\ &= \sum_{l=1}^{m} P(\mathcal{T}_{p(i)\backslash i} \mid \gamma_{p(i)} = l, \boldsymbol{\theta}) \varepsilon^{kl}_{i,p(i)} \alpha^*_{p(i)}(l) \\ &= \sum_{l=1}^{m} \beta^*_{p(i)\backslash i}(l) \varepsilon^{kl}_{i,p(i)} \alpha^*_{p(i)}(l) \qquad \text{(equation (25) in Crouse et al. (1998))}\end{aligned}$$

where the second last step is due to conditional independence given $\gamma_{p(i)}$. The two quantities required for evaluating the E-step quantities are the the joint distributions of all WCs with each state. These are calculated as follows:

$$\begin{aligned}
P(\gamma_i = k, \mathcal{T}_1 \mid \boldsymbol{\theta}) &= P(\gamma_i = k, \mathcal{T}_{1\backslash i}, \mathcal{T}_i \mid \boldsymbol{\theta}) \\
&= P(\mathcal{T}_i \mid \gamma_i = k, \mathcal{T}_{1\backslash i}, \boldsymbol{\theta}) P(\gamma_i = k, \mathcal{T}_{1\backslash i} \mid \boldsymbol{\theta}) \\
&= P(\mathcal{T}_i \mid \gamma_i = k, \boldsymbol{\theta}) P(\gamma_i = k, \mathcal{T}_{1\backslash i} \mid \boldsymbol{\theta}) \\
&= \beta_i^*(k) \alpha_i^*(k) \\
\therefore P(\mathbf{y} \mid \boldsymbol{\theta}) &= P(\mathcal{T}_1 \mid \boldsymbol{\theta}) \\
&= \sum_{k=1}^{m} P(\gamma_i = k, \mathcal{T}_1 \mid \boldsymbol{\theta}) \\
&= \sum_{k=1}^{m} \beta_i^*(k) \alpha_i^*(k)
\end{aligned}$$

and for the joint child-parent state and WC probability:

$$\begin{aligned}
P(\gamma_i = k, \gamma_{p(i)} = l, \mathcal{T}_1 \mid \boldsymbol{\theta}) &= P(\gamma_i = k, \gamma_{p(i)} = l, \mathcal{T}_{1\backslash p(i)}, \mathcal{T}_{p(i)\backslash i}, \mathcal{T}_i \mid \boldsymbol{\theta}) \\
&= P(\gamma_{p(i)} = l, \mathcal{T}_{1\backslash p(i)} \mid \boldsymbol{\theta}) P(\gamma_i = k, \mathcal{T}_{p(i)\backslash i}, \mathcal{T}_i \mid \gamma_{p(i)} = l, \mathcal{T}_{1\backslash p(i)}, \boldsymbol{\theta}) \\
&= \alpha_{p(i)}^*(l) P(\gamma_i = k, \mathcal{T}_{p(i)\backslash i}, \mathcal{T}_i \mid \gamma_{p(i)} = l, \boldsymbol{\theta}) \\
&= \alpha_{p(i)}^*(l) P(\mathcal{T}_{p(i)\backslash i}, \mathcal{T}_i \mid \gamma_i = k, \gamma_{p(i)} = l, \boldsymbol{\theta}) P(\gamma_i = k \mid \gamma_{p(i)} = l, \boldsymbol{\theta}) \\
&= \alpha_{p(i)}^*(l) P(\mathcal{T}_{p(i)\backslash i}, \mathcal{T}_i \mid \gamma_i = k, \gamma_{p(i)} = l, \boldsymbol{\theta}) \varepsilon_{i,p(i)}^{kl} \\
&= \alpha_{p(i)}^*(l) P(\mathcal{T}_{p(i)\backslash i} \mid \gamma_i = k, \gamma_{p(i)} = l, \boldsymbol{\theta}) P(\mathcal{T}_i \mid \gamma_i = k, \gamma_{p(i)} = l, \boldsymbol{\theta}) \varepsilon_{i,p(i)}^{kl} \\
&= \alpha_{p(i)}^*(l) P(\mathcal{T}_{p(i)\backslash i} \mid \gamma_{p(i)} = l, \boldsymbol{\theta}) P(\mathcal{T}_i \mid \gamma_i = k, \boldsymbol{\theta}) \varepsilon_{i,p(i)}^{kl} \\
&= \alpha_{p(i)}^*(l) \beta_{p(i)\backslash i}^*(l) \beta_i^*(k) \varepsilon_{i,p(i)}^{kl}
\end{aligned}$$

with most of the conditioning simplifications coming due to conditional independence on the node's state.

The M-step quantities can be derived in closed form by using a maximisation procedure such as constrained optimisation with the Lagrangian. Note also that the set of π's can be optimised independently of the set of ε's, as these hyperparameters lie in separate terms of the expression $Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)})$.

B.2 Alignment of amended algorithm with Crouse et al. (1998)

To show that our algorithm's outputs align with those in Crouse et al. (1998), we focus on illustrating how the quantities (2.14) – (2.17) deviate from the original by stepping through the first iteration. We then show that, when calculating (2.12) and (2.13), we get the required cancellation such that these quantities align with those from Crouse et al. (1998).

For all nodes in the lowest scale, where $s = J$, we have:

$$\beta_i(m) = \mathrm{BF}_i(y, g)$$

Moving up the tree to the next highest scale, $s = J - 1$, we have:

$$\begin{aligned}
\beta_{i,p(i)}(m) &= \sum_{n=0}^{1} \varepsilon_{i,p(i)}^{nm} \beta_i(n) \\
&= \sum_{n=0}^{1} P(\gamma_i = n \mid \gamma_{p(i)} = m, \mathbf{g}, \boldsymbol{\theta}) \frac{P(y_i \mid \gamma_i = n, \mathbf{g})}{P(y_i \mid \gamma_i = 0, \mathbf{g})} \\
&= \frac{\beta^*_{i,p(i)}(m)}{P(y_i \mid \gamma_i = 0, \mathbf{g})} \\
\beta_{p(i)}(m) &= \mathrm{BF}_{p(i)}(y, g) \times \prod_{j \in c(p(i))} \beta_{j,p(j)}(m) \\
&= \frac{P(y_{p(i)} \mid \gamma_{p(i)} = m, \mathbf{g})}{P(y_{p(i)} \mid \gamma_{p(i)} = 0, \mathbf{g})} \times \prod_{j \in c(p(i))} \frac{\beta^*_{j,p(j)}(m)}{P(y_j \mid \gamma_j = 0, \mathbf{g})} \\
&= \frac{P(y_{p(i)} \mid \gamma_{p(i)} = m, \mathbf{g})}{P(y_{p(i)} \mid \gamma_{p(i)} = 0, \mathbf{g})} \times \prod_{j \in c(p(i))} \frac{P(\mathcal{T}_j \mid \gamma_{p(i)} = m, \mathbf{g})}{P(y_j \mid \gamma_j = 0, \mathbf{g})} && (*) \\
&= \frac{P(\mathcal{T}_{p(i)} \mid \gamma_{p(i)} = m, \mathbf{g})}{\prod_{j \in \nu_{p(i)}} P(y_j \mid \gamma_j = 0, \mathbf{g})}
\end{aligned}$$

where we have used the fact that for $j \in c(p(i)), p(j) = p(i)$ in (*). Note that the denominator is the product of each of $\nu_{p(i)}$'s individual WC densities, conditioned on each individual node's $\gamma = 0$, as opposed to a vector of WCs, conditioned on one node's state.

When we evaluate the $\beta_{i,p(i)}(m)$ quantity as we move up the tree, we note that each $\beta_i(n)$ expression brings, in its denominator, the product of all WC probabilities, conditioned on $\gamma_j = 0$, for all nodes from the subtree rooted at node i. Moving up the tree to $S = J - 2$, We evaluate the $\beta_{i,p(i)}(m)$ term at this scale to illustrate, noting that the node $p(i)$ at the old scale is now node i:

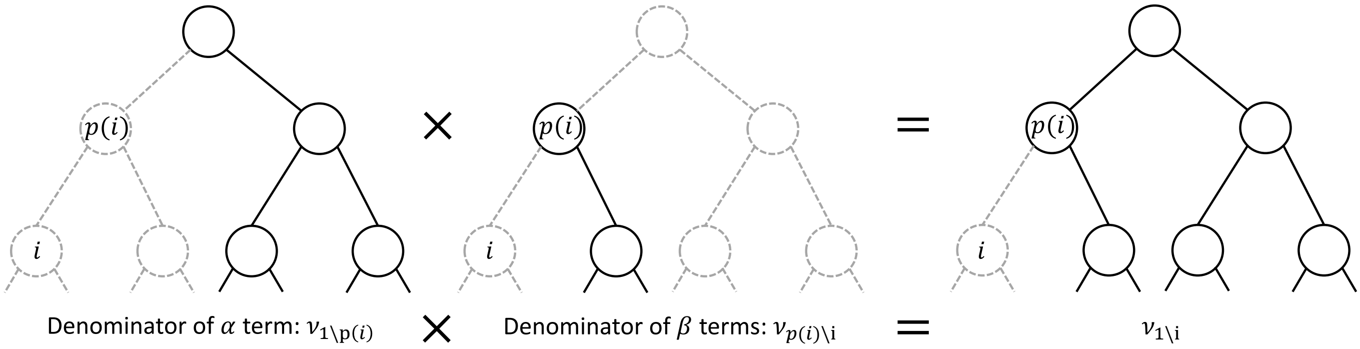
$$\begin{aligned}\beta_{i,p(i)}(m) &= \sum_{n=0}^{1} \varepsilon_{i,p(i)}^{nm} \beta_i(n) \\ &= \sum_{n=0}^{1} P(\gamma_i = n \mid \gamma_{p(i)} = m, \mathbf{g}, \boldsymbol{\theta}) \frac{P(\mathcal{T}_i \mid \gamma_i = n, \mathbf{g})}{\prod_{j \in \nu_i} P(y_j \mid \gamma_j = 0, \mathbf{g})} \\ &= \frac{\beta^*_{i,p(i)}(m)}{\prod_{j \in \nu_i} P(y_j \mid \gamma_j = 0, \mathbf{g})}\end{aligned}$$

This gets carried over to our expression for the $\beta_{p(i)}(m)$, where we recursively multiply on the parent's WC probability, conditional on $\gamma_{p(i)} = m$ on the numerator, and conditional on $\gamma_{p(i)} = 0$ in the denominator. We evaluate $\beta_i(m)$ at this scale to confirm this:

$$\begin{aligned}\beta_{p(i)}(m) &= \mathrm{BF}_{p(i)}(y, g) \times \prod_{j \in c(p(i))} \beta_{j,p(j)}(m) \\ &= \frac{P(y_{p(i)} \mid \gamma_{p(i)} = m, \mathbf{g})}{P(y_{p(i)} \mid \gamma_{p(i)} = 0, \mathbf{g})} \times \prod_{j \in c(p(i))} \frac{\beta^*_{j,p(j)}(m)}{\prod_{k \in \nu_j} P(y_k \mid \gamma_k = 0, \mathbf{g})} \\ &= \frac{\beta^*_{p(i)}(m)}{P(y_{p(i)} \mid \gamma_{p(i)} = 0, \mathbf{g}) \prod_{j \in c(p(i))} \prod_{k \in \nu_j} P(y_k \mid \gamma_k = 0, \mathbf{g})} \\ &= \frac{\beta^*_{p(i)}(m)}{\prod_{j \in \nu_{p(i)}} P(y_j \mid \gamma_j = 0, \mathbf{g})}\end{aligned}$$

where the denominator in the last step follows as, for each of i's children, $j \in c(p(i))$, all the nodes in each subtree rooted at each j, ν_j, are mutually exclusive. Hence the product of all the terms in the denominator results in the product of densities of the individual WCs contained in the subtree rooted at $p(i)$ recursively, until $s = 1$.

(2.16) follows due to cancellation between the overlapping products of the individual densities at the subtrees rooted at $p(i)$ and i. (2.17) follows as the separate products in the denominators from the augmented α and β terms are mutually exclusive, and constitute the whole tree, minus the subtree rooted at i. In other words, the denominator of the α term has WCs from $\nu_{1\backslash p(i)}$, and, β's denominators are the WCs from the $\nu_{p(i)\backslash i}$ terms. Hence, multiplied together, we get the probabilities from the $\nu_{1\backslash i}$ WCs.


p(i)
i
×
p(i)
i
=
p(i)
i
Denominator of α term: ν1\p(i)
×
Denominator of β terms: νp(i)\i
=
ν1\i

From this, we can derive our two desired quantities. We see that, even though we have augmented the intermediate quantities slightly to use BFs, the cancelling means the results are equivalent to those desired from Crouse et al. (1998). For the quantity in (2.12):

$$\begin{aligned}
P(\gamma_i = m \mid \mathbf{y}, \boldsymbol{\theta}) &= \frac{\alpha_i^*(m)\beta_i^*(m)}{\sum_{n=0}^{1} \alpha_i^*(n)\beta_i^*(n)} \\
&= \frac{\alpha_i(m)\beta_i(m)}{\prod_{j\in\nu_1} P(y_j \mid \gamma_j = 0, \mathbf{g})} \frac{\prod_{j\in\nu_1} P(y_j \mid \gamma_j = 0, \mathbf{g})}{\sum_{n=0}^{1} \alpha_i(n)\beta_i(n)} \\
&= \frac{\alpha_i(m)\beta_i(m)}{\sum_{n=0}^{1} \alpha_i(n)\beta_i(n)}
\end{aligned}$$

For (2.13):

$$\begin{aligned}
P(\gamma_i = m, \gamma_{p(i)} = n \mid \mathbf{y}, \boldsymbol{\theta}) &= \frac{\beta_i^*(m)\varepsilon_{i,p(i)}^{mn}\alpha_{p(i)}^*(n)\beta_{p(i)\backslash i}^*(n)}{\sum_{n=0}^{1} \alpha_i^*(n)\beta_i^*(n)} \\
&= \frac{\beta_i(m)\varepsilon_{i,p(i)}^{mn}\alpha_{p(i)}(n)\beta_{p(i)\backslash i}(n)}{\prod_{j\in\nu_1} P(y_j \mid \gamma_j = 0, \mathbf{g})} \frac{\prod_{j\in\nu_1} P(y_j \mid \gamma_j = 0, \mathbf{g})}{\sum_{n=0}^{1} \alpha_i(n)\beta_i(n)} \\
&= \frac{\beta_i(m)\varepsilon_{i,p(i)}^{mn}\alpha_{p(i)}(n)\beta_{p(i)\backslash i}(n)}{\sum_{n=0}^{1} \alpha_i(n)\beta_i(n)}
\end{aligned}$$

Appendix C

Wavelet-based model: analysis

The *WaveQTL* repository can be found at: `https://github.com/heejungshim/WaveQTL`. The WIP repository which includes the *WaveQTL-HMT* code, as well as all analysis performed in this thesis can be found here: `https://github.com/blaw36/Masters_Project`. Featured analyses and plots can be found an interactive webpage format here: `https://blaw36.github.io/Masters_Project/`.

C.1 Implementation validation – simulation procedure

Our aim is to simulate an $N \times B$ matrix of WCs, generated from preset hyperparameters which our algorithm will attempt to retrieve.

1. Initialise a specific $\boldsymbol{\theta}_{sim} = (\pi_{1,1}, \varepsilon^{11}, \varepsilon^{10})$ to simulate data from.

2. Generate a sequence of $\gamma_{s,l}$ from $\boldsymbol{\theta}_{sim}$.

3. Set $\mu_{s,l} = 0$ for the purposes of the simulation. Set $\beta_{s,l} = x$ where x is a preset constant.

4. Represent each individual's genotype by drawing $g^i \overset{d}{\sim} Binomial(2, p)$ where $p \in [0, 1]$ is a preset probability of success.

5. Draw the WC noise parameter, $\varepsilon^i_{s,l} \overset{d}{\sim} N(0, \sigma^2)$. In our simulation, we set σ^2 as a fixed constant, allowing us to vary the level of noise in our simulation.

6. Construct a matrix of simulated WCs; $y^i_{s,l} = \beta_{s,l} g^i + \varepsilon^i_{s,l}$.

7. Write the simulated g^i sequence to file, and run *WaveQTL-HMT* on the simulated WC and genotype data to output estimates of $\boldsymbol{\theta}_{sim}$.

8. Repeat steps 2-7 until the number of desired simulations has been run.

We ran simulations with $\beta_{s,l} = 2, \sigma^2 = 1$ and $\pi_{1,1} = 1$ to ensure there was a clear signal in one half of the dataset for simplicity. This was the only hyperparameter we did not experiment with, but we still evaluated whether our algorithm could extract this value successfully.

C.2 Implementation validation – further results

Here are some further results from the simulations performed in Section 3.3. We ran the following combinations of hyperparameters, in addition to the (0.75,0.25) run in Section 3.3:

Combination	ε^{11}	ε^{10}
1	0.9	0.1
2	0.5	0.5
3	0.9	0.9
4	0.75	0.75

so that we can test how our algorithm performs when these two hyperparameters are similar or different to each other. Results are as follows:

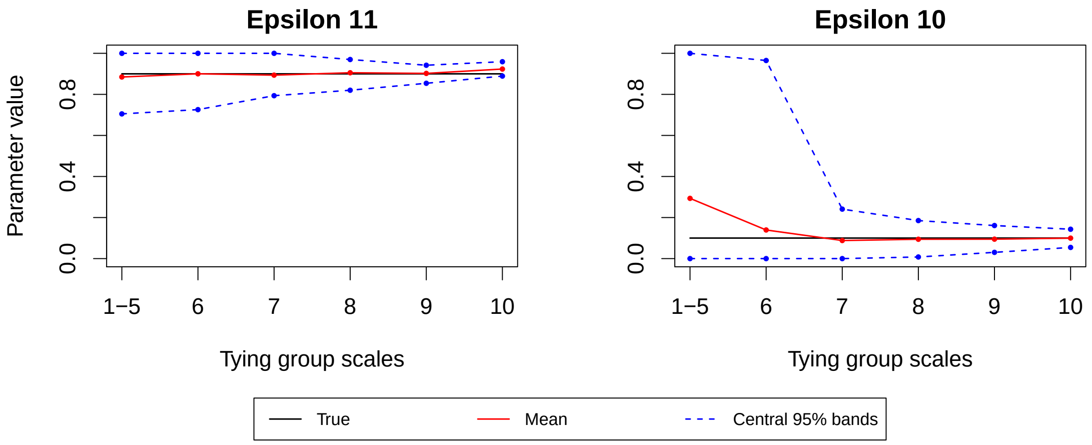


Figure C.1: Simulation results for combination 1; (0.9,0.1)

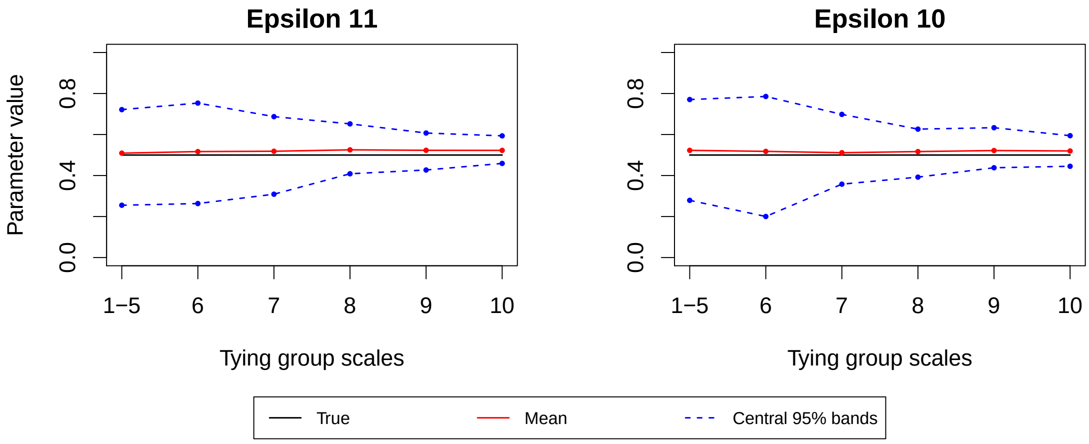


Figure C.2: Simulation results for combination 2; (0.5,0.5)

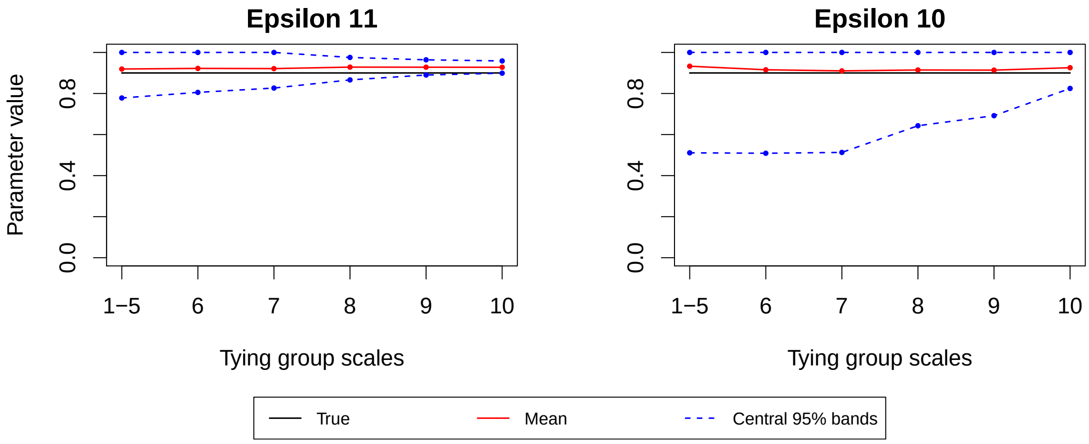


Figure C.3: Simulation results for combination 3; (0.9,0.9)

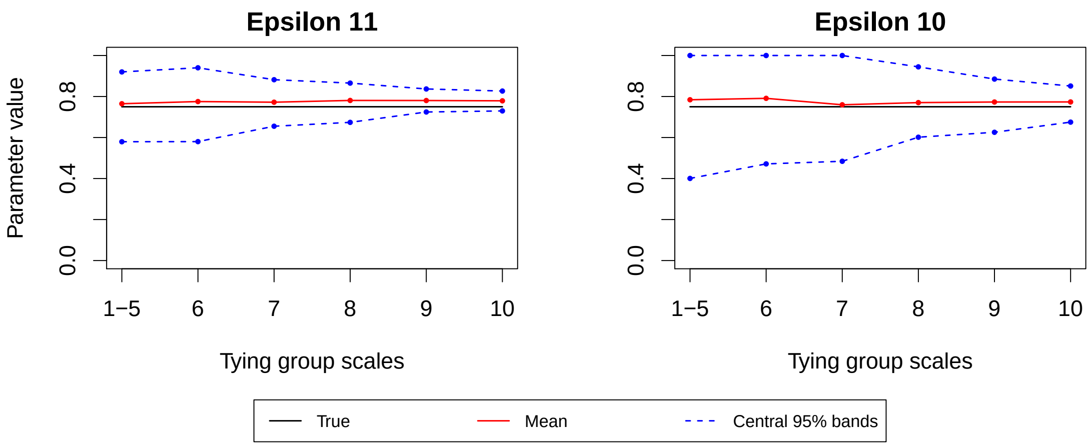


Figure C.4: Simulation results for combination 4; (0.75,0.75)

The 95% bands included the true hyperparameters for ε in all the combinations, and the same applied for $\pi_{1,1}$, whose results are not displayed here.

C.3 Simulated data - simulation procedure

Our objective is to simulate the counts of N individuals across B base locations, where the number of counts at each base correspond to the actual counts in the DNase-seq dataset. Based on our real dataset, we know $(\overline{d}_1, \ldots, \overline{d}_B)$, where $\overline{d}_b = \frac{1}{N}\sum_{i=1}^{N} d_{b,i}$, a vector of the average counts for each individual at each base. To align the expected number of simulated counts per individual with the empirical sample average counts per individual observed in the real dataset, we could draw from a Binomial distribution as follows:

$$d_{b,i}^{(sim)} \overset{d}{\sim} \text{Binomial}(\widetilde{d}_b, 1/N) \tag{C.1}$$

where:

$$\widetilde{d}_b = \sum_{i=1}^{N} d_{b,i} \tag{C.2}$$

ensuring that, in expectation, $\mathbb{E}[d_{b,i}^{(sim)}] = \frac{1}{N}\widetilde{d}_b = \overline{d}_b$ as required.

To show the impact of an effect size, we sample counts from two groups of individuals, g_1 and g_2, each with its own corresponding proportion, p_1 and p_2. To incorporate the additive effects output from our model into p_1 and p_2, we use a ratio. Denoting e_b as the effect size to be used in the proportions, and with $(\beta_1, \ldots, \beta_B)$ being the estimated effect size at each base from the DNase-seq data, we have:

$$\frac{1}{N}\widetilde{d}_b + \beta_b = \frac{1}{N}\widetilde{d}_b e_b \tag{C.3}$$

$$\therefore e_b = \begin{cases} 1 + \frac{N\beta_b}{\widetilde{d}_b}, & \widetilde{d}_b \neq 0 \\ 1, & \widetilde{d}_b = 0 \end{cases} \tag{C.4}$$

where $e_b = 1$ indicates no difference between the groups at base b.

Now that we deal with two groups, we want to draw from two distributions with proportions such that $p_1 + p_2 = \frac{2}{N}$, but where p_1 and p_2 vary depending on the effect size at a given base, hence the ‘thinning’. A simple way to incorporate the effect size from (C.4) whilst holding the overall sum of proportions constant is:

$$p_{1,b} = \frac{2}{N}\frac{1}{1 + e_b} \tag{C.5}$$

$$p_{2,b} = \frac{2}{N}\frac{e_b}{1 + e_b} \tag{C.6}$$

which includes the case where there is no effect size as a special case ($e_b = 1 \Rightarrow p_{1,b} = p_{2,b} = \frac{1}{N}$).

In practice, the effect size, β_b, is rarely in the same scale as the counts for a variety of reasons such as standardisation and quantile transforms. Additionally, when the sequencing count is too small at a particular base, we have high sampling variability. We include two scaling parameters, c_β and c_d, to control these quantities. In addition, to capture over-dispersion, we sample from a Beta-binomial distribution.

In summary, we have three parameters we can adjust: c_d, c_β, ξ:

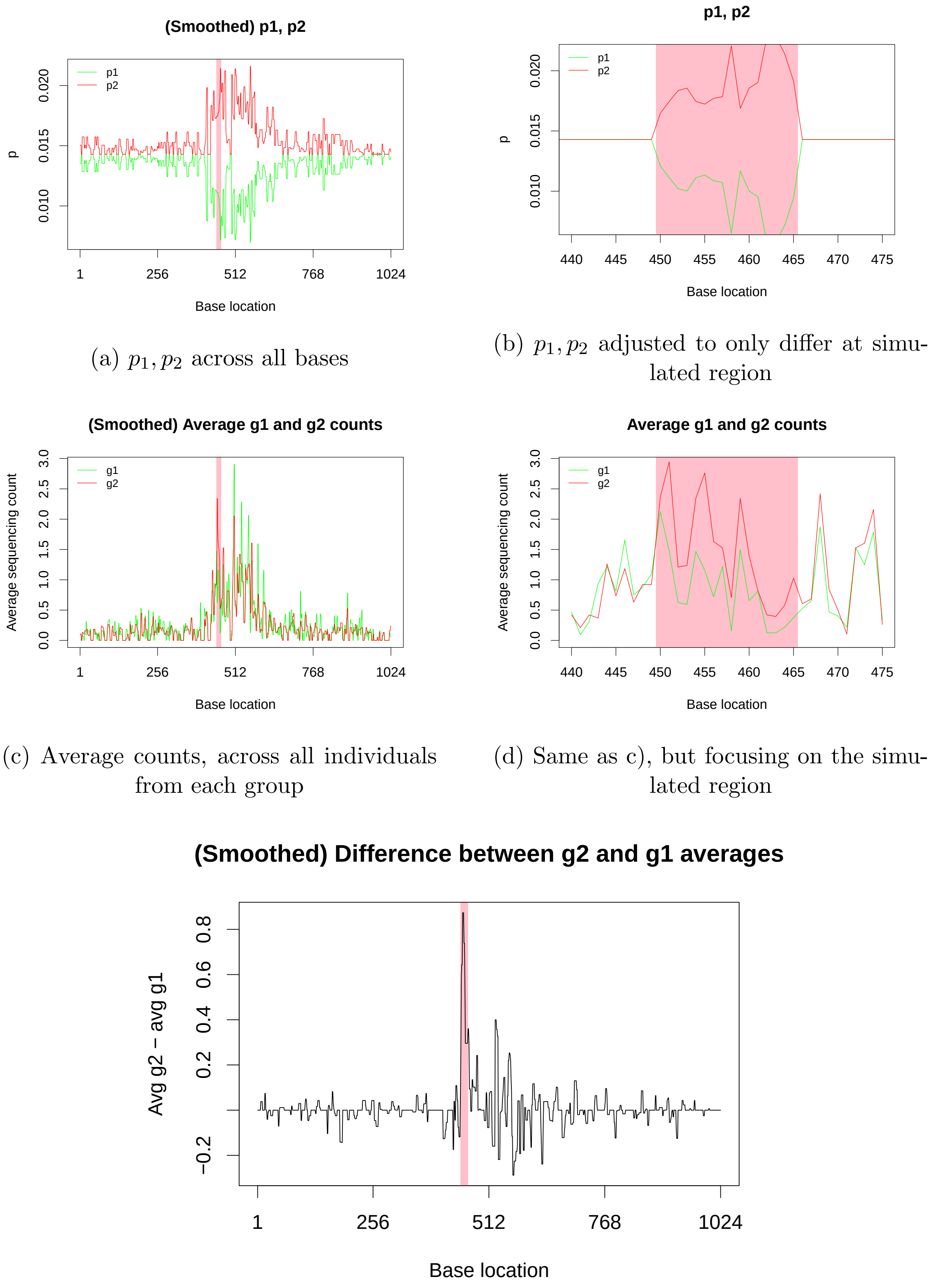
$$e_b = \begin{cases} 1 + Nc_\beta \beta_b / \widetilde{d}_b, & \widetilde{d}_b \neq 0 \\ 1, & \widetilde{d}_b = 0 \end{cases} \tag{C.7}$$

$$d_{b,i_1}^{(sim)} \overset{d}{\sim} \text{Beta-Binomial}(c_d \widetilde{d}_b, p_{1,b}, \xi) \tag{C.8}$$

$$d_{b,i_2}^{(sim)} \overset{d}{\sim} \text{Beta-Binomial}(c_d \widetilde{d}_b, p_{2,b}, \xi) \tag{C.9}$$

where i_j represents an individual in group j. The Beta-Binomial representation used above is Beta-Binomial(n, m, s), where n is the number of trials, m is the probability of success, and s is an over-dispersion parameter where, as s increases, variance decreases (and becomes equivalent to Binomial variance as $s \to \infty$). To illustrate how this process works, here are some plots which show examples of how the effect sizes translate into proportions, and the sequences which result from those proportions.

Regions in pink denote the locations where effects are generated. These plots were generated from an effect length 16 with the following settings: $c_\beta = 8 \times 10^7, c_d = 10, \xi = 70$ with quantile-transformed WCs, and the same tree tying convention as per the simulations in Section 3.3.2. Some plots have been smoothed (using the `smooth` command in R) to improve readability.



(a) p_1, p_2 across all bases

(b) p_1, p_2 adjusted to only differ at simulated region

(c) Average counts, across all individuals from each group

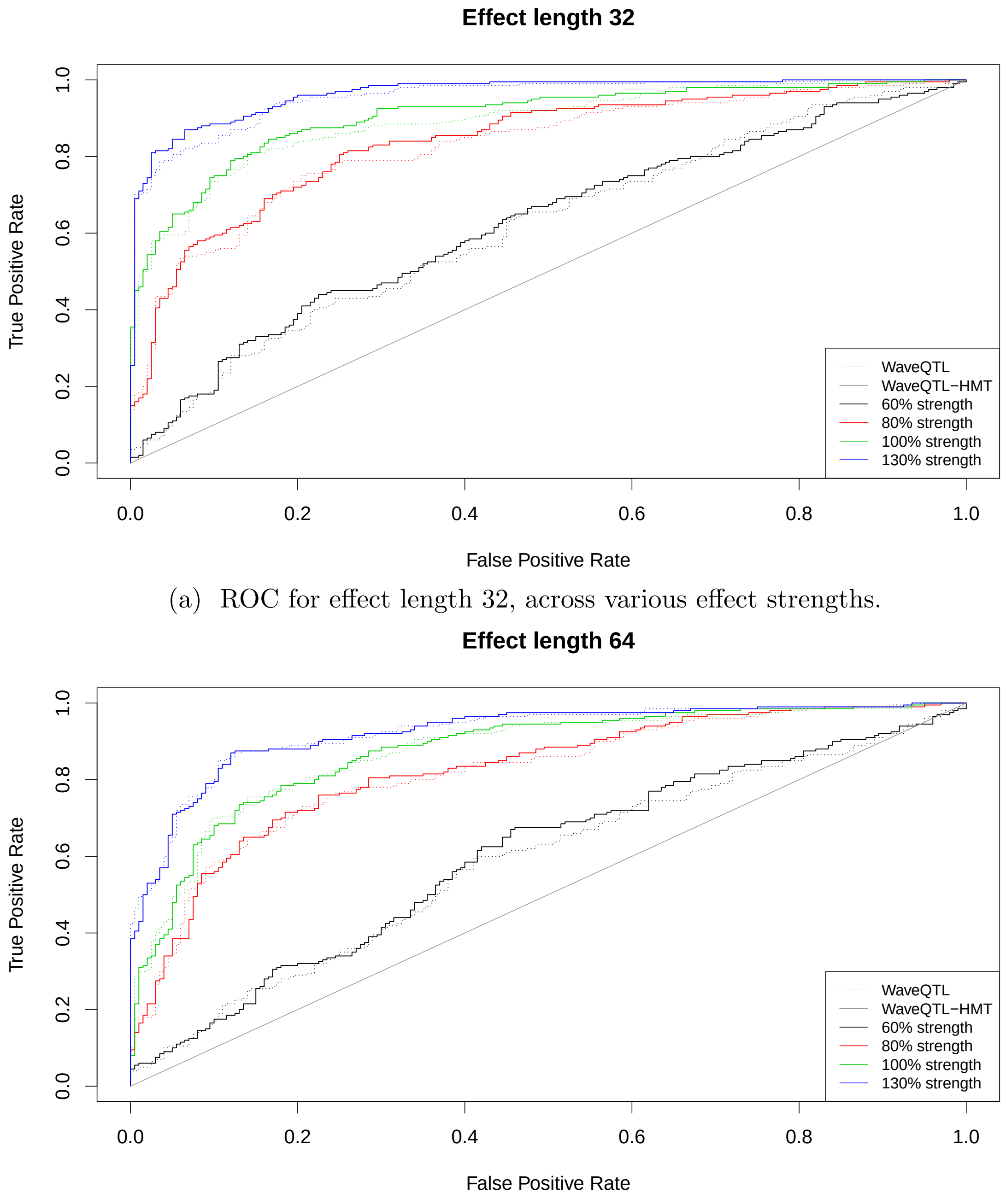
(d) Same as c), but focusing on the simulated region

(e) Difference between average g_2 and g_1 counts

Figure C.5: Length 16 simulation diagnostics

C.4 Simulated data - further results

From the method presented in Section 3.4.1, here are the respective ROC for effect lengths 32 and 64:



(a) ROC for effect length 32, across various effect strengths.

(b) ROC for effect length 64, across various effect strengths.

Figure C.6

Appendix D

Poisson-based model with HMT

D.1 Likelihood test statistic derivation

In this section, $\widehat{\beta}_{s,l}$ and $\mathrm{se}(\widehat{\beta}_{s,l})^2$ represent the MLE and standard error of $\beta_{s,l}$ respectively, with the same notation applying for $\mu^*_{s,l}$. As explained in Shim et al. (in preparation) and Xing (2016), the Laplace method and re-parameterisation means that the likelihood for $(\beta_{s,l}, \mu_{s,l})$ can be approximated as:

$$\mathcal{L}((\beta_{s,l}, \mu_{s,l}); \mathbf{x}_{s,l}, \mathbf{g}) \approx h(\widehat{\mu^*}_{s,l} \mid \mu^*_{s,l}, \mathrm{se}(\widehat{\mu}^*_{s,l})^2) \times h(\widehat{\beta}_{s,l} \mid \beta_{s,l}, \mathrm{se}(\widehat{\beta}_{s,l})^2) \tag{D.1}$$

where $h(\theta; a, b)$ is the PDF of a univariate normal distribution, $\mathbf{x}_{s,l}$ are quantities related to counts in the multiscale space, and (D.1) can be claimed asymptotically.

We start by evaluating the numerator of (4.22):

$$\begin{aligned} &P(\mathbf{x} \mid \mathbf{g}, \boldsymbol{\theta}, \boldsymbol{\theta}^{(\mu)}) \\ &\quad = \sum_{\boldsymbol{\gamma}^{(\boldsymbol{\mu})}} \sum_{\boldsymbol{\gamma}} P(\mathbf{x} \mid \mathbf{g}, \boldsymbol{\theta}, \boldsymbol{\theta}^{(\mu)}, \boldsymbol{\gamma}, \boldsymbol{\gamma}^{(\boldsymbol{\mu})}) P(\boldsymbol{\gamma} \mid \mathbf{g}, \boldsymbol{\theta}) P(\boldsymbol{\gamma}^{(\boldsymbol{\mu})} \mid \mathbf{g}, \boldsymbol{\theta}^{(\mu)}) && \text{(D.2)} \\ &\quad = \sum_{\boldsymbol{\gamma}^{(\boldsymbol{\mu})}} \sum_{\boldsymbol{\gamma}} \prod_{s,l} \left[P(x_{s,l} \mid \mathbf{g}, \boldsymbol{\theta}, \boldsymbol{\theta}^{(\mu)}, \gamma_{s,l}, \gamma^{(\mu)}_{s,l}) \right] \times P(\boldsymbol{\gamma} \mid \boldsymbol{\theta}) P(\boldsymbol{\gamma}^{(\boldsymbol{\mu})} \mid \boldsymbol{\theta}^{(\mu)}) && \text{(D.3)} \end{aligned}$$

due to independence when conditioned on the states. Evaluating the term in the square brackets yields:

$$\begin{aligned} P(x_{s,l}\ |\mathbf{g},\boldsymbol{\theta},\boldsymbol{\theta}^{(\mu)},\gamma_{s,l},\gamma_{s,l}^{(\mu)}) & \\ = \int\int P(x_{s,l}\mid \mathbf{g},\boldsymbol{\theta},\boldsymbol{\theta}^{(\mu)},\gamma_{s,l},\gamma_{s,l}^{(\mu)},\mu_{s,l}^{*},\beta_{s,l})P(\mu_{s,l}^{*}\mid \boldsymbol{\theta},\gamma_{s,l}^{(\mu)})P(\beta_{s,l}\mid \boldsymbol{\theta},\gamma_{s,l})d\mu_{s,l}^{*}d\beta_{s,l} & \end{aligned} \tag{D.4}$$

Note that the first term in the above integral is the (reparameterised) likelihood with the random effect $u_{s,l}^{i}$ integrated out, which was approximated by the Laplace method:

$$\approx \int\int P(\widehat{\beta}_{s,l}\mid \beta_{s,l},\mathrm{se}(\widehat{\beta}_{s,l})^2,\gamma_{s,l})P(\beta_{s,l}\mid \boldsymbol{\theta},\gamma_{s,l})\times\dots$$
$$P(\widehat{\mu}_{s,l}^{*}\mid \mu_{s,l}^{*},\mathrm{se}(\widehat{\mu}_{s,l}^{*})^2,\gamma_{s,l}^{(\mu)})P(\mu_{s,l}^{*}\mid \boldsymbol{\theta},\gamma_{s,l}^{(\mu)})d\mu_{s,l}^{*}d\beta_{s,l} \tag{D.5}$$
$$= \int P(\widehat{\beta}_{s,l}\mid \beta_{s,l},\mathrm{se}(\widehat{\beta}_{s,l})^2,\gamma_{s,l})P(\beta_{s,l}\mid \boldsymbol{\theta},\gamma_{s,l})d\beta_{s,l}\times\dots$$
$$\int P(\widehat{\mu}_{s,l}^{*}\mid \mu_{s,l}^{*},\mathrm{se}(\widehat{\mu}_{s,l}^{*})^2,\gamma_{s,l}^{(\mu)})P(\mu_{s,l}^{*}\mid \boldsymbol{\theta},\gamma_{s,l}^{(\mu)})d\mu_{s,l}^{*} \tag{D.6}$$
$$= P(\widehat{\beta}_{s,l}\mid \mathrm{se}(\widehat{\beta}_{s,l})^2,\gamma_{s,l},\boldsymbol{\theta})P(\widehat{\mu}_{s,l}^{*}\mid \mathrm{se}(\widehat{\mu}_{s,l}^{*})^2,\gamma_{s,l}^{(\mu)},\boldsymbol{\theta}^{(\mu)}) \tag{D.7}$$

Substituting this quantity back into (D.3), we get:

$$P(\mathbf{x}\mid \mathbf{g},\boldsymbol{\theta},\boldsymbol{\theta}^{(\mu)}) = \sum_{\boldsymbol{\gamma}^{(\boldsymbol{\mu})}}\sum_{\boldsymbol{\gamma}}\Big[\prod_{s,l}P(\widehat{\beta}_{s,l}\mid \mathrm{se}(\widehat{\beta}_{s,l})^2,\gamma_{s,l},\boldsymbol{\theta})P(\widehat{\mu}_{s,l}^{*}\mid \mathrm{se}(\widehat{\mu}_{s,l}^{*})^2,\gamma_{s,l}^{(\mu)},\boldsymbol{\theta}^{(\mu)})\Big]\times\dots$$
$$P(\boldsymbol{\gamma}\mid \boldsymbol{\theta})P(\boldsymbol{\gamma}^{(\boldsymbol{\mu})}\mid \boldsymbol{\theta}^{(\mu)}) \tag{D.8}$$
$$= \sum_{\boldsymbol{\gamma}^{(\boldsymbol{\mu})}}\Big[\prod_{s,l}P(\widehat{\mu}_{s,l}^{*}\mid \mathrm{se}(\widehat{\mu}_{s,l}^{*})^2,\gamma_{s,l}^{(\mu)},\boldsymbol{\theta}^{(\mu)})\Big]P(\boldsymbol{\gamma}^{(\boldsymbol{\mu})}\mid \boldsymbol{\theta}^{(\mu)})\times\dots$$
$$\sum_{\boldsymbol{\gamma}}\Big[\prod_{s,l}P(\widehat{\beta}_{s,l}\mid \mathrm{se}(\widehat{\beta}_{s,l})^2,\gamma_{s,l},\boldsymbol{\theta})\Big]P(\boldsymbol{\gamma}\mid \boldsymbol{\theta}) \tag{D.9}$$
$$= \sum_{\boldsymbol{\gamma}^{(\boldsymbol{\mu})}}P(\widehat{\boldsymbol{\mu}}^{*}\mid \mathrm{se}(\widehat{\boldsymbol{\mu}}^{*})^2,\boldsymbol{\gamma}^{(\boldsymbol{\mu})},\boldsymbol{\theta}^{(\mu)})P(\boldsymbol{\gamma}^{(\boldsymbol{\mu})}\mid \boldsymbol{\theta}^{(\mu)})\times\dots$$
$$\sum_{\boldsymbol{\gamma}}P(\widehat{\boldsymbol{\beta}}\mid \mathrm{se}(\widehat{\boldsymbol{\beta}})^2,\boldsymbol{\gamma},\boldsymbol{\theta})P(\boldsymbol{\gamma}\mid \boldsymbol{\theta}) \tag{D.10}$$
$$= P(\widehat{\boldsymbol{\mu}}^{*}\mid \mathrm{se}(\widehat{\boldsymbol{\mu}}^{*})^2,\boldsymbol{\theta}^{(\mu)})P(\widehat{\boldsymbol{\beta}}\mid \mathrm{se}(\widehat{\boldsymbol{\beta}})^2,\boldsymbol{\theta}) \tag{D.11}$$

where $\widehat{\boldsymbol{\beta}} = (\widehat{\beta}_{1,1}, \ldots, \widehat{\beta}_{J,2^{J-1}}), \mathrm{se}(\widehat{\boldsymbol{\beta}})^2 = (\mathrm{se}(\widehat{\beta}_{1,1})^2, \ldots, \mathrm{se}(\widehat{\beta}_{J,2^{J-1}})^2)$ and respectively for $\widehat{\mu}^*$ as we only consider the tree rooted at the node where $s = 1, l = 1$ for now, as we did in the HMT analysis in the Wavelet-based model. Continuing without the coefficient at the zero-th scale, to evaluate our likelihood ratio at the MLEs of the hyperparameter set, $\boldsymbol{\theta}$, we substitute in (D.11):

$$\widehat{\Lambda}(\mathbf{x}, \mathbf{g}) = \frac{P(\widehat{\boldsymbol{\mu}}^* \mid \mathrm{se}(\widehat{\boldsymbol{\mu}}^*)^2, \widehat{\boldsymbol{\theta}}^{(\mu)})P(\widehat{\boldsymbol{\beta}} \mid \mathrm{se}(\widehat{\boldsymbol{\beta}})^2, \widehat{\boldsymbol{\theta}})}{P(\widehat{\boldsymbol{\mu}}^* \mid \mathrm{se}(\widehat{\boldsymbol{\mu}}^*)^2, \widehat{\boldsymbol{\theta}}^{(\mu)})P(\widehat{\boldsymbol{\beta}} \mid \mathrm{se}(\widehat{\boldsymbol{\beta}})^2, \boldsymbol{\theta}_0)} \tag{D.12}$$

$$= \frac{P(\widehat{\boldsymbol{\beta}} \mid \mathrm{se}(\widehat{\boldsymbol{\beta}})^2, \widehat{\boldsymbol{\theta}})}{P(\widehat{\boldsymbol{\beta}} \mid \mathrm{se}(\widehat{\boldsymbol{\beta}})^2, \boldsymbol{\theta}_0)} \tag{D.13}$$

$$= \frac{\sum_{k=0}^{M} P(\widehat{\boldsymbol{\beta}} \mid \mathrm{se}(\widehat{\boldsymbol{\beta}})^2, \gamma_{1,1} = k)P(\gamma_{1,1} = k \mid \widehat{\boldsymbol{\theta}})}{P(\widehat{\boldsymbol{\beta}} \mid \mathrm{se}(\widehat{\boldsymbol{\beta}})^2, \gamma_{1,1} = 0)} \tag{D.14}$$

noting that $P(\gamma_{1,1} = k \mid \boldsymbol{\theta}_0) = 0, k = 1, \ldots, M$, and:

$$= \sum_{k=0}^{M} \frac{P(\widehat{\boldsymbol{\beta}} \mid \mathrm{se}(\widehat{\boldsymbol{\beta}})^2, \gamma_{1,1} = k)\pi_{1,1}^{(k)}}{\prod_{j \in \nu_{1,1}} P(\widehat{\beta}_j \mid \mathrm{se}(\widehat{\beta}_j)^2, \gamma_j = 0)} \tag{D.15}$$

which can be evaluated by recursive quantities from our Upward-Downward algorithm, as mentioned in (4.23):

$$\therefore \widehat{\Lambda}(\mathbf{x}, \mathbf{g}) = \sum_{k=0}^{M} \beta_{1,1}^{(EM)} \alpha_{1,1}^{(EM)} \tag{D.16}$$

As (D.15) only requires us to evaluate quantities related to β, (4.23) only uses outputs from the Upward-Downward algorithm when applied to the β and γ parameters, not the corresponding μ^* parameters. Note also, that analogous to the Wavelet-based models, (4.23) accounts for both the numerator and denominator of equation (D.15), as our algorithm admits $\mathrm{ABF}_{s,l}(\sigma_m^2)$ as inputs, rather than just $P(\widehat{\beta}_{s,l} \mid \mathrm{se}(\widehat{\beta}_{s,l})^2, \gamma_{s,l})$.

D.2 Expectations and Variances for mixture distributions

For a random variable, $Y \mid Z$, where $Z \in \{1, \dots, M\}$

$$\begin{aligned} \mathbb{E}[Y \mid Z = m] &= \mu_m & \text{(D.17)} \\ V(Y \mid Z = m) &= \sigma_m^2 & \text{(D.18)} \\ P(Z = m) &= \pi_m, \text{ where } \sum_{m=1}^{M} \pi_m = 1 & \text{(D.19)} \end{aligned}$$

we have:

$$\begin{aligned} \mathbb{E}[Y] &= \mathbb{E}[\mathbb{E}[Y \mid Z]] & \text{(D.20)} \\ &= \sum_{m=1}^{M} \pi_m \mu_m & \text{(D.21)} \\ \mathbb{E}[Y^2] &= \mathbb{E}[\mathbb{E}[Y^2 \mid Z]] & \text{(D.22)} \\ &= \sum_{m=1}^{M} \pi_m (\mu_m^2 + \sigma_m^2) & \text{(D.23)} \\ V(Y) &= \mathbb{E}[Y^2] - \mathbb{E}[Y]^2 & \text{(D.24)} \\ &= \sum_{m=1}^{M} \pi_m (\mu_m^2 + \sigma_m^2) - \big(\sum_{m=1}^{M} \pi_m \mu_m \big)^2 & \text{(D.25)} \end{aligned}$$

alternatively,

$$\begin{aligned} V(Y) &= \mathbb{E}[V(Y \mid Z)] + V(\mathbb{E}[Y \mid Z]) & \text{(D.26)} \\ &= \sum_{m=1}^{M} \pi_m \sigma_m^2 + (\mathbb{E}[\mathbb{E}[Y \mid Z]^2] - \mathbb{E}[\mathbb{E}[Y \mid Z]]^2) & \text{(D.27)} \\ &= \sum_{m=1}^{M} \pi_m \sigma_m^2 + \Big(\sum_{m=1}^{M} \pi_m \mu_m^2 - \big(\sum_{m=1}^{M} \pi_m \mu_m \big)^2 \Big) & \text{(D.28)} \\ &= \sum_{m=1}^{M} \pi_m (\mu_m^2 + \sigma_m^2) - \big(\sum_{m=1}^{M} \pi_m \mu_m \big)^2 & \text{(D.29)} \end{aligned}$$

D.3 DP algorithms for approximating (pointwise) posterior quantities

In this section, we consider only the example of calculating posterior quantities for the intensity at the leftmost base. We exclude the scaling coefficient, which has already been accounted for. The scale-location indices represent the 'sequence' of nodes where p quantities are required, $\{(1,1),(2,1),\ldots,(J,1)\}$, which we re-index using just the scale, $(1,2,\ldots,J)$ for convenience. Define:

$$\eta_s := \log p_s^1 - \log p_s^0$$

D.3.1 Posterior mean

Here is an alternative algorithm which uses DP for calculating the posterior means. It is required to evaluate the intermediate quantities in the DP algorithm for the posterior variance, as they rely on expectations. The quantity we are after to calculate the posterior mean for the leftmost base is:

$$\mathbb{E}\Big[\sum_{s=1}^{J}\eta_s \mid \mathbf{D},\mathbf{g}\Big] \tag{D.30}$$

To simplify notation, we drop the conditioning on $\mathbf{D},\mathbf{g}$. The algorithm is as follows:

0. **Initialise:** At the finest scale, $j = J$ (bottom of tree), across all states, $m = 0,\ldots,M$, calculate:

$$\mathbb{E}[\eta_J \mid \gamma_J = m, \gamma_J^{(\mu)} = a] \tag{D.31}$$

1. At scale j, noting that η_{j-1} is parent of η_j, calculate, for all states, $m = 0, \dots, M$:

$$\mathbb{E}\Big[\eta_{j-1} + \sum_{s=j}^{J} \eta_s \mid \gamma_{j-1} = m, \gamma_{j-1}^{(\mu)} = a\Big]$$

$$= \mathbb{E}\Big[\mathbb{E}\big[\eta_{j-1} + \sum_{s=j}^{J} \eta_s \mid \gamma_{j-1} = m, \gamma_{j-1}^{(\mu)} = a, \gamma_j, \gamma_j^{(\mu)}\big]\Big] \tag{D.32}$$

$$= \sum_{b=0}^{M} \sum_{n=0}^{M} \Big[\mathbb{E}\big[\eta_{j-1} + \sum_{s=j}^{J} \eta_s \mid \gamma_{j-1} = m, \gamma_{j-1}^{(\mu)} = a, \gamma_j = n, \gamma_j^{(\mu)} = b\big] \times \dots$$

$$P(\gamma_j = n \mid \gamma_{j-1} = m) P(\gamma_j^{(\mu)} = b \mid \gamma_{j-1}^{(\mu)} = a)\Big] \tag{D.33}$$

Noting that γ and $\gamma^{(\mu)}$ are independent, we have:

$$\mathbb{E}\big[\eta_{j-1} + \sum_{s=j}^{J} \eta_s \mid \gamma_{j-1} = m, \gamma_{j-1}^{(\mu)} = a, \gamma_j = n, \gamma_j^{(\mu)} = b\big] =$$

$$\mathbb{E}\big[\eta_{j-1} \mid \gamma_{j-1} = m, \gamma_{j-1}^{(\mu)} = a\big] + \mathbb{E}\big[\sum_{s=j}^{J} \eta_s \mid \gamma_j = n, \gamma_j^{(\mu)} = b\big] \tag{D.34}$$

where the second term in both of these expressions has been evaluated in previous iterations of this algorithm.

2. Set $j := j - 1$

3. If $j = 1$ (coarsest scale), then stop, else return to step 1.

4. To get the desired quantity, marginalise over all states:

$$\mathbb{E}\Big[\sum_{s=1}^{J} \eta_s\Big] = \sum_{a=0}^{M} \sum_{m=0}^{M} \Big[\mathbb{E}\big[\sum_{s=1}^{J} \eta_s \mid \gamma_J = m, \gamma_J^{(\mu)} = a\big] P(\gamma_J = m) P(\gamma_J^{(\mu)} = a)\Big] \tag{D.35}$$

D.3.2 Posterior variance

The quantity we are after is:

$$V\Big(\sum_{s=1}^{J} \eta_s \mid \mathbf{D}, \mathbf{g}\Big) \tag{D.36}$$

To simplify notation, we drop the conditioning on $\mathbf{D}, \mathbf{g}$. The algorithm is as follows:

0. **Initialise:** At the finest scale, $j = J$ (bottom of tree), across all states, $m = 0, \ldots, M$, calculate:

$$V(\eta_J \mid \gamma_J = m, \gamma_J^{(\mu)} = a) \tag{D.37}$$

1. At scale j, noting that η_{j-1} is parent of η_j, calculate, for all states, $m = 0, \ldots, M$:

$$\begin{aligned} V\Big(\eta_{j-1} + \sum_{s=j}^{J} \eta_s \mid \gamma_{j-1} = m, \gamma_{j-1}^{(\mu)} = a\Big) = \\ V\Big(\mathbb{E}\big[\eta_{j-1} + \sum_{s=j}^{J} \eta_s \mid \gamma_{j-1} = m, \gamma_{j-1}^{(\mu)} = a, \gamma_j, \gamma_j^{(\mu)}\big]\Big) + \ldots \\ \mathbb{E}\Big[V\big(\eta_{j-1} + \sum_{s=j}^{J} \eta_s \mid \gamma_{j-1} = m, \gamma_{j-1}^{(\mu)} = a, \gamma_j, \gamma_j^{(\mu)}\big)\Big] \end{aligned} \tag{D.38}$$

To simplify notation in this section, we denote:

$$\pi_{j|j-1}(n) := P(\gamma_j = n \mid \gamma_{j-1} = m) \tag{D.39}$$

$$\pi_{j|j-1}^{(\mu)}(b) := P(\gamma_j^{(\mu)} = b \mid \gamma_{j-1}^{(\mu)} = a) \tag{D.40}$$

$$\mu_{j-1}(\gamma_j, \gamma_j^{(\mu)}) := \mathbb{E}\Big[\eta_{j-1} + \sum_{s=j}^{J} \eta_s \mid \gamma_{j-1} = m, \gamma_{j-1}^{(\mu)} = a, \gamma_j, \gamma_j^{(\mu)}\Big] \tag{D.41}$$

$$\sigma_{j-1}^2(\gamma_j, \gamma_j^{(\mu)}) := V\Big(\eta_{j-1} + \sum_{s=j}^{J} \eta_s \mid \gamma_{j-1} = m, \gamma_{j-1}^{(\mu)} = a, \gamma_j, \gamma_j^{(\mu)}\Big) \tag{D.42}$$

and the means and variances conditioned on specific states are notated as:

$$\mu_{j-1}(\gamma_j = n, \gamma_j^{(\mu)} = b) := \mathbb{E}\Big[\eta_{j-1} + \sum_{s=j}^{J} \eta_s \mid \gamma_{j-1} = m, \gamma_{j-1}^{(\mu)} = a, \gamma_j = n, \gamma_j^{(\mu)} = b\Big] \tag{D.43}$$

$$\sigma_{j-1}^2(\gamma_j = n, \gamma_j^{(\mu)} = b) := V\Big(\eta_{j-1} + \sum_{s=j}^{J} \eta_s \mid \gamma_{j-1} = m, \gamma_{j-1}^{(\mu)} = a, \gamma_j = n, \gamma_j^{(\mu)} = b\Big) \tag{D.44}$$

Noting that γ and $\gamma^{(\mu)}$ are independent, we have:

$$\begin{aligned} V\Big(\mu_{j-1}(\gamma_j, \gamma_j^{(\mu)})\Big) = & \\ & \sum_{b=0}^{M} \sum_{n=0}^{M} \pi_{j|j-1}(b)^{(\mu)} \pi_{j|j-1}(n) \Big[\mu_{j-1}(\gamma_j = n, \gamma_j^{(\mu)} = b)^2 + \sigma_{j-1}^2(\gamma_j = n, \gamma_j^{(\mu)} = b)\Big] - \dots \\ & \left[\sum_{b=0}^{M} \sum_{n=0}^{M} \pi_{j|j-1}(b)^{(\mu)} \pi_{j|j-1}(n) \mu_{j-1}(\gamma_j = n, \gamma_j^{(\mu)} = b)\right]^2 \end{aligned} \tag{D.45}$$

$$\begin{aligned} E\Big[\sigma_{j-1}^2(\gamma_j, \gamma_j^{(\mu)})\Big] = & \\ & \sum_{b=0}^{M} \sum_{n=0}^{M} \pi_{j|j-1}(b)^{(\mu)} \pi_{j|j-1}(n) \sigma_{j-1}^2(\gamma_j = n, \gamma_j^{(\mu)} = b) \end{aligned} \tag{D.46}$$

which is a standard result – see Appendix D for details. Both of (D.39) and (D.40) are output from the Upward-Downward algorithm, whilst for (D.43) and (D.44), due to conditional independence, we have:

$$\mathbb{E}\Big[\eta_{j-1} + \sum_{s=j}^{J} \eta_s \mid \gamma_{j-1} = m, \gamma_{j-1}^{(\mu)} = a, \gamma_j = n, \gamma_j^{(\mu)} = b\Big] = \\ \mathbb{E}\Big[\eta_{j-1} \mid \gamma_{j-1} = m, \gamma_{j-1}^{(\mu)} = a\Big] + \mathbb{E}\Big[\sum_{s=j}^{J} \eta_s \mid \gamma_j = n, \gamma_j^{(\mu)} = b\Big] \tag{D.47}$$

$$V\Big(\eta_{j-1} + \sum_{s=j}^{J} \eta_s \mid \gamma_{j-1} = m, \gamma_{j-1}^{(\mu)} = a, \gamma_j = n, \gamma_j^{(\mu)} = b\Big) = \\ V\Big(\eta_{j-1} \mid \gamma_{j-1} = m, \gamma_{j-1}^{(\mu)} = a\Big) + V\Big(\sum_{s=j}^{J} \eta_s \mid \gamma_j = n, \gamma_j^{(\mu)} = b\Big) \tag{D.48}$$

where the second term in both of these expressions has been evaluated in previous iterations of this algorithm, or the algorithm when applied to expectations.

2. Set $j := j - 1$

3. If $j = 1$ (coarsest scale), then stop, else return to step 1.

4. To get the desired quantity, marginalise over all states:

$$V\Big(\sum_{s=1}^{J} \eta_s\Big) = \sum_{a=0}^{M} \sum_{m=0}^{M} \Big[V\Big(\sum_{s=1}^{J} \eta_s \mid \gamma_J = m, \gamma_J^{(\mu)} = a\Big) P(\gamma_J = m) P(\gamma_J^{(\mu)} = a)\Big] \tag{D.49}$$

Bibliography

F. Abramovich, T. Sapatinas, and B. W. Silverman. Wavelet thresholding via a bayesian approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(4):725–749, 1998.

F. W. Albert and L. Kruglyak. The role of regulatory variation in complex traits and disease. *Nature Reviews Genetics*, 16(4):197, 2015.

M. S. Crouse, R. D. Nowak, and R. G. Baraniuk. Wavelet-based statistical signal processing using hidden markov models. *IEEE Transactions on signal processing*, 46(4):886–902, 1998.

I. Daubechies. *Ten lectures on wavelets*, volume 61. Siam, 1992.

J. F. Degner, A. A. Pai, R. Pique-Regi, J.-B. Veyrieras, D. J. Gaffney, J. K. Pickrell, S. De Leon, K. Michelini, N. Lewellen, G. E. Crawford, et al. Dnase i sensitivity qtls are a major determinant of human expression variation. *Nature*, 482(7385): 390, 2012.

D. L. Donoho and I. M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the american statistical association*, 90(432):1200–1224, 1995.

R. E. Kass and A. E. Raftery. Bayes factors. *Journal of the american statistical association*, 90(430):773–795, 1995.

E. D. Kolaczyk. Bayesian multiscale models for poisson processes. *Journal of the American Statistical Association*, 94(447):920–933, 1999.

L. Ma and J. Soriano. Analysis of distributional variation through graphical multiscale beta-binomial models. *Journal of Computational and Graphical Statistics*, 27(3):529–541, 2018.

S. G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (7): 674–693, 1989.

J. S. Morris and R. J. Carroll. Wavelet-based functional mixed models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(2):179–199, 2006.

L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

O. Ronen, J. Rohlicek, and M. Ostendorf. Parameter estimation of dependence tree models using the em algorithm. *IEEE Signal Processing Letters*, 2(8):157–159, 1995.

B. Servin and M. Stephens. Imputation-based analysis of association studies: candidate regions and quantitative traits. *PLoS genetics*, 3(7):e114, 2007.

H. Shim and M. Stephens. Wavelet-based genetic association analysis of functional phenotypes arising from high-throughput sequencing assays. *The annals of applied statistics*, 9(2):655, 2015.

H. Shim, Z. Xing, E. Pantaleo, and M. Stephens. Bayesian multi-scale models for detecting differences in high-thoughput sequencing data between multiple groups and their applications in small sample sizes. in preparation.

M. Stephens. False discovery rates: a new deal. *Biostatistics*, 18(2):275–294, 2016.

D. C. Thomas et al. *Statistical methods in genetic epidemiology.* Oxford University Press, 2004.

K. Timmerman and R. D. Nowak. Multiscale modeling and estimation of poisson processes with application to photon-limited imaging. *IEEE Transactions on Information Theory*, 45(3):846–842, 1999.

Z. Xing. *Poisson multiscale methods for high-throughput sequencing data.* PhD thesis, The University of Chicago, 2016.