Simulation Study of Markov Chain Composite Likelihood and its Application in Recombination Model

Grace E. Rhodes

Advisor:
Marie Ozanne, Ph.D. (*Mount Holyoke College*)

Thesis Committee:
Jianping Sun, Ph.D. (*University of North Carolina at Greensboro*),
Timothy Chumley, Ph.D. (*Mount Holyoke College*) ,
and
Mara Breen, Ph.D. (*Mount Holyoke College*)

A thesis submitted to the Department of Mathematics
and Statistics in partial fulfillment of the requirements for
the degree of Bachelor of Arts in Statistics.

Department of Mathematics and Statistics
Mount Holyoke College
South Hadley, MA 01075
May 2022

Abstract

DNA sequencing technologies are rapidly advancing, allowing researchers access to data which is both high quality and highly detailed. In particular, these technologies are able to record the allele found at single nucleotide polymorphism (SNP) sites on individual haplotypes. A central goal for SNP data is SNP mapping, which would facilitate advancements in genetics, including hierarchical trees that enrich our understanding of human evolutionary history. Though geneticists have detailed SNP data on extant humans, this is not the case for previous generations, necessitating estimation backward in time. There is a need for statistical methods that perform this estimation. The statistical question is: If we observe n current descendant binary sequences with length L, how can we estimate the unknown ancestor distribution while considering biological complexities? Recombination, a biological complexity involving an exchange of genetic material between chromosomes, can give descendants haplotypes which don't match ancestral chromosomes. Sun (2011) proposed a Recombination Model which estimates the unknown ancestral distribution while considering a fixed probability of recombination. Markov chain composite likelihood (MCCL) is used to obtain estimates of the population frequency with which the ancestor will have a given binary sequence. Under the assumption that both ancestor and descendant sequences follow an order-m Markov Chain structure, hierarchical estimation is used to estimate the joint distribution from marginal estimates.

Here, we run simulations for this estimator and focus on the use of MCCL and selection of fixed quantities. Our data-generating mechanism will be done via resampling using data from the International HapMap Project, allowing sample proportions to simulate a true ancestor distribution. Performance measures will include bias and standard error of both joint and marginal estimates, bootstrapped confidence intervals, and total density correctly assigned to true non-zero probability sequences. Marginal distribution results show that the method provides estimates with low bias and standard error, but show evidence of a directional effect of the use of MCCL such that both bias and standard error increase for sites further from the start of the chain. Joint distribution results show a trade-off between bias and standard error; increasing m decreases the bias but increases the standard error. The joint density sums show that nearly all of the density is assigned to either true non-zero sequences or sequences which are 85 % similar. Finally, to make this methodology accessible, an R package **recombinationMCCL** is currently under development with a preliminary version available on Github.

Acknowledgements

I would first like to thank my two amazing mentors on this project, Marie Ozanne and Jianping Sun. Professor Ozanne is incredibly inspiring to me as a professor and a researcher, and I'm so grateful to have had her as a thesis advisor. Working with Professor Ozanne has made me a better writer and a better student, and her mentorship was instrumental to me continuing to study Biostatistics as a PhD student next fall at Duke University. I began work on this project with Dr. Sun last summer as part of an REU at University of North Carolina at Greensboro, and I'm very grateful to her for the opportunity to contribute to her project. Working with Dr. Sun pushed me as a researcher and her mentorship made me much more comfortable with R coding.

I would also like to thank the other two members of my committee, Tim Chumley and Mara Breen. I'm so grateful to them both for taking the time to provide feedback on this project. I had the opportunity to take multiple math classes with Professor Chumley and his dedication to his students made me feel more comfortable with advanced math classes. Professor Breen's Intro to Psychology class was my first college course I attended, so being able to receive her feedback on this project feels full circle.

I'd next like to thank my parents and my brother Henry for their constant support. Thank you for the emotional support pictures of cats, I hope I continue making you guys proud! And of course thank you to all of my adorable cats: Moxie Doodle, Peanut, Bert Bungus, the late Mort, the late Spunky Monkey, and the late Snarf.

Lastly, a huge thank you to my lovely friends: Aya, Cydney, Dani, Inga, Kaitlin, Margaret, and Olivia. Thank you for listening to many rants about code, editing my grammar, and being an audience for many practice presentations.

Work on this project began at the 2021 Research Experience for Undergraduates in Complex Data Analysis using Statistical and Machine Learning Tools at the University of North Carolina at Greensboro. Dr. Sat Gupta was the program head, and Dr. Jianping Sun was my faculty mentor. This REU was funded by an NSF grant (DMS-1950549)

Contents

Chapter 1

Introduction

DNA sequencing technologies are rapidly advancing, allowing researchers access to data which is both high quality and highly specific. The level of specificity now available is the result of two important improvements. The first is the detection of variations in nucleotide bases; in particular, these technologies are able to sequence single nucleotide polymorphisms (SNPs). For the vast majority of sites on the human genome, all humans will have the same nucleotide base (A, C, T, or G). Single nucleotide polymorphisms refer to the sites on the genome where there is variation among humans such that there is a major and minor allele (Philips and Milo 2015). For example, it may be that 2% of the population has A while 98% has G; in this case A is the minor allele and G is the major allele. To be classified as a SNP, at least 1% of the population must have the minor allele (Nature Education 2014). The ability to accurately collect data on these SNP sites is an important advancement for genetics research because these SNPs implicitly provide information about the processes that produce variation in the human genome.

The second advancement is the phasing of these variants into haplotypes. A haplotype refers broadly to the genetic material that is inherited together

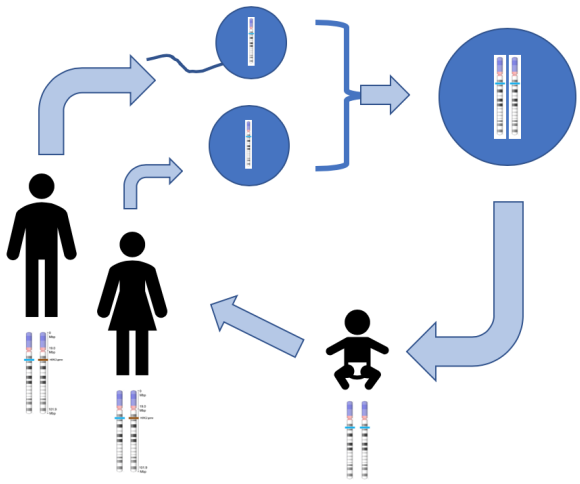

Figure 1.1: Figure from Bartee, Shriner, and Creech 2017 showing genetic inheritance. The offspring inherits one haplotype from each parent.

from a single parent, while genotype refers to all of an organism's genetic material. The term *haplotype* can be defined in multiple ways, but we can state broadly that genotypes are composed of two haplotypes, one inherited from each biological parent (Figure 1.1). In some cases, the haplotype may refer only to the genes which are inherited together rather than all of the genetic material (Nature Education 2022). Alternatively, the haplotype can refer to the set of DNA variations or polymorphisms which are inherited together (National Human Genome Research Institute 2022b). The commonality to all of these definitions is that the term *haplotype* refers to the genetic material that is passed together from a single parent to their offspring. Here, we will use haplotype to refer to the polymorphisms that are inherited together. For an individual, the ability to phase their recorded SNPs into haplotypes means that we know which SNPs were passed from the mother and which were passed from the father. This tells us a great deal about genetic inheritance and genetic lineages.

The result is haplotype SNP data, or data that encodes the sites on the chro-

mosome where there exists variation that were inherited together. This data allows researchers to answer questions about variation in the human genome and its relation to genetic lineages.

Variation in human genetics is produced by processes such as mutation and recombination. Thus, analysis of haplotype SNP data can be enriched by considering these processes. Mutation occurs at a single site on the genome during the process of transcription and results in a change in the nucleotide base at that one site relative to the haplotype inherited from the parent (Bartee, Shriner, and Creech 2017). For example, if at site 1 the parent has nucleotide base A this may become nucleotide base G if there is a mutation.

Recombination events, on the other hand, occur during meiosis. In this process there is a fusing between two chromosomes that results in an exchange of genetic material (Harold L.K. Whitehouse 1982; Philips and Milo 2015). For a visualization of this process, see Figure 1.2. The result is that the descendant may inherit a haplotype which does not match any of the haplotypes held by their parents. For example, suppose that one parent has the sequence $AAAAA$ and the other has the sequence $GGGGG$. If a recombination event occurs that splits the chromosome between sites 2 and 3, then one of the descendant's haplotypes may be $AAGGG$. It has been estimated that recombination events occur about once or twice per chromosome per replication (Philips and Milo 2015).

Lastly, another benefit of this data is that it is easily accessible to the scientific community. Several databases of haplotype SNP data have been made publicly available, including the International HapMap Project data (International HapMap Project 2009). The International HapMap Project is a collaborative project "among researchers at academic centers, non-profit biomedical research

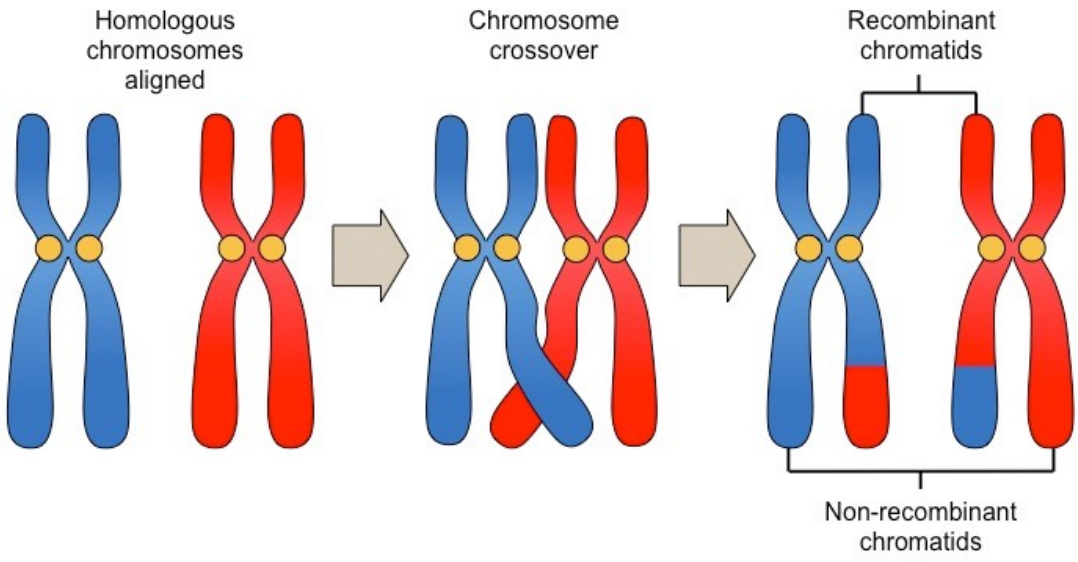

Figure 1.2: Visualization of a recombination event from Cornell 2016. During meiosis, two chromosomes fuse together, called a crossover event, resulting in an exchange of genetic material.

groups and private companies in Japan, the United Kingdom, Canada, China, Nigeria and the United States." (National Human Genome Research Institute 2022a) The project collected SNP data phased into haplotypes from people belonging to 11 ancestry groups around the world. The goal for the project was to develop a haplotype map for the human genome. Another notable source of publicly available SNP data is the 1000 Genomes Project (1000 Genomes 2022).

The degree of specificity, quality, and accessibility of these data mean that they have the potential to be used by researchers to answer a wide variety of research questions.

1.1 Scientific Questions

A current goal in genetics research is to create maps of these SNPs. These maps would record the nucleotide base and the location of the SNP on the human genome, allowing researchers to identify the SNPs associated with diseases. Diabetes, cancer, heart disease, stroke, depression and asthma, for example, have all been shown to have genetic predispositions (National Human Genome Research Institute 2022a). Genome-wide association studies (GWAs), which identify sites on the genome associated with disease, are currently a very active area of research. In addition, it has been shown that variations in DNA can affect the development of disease, suggesting that SNPs play a role in predicting disease. Maps of SNPs would facilitate identifying the SNPs associated with an increased likelihood of developing a disease.

Mapping SNPs on the human genome can then be used towards another goal: constructing hierarchical trees. These trees would record the passing of SNPs from parent to offspring across several generations, showing the genetic lineages that came from a common ancestor. These hierarchical trees would enrich our understanding of human evolutionary history. For example, it is at this point unknown what the genome of the common human ancestor may have looked like. These hierarchical trees may also play a role in identifying genetic lineages which are susceptible to disease.

A current barrier in constructing these hierarchical trees is the fact that, while we have highly detailed data for extant humans, this is not the case for their ancestors. Hence, there is a need to estimate backward across generations. Specifically, there is a need for methods that can produce estimates of ancestor DNA sequences based on observed descendant sequences to fill in for

this technological gap.

1.2 Statistical Question

The process of estimating ancestors' DNA can be approached as a statistical question. Mathematically, the building of these hierarchical trees is equivalent to estimating the unknown distribution of ancestral SNP sequences from current descendants because, in both cases, we are looking backwards from the current state of the SNP sequences. Suppose that we observe n descendants' SNP sequences of length L. There is a need for statistical methods which can estimate the unknown distribution of ancestral SNP sequences from observed descendants while considering biological complexities such as recombination and mutation.

1.3 Overview of Previous Models

The first statistical model to address this question was the Ancestral Mixture Model, proposed by Chen and B. G. Lindsay 2006. In this model, Chen and Lindsay extend the idea of a multivariate normal mixture model to be used with binary sequence data, which represent SNP sequences, to estimate the population structure at fixed time points in the past.

A mixture model is a class of probabilistic models which can be used to represent sub-populations within an overall population. These sub-populations are not required to be measured in the observed data, meaning that they may be latent sub-populations. Mixture models come from machine learning, and these models are used to generate k clusters of the feature data where k is dependent

on the state of the machine. These models include a mixture distribution, which specifies how observations from the overall population are distributed. Thus, one type of mixture model is a multivariate normal mixture model, where the mixture distribution is assumed to be multivariate normal. Multivariate normal mixture models follow a Bayesian approach; we have a vector of unknown parameters, which we assume to be multivariate normal distributions, and provide initial estimates for these parameters which are then updated using an EM algorithm. The estimated parameters provide information about how the features of the data can be used to cluster the data. For example, a normal mixture model can be used to explain housing prices. House prices vary widely across the housing market, but the prices tend to be similar and predictable for houses of the same type in the same neighborhood. We could develop a mixture model with K components such that each component is a house type-neighborhood combination (e.g. three bedroom house in neighborhood B). We assume that each component is distributed normally with unknown mean and variance. If we fit this model to observed house prices, then an EM algorithm will tend to cluster houses of the same type in the same neighborhood and provide estimates of the spread of data in each of these clusters. For more details on mixture models, see Wikipedia 2021b.

Chen and B. G. Lindsay 2006 extend the multivariate normal mixture model to be used with discrete data. In particular, their discrete data are binary sequences of length L which represent SNP sequences. They argue that, at a SNP site, there are typically only two nucleotide bases which will be possible at that site (A or G at a purine site, and T or C at a pyrimidine site) so the data can be expressed as binary sequences. For more details on this, see Section 2.1. The data are therefore a discrete sample space of size 2^L. The

Ancestral Mixture Model uses mixture analysis to build hierarchical trees of SNP inheritance by identifying clusters to which we can assign any descendant in a probabilistic way while accounting for the rate of mutation.

Let $X_1, X_2, \ldots, X_n$ be observed binary sequences with length L, which represent descendant sequences. Further, let μ denote the ancestor binary sequences and let p denote the mutation rate. Chen and B. G. Lindsay 2006 define a random variable Θ which has distribution Q. Q will be a discrete distribution with K units of support each of which correspond to an ancestor binary sequence $\mu_1, \mu_2, \ldots, \mu_K$. These ancestors can be thought of in context as lineage groups. Then each observation X_n can be thought of as a random draw Θ from Q plus an error term, ϵ. Differences between the ancestor μ and an observation X will be the result of mutation. Thus, we define the mutation kernel which is dependent on the mutation rate p. The mutation kernel is used to determine the error. Then each descendant can be represented by $X = \Theta + \epsilon$.

This model can be used to estimate hierarchical trees. Allow Q to have an arbitrary number of components, or an arbitrary number of ancestors. Use p as a sieve parameter, or a parameter that captures evolutionary time. Then measure time in the hierarchical trees as the amount of mutations that occur. It is clear that, the further back we go in evolutionary history, the fewer ancestors there will be. An EM algorithm is used to estimate the K clusters which represent lineage groups. Then, allowing mutations to accumulate over generations through the additive properties of the sieve parameter, these clusters can be used to build hierarchical trees. This is done by identifying the points at which a mutation caused a lineage group to "split".

Chen and B. G. Lindsay 2006 include a case study of their method using mitochondrial DNA data. The hierarchical tree generated for this data can be

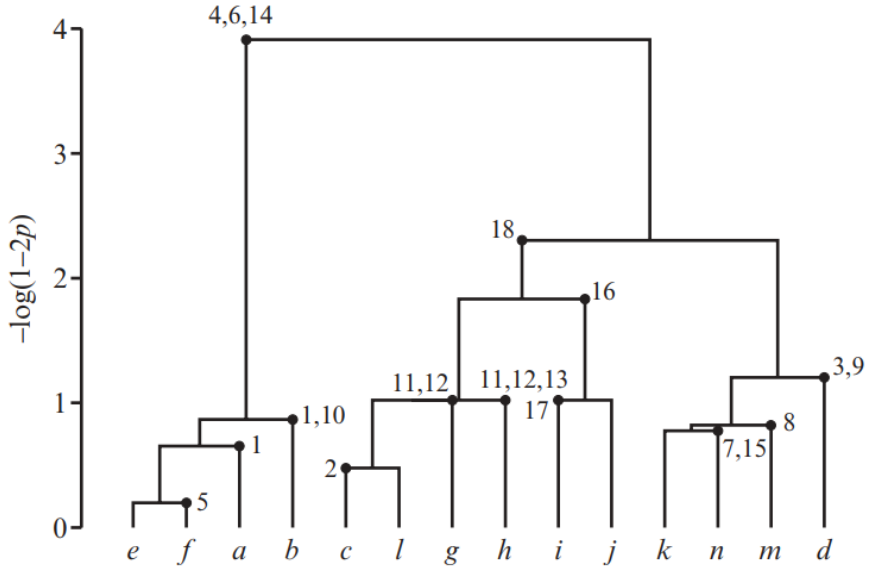

Figure 1.3: Hierarchical Tree built for case study using mitochondrial DNA data from Chen and B. G. Lindsay 2006. The y-axis represents time measures in the amount of mutation. The x-axis represents the estimated clusters, which are identified lineages. This case study identified 14 lineages. The points where the branches meet represent two lineages merging due to reaching the mutation that created those two lineages. For example, lineages c and l merge at $p = 0.19$ because it was at this point that there was a mutation at site 2 which created lineage l from lineage c.

seen in Figure 1.3. Observe that the y-axis indexes evolutionary time measured in the amount of mutations, and the x-axis indexes identified lineage groups. At $-\log(1-2p) = 0$, the method identifies 14 lineage groups. Going backwards in time to $-\log(1-2p) = 4$, we can see that these 14 lineage groups are traced back to one ancestor. At $-\log(1-2p) = 4$, mutations on sites $4, 6, 14$ create two lineage groups from this common ancestor. The left lineage group experiences another mutation when $-\log(1-2p) = 1$, at sites 1 and 10, creating two more lineage groups. Similar processes occur in the right lineage group until 14 lineage groups are formed.

The Ancestral Mixture Model proposed by Chen and B. G. Lindsay 2006

had several important contributions to the statistical question. First, they suggest representing the SNP sequences as binary sequences which are more easily handled mathematically. Second, their model allows mutations to accumulate over generations which is biologically accurate. Third, their model easily lends hierarchical trees which are important to answering scientific questions which include understanding human evolutionary history and identifying lineages susceptible to certain diseases. However, the Ancestral Mixture Model fails to account for recombination. Because recombination events can change large portions of the sequence, this is a biological complexity that is important to consider when estimating ancestors' DNA from their descendants'.

1.4 Present Study

The Recombination Model, proposed by Jianping Sun 2011, built on the work in Chen and B. G. Lindsay 2006 by proposing a method that focuses on the probability of recombination rather than the probability of mutation. Let $X_1, X_2, \ldots, X_n$ be a sample of observed descendant sequences, each of length L, such that $X_i = (x_1, x_2, \ldots, x_L)$ where $x_s \in \{0, 1\}$ for each $s = 1, \ldots, L$ and for all $i = 1, \ldots, n$. Stated intuitively, this is a sample of n observed descendant binary sequences which represent sequences of L SNPs such that 0 represents the major allele and 1 represents the minor allele. Let q denote the probability of recombination occurring between any two given sites during the passing of SNPs from ancestor to descendant. Since, as discussed above, if a recombination event occurs in the passing of genetic material the descendant's haplotype will not exactly match either of their ancestors' haplotypes, it is important to consider q when estimating the ancestor's SNP sequence from

their descendants'. Let $\pi_{1,\ldots,L}(x_1,\ldots,x_L)$ denote the population frequency with which the ancestor has the binary sequence $(x_1,\ldots,x_L)$ on sites $1,\ldots,L$. The Recombination Model implements Markov chain composite likelihood to use the observations $X_1, X_2, \ldots, X_n$ and the probability of recombination q to estimate the $\pi_{1,\ldots,L}(x_1,\ldots,x_L)$. More details on this method are provided in Chapter 2.

In this project, we conduct a simulation study to assess the performance of the estimator proposed by Jianping Sun 2011. We analyze both the marginal distribution estimates and the joint distribution estimates provided by the method. In particular, we focus on assessing the influence of the selected values for q and m, and the influence of the use of Markov chain composite likelihood, and interpret these results in a genetic context. The simulation design is discussed in Chapter 3 and the simulation results are discussed in Chapter 4.

Lastly, we discuss the `R` package currently in development to make this methodology accessible. Details on the functions available in the **recombinationMCCL** package are provided in Chapter 5, and a preliminary version of the package is available through Github at rhode22g/recombinationMCCL.

Chapter 2

Overview of Recombination Model

Here, we present an overview of the key elements of the Recombination Model. For more details and for proofs of methods, see Jianping Sun 2011.

2.1 Model Formulation

Suppose that, for a given descendant, we observe L SNP sites on one of their haplotypes. These will each be one of the four nucleotide bases that make up strands of DNA: adenine (A), cytosine (C), guanine (G), and thymine (T). A and G are the purine bases, and C and T are the pyrimidine bases. These form two sets of complementary base pairs such that a purine must bind with a pyrimidine: A will always bind with T and C will always bind with G. This means that at a given SNP site s, there are at most two possibilities for which allele will be present. Recall that for SNP sites, there will be a major and minor allele. Thus, we can simplify this descendant's sequence of SNP sites by

regarding it as a binary sequence where 0 corresponds to the major allele and 1 corresponds to the minor allele.

Let $X = (x_1, \ldots, x_L)$ denote the binary SNP sequence for one descendant on L sites such that $x_s \in \{0, 1\}$ for $s = 1, \ldots, L$. To begin writing a likelihood statement involving this descendant's observed binary sequence, we must begin by considering the probability of observing this particular descendant $P(X = (x_1, \ldots, x_L))$. If we let the possibility of a recombination event occurring be zero, then this probability is dependent only on the probability of their ancestor having that sequence. Let $\pi_{1,\ldots,L}(x_1, \ldots, x_L)$ denote the population frequency with which the ancestor has the sequence $(x_1, \ldots, x_L)$ on sites $s = 1, \ldots, L$. Then under the condition where the probability of recombination is zero,

$$P(X = (x_1, \ldots, x_L)) = \pi_{1,\ldots,L}(x_1, \ldots, x_L).$$

We now consider the case where the probability of recombination is non-zero. Let q denote the probability of recombination. We assume that q is constant for all sites. We further assume that the probability of recombination between consecutive sites j, k is independent of the probability of recombination between consecutive sites g, h where $j, k, g, h \in [0, L]$ and $j \neq k, g \neq h$. Then there are $L - 1$ locations where recombination is possible, and 2^{L-1} recombination possibilities.

To state explicitly what is meant by "recombination possibility," consider a sequence such that $L = 3$. We can state generally that $X = (x_1, x_2, x_3)$. We must consider every possible way that this descendant could have this specific sequence by taking into consideration all of the possible "recombination/ no recombination" scenarios. There are $2^{3-1} = 4$ possible scenarios: between

sites 1 and 2 there are two possibilities (recombination and no recombination) and between sites 2 and 3 there are two possibilities (recombination and no recombination). Only one of these will have actually occurred, so these are disjoint events each corresponding to a recombination possibility (see Figure 2.1.)

$s = 1, s+1 = 2, s+2 = 3$				
Sites	Recombination	Sites	Recombination	Expression
s, $s+1$	$\mathbf{q}$	$s+1$, $s+2$	$\mathbf{q}$	$q^2\pi_1(x_1)\pi_2(x_2)\pi_3(x_3)$
			$(\mathbf{1-q})$	$q(1-q)\pi_1(x_1)\pi_{2,3}(x_2,x_3)$
	$(\mathbf{1-q})$	$s+1$, $s+2$	$\mathbf{q}$	$(1-q)q\pi_{1,2}(x_1,x_2)\pi_3(x_3)$
			$(\mathbf{1-q})$	$(1-q)^2\pi_{1,2,3}(x_1,x_2,x_3)$

Table 2.1: Recombination possibilities for $L = 3$. In these possibilities, we observe that the results contain marginal probabilities– let $\pi_{1,2}(x_1, x_2) = \sum_{x_3} \pi_{1,2,3}(x_1, x_2, x_3)$. For an illustrative example, consider the second expression. This is the instance where recombination does occur between sites 1 and 2, and recombination doesn't occur between sites 2 and 3. This is captured by $q(1-q)$. This means that site 1 is inherited separately from sites 2 and 3. Then we need to include the marginal probability of the ancestor having x_1 on site 1 $\pi_1(x_1)$ and the marginal probability of the ancestor having (x_2, x_3) on sites 2 and 3 $\pi_{2,3}(x_2, x_3)$.

Then $P(X = (x_1, x_2, x_3))$ is equal to the sum of these disjoint events. To continue with our example of $L = 3$,

$$\begin{aligned} f(x_1, x_2, x_3) &= P(X = (x_1, x_2, x_3)) \\ &= (1-q)^2\pi_{1,2,3}(x_1, x_2, x_3) + (1-q)q\pi_{1,2}(x_1, x_2)\pi_3(x_3) \\ &\quad + q(1-q)\pi_1(x_1)\pi_{2,3}(x_2, x_3) + q^2\pi_1(x_1)\pi_2(x_2)\pi_3(x_3). \end{aligned}$$

Thus, the probability of observing a descendant with a specific sequence is governed by the probability of the ancestor having segments of or the whole of that sequence and the probability of recombination. Generalized to a sequence

of length L, the Recombination Model is given by Equation (2.1) below.

$$
\begin{aligned}
&f(x_1, x_2, ..., x_L) = P(X = (x_1, x_2, ..., x_L)) \qquad (2.1)\\
&= (1-q)^{L-1}\pi_{1,2,...,L}(x_1, x_2, ..., x_L)\\
&+ q(1-q)^{L-2}\sum_{s=1}^{L-1}[\pi_{1,...,s}(x_1, ..., x_s)\pi_{s+1,...,L}(x_{s+1}, ..., x_L)]\\
&+ q^2(1-q)^{L-3}\sum_{s_1=1}^{L-2}\sum_{s_2=s_1+1}^{L-1}[\pi_{1,...,s_1}(x_1, \ldots, x_{s_1}\pi_{s_1+1,...,s_2}(x_{s_1+1}, \ldots, x_{s_2})\\
&\pi_{s_2+1,...,L}(x_{s_2+1}, \ldots, x_L)]\\
&\vdots\\
&+ q^{L-1}\pi_1(x_1)\pi_2(x_2)...\pi_L(x_L)
\end{aligned}
$$

Recall that our aim is to estimate the unknown ancestral distribution. Thus, from Equation (2.1), we must estimate each $\pi_{1,...,L}(x_1, \ldots, x_L)$. Suppose that we observe n descendant binary sequences. Let $n_{1,...,L}(x_1, \ldots, x_L)$ denote the number of descendants in the sample that have the sequence $(x_1, \ldots, x_L)$ on sites $1, \ldots, L$. Then we have the following log-likelihood statement,

$$\ell(\pi|\mathbf{X}) = \sum_{x_1,...,x_L} n_{1,2,...,L}(x_1, x_2, ..., x_L)\log[P(X = (x_1, x_2, ..., x_L))]. \qquad (2.2)$$

2.2 Computation Problem

We aim to estimate each $\pi_{1,...,L}(x_1, \ldots, x_L)$ from Equation (2.2). Since we are able to write this log-likelihood statement explicitly, this would ideally be a matter of maximizing the log-likelihood. However, a maximum likelihood estimation (MLE) approach is not computationally feasible due to the

long sequence lengths that are inherent to genetics data. For example, 116415 SNP sites have been observed on Chromosome 1 according to data by the International HapMap Project (International HapMap Project 2009). With 2^{L-1} recombination possibilities encapsulated by Equation (2.1), it is clear that maximizing Equation (2.2) quickly becomes infeasible as we increase L.

Thus, it is necessary to suggest another method of estimation. Jianping Sun 2011 proposes use of composite likelihood, which has been used previously for high-dimensional data by B. Lindsay 1988. Composite likelihood is a type of "pseudo-likelihood" which replaces the full likelihood with a product of component likelihoods. These component likelihoods are conditional or marginal events for which we can obtain the log-likelihood, and the component likelihoods used are chosen depending on the context (Varin, Nancy Reid, and Firth 2011). Composite likelihood can be defined as,

$$\mathcal{L}_C(\theta, y) = \prod_{k=1}^{K} \mathcal{L}_k(\theta, y)^{w(k)}.$$

Denote $f(y, \theta)$ as the probability density function associated with an m-dimensional vector random variable $\mathbf{Y}$ for some unknown p-dimensional parameter vector θ; it's associated likelihood is $\mathcal{L}_C(\theta, y)$. Let $\{\mathcal{A}_1, \ldots, \mathcal{A}_K\}$ be a set of marginal or conditional events, and let their likelihoods be $\mathcal{L}_K(\theta, y) \propto f(y \in \mathcal{A}_L, \theta)$. Let $w(k)$ be chosen non-negative weights, which can be chosen to be equal (Varin, Nancy Reid, and Firth 2011; B. Lindsay 1988). Maximum composite likelihood estimation has been shown to be a consistent method of estimation, but does involve some loss of efficiency (Xu and N. Reid 2011).

Furthermore, in the case where conditional component likelihoods are used, Besag 1974 introduced the following specific notation in the context of spatial

applications,

$$\mathcal{L}_C(\theta, y) = \prod_{r=1}^{m} f(y_r | \{y_s : y_s \text{ is a neighbor of } y_r\}, \theta)$$

as well as introducing the use of Markov fields to derive the conditional component likelihoods.

Composite likelihood has been used previously in statistical genetics applications, particularly in population genetics problems (*for an overview see* Larribe and Fearnhead 2011). These applications include composite likelihood estimation of recombination rates, and estimating genetic maps of SNPs. The use of composite likelihood in similar genetics problems support its use here. Based on the existing body of literature on composite likelihood, Jianping Sun 2011 proposes the use of a Markov chain to derive conditional component likelihoods.

A Markov chain is a Stochastic Process with the property that the future, conditioned on the present, is independent of the past (Dobrow 2016). We can re-frame this definition to be in terms of SNP sites rather than time indices. Then a Markov chain applied to SNPs is a Stochastic Process with the property that the next SNP sites, conditioned on the present SNP sites, is independent of the previous SNP sites. This implies that for the next SNP site, it is dependent on some number of SNP sites m before it but independent of sites $1, \ldots, m-1$ and of sites after it. It has been established in the genetics literature that there is dependence between sites on the chromosome, and that the dependence is governed by physical proximity. For example, Lee 2016 established via a probabilistic model that SNPs are distributed across the genome in clusters as opposed to previous assumptions of random distribution. The suc-

cess of this model in explaining HapMap SNP data suggests that SNPs tend to have a dependence on the SNP sites that are within some measure of physical proximity. Similar methods of conditioning on SNP sites based on measures of physical distance, such as splitting the data into k consecutive SNPs, are described in Larribe and Fearnhead 2011.

This supports imposing conditional probabilities that captures this relationship between SNP sites by physical distance. Recall, however, that our aim is to simplify a complex probability statement. Conditioning on sites in both directions, left and right, would introduce an additional source of complexity. Jianping Sun 2011 thus makes the simplifying assumption that a SNP site is only dependent on the sites to the left of it. We will explore the effects of this assumption in our simulation (Chapter 4). We can condition on the previous SNP sites within the physical distances that govern these clusters, but assume independence from the sites outside of these clusters.

We assume then that the descendant binary sequences follow a Markov chain. Implementing a Markov chain structure to Equation 2.1 produces a product of conditional probabilities, which will be true likelihoods when our dependence assumption is valid, whose product will form a conditional likelihood.

2.3 Markov Chain Composite Likelihood

We will assume that the n descendant binary sequences each follow an order-m Markov Chain. That is, we assume that each x_{s+m} is conditional on the $x_s, \ldots, x_{s+m-1}$ (the m previous sites). Then we take the probability

statement to be equal to

$$P(X = (x_1, \ldots, x_L)) =$$
$$P(x_1, \ldots, x_m)P(x_{m+1}|x_1, \ldots, x_m)P(x_{m+2}|x_2, \ldots, x_{m+1}) \cdots P(x_L|x_{L-m}, \ldots, x_{L-1})$$

which, combined with the formula for conditional probability, allows us to rewrite our new log-likelihood statement to be Equation (2.3).

$$\begin{aligned} \ell(\pi|\mathbf{X}) = & \sum_{x_1,\ldots,x_L} \{n_{1,2,\ldots,L}(x_1, x_2, ..., x_L) \\ & \log[P(X_1, ..., X_m)P(X_{m+1}|X_1, ..., X_m) \cdots P(X_L|X_{L-m}, ..., X_{L-1})]\} \\ = & \sum_{x_1,\ldots,x_L} \{n_{1,2,\ldots,L}(x_1, x_2, ..., x_L) \\ & \log[\frac{f(x_1, ..., x_{m+1}) \cdot f(x_2, ..., x_{m+2}) \cdots f(x_{L-m}, ..., x_L)}{f(x_2, ..., x_{m+1}) \cdots f(x_{L-m}, ..., x_{L-1})}]\} \end{aligned} \tag{2.3}$$

The most significant change following this reformulation is that we are no longer able to directly obtain estimates of the joint distribution, $\pi_{1,\ldots,L}(x_1, \ldots, x_L)$. The numerator of Equation (2.3) contains a product of functions of the $(m+1)$-wise marginal distributions. The denominator contains a product of functions of the m-wise marginal distributions, each of which can be rewritten as a sum involving the $(m+1)$-wise marginal distributions (e.g., $f(x_2, \ldots, x_{m+1}) = \sum_{x_1} f(x_1, \ldots, x_{m+1})$).

Thus, we now aim to obtain estimates of the $(m+1)$-wise marginal distributions from Equation (2.3). This successfully reduces the computational burden of the problem. When $m + 1 < L$, $2^m < 2^{L-1}$ and there are fewer recombination possibilities to contend with within each marginal distribution estimation process. However, this means that we require a method to reconstruct the

joint distribution from these estimated marginal distributions. So, we will assume also that the ancestral binary sequences follow an order-m Markov chain. Then, for each $\hat{\pi}_{1,\ldots,L}(x_1,\ldots,x_L)$,

$$\begin{aligned}&\hat{\pi}_{1,\ldots,L}(x_1,...,x_L) \qquad (2.4)\\ &= \hat{\pi}_{1,\ldots,m}(x_1,...,x_m)\cdot\hat{\pi}_{m+1|1,\ldots,m}(x_{m+1}|x_1,...,x_m)\cdots\hat{\pi}_{L|L-m,\ldots,L-1}(x_L|x_{L-m},...,x_{L-1})\\ &= \frac{\hat{\pi}_{1,\ldots,m+1}(x_1,...,x_{m+1})\hat{\pi}_{2,\ldots,m+2}(x_2,...,x_{m+2})\cdots\hat{\pi}_{L-m,\ldots,L}(x_{L-m},...,x_L)}{\hat{\pi}_{2,\ldots,m+1}(x_2,...,x_{m+1})\cdots\hat{\pi}_{L-m,\ldots,L-1}(x_{L-m},...,x_{L-1})}.\end{aligned}$$

Note that the numerator is composed of a product of the $(m+1)$-wise marginal distributions, and the denominator is composed of the m-wise marginal distributions. Recall that the m-wise marginal distributions can be rewritten as sums of the $(m+1)$-wise marginal distributions: $\pi_{s,\ldots,m}(x_s,\ldots,x_{s+m}) = \sum_{x_{s+m+1}}\pi_{s,\ldots,s+m+1}(x_s,\ldots,x_{s+m+1})$. Therefore, the goal is to estimate these $(m+1)$-wise marginal distributions.

For an illustrative example, consider a sequence length of $L=6$. Suppose that we are interested in the probability that descendant X has the sequence

$$X = (0,1,1,0,1,1).$$

Further, suppose that we have chosen to implement an order-2 Markov Chain for this sequence. Then the probability that we observe descendant X is

$$\begin{aligned}P(X=(0,1,1,0,1,1)) = P((x_1,x_2)=(0,1))&P(x_3=1|x_1=0,x_2=1)\\ &P(x_4=0|x_2=1,x_3=1)P(x_5=1|x_3=1,x_4=0)\\ &P(x_6=1|x_4=0,x_5=1).\end{aligned}$$

Suppose that we now have a sample of descendant sequences. Then the log-likelihood that the ancestor has the sequence $(0, 1, 1, 0, 1, 1)$ is

$$\begin{aligned}\ell(\pi|\mathbf{X}) = n_{1,\ldots,6}(0,1,1,0,1,1)\log\{&P((x_1,x_2)=(0,1))P(x_3=1|x_1=0,x_2=1)\\ &P(x_4=0|x_2=1,x_3=1)P(x_5=1|x_3=1,x_4=0)\\ &P(x_6=1|x_4=0,x_5=1)\}\end{aligned}$$

which will allow us to obtain estimates of $\hat{\pi}_{1,2,3}(0,1,1)$, $\hat{\pi}_{2,3,4}(1,1,0)$, $\hat{\pi}_{3,4,5}(1,0,1)$, and $\hat{\pi}_{4,5,6}(0,1,1)$.

Lastly, we can reconstruct the probability that the ancestor has this sequence, $\pi_{1,\ldots,6}(0,1,1,0,1,1)$ with

$$\hat{\pi}_{1,\ldots,6}(0,1,1,0,1,1) = \frac{\hat{\pi}_{1,2,3}(0,1,1)\hat{\pi}_{2,3,4}(1,1,0)\hat{\pi}_{3,4,5}(1,0,1)\hat{\pi}_{4,5,6}(0,1,1)}{\hat{\pi}_{2,3}(1,1)\hat{\pi}_{3,4}(1,0)\hat{\pi}_{4,5}(0,1)}.$$

Additional barriers remain in using a maximum likelihood estimation approach for these estimates. The first is that we must impose complex parameter space constraints to ensure that estimates are valid probabilities. The second is that each $(m+1)$-wise estimate will be a function of the lower-order marginal distributions, each of which we do not yet have estimates. The estimation process then will proceed as a hierarchical estimator with reparameterization.

2.4 Reparameterization

We first present the three parameter space constraints. The first is the *between zero and one constraint*; we require that for each $s = 1, ..., L - m$ the

estimates probabilities must fall between 0 and 1,

$$0 \leq \pi_{s,\ldots,s+m}(x_s, ..., x_{s+m}) \leq 1.$$

The second is the *sum to one constraint*. We require that for each $s = 1, ..., L - m$ the estimated marginal distributions will sum to 1,

$$\sum_{x_s=0}^{1} \cdots \sum_{x_{s+m}=0}^{1} \pi_{s,\ldots,s+m}(x_s, ..., x_{s+m}) = 1.$$

The third and last is the *lower-order margin consistency property*. This means that for each $s = 1, ..., L - m - 1$, the estimated marginal distributions must satisfy,

$$\begin{aligned}\sum_{x_s=0}^{1} \pi_{s,\ldots,s+m}(x_s, ..., x_{s+m}) &= \sum_{x_{s+m+1}=0}^{1} \pi_{s+1,\ldots,s+m}(x_{s+1}, ..., x_{s+m+1}) \\ &= \pi_{s+1,\ldots,s+m}(x_{s+1}, \ldots, x_{s+m}).\end{aligned}$$

The introduction of these necessary parameter space constraints adds additional complexity to the estimation process. Namely, it reduces the number of unknown parameters that are free parameters. For an $(m + 1)$-wise marginal distribution, there will be 2^{m+1} total unknown parameters, $3 \cdot 2^{m-2}$ constraints, and thus only 2^{m-1} free parameters. The estimation process should then proceed such that we are obtaining estimates for the free parameters as well as obtaining equations for the constrained parameters as functions of the free parameters.

2.4.1 General Case Reparameterization

We will let ϕ_s denote the free parameter at site s. We arbitrarily select the 2^{m-1} free parameters such that each ϕ_s corresponds to a parameter

$$\pi_{s,\ldots,s+m}(0, x_{s+1}, x_{s+2}, \ldots, x_{s+m-2}, x_{s+m-1}, 0).$$

Denote this by $\phi_s(\underline{x})$ where $\underline{x} = (x_{s+1}, x_{s+2}, \ldots, x_{s+m-2}, x_{s+m-1})$ for each x_{s+1}, $x_{s+2}, \ldots, x_{s+m-2}$, $x_{s+m-1} \in \{0, 1\}$. Then the constrained parameters will be each of

$$\begin{cases} \pi_{s,\ldots,s+m}(0, x_{s+1}, x_{s+2}, \ldots, x_{s+m-2}, x_{s+m-1}, 1) \\ \pi_{s,\ldots,s+m}(1, x_{s+1}, x_{s+2}, \ldots, x_{s+m-2}, x_{s+m-1}, 0) \\ \pi_{s,\ldots,s+m}(1, x_{s+1}, x_{s+2}, \ldots, x_{s+m-2}, x_{s+m-1}, 1) \end{cases}$$

giving 2^{m-1} groups of 4 parameters, each with one free parameter and three constrained parameters. We will then calculate each constrained parameter as a function of the free parameter,

$$\begin{cases} \hat{\pi}_{s,\ldots,s+m}(0, \underline{x}, 1) = \hat{\pi}_{s,\ldots,s+m-1}(0, \underline{x}) - \phi_s(\underline{x}) \\ \hat{\pi}_{s,\ldots,s+m}(1, \underline{x}, 0) = \hat{\pi}_{s+1,\ldots,s+m}(\underline{x}, 0) - \phi_s(\underline{x}) \\ \hat{\pi}_{s,\ldots,s+m}(1, \underline{x}, 1) = \hat{\pi}_{s+1,\ldots,s+m-1}(\underline{x}) - \hat{\pi}_{s,\ldots,s+m-1}(0, \underline{x}) - \hat{\pi}_{s+1,\ldots,s+m}(\underline{x}, 0) + \phi_s(\underline{x}). \end{cases}$$

To ensure that each of the 2^{m+1} parameters satisfies the between 0 and 1

constraint, we constrain the the free parameter $\phi_s(\underline{x})$ such that

$$\phi_s(\underline{x}) \in [L_m, U_m] \tag{2.5}$$

where,

$$L_m = \max\{0, \hat{\pi}_{s,\ldots,s+m-1}(0,\underline{x}) + \hat{\pi}_{s+1,\ldots,s+m}(\underline{x},0) - \hat{\pi}_{s+1,\ldots,s+m-1}(\underline{x})\},$$

and,

$$U_m = \min\{\hat{\pi}_{s,\ldots,s+m-1}(0,\underline{x}), \hat{\pi}_{s+1,\ldots,s+m}(\underline{x},0)\}.$$

To build intuition for the derivation of these equations, we will outline explicitly the reparameterizations for the cases where $m = 1, 2, 3$ (which will from now on be referred to as the *pairwise*, *threewise*, and *fourwise* cases respectively).

2.4.2 Pairwise Reparameterization

Consider the case where $m = 1$ and we are calculating pairwise marginal estimates. There will be four unknown parameters in total, and one free parameter. Let $\phi_s = \pi_{s,s+1}(0,0)$. We then need equations for $\pi_{s,s+1}(0,1)$, $\pi_{s,s+1}(1,0)$, $\pi_{s,s+1}(1,1)$ as functions of ϕ_s. To illustrate the derivation of these equations, consider an illustration of the relevant parameter space (Figure 2.1).

Then for the pairwise marginal estimates, the constrained parameters can be rewritten as follows

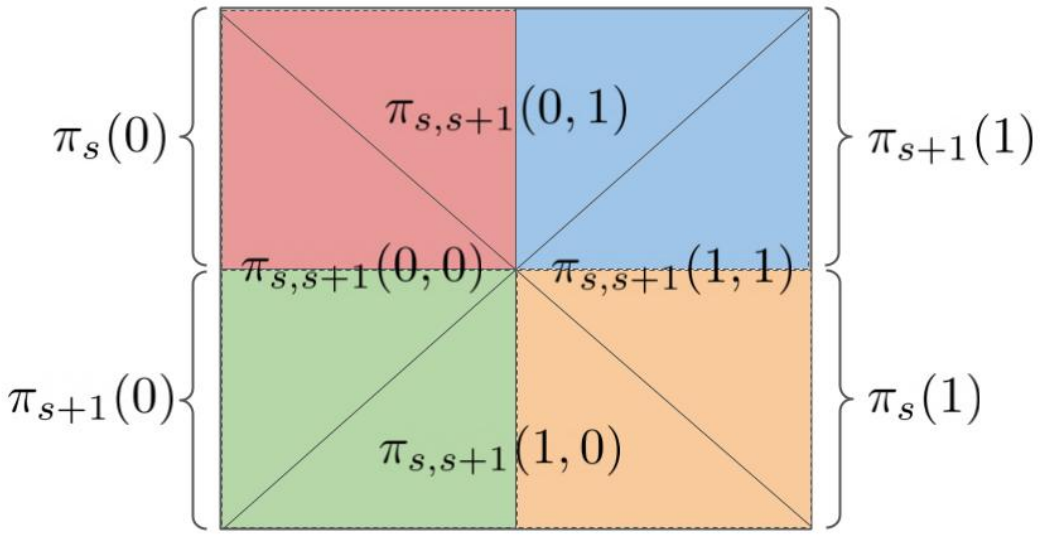

Figure 2.1: Pairwise Parameter Space, site s. We can see in this diagram that the pairwise parameters' spaces consist of the onewise parameters' spaces. For example, the space for $\pi_{s,s+1}(0,1)$ sits inside the space for $\pi_s(0)$ and $\pi_{s+1}(1)$ since $\pi_{s+1}(1) = \sum_{x_s} \pi_{s,s+1}(x_s, 1)$ and $\pi_s(0) = \sum_{x_{s+1}} \pi_{s,s+1}(0, x_{s+1})$. We can also see that some of the pairwise parameters sit inside the same space of a onewise parameter. For example, both $\pi_{s,s+1}(0,0)$ and $\pi_{s,s+1}(0,1)$ sit inside the space of $\pi_s(0)$ since $\pi_s(0) = \sum_{x_{s+1}} \pi_{s,s+1}(0, x_{s+1}) = \pi_{s,s+1}(0,0) + \pi_{s,s+1}(0,1)$. This tells us that we can calculate the value of $\pi_{s,s+1}(0,1)$ by taking the estimate $\hat{\pi}_s(0)$ and subtracting off the estimate $\hat{\pi}_{s,s+1}(0,0)$. Similar arguments can be made for each of the constrained pairwise parameters, giving us equations for each of these parameters in terms of the free parameter ϕ_s.

$$\begin{cases} \hat{\pi}_{s,s+1}(0,1) = \hat{\pi}_s(0) - \phi_s \\ \hat{\pi}_{s,s+1}(1,0) = \hat{\pi}_{s+1}(0) - \phi_s \\ \hat{\pi}_{s,s+1}(1,1) = 1 - \hat{\pi}_s(0) - \hat{\pi}_{s+1}(0) + \phi_s. \end{cases}$$

In addition, to ensure that each pairwise marginal estimate is between 0 and 1, we place the following bounds on ϕ_s,

$$\phi_s \in [L_1, U_1] \tag{2.6}$$

where,

$$L_1 = \max\{0, \hat{\pi}_s(0) - \hat{\pi}_{s+1}(0) - 1\}$$

and

$$U_1 = \min\{\hat{\pi}_s(0), \hat{\pi}_{s+1}(0)\}.$$

2.4.3 Threewise Reparameterization

Consider next the case where $m = 2$ and we are calculating threewise marginal estimates. There will be eight unknown parameters in total, and two free parameters. Let $\phi_s(0) = \pi_{s,s+1,s+2}(0,0,0)$ and $\phi_s(1) = \pi_{s,s+1,s+2}(0,1,0)$. The first group of constrained parameters will be $\pi_{s,s+1,s+2}(0,0,1)$, $\pi_{s,s+1,s+2}(1,0,0)$, and $\pi_{s,s+1,s+2}(1,0,1)$. Let this be Group 1. We then need to derive equations for the three constrained parameters in Group 1 as functions of $\phi_s(0)$. The second group of constrained parameters will be $\pi_{s,s+1,s+2}(0,1,1)$, $\pi_{s,s+1,s+2}(1,1,0)$, and $\pi_{s,s+1,s+2}(1,1,1)$. Let this be Group 2. We then need to derive equations

for the three constrained parameters in Group 2 as functions of $\phi_s(1)$.

This parameter space will involve both the pairwise parameter space and the onewise parameter space. At this dimension, the parameter space is too complex to illustrate. However, the derivation of the equations will follow the same intuition built in the pairwise case. For example, using the lower-order margin consistency property we have that

$$\begin{aligned}\pi_{s,s+1}(0,0) &= \sum_{x_{s+2}} \pi_{s,s+1,s+2}(0,0,x_{s+2}) \\ &= \pi_{s,s+1,s+2}(0,0,0) + \pi_{s,s+1,s+2}(0,0,1)\end{aligned}$$

which tells us that both $\pi_{s,s+1,s+2}(0,0,0)$ and $\pi_{s,s+1,s+2}(0,0,1)$ will sit inside the parameter space of $\pi_{s,s+1}(0,0)$. Similarly,

$$\pi_{s,s+1}(0,0) = \sum_{x_{s+2}} \pi_{s,s+1,s+2}(0,0,x_{s+2})$$

and

$$\begin{aligned}\pi_{s+1,s+2}(0,1) &= \sum_{x_s} \pi_{s,s+1,s+2}(x_s,0,1) \\ &= \pi_{s,s+1,s+2}(0,0,1) + \pi_{s,s+1,s+2}(1,0,1)\end{aligned}$$

so $\pi_{s,s+1,s+2}(0,0,1)$ sits inside both $\pi_{s,s+1}(0,0)$ and $\pi_{s+1,s+2}(0,1)$. Similar arguments can be made for the other threewise parameters.

Then for the threewise marginal estimates, the constrained parameters can be rewritten as follows,

Group 1:

$$\begin{cases} \hat{\pi}_{s,s+1,s+2}(0,0,0) = \phi_s(0) \\ \hat{\pi}_{s,s+1.s+2}(0,0,1) = \hat{\pi}_{s,s+1}(0,0) - \phi_s(0) \\ \hat{\pi}_{s,s+1,s+2}(1,0,0) = \hat{\pi}_{s+1,s+2}(0,0) - \phi_s(0) \\ \hat{\pi}_{s,s+1,s+2}(1,0,1) = \hat{\pi}_{s+1}(0) - \hat{\pi}_{s+1,s+2}(0,0) - \hat{\pi}_{s,s+1}(0,0) + \phi_s(0) \end{cases}$$

Group 2:

$$\begin{cases} \hat{\pi}_{s,s+1,s+2}(0,1,0) = \phi_s(1) \\ \hat{\pi}_{s,s+1.s+2}(0,1,1) = \hat{\pi}_{s,s+1}(0,1) - \phi_s(1) \\ \hat{\pi}_{s,s+1,s+2}(1,1,0) = \hat{\pi}_{s+1,s+2}(1,0) - \phi_s(1) \\ \hat{\pi}_{s,s+1,s+2}(1,1,1) = \hat{\pi}_{s+1}(1) - \hat{\pi}_{s+1,s+2}(1,0) - \hat{\pi}_{s,s+1}(0,1) + \phi_s(1). \end{cases}$$

In addition, to ensure that each threewise marginal estimate is between 0 and 1, we place the following bounds on the free parameters

$$\phi_s(0) \in [L_{2,1}, U_{2,1}], \text{and } \phi_s(1) \in [L_{2,2}, U_{2,2}] \tag{2.7}$$

where,

$$L_{2,1} = \max\{0, \hat{\pi}_{s,s+1}(0,0) + \hat{\pi}_{s+1,s+2}(0,0) - \hat{\pi}_{s+1}(0)\},$$

$$U_{2,1} = \min\{\hat{\pi}_{s,s+1}(0,0), \hat{\pi}_{s+1,s+2}(0,0)\},$$

$$L_{2,2} = \max\{0, \hat{\pi}_{s,s+1}(0,1) + \hat{\pi}_{s+1,s+2}(1,0) - \hat{\pi}_{s+1}(1)\},$$

and,

$$U_{2,2} = \min\{\hat{\pi}_{s,s+1}(0,1), \hat{\pi}_{s+1,s+2}(1,0)\}.$$

2.4.4 Fourwise Reparameterization

Consider next the case where $m = 3$ and we are calculating fourwise marginal estimates. There will be sixteen unknown parameters in total, and four free parameters. Let

$$\phi_s(0,0) = \pi_{s,s+1,s+2,s+3}(0,0,0,0),$$
$$\phi_s(0,1) = \pi_{s,s+1,s+2,s+3}(0,0,1,0),$$
$$\phi_s(1,0) = \pi_{s,s+1,s+2,s+3}(0,1,0,0),$$
$$\phi_s(1,1) = \pi_{s,s+1,s+2,s+3}(0,1,1,0).$$

The first group of constrained parameters will be $\pi_{s,s+1,s+2,s+3}(0,0,0,1)$, $\pi_{s,s+1,s+2,s+3}(1,0,0,0)$, and $\pi_{s,s+1,s+2,s+3}(1,0,0,1)$. Let this be Group 1. We then need to derive equations for the three constrained parameters in Group 1 as functions $\phi_s(0,0)$. The second group of constrained parameters will be $\pi_{s,s+1,s+2,s+3}(0,0,1,1)$, $\pi_{s,s+1,s+2,s+3}(1,0,1,0)$, and $\pi_{s,s+1,s+2,s+3}(1,0,1,1)$. Let this be Group 2. We then need to derive equations for the three constrained parameters in Group 2 as functions of $\phi_s(0,1)$. The third group of constrained parameters will be $\pi_{s,s+1,s+2,s+3}(0,1,0,1)$, $\pi_{s,s+1,s+2,s+3}(1,1,0,0)$, and $\pi_{s,s+1,s+2,s+3}(1,1,0,1)$. Let this be Group 3. We then need to derive equations for the three constrained parameters in Group 3 as functions of $\phi_s(1,0)$. The fourth group of constrained parameters will be $\pi_{s,s+1,s+2,s+3}(0,1,1,1)$, $\pi_{s,s+1,s+2,s+3}(1,1,1,0)$, and $\pi_{s,s+1,s+2,s+3}(1,1,1,1)$. Let this be Group 4. We then need to derive equations for the three constrained parameters in Group 4 as functions of $\phi_s(1,1)$.

The fourwise parameter space will involve the threewise parameter space, the pairwise parameter space, and the onewise parameter space. Thus, as was

the case for the threewise derivations, the parameter space is too complex to illustrate. However, we can once again use the intuition observed in the pairwise case.

For example, using the lower-order margin consistency property we have that

$$\begin{aligned}\pi_{s,s+1,s+2}(0,0,0) &= \sum_{x_{s+3}} \pi_{s,s+1,s+2,s+3}(0,0,0,x_{s+3}) \\ &= \pi_{s,s+1,s+2,s+3}(0,0,0,0) + \pi_{s,s+1,s+2,s+3}(0,0,0,1)\end{aligned}$$

which tells us that both $\pi_{s,s+1,s+2,s+3}(0,0,0,0)$ and $\pi_{s,s+1,s+2,s+3}(0,0,0,1)$ will sit inside the parameter space for $\pi_{s,s+1,s+2}(0,0,0)$. Similarly,

$$\pi_{s,s+1,s+2}(0,0,0) = \sum_{x_{s+3}} \pi_{s,s+1,s+2,s+3}(0,0,0,x_{s+3})$$

and

$$\begin{aligned}\pi_{s+1,s+2,s+3}(0,0,1) &= \sum_{x_s} \pi_{s,s+1,s+2,s+3}(x_s,0,0,1) \\ &= \pi_{s,s+1,s+2,s+3}(0,0,0,1) + \pi_{s,s+1,s+2,s+3}(1,0,0,1)\end{aligned}$$

so $\pi_{s,s+1,s+2,s+3}(0,0,0,1)$ sits inside both $\pi_{s,s+1,s+2}(0,0,0)$ and $\pi_{s+1,s+2,s+3}(0,0,1)$. Similar arguments can be made for the other fourwise parameters.

Then for the fourwise marginal estimates, the constrained parameters can be rewritten as follows,

Group 1:

$$\begin{cases} \hat{\pi}_{s,\ldots,s+3}(0,0,0,0) = \phi_s(0,0) \\ \hat{\pi}_{s,\ldots,s+3}(0,0,0,1) = \hat{\pi}_{s,s+1,s+2}(0,0,0) - \phi_s(0,0) \\ \hat{\pi}_{s,\ldots,s+3}(1,0,0,0) = \hat{\pi}_{s+1,s+2,s+3}(0,0,0) - \phi_s(0,0) \\ \hat{\pi}_{s,\ldots,s+3}(1,0,0,1) = \hat{\pi}_{s+1,s+2}(0,0) - \hat{\pi}_{s+1,s+2,s+3}(0,0,0) - \hat{\pi}_{s,s+1,s+2}(0,0,0) + \phi_s(0,0) \end{cases}$$

Group 2:

$$\begin{cases} \hat{\pi}_{s,\ldots,s+3}(0,0,1,0) = \phi_s(0,1) \\ \hat{\pi}_{s,\ldots,s+3}(0,0,1,1) = \hat{\pi}_{s,s+1,s+2}(0,0,1) - \phi_s(0,1) \\ \hat{\pi}_{s,\ldots,s+3}(1,0,1,0) = \hat{\pi}_{s+1,s+2,s+3}(0,1,0) - \phi_s(0,1) \\ \hat{\pi}_{s,\ldots,s+3}(1,0,1,1) = \hat{\pi}_{s+1,s+2}(0,1) - \hat{\pi}_{s+1,s+2,s+3}(0,1,0) - \hat{\pi}_{s,s+1,s+2}(0,0,1) + \phi_s(0,1) \end{cases}$$

Group 3:

$$\begin{cases} \hat{\pi}_{s,\ldots,s+3}(0,1,0,0) = \phi_s(1,0) \\ \hat{\pi}_{s,\ldots,s+3}(0,1,0,1) = \hat{\pi}_{s,s+1,s+2}(0,1,0) - \phi_s(1,0) \\ \hat{\pi}_{s,\ldots,s+3}(1,1,0,0) = \hat{\pi}_{s+1,s+2,s+3}(1,0,0) - \phi_s(1,0) \\ \hat{\pi}_{s,\ldots,s+3}(1,1,0,1) = \hat{\pi}_{s+1,s+2}(1,0) - \hat{\pi}_{s+1,s+2,s+3}(1,0,0) - \hat{\pi}_{s,s+1,s+2}(0,1,0) + \phi_s(1,0) \end{cases}$$

Group 4:

$$\begin{cases} \hat{\pi}_{s,\ldots,s+3}(0,1,1,0) = \phi_s(1,1) \\ \hat{\pi}_{s,\ldots,s+3}(0,1,1,1) = \hat{\pi}_{s,s+1,s+2}(0,1,1) - \phi_s(1,1) \\ \hat{\pi}_{s,\ldots,s+3}(1,1,1,0) = \hat{\pi}_{s+1,s+2,s+3}(1,1,0) - \phi_s(1,1) \\ \hat{\pi}_{s,\ldots,s+3}(1,1,1,1) = \hat{\pi}_{s+1,s+2}(1,1) - \hat{\pi}_{s+1,s+2,s+3}(1,1,0) - \hat{\pi}_{s,s+1,s+2}(0,1,1) + \phi_s(1,1). \end{cases}$$

To ensure that each fourwise estimate will satisfy the constraint of being between zero and one, we also impose the following bounds on each of the free parameters,

$$\phi_s(0,0) \in [L_{3,1}, U_{3,1}],\ \phi_s(0,1) \in [L_{3,2}, U_{3,2}], \tag{2.8}$$
$$\phi_s(1,0) \in [L_{3,3}, U_{3,3}],\ \text{and},\ \phi_s(1,1) \in [L_{3,4}, U_{3,4}]$$

where,

$$L_{3,1} = \max\{0, \hat{\pi}_{s,s+1,s+2}(0,0,0) + \hat{\pi}_{s+1,s+2,s+3}(0,0,0) - \hat{\pi}_{s+1,s+2}(0,0)\},$$
$$U_{3,1} = \min\{\hat{\pi}_{s,s+1,s+2}(0,0,0), \hat{\pi}_{s+1,s+2,s+3}(0,0,0)\},$$
$$L_{3,2} = \max\{0, \hat{\pi}_{s,s+1,s+2}(0,0,1) + \hat{\pi}_{s+1,s+2,s+3}(0,1,0) - \hat{\pi}_{s+1,s+2}(0,1)\},$$
$$U_{3,2} = \min\{\hat{\pi}_{s,s+1,s+2}(0,0,1), \hat{\pi}_{s+1,s+2,s+3}(0,1,0)\},$$
$$L_{3,3} = \max\{0, \hat{\pi}_{s,s+1,s+2}(0,1,0) + \hat{\pi}_{s+1,s+2,s+3}(1,0,0) - \hat{\pi}_{s+1,s+2}(1,0)\},$$
$$U_{3,3} = \min\{\hat{\pi}_{s,s+1,s+2}(0,1,0), \hat{\pi}_{s+1,s+2,s+3}(1,0,0)\},$$
$$L_{3,4} = \max\{0, \hat{\pi}_{s,s+1,s+2}(0,1,1) + \hat{\pi}_{s+1,s+2,s+3}(1,1,0) - \hat{\pi}_{s+1,s+2}(1,1)\},$$

and,

$$U_{3,4} = \min\{\hat{\pi}_{s,s+1,s+2}(0,1,1), \hat{\pi}_{s+1,s+2,s+3}(1,1,0)\}.$$

The reparameterization can be carried out for values $m > 3$ following the general forms presented in Section 2.4.1.

2.5 Hierarchical Estimator

Recall that the goal is to be able to calculate the $(m+1)$-wise marginal distributions in order to reconstruct the joint ancestral distribution. With the introduction of this reparameterization process, the estimation of these $(m+1)$-wise marginal distributions will involve obtaining estimates of the 2^{m-1} free parameters. To estimate these free parameters, we use order-m MCCL to maximize the log-likelihood for these free parameters within the parameter space (Equation 2.3).

For the pairwise marginals, as an example, the log-likelihood in terms of the free parameter ϕ_s is,

$$\begin{aligned}\ell(\phi_s) &= \sum_{x_s=0}^{1}\sum_{x_{s+1}=0}^{1} n_{s,s+1}\ \log\ f(x_s, x_{s+1}) \qquad (2.9)\\ &= n_{s,s+1}(0,0)\ \log\ \{(1-q)\phi_s + q\hat{\pi}_s(0)\hat{\pi}_{s+1}(0)\}\\ &+ n_{s,s+1}(0,1)\ \log\ \{(1-q)[\hat{\pi}_s(0) - \phi_s] + q\hat{\pi}_s(0)\hat{\pi}_{s+1}(1)\}\\ &+ n_{s,s+1}(1,0)\ \log\ \{(1-q)[\hat{\pi}_{s+1}(0) - \phi_s] + q\hat{\pi}_s(1)\hat{\pi}_{s+1}(0)\}\\ &+ n_{s,s+1}(1,1)\ \log\ \{(1-q)[1 - \hat{\pi}_s(0) - \hat{\pi}_{s+1}(0) + \phi_s] + q\hat{\pi}_s(1)\hat{\pi}_{s+1}(1)\}.\end{aligned}$$

Observe that this log-likelihood depends on the free parameter ϕ_s, the sample values $n_{s,s+1}(x_s, x_{s+1})$, and the onewise marginal estimates.

For the threewise marginals the log-likelihood in terms of the free parame-

ters $\phi_s(0)$ and $\phi_s(1)$ is,

$$\ell(\phi_s(0), \phi_s(1)) = \sum_{x_s} \sum_{x_{s+1}} \sum_{x_{s+2}} n_{s,s+1,s+2}(x_s, x_{s+1}, x_{s+2}) \log f(x_s, x_{s+1}, x_{s+2}) \tag{2.10}$$

where $f(x_s, x_{s+1}, x_{s+2})$ is calculated by Equation 2.1. Note that, as was true of the pairwise log-likelihood, the threewise log-likelihood will depend on the free parameters, sample values, and lower-order marginal estimates– both pairwise and onewise marginal estimates.

For the fourwise marginals the log-likelihood in terms of the free parameters $\phi_s(0,0)$, $\phi_s(0,1)$, $\phi_s(1,0)$, and $\phi_s(1,1)$ is,

$$\ell(\phi_s(0,0), \phi_s(0,1), \phi_s(1,0), \phi_s(1,1)) = \tag{2.11}$$
$$\sum_{x_s} \cdots \sum_{x_{s+3}} n_{s,\ldots,s+3}(x_s, \ldots, x_{s+3}) \log f(x_s, \ldots, x_{s+3})$$

where $f(x_s, \ldots, x_{s+3})$ is calculated by Equation 2.1. Note that the fourwise log-likelihood will depend on the free parameters, sample values, and lower-order marginal estimates – onewise, pairwise, and threewise marginal estimates.

Furthermore, it can be shown that each corresponding order-m log MCCL is separable. In particular, it can be separated into terms for each free parameter $\phi_s(\underline{x})$. The free parameters being separable in the log likelihood function simplifies the process of maximizing the likelihood since each free parameter can be maximized separately. Then the separate log likelihood statements is a

sum of terms as follows,

$$\ell(\phi_s(\underline{x})) = \sum_{x_s} \cdots \sum_{x_{s+m}} n_{s,\underline{x},s+m}(x_s, \underline{x}, x_{s+m}) \log f(x_s, \underline{x}, x_{s+m}) \tag{2.12}$$

where $f(x_s, \underline{x}, x_{s+3})$ is calculated by Equation 2.1.

The question then becomes how to maximize these log Markov chain composite likelihood statements following the form of Equation 2.12. Jianping Sun 2011 focuses on hierarchical estimation. This is a quick, though possibly inefficient, method to maximize the Markov chain composite likelihood equations for each free parameter. In this method we will begin by estimating marginal ancestral distributions for small margins then use these fixed estimates to obtain estimates for the desired marginal ancestral distribution and reconstruct the joint distribution. We first estimate the onewise marginal ancestral distribution. We then fix these onewise estimates and estimate the pairwise marginal ancestral distribution given these fixed estimates. The pairwise and onewise marginal estimates are then fixed, and the threewise marginal ancestral distribution is estimated given these fixed estimates. We follow this process hierarchically until we have estimated the desired $(m+1)$-wise marginal distribution. This hierarchical structure follows naturally from our observation that in Equations 2.9, 2.10, and 2.11 there is a dependence on the lower-order marginal estimates.

How the $(m+1)$-wise marginal estimates are obtained depends on the current value of m. These different processes fall into three categories: $m = 0$, $m = 1$, and $m \geq 2$.

2.5.1 Order-0 Markov Chain

We have given little consideration to the case where $m = 0$ up to this point due to the fact that the joint cannot be reconstructed from the onewise marginal distributions alone. However, we have seen but not explicitly noted that we will need to obtain these onewise marginal estimates; for example, to calculate the constrained parameters and free parameter bounds for the pairwise marginals we must have fixed onewise marginal estimates.

In this case, there are two unknown parameters for site s: $\pi_s(0)$ and $\pi_s(1)$. The log MCCL for the onewise marginal estimates can be simply maximized by the sample proportions:

$$\hat{\pi}_s(x_s) = \frac{n_s(x_s)}{n} \tag{2.13}$$

which will automatically satisfy all parameter space constraints. We fix these onewise estimates for use in the higher order marginal estimates.

2.5.2 Order-1 Markov Chain

Recall that, for the pairwise marginal case, we need to estimate the parameter ϕ_s by maximizing Equation 2.9. This can be accomplished by solving for ϕ_s in the score equation,

$$\frac{d}{d\phi_s}\ell(\phi_s) = 0.$$

It can be shown that $\ell(\phi_s)$ is a concave function with respect to ϕ_s (see Jianping Sun 2011 for proof). Therefore there will exist a local maximum for

$\ell(\phi_s)$. It can be shown that the score equation achieves zero when,

$$\hat{\phi}_s = \frac{1}{1-q}\left[\frac{n_{s,s+1}(0,0)}{n} - q\hat{\pi}_s(0)\hat{\pi}_{s+1}(0)\right]. \tag{2.14}$$

However, this MLE will not necessarily give an estimate that satisfies the bounds placed on ϕ_s in Equation (2.6). It can be shown that in the case where $\hat{\phi}_s$ is not in the interval defined in (2.6), the desired estimate should be the interval bound that is closest to $\hat{\phi}_s$ (For the proof of this property, see Jianping Sun 2011).

Then the hierarchical estimator for the order-1 Markov Chain can be summarized as follows,

1. Fix onewise margins according to Equation (2.13) for $s = 1, \ldots, L$.
2. Calculate the free parameter $\hat{\phi}_s$ using Equation (2.14).
3. Determine if $\hat{\phi}_s$ is in the interval defined in Equation (2.6). If it is in the interval, then $\hat{\phi}_s$ is the estimate. If it is not in the interval, then the estimate is the interval bound closest to $\hat{\phi}_s$.
4. Calculate the constrained parameters, $\pi_{s,s+1}(0,1)$, $\pi_{s,s+1}(1,0)$, and $\pi_{s,s+1}(1,1)$, through reparameterization.

2.5.3 Order-m Markov Chain, $m \geq 2$

Order-2 Markov Chain

Similarly, for the threewise marginal case, we aim to estimate $\phi_s(0)$ and $\phi_s(1)$ by maximizing Equation (2.10). We know that these two parameters are

separable, and therefore

$$\ell(\phi_s(0), \phi_s(1)) = \ell(\phi_s(0)) + \ell(\phi_s(1))$$

where

$$\ell(\phi_s(i)) = \sum_{x_s} \sum_{x_{s+2}} [n_{s,s+1,s+2}(x_s, i, x_{s+2}) \log f(x_s, x_{s+1} = i, x_{s+2})] \,.$$

Then for each $\phi_s(i), i \in \{0, 1\}$ we solve the score equation

$$\frac{d}{d\phi_s(i)} \ell(\phi_s(i)) = 0.$$

It can be shown that $\frac{d^2}{d\phi_s(i)^2} \leq 0$ which implies that the score equation is concave and thus that the solution to the score equation will be a local maximum of $\ell(\phi_s(i))$. (For a proof of this property, see Jianping Sun 2011). Then it can be shown that the solution to the score equation is,

$$\hat{\phi}_s(i) = \frac{1}{(1-q)^2} \Bigg[\frac{n_{s,s+1,s+2}(0,i,0)}{n} - \quad q(1-q)[\hat{\pi}_s(0)\hat{\pi}_{s+1,s+2}(i,0) + \hat{\pi}_{s,s+1}(0,i)\hat{\pi}_{s+2}(0)] \tag{2.15}$$

$$- q^2 \hat{\pi}_s(0)\hat{\pi}_{s+1}(i)\hat{\pi}_{s+2}(0) \Bigg]$$

where $\hat{\pi}_s(x_s)$ is calculated by Equation (2.13) and $\hat{\pi}_{s,s+1}(x_s, x_{s+1})$ is calculated by Equation (2.14).

However, Equation (2.15) is not a solution forced by constraint. If the $\hat{\phi}_s$ from Equation (2.14) is not in the parameter space defined in Equation (2.6) and we take the closest bound as our estimate of this free parameter, then

Equation (2.15) will have no solution. Therefore, the Order-1 Markov Chain Hierarchical Estimator process is not generalizable to the Order-2 Markov Chain Hierarchical Estimator case.

However, Jianping Sun 2011 shows that the score equation can be equivalently solved by finding the roots of a cubic equation of $\phi_s(i)$,

$$A_1^3\phi_s(i)^3 + A_2^3\phi_s(i)^2 + A_3^3\phi_s(i) + A_4^3 = 0. \tag{2.16}$$

See Section 7.1 for explicit forms of $A_1^3, A_2^3, A_3^3, A_4^3$,.

For any cubic equation with real coefficients, there will exist either one real root and two conjugate complex roots, or three real roots. It can be shown that, if Equation (2.16) has three real roots, only one of these real roots is eligible to be a candidate estimate for $\phi_s(i)$ (i.e. $f(x_s, i, x_{s+2}) \geq 0$ and $x_{s+1} \in \{0, 1\}$) and this will be the middle root of the three roots (Jianping Sun 2011).

We must also ensure that the candidate estimate satisfies the bounds defined in Equation 2.7. Jianping Sun 2011 shows that, if the candidate estimate is not within the specified bounds, then the estimate should be taken to be the closest interval bound.

Therefore, the hierarchical estimator for the Order-2 Markov Chain can be summarized as follows,

1. Fix onewise margins according to Equation (2.13) for $s = 1, \ldots, L$. Fix pairwise margins according to Equation (2.14) for $s = 1, \ldots, L-1$.

2. Find all of the real roots of Equation (2.16).

3. If there is only one real root, then this is the candidate estimate of $\phi_s(i)$. If there are three real roots, then the middle root is the candidate estimate of $\phi_s(i)$

4. Determine if $\hat{\phi}_s(i)$ is in the interval defined in Equation (2.7). If it is in the interval, then $\hat{\phi}_s(i)$ is the estimate. If it is not in the interval, then the estimate is the interval bound closest to $\hat{\phi}_s(i)$.

5. Calculate the constrained parameters through reparameterization.

Order-3 Markov Chain

For the fourwise marginal case, we aim to estimate $\phi_s(0,0)$, $\phi_s(0,1)$, $\phi_s(1,0)$, and $\phi_s(1,1)$ by maximizing Equation (2.11). Since the parameters are separable, we take

$$\begin{aligned}\ell(\phi_s(0,0), \phi_s(0,1), \phi_s(1,0), \phi_s(1,1)) =& \ell(\phi_s(0,0)) + \ell(\phi_s(0,1)) \\ &+ \ell(\phi_s(1,0)) + \ell(\phi_s(1,1))\end{aligned}$$

where

$$\ell(\phi_s(i,j) = \sum_{x_s} \sum_{x_{s+3}} [n_{s,s+1,s+2,s+3}(x_s, i, j, x_{s+3}) \log f(x_s, i, j, x_{s+3})]$$

for $i, j \in \{0, 1\}$.

Then for each $i, j \in \{0, 1\}$ we solve the score equation

$$\frac{d}{d\phi_x(i,j))} \ell(\phi_s(i,j) = 0.$$

It can be shown that $\frac{d^2}{d\phi_s(i,j)^2} \leq 0$ which implies that the score equation is concave and thus that the solution to the score equation will be a local maximum of $\ell(\phi_s(i,j))$. As occurred in the threewise margin case, using conditional maximum likelihood estimation becomes problematic under conditions where we take our estimates of lower order margin estimates through a process other than MLE (such as taking an interval bound). Therefore, this approach will not be tenable for the fourwise margin case. Like the threewise margin case, however, the score equation can be equivalently solved by finding the roots of a cubic equation of $\phi_s(i,j)$,

$$A_1^4 \phi_s(i,j)^3 + A_2^4 \phi_s(i,j)^2 + A_3^4 \phi_s(i,j) + A_4^4 = 0. \tag{2.17}$$

See Section 7.2 for explicit forms of A_1^4, A_2^4, A_3^4, A_4^4

By the same arguments used in the threewise margin case, we choose the candidate estimate from the roots of Equation (2.17) by either taking the one real root, or the middle root of the three real roots. Additionally, we constrain this estimate to the interval given in Equation 2.8 in the same way as described

above. Therefore, the hierarchical estimator for the Order-3 Markov Chain can be summarized as follows,

1. Fix onewise margins according to Equation (2.13) for $s = 1, \ldots, L$. Fix pairwise margins according to Equation (2.14) for $s = 1, \ldots, L-1$. Fix threewise margins using the roots of Equation (2.16) for $s = 1, \ldots, L-2$.

2. Find all of the real roots of Equation (2.17).

3. If there is only one real root, then this is the candidate estimate of $\phi_s(i,j)$. If there are three real roots, then the middle root is the candidate estimate of $\phi_s(i,j)$

4. Determine if $\hat{\phi}_s(i,j)$ is in the interval defined in Equation (2.8). If it is in the interval, then $\hat{\phi}_s(i,j)$ is the estimate. If it is not in the interval, then the estimate is the interval bound closest to $\hat{\phi}_s(i,j)$.

5. Calculate the constrained parameters through reparameterization.

Order-m Markov Chain

The hierarchical estimator outlined above for the threewise and fourwise marginal cases can be directly generalized to estimate any $(m+1)$-wise margins where $m \geq 2$.

We aim to estimate each $\phi_s(\underline{x})$ by maximizing Equation (2.12). Since the parameters are separable, for each $\phi_s(\underline{x})$ we solve the score equation

$$\frac{d}{d\phi_x(\underline{x})}\ell(\phi_s(\underline{x})) = 0.$$

It can be shown that $\frac{d^2}{d\phi_x(\underline{x})^2}\ell(\phi_s(\underline{x})) \leq 0$ indicating that the score equation is concave and thus that the solution to the score equation will be a local maximum of $\ell(\phi_s(\underline{x}))$.

This can be solved equivalently by solving a cubic equation of $\phi_s(\underline{x})$,

$$A_1^m\phi_s(\underline{x})^3 + A_2^m\phi_s(\underline{x})^2 + A_3^m\phi_s(\underline{x}) + A_4^m = 0. \tag{2.18}$$

See Section 7.3 for explicit forms of A_1^m, A_2^m, A_3^m, A_4^m.

We choose the candidate estimate from the roots of Equation (2.18) by either taking the one real root, of the middle of the three real roots as described above. We additionally constrain the candidate estimate to the interval specified in Equation 2.5 in the same method described above. Therefore, the hierarchical estimator for the Order-m Markov Chain where $m \geq 2$ can be summarized as follows,

1. Fix estimates for all the lower order marginal ancestral distributions.
2. Find all of the real roots of Equation (2.18).
3. If there is only one real root, then this is the candidate estimate of $\phi_s(\underline{x})$. If there are three real roots, then the middle root is the candidate estimate of $\phi_s(\underline{x})$
4. Determine if $\hat{\phi}_s(\underline{x})$ is in the interval defined in Equation (2.5). If it is in the interval, then $\hat{\phi}_s(\underline{x})$ is the estimate. If it is not in the interval, then the estimate is the interval bound closest to $\hat{\phi}_s(\underline{x})$.
5. Calculate the constrained parameters through reparameterization.

These processes of hierarchical estimation are used to obtain estimates of the selected $(m+1)-$ wise marginal ancestral distribution. These estimates are then used to reconstruct the joint ancestral distribution by Equation (2.4).

Chapter 3

Simulation Design

What remains unknown at the conclusion of Jianping Sun 2011 is how the Markov chain composite likelihood hierarchical estimation for the Recombination Model performs estimating a known, true ancestor distribution of possible binary sequences. Concerns include the effect of reformulating as Markov chain composite likelihood relative to a classical likelihood estimation approach. It has been shown that maximum composite likelihood estimates are consistent, but involve a loss of efficiency relative to the full likelihood (Xu and N. Reid 2011). In addition, the use of Markov chain composite likelihood means that we are concerned both with the estimator performance for the marginal distribution estimates and the joint distribution estimates. The impact of selected fixed quantities, the recombination probability q and the Markov chain order m, on estimator performance is also an open question.

Structure for this simulation design comes from the ADEMP approach proposed by Morris, White, and Crowther 2019. In particular, we focus on their recommendations for simulation design and presentation of results for numerical analysis assessing estimator performance.

3.1 Aims of Simulation

This simulation will be conducted by simulating a true ancestor distribution and comparing the estimated probabilities obtained through the estimation process proposed by Jianping Sun 2011 to these true probabilities. We will consider a successful comparison to be a small bias between the average estimated probability and the true probability for a given sequence, small empirical standard error across replications, and a sufficient portion of the density of the joint distribution being assigned to sequences which have a non-zero probability on the true ancestor distribution. Furthermore, we aim to examine quantities in the model which may impact performance of the estimator. These will include the length of the SNP haplotype sequence taken for each descendent L, the selected Markov chain order m, and the fixed recombination probability q. Finally, we aim to assess whether the estimator's performance is impacted by the use of Markov chain composite likelihood. In the present analysis, we are unable to compare the estimates produced from MCCL with estimates produced by MLE, so we instead look for potential effects of modeling choices made in the implementation of MCCL. In particular, the Markov chain was chosen to condition from left-to-right, so we will examine whether this choice impacts estimates through a directional effect.

For example, suppose we take an order-2 Markov chain in our implementation. Then we would condition site 10 on sites 8 and 9, site 11 on sites 9 and 10, site 12 on site 10 and 11, and so on and so forth. As previously discussed, there is biological motivation for including a dependence structure on these SNP sites. Previous literature has shown that there is dependence between SNP sites governed by physical distance. However, this does not nec-

essarily indicate a left-to-right dependence. This directional dependence was chosen for its beneficial mathematical and computational properties. In reality, ordering of SNP sites on the haplotype depends on the physical orientation of the haplotype during sequencing. The dependence between SNP sites governed by physical distance has been shown to occur in clusters, which could further indicate that the dependence being on the m sites before and after site s rather than the m sites before only would be more biologically accurate (but more mathematically intensive) (Lee 2016). Therefore, we will assess whether there is a detectable directional effect on estimates as a result of this modeling choice. This potential directional effect could include the accumulation of bias or standard error for sites further right on the chain.

3.2 Data-Generating Mechanism

The general structure for our data-generating mechanism will be a resampling study. International HapMap Project data will be used to simulate a true ancestor distribution. Samples of 100 descendants will be simulated from this true ancestor distribution.

As was previously introduced, the International HapMap Project is a source of publicly available SNP data for both genotypes and haplotypes from 11 global ancestry groups for trios (two parents and a child), duos (one parent and a child), and unrelated individuals (International HapMap Project, 2011). The phased data files removed the children's haplotypes, giving us a file of unrelated haplotypes for use in simulating an ancestor distribution. One global ancestry group sampled was the Yoruba people from Ibadan, Nigeria (YRI). This ancestry group was selected for its large sample size after phasing. In

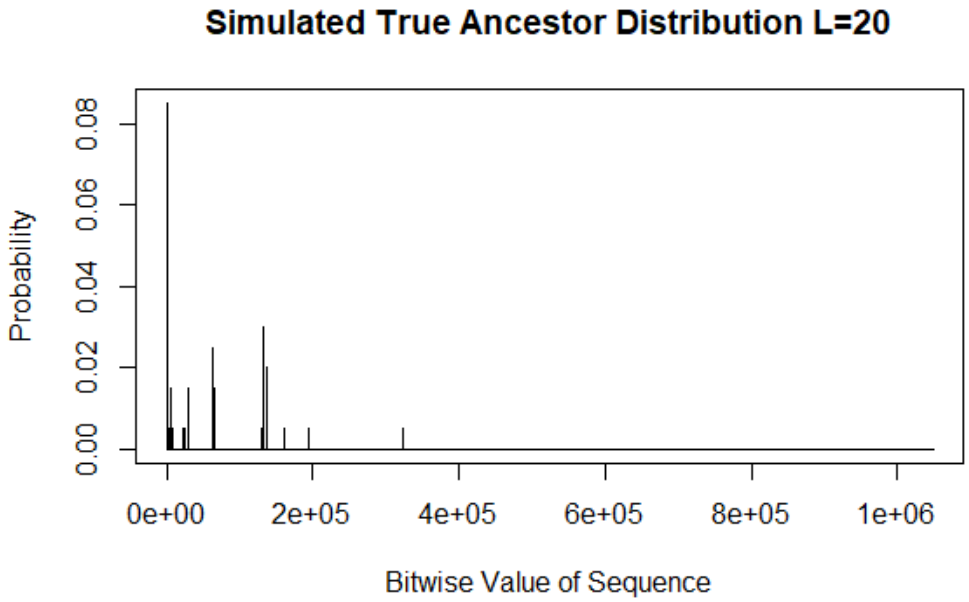

Figure 3.1: For the purposes of this analysis, we will use the following as our Simulated True Ancestor distribution. The Bitwise Value of each of the 2^20 possible sequences was calculated as $(x_1, \ldots, x_L) = \sum_{s=1}^{L} x_s \cdot 2^{L-s}$. Therefore, the possible sequences take on bitwise values from 0 to $2^L - 1$ where $(0, 0, \ldots, 0, 0)$ has bitwise value of 0 and $(1, 1, \ldots, 1, 1)$ has the greatest bitwise value. Furthermore, for each sequence we simulated it's probability to be it's sample proportion within the YRI Trios International HapMap Project data. This gave us 91 sequences with a true non-zero probability. See 7.4 for a table of these probabilities.

particular, we select 200 haplotypes from the YRI population trios data. Using major and minor alleles, these haplotypes were re-coded as binary sequences where 0 is the major allele.

For this simulation, we will use a sequence length of $L = 20$. So our simulated true ancestor distribution will be comprised of $2^{20} = 1,048,576$ possible binary sequences. Then for each sequence $(x_1, x_2, \ldots, x_{19}, x_{20})$, $x_s \in \{0, 1\}$ for $s \in \{1, 2, \ldots, 19, 20\}$, the probability $\pi_{1,2,\ldots,19,20}(x_1, x_2, \ldots, x_{19}, x_{20})$ will be taken as the sample proportion in the selected HapMap data (Figure 3.1). This

resulted in a true distribution that has 91 sequences with non-zero probabilities (see Section 7.4 for a table of these true non-zero probabilities).

We then generate samples of descendant sequences, also of length 20, from this true ancestor distribution. These samples will each contain 100 descendant sequences. For a given descendant sequence, we begin by randomly selecting one of the possible ancestor sequences as a "starting point." The probability of a given ancestor sequence being chosen will be its simulated probability on the true ancestor distribution. Then, for each of the 19 locations on that sequence where a recombination event might occur, we take a draw from a Bernoulli distribution where the success probability is the probability of recombination, q. If the draw is a 1, then we simulate a recombination event occurring at that location by replacing the rest of the sequence with that portion of another randomly selected ancestor sequence. This process is repeated for each simulated descendant sequence.

For example, suppose that we are simulating a descendant sequence of length $L = 10$ and a recombination probability fixed at $q = 0.01$. Assume that we randomly select the starting point sequence to be

$$(0, 0, 0, 0, 1, 0, 0, 1, 0, 0).$$

Then we are taking 9 draws from a Bernoulli distributions with parameter $p = q = 0.01$ to determine whether there are recombination events. Assume that our first two draws are 0, but our third draw is a 1. Then we randomly select another possible sequence, such as

$$(0, 0, 1, 1, 0, 0, 1, 1, 0, 1).$$

Starting from site 4, we replace $(x_4, x_5, \ldots, x_9, x_{10})$ in the first randomly selected sequence with those sites from the second randomly selected sequence. Then our descendant, assuming the remaining 6 draws are 0, will be

$$(0, 0, 0, \mathbf{1}, \mathbf{0}, \mathbf{0}, \mathbf{1}, \mathbf{1}, \mathbf{0}, \mathbf{1}).$$

We will conduct 100 replications for each simulation, generating a sample of 100 descendants in this same manner for each replication.

3.3 Estimand Targeted in Analysis

The samples of descendant samples will be used to estimate the ancestor distribution following the hierarchical estimation process proposed by Jianping Sun 2011 and outlined in Chapter 2. In particular, we will carry out this estimation for three different values of Markov chain order: $m = 1, 2, 3$. These correspond to pairwise marginal estimates, threewise marginal estimates, and fourwise marginal estimates respectively. We will examine the estimates for the free parameters in each of these three cases– $\hat{\phi}_s$ for the pairwise marginals, $\hat{\phi}_s(0)$ and $\hat{\phi}_s(1)$ for the threewise marginals, and $\hat{\phi}_s(0,0)$, $\hat{\phi}_s(0,1)$, $\hat{\phi}_s(1,0)$, and $\hat{\phi}_s(1,1)$ for the fourwise marginals. These will be the marginal estimates targeted in this analysis. We will compare these estimates to the respective true values calculated from the true ancestor distribution.

We will also be targeting the joint distribution estimates reconstructed from each of these three sets of marginal estimates according to Equation (2.4). This will be a total of 1,048,576 estimates. To narrow down the number of estimates to analyze, we will look specifically at the estimates for the 91 sequences as-

signed a true non-zero probability. See Section 7.4 for these simulated probabilities.

3.4 Methods

For our simulations, we will be holding fixed the number of descendants in the sample, n, and the SNP sequence length, L. We will be using samples of $n = 100$ and a sequence length of $L = 20$. We will be varying two parameters: the selected Markov chain order, m, and the selected recombination probability, q.

Theoretical derivations tell us that we should expect efficiency to increase as we increase the Markov chain order (Xu and N. Reid 2011; Jianping Sun 2011). Therefore, we would ideally be able to take relatively large values of m compared to our implemented sequence length. We are, however, limited by the intensive nature of the programming. As a result of the hierarchical estimation, to carry out the estimation for a given Markov chain order m, we must also have programmed functions that output estimates for $m = 0, 1, \ldots, m-1$. Therefore, for this set of simulations we will focus on three Markov chain order values $m = 1, 2, 3$.

We will also select values for q to implement in the simulations, and vary these values factorially with the selected Markov chain order values. The recombination probabilities used should be biologically realistic, as well as capturing the full range of values that may arise so that we can assess whether the method is robust to extreme small or large probabilities.

Several methods have been implemented to analyze meiotic recombination and estimate the probability of recombination events. "Low-resolution" meth-

ods include direct visualization of recombination intermediates analysis of genetic linkage maps, and "high resolution" methods include sperm genotyping and the reconstruction of high resolution recombination rate profiles from genetic variation data. High-resolution methods are able to capture individual functional units, recombination hotspots, where the probability of recombination is higher on some regions of the genome. Differences in resolution between methods means that there is a significant degree of variability in reported recombination probabilities (Khil and Camerini-Otero 2009). Philips and Milo 2015 reports that the average rate of recombination is about one in one hundred per generation for every million base pairs. The variation in the rate of recombination has also been shown to scale inversely with the length of the genome, and that the rate of recombination tends to be about 50% higher in human females than males. This gives a "rule of thumb" that, across species there will be one or two recombination events per chromosome per replication (Philips and Milo 2015). Relative to this average, research suggests that recombination probability is not uniform across the human genome. "Hotspots" are thought to arise from recombination probabilities being higher in chromosomes that have not seen significant alteration throughout the course of human evolution, and "coldspots" are thought to arise from the process of recombination suppression which may play a role in speciation through the reduction of gene flow (Farré, Micheletti, and Ruiz-Herrera 2013). Providing evidence for an upper bound is the idea that for genes that are close together on the same chromosome and do not assort independently, called linked genes, the recombination frequency will be less than 0.50 (Wikipedia 2021a). Based on this evidence we want to capture recombination probabilities around 1% and not exceeding 50%.

We select four values for the recombination probability $q = 0.005, 0.01, 0.05, 0.1$. We will vary these values factorially with the chosen Markov chain order values $m = 1, 2, 3$ giving us 12 simulations (see Table 3.1).

Simulation	m**-value**	q**-value**
1	1	0.005
2	1	0.01
3	1	0.05
4	1	0.1
5	2	0.005
6	2	0.01
7	2	0.05
8	2	0.1
9	3	0.005
10	3	0.01
11	3	0.05
12	3	0.1

Table 3.1: For this analysis, we will be running twelve simulations in total as we will be implementing three levels of m and four levels of q.

3.5 Performance Measures

Morris, White, and Crowther 2019 analyzed a sample of articles wherein a simulation study was implemented to evaluate statistical methods. They report that, among those examining the performance of an estimator, 92% reported the bias, 62% reported the coverage of confidence intervals, 50% reported the empirical standard error, and 35% reported the mean squared error. We will be using bias, empirical standard error, and mean squared error to assess the accuracy of the estimator relative to our simulated true ancestor distribution. The asymptotic distribution of the estimator proposed by Jianping Sun 2011 is at this point unknown, so we are unable to consider coverage of traditional para-

metric confidence intervals in the present study. We will instead implement 95% bootstrapped confidence intervals to further assess estimator accuracy. Bootstrap confidence intervals were chosen to allow for potentially non-symmetric intervals.

In addition to the accuracy of the estimated probabilities which will be encompassed by these measures, we also want to assess the accuracy of the particular sequences that are being assigned a non-zero probability. For example, with the reconstruction of the joint distribution there may be conflation between

$$(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)$$

and

$$(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1)$$

given that our simulated true joint distribution has 1,048,576 total possible sequences but only 91 have been assigned a non-zero probability. We will examine this by looking at the proportion of the total density that is assigned to the sequences with a true non-zero probability and their neighboring or similar sequences. Which sequences qualify as neighbors will be determined using measures of Hamming distance.

3.5.1 Bias

We will calculate the bias of our estimates across the 100 replications implemented in the simulations to assess how close on average the estimated probabilities are to the simulated true probabilities. Ideally, the estimator will show minimal bias.

We will have estimates of the joint distribution from an order-1 Markov chain reconstruction, order-2 Markov chain reconstruction, and an order-3 Markov chain. For the joint estimates from each of these reconstructions, we will calculate the bias relative to the true ancestor distribution. The expectation of the estimated probability is calculated as the mean from 100 replications,

$$Bias(\hat{\pi}_{1,\ldots,20}(x_1,\ldots,x_{20})) = \mathrm{E}(\hat{\pi}_{1,\ldots,20}(x_1,\ldots,x_{20})) - \pi_{1,\ldots,20}(x_1,\ldots,x_{20}).$$

The joint distribution includes 1,048,576 possible sequences, so for ease of interpretation we will focus on the 91 sequences which have been assigned a true non-zero probability. See Section 7.4 for a table of these sequences and their corresponding true probabilities.

We will also have estimates of the pairwise marginal distribution, threewise marginal distribution, and fourwise marginal distribution. To calculate the bias for our marginal estimates, we must calculate the true marginal probabilities. For ease of interpretation, we will focus on the estimates of the free parameters $\phi_s(\underline{x})$; the remaining parameters are functions of these estimates. This will aid in interpretation because, even for the pairwise estimates where there are only four possible sequences, we have estimates for 19 sites giving 76 estimates in total. Looking only at the free parameter reduces this to 19 estimates for the pairwise case, 36 estimates for the threewise case, and 68 estimates for the fourwise case.

For the pairwise case, we calculate the true values of $\phi_s = \pi_{s,s+1}(0,0)$ by taking the sum of the probabilities on each sequence on the joint distribution

where $x_s = 0$ and $x_{s+1} = 0$,

$$\pi_{s,s+1}(0,0) = \sum_{x_s} \cdots \sum_{x_{s-1}} \sum_{x_{s+2}} \cdots \sum_{x_{20}} \pi_{1,\ldots,s,s+1,\ldots,20}(x_1, \ldots, 0, 0, \ldots, x_{20})$$

for $s = 1, \ldots, 19$. See Table 7.2 in Appendix for these calculated probabilities. Consequently, we calculate the bias across the 100 replications using these true pairwise marginal probabilities as,

$$Bias(\hat{\phi}_s) = \mathrm{E}(\hat{\phi}_s) - \phi_s.$$

For the threewise case, we calculate the true values of $\phi_s(0) = \pi_{s,s+1,s+2}(0,0,0)$ and $\phi_s(1) = \pi_{s,s+1,s+2}(0,1,0)$ by taking the sum of the probabilities on each sequence on the joint distribution where $x_s = 0, x_{s+1} = 0, x_{s+2} = 0$ or $x_s = 0, x_{s+1} = 1, x_{s+2} = 0$ respectively:

$$\pi_{s,s+1,s+2}(0,0,0) = \sum_{x_s} \cdots \sum_{x_{s-1}} \sum_{x_{s+3}} \cdots \sum_{x_{20}} \pi_{1,\ldots,s,s+1,s+2\ldots,20}(x_1, \ldots, 0, 0, 0 \ldots, x_{20})$$

$$\pi_{s,s+1,s+2}(0,1,0) = \sum_{x_s} \cdots \sum_{x_{s-1}} \sum_{x_{s+3}} \cdots \sum_{x_{20}} \pi_{1,\ldots,s,s+1,s+2\ldots,20}(x_1, \ldots, 0, 1, 0 \ldots, x_{20})$$

for $s = 1, \ldots, 18$. See Table 7.3 in Appendix for these calculated probabilities. Consequently, we calculate the bias across the 100 replications using the calculated true threewise marginal probabilities as,

$$Bias(\hat{\phi}_s(i)) = \mathrm{E}(\hat{\phi}_s(i)) - \phi_s(i).$$

For the fourwise case, we calculate the true values

1. $\phi_s(0,0) = \pi_{s,s+1,s+2,s+3}(0,0,0,0)$,

2. $\phi_s(0,1) = \pi_{s,s+1,s+2,s+3}(0,0,1,0)$,

3. $\phi_s(1,0) = \pi_{s,s+1,s+2,s+3}(0,1,0,0)$, and

4. $\phi_s(1,1) = \pi_{s,s+1,s+2,s+3}(0,1,1,0)$

by taking the sum of the probabilities on each sequence on the joint distribution where (**1**) $x_s = 0, x_{s+1} = 0, x_{s+2} = 0, x_{s+3} = 0$ for $\phi_s(0,0)$, (**2**) $x_s = 0, x_{s+1} = 0, x_{s+2} = 1, x_{s+3} = 0$ for $\phi_s(0,1)$, (**3**) $x_s = 0, x_{s+1} = 1, x_{s+2} = 0, x_{s+3} = 0$ for $\phi_s(1,0)$, and (**4**) $x_s = 0, x_{s+1} = 1, x_{s+2} = 1, x_{s+3} + 0$ for $\phi_s(1,1)$. This gives,

$$\pi_{s,s+1,s+2,s+3}(0,0,0,0) = \sum_{x_s} \cdots \sum_{x_{s-1}} \sum_{x_{s+4}} \cdots \sum_{x_{20}} \pi_{1,\ldots,s,s+1,s+2,s+3\ldots,20}(x_1,\ldots,0,0,0,0,\ldots,x_{20})$$

$$\pi_{s,s+1,s+2,s+3}(0,0,1,0) = \sum_{x_s} \cdots \sum_{x_{s-1}} \sum_{x_{s+4}} \cdots \sum_{x_{20}} \pi_{1,\ldots,s,s+1,s+2,s+3\ldots,20}(x_1,\ldots,0,0,1,0\ldots,x_{20})$$

$$\pi_{s,s+1,s+2,s+3}(0,1,0,0) = \sum_{x_s} \cdots \sum_{x_{s-1}} \sum_{x_{s+4}} \cdots \sum_{x_{20}} \pi_{1,\ldots,s,s+1,s+2,s+3\ldots,20}(x_1,\ldots,0,1,0,0,\ldots,x_{20})$$

$$\pi_{s,s+1,s+2,s+3}(0,1,1,0) = \sum_{x_s} \cdots \sum_{x_{s-1}} \sum_{x_{s+4}} \cdots \sum_{x_{20}} \pi_{1,\ldots,s,s+1,s+2,s+3\ldots,20}(x_1,\ldots,0,1,1,0,\ldots,x_{20})$$

for $s = 1,\ldots,17$. See Table 7.4 in Appendix for these calculated probabilities. Consequently, we calculate the bias across the 100 replications using these true fourwise marginal probabilities as,

$$Bias(\hat{\phi}_s(\underline{x})) = \mathrm{E}(\hat{\phi}_s(\underline{x})) - \phi_s(\underline{x}).$$

3.5.2 Empirical Standard Error

We will calculate the empirical standard error of our estimates across the 100 replications implemented in our simulations in order to assess variability in our estimates. Ideally, the estimator will demonstrate a low empirical standard error.

We will calculate the standard error for the joint distribution estimates from the order-1 Markov chain reconstruction, order-2 Markov chain reconstruction, and order-3 Markov chain reconstruction. By the same reasoning as was given for the bias calculations, we will look at the empirical standard error for the 91 sequences assigned a non-zero probability on the simulated true ancestor distribution.

We will also calculate the standard error for the pairwise, threewise, and fourwise marginal estimates. By the same reasoning as was given for the bias calculations, we will look at the standard error for the free parameters only.

Note that these measures of bias and empirical standard error will allow us to calculate the mean squared error for the marginal and joint distribution estimates as well since it can be shown that

$$MSE(\hat{\phi}_s(\underline{x})) = Var(\hat{\phi}_s(\underline{x})) + Bias^2(\hat{\phi}_s(\underline{x})).$$

3.5.3 Bootstrap Confidence Intervals

For each of the free parameters in each simulation, we will calculate bootstrapped confidence intervals with a confidence level of $\alpha = 0.95$. For each marginal free parameter at each site, we have a sample of 100 estimated probabilities. That is, we have

$$\phi_s(\underline{x})^1, \phi_s(\underline{x})^2, \ldots, \phi_s(\underline{x})^{100}$$

for $s = 1, \ldots, L-m$. We will take 100 re-samplings of size 100 for each marginal free parameter at each site. Then in each re-sampling, we will have

$$\phi_s(\underline{x})^{1*}, \phi_s(\underline{x})^{2*}, \ldots, \phi_s(\underline{x})^{100*}$$

for $s = 1, \ldots, L - m$. For each re-sampling, we will calculate the difference between the average estimate calculated from the original sample ($\bar{x}$) and the average estimate calculated from the re-sampling ($\bar{x}^*$). We will denote the difference between these values with δ, where

$$\delta = \bar{x}^* - \bar{x}.$$

The distribution of δ will therefore tell us about how much $\bar{x}^*$ varies around $\bar{x}$. Then we find $\delta_{0.025}$ and $\delta_{0.975}$ so that

$$P(\delta_{0.025} \leq \bar{x}^* - \bar{x} \leq \delta_{0.975} | \bar{x}) = 0.95$$

and equivalently,

$$P(\bar{x}^* - \delta_{0.975} \leq \bar{x} \leq \bar{x}^* - \delta_{0.025} | \bar{x}) = 0.95.$$

This gives us our 95 % confidence interval, $[\bar{x}^* - \delta_{0.975}, \bar{x}^* - \delta_{0.025}]$. We will use these bootstrapped confidence intervals to determine whether our known true marginal free parameter probabilities are within these 95% confidence intervals. If the estimator is accurate, we would expect that the true marginal free parameter probabilities tend to be within this interval.

3.5.4 Neighboring Sequences

As mentioned above, we are interested not only in the probability estimated for a given sequence on the joint distribution being close to the true probability on average but also in which sequences are being assigned a portion of the density of the joint distribution (*i.e. assigned some non-zero probability*). Our simulated true ancestor distribution assigns density to only 91 of the 1,048,576 possible sequences, so it is likely that the estimated joint distributions on average will assign some density to true zero probability sequences. We therefore require a method to relate the true non-zero probability sequences and the true zero probability sequences.

Hamming distances have been employed in previous research to quantify the similarity or dissimilarity between DNA, RNA, or SNP sequences. The Hamming distance between two sequences X_1 and X_2 is defined as the number of indices in the sequences that differ between the two sequences,

$$D_H(X_1, X_2) = \sum_{i=1}^{L} (X_{1i} \neq X_{2i}).$$

It is often implemented in genetics problems where there is a need to identify pairs or groups of sequences that are similar, to study viral transmission with intra-host viral variants, or to reduce dimensionality of data by combining information from many SNP sequences determined to be in the same group (Tsyvina et al. 2018; Wang, Kao, and Hsiao 2015). These groups or pairs are often defined by setting a threshold t such that X_1 and X_2 are classified as in the same group if and only if $D_H(X_1, X_2) \leq t$. This threshold can be selected through processes such as a clustering algorithm, or can be set manually by the researcher (Wang, Kao, and Hsiao 2015).

We will use this concept to determine which sequences should be considered "neighbors" to the true non-zero probability sequences. For example, if we set $t = 2$ then this would mean that we believe a sequence should be considered a neighbor to a true non-zero probability sequences if it is different on only two of the 20 sites. One true non-zero probability sequence is

$$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0).$$

For $t = 2$, we would consider

$$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1)$$

and

$$(1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$$

to be in the set of neighbor sequences, but

$$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1)$$

would not be in this set.

Tsyvina et al. 2018, for sequences of length 264 nucleotides, set the threshold t manually to be 3.77% of the sequence, or 10 nucleotides. We aim to implement a similar idea, but we are limited in the degree of specificity that we are able to consider because our sequence length is much shorter; our sequences are 20 SNP sites long, so each that each site that differs corresponds to a 5% difference between the two sequences. The threshold choice for Wang, Kao, and Hsiao 2015 was also motivated by "empirically validated recommended thresh-

old for separation between epidemiologically related and unrelated intra-host HCV HVR1 populations." Such a recommended threshold has not been validated for comparing sequences in the context of estimating the distribution of possible ancestor sequences. We can, however, borrow from this epidemiological threshold the idea of relating the set threshold with the percentage of the sequences that differ.

We will calculate the total density from the joint distribution assigned to the 91 true non-zero sequences, and compare this sum to that of three sets of neighboring sequences. The first will be the 95% Neighbors sequences which are within a Hamming distance of 1 of a true non-zero sequence. The second will be the 90% Neighbors sequences which are within a Hamming distance of 2 of a true non-zero sequence. The third will be the 85% Neighbors sequences which are within a Hamming distance of 3 of a true non-zero sequences. See Table 3.2 for details on the resulting groups. Ideally, the estimator will assign the majority of the density of the joint distribution to the true non-zero sequences, but the comparison of these various joint density sums will allow us to assess whether, in cases where some portion of that density is not assigned to a true non-zero sequence, the misplaced density is assigned to a sequence that is similar to a true non-zero sequence among different thresholds of similarity.

Group	**Criteria**	**Num. Sequences in Group**
True Non-Zeros Only	$\pi_{1,\ldots,20}(x_1, \ldots, x_{20}) \neq 0$	91
True Non-Zeros and 95% Neighbors	$d_H(X_{true}, X_i) \leq 1$	1564
True Non-Zeros and 90% Neighbors	$d_H(X_{true}, X_i) \leq 2$	12070
True Non-Zeros and 85% Neighbors	$d_H(X_{true}, X_i) \leq 3$	55374

Table 3.2: We use Hamming distance thresholds to define sets of sequences which are "neighbors" with the true non-zero sequences. The Hamming distance between two sequences denotes the number of indices which differ between the two sequences. Let $d_H(X_{true}, X_i)$ denote the Hamming distance between one of the true non-zero sequences, X_{true}, and one of the true zero sequences, X_i. We will investigate four different sets of sequences. The group "True Non-Zeros Only" is only the 91 true non-zero sequences. The group "True Non-Zeros and 95% Neighbors" is the 91 true non-zero sequences and the sequences that are within a Hamming distance of 1 to a true non-zero sequence (this means that 19 of 20 sites are the same, so they are 95% similar). The group "True Non-Zeros and 90% Neighbors" is the 91 true non-zero sequences and the sequences that are within a Hamming distance of 2 to a true non-zero sequence. The group "True Non-Zeros and 85% Neighbors" is the 91 true non-zero sequences and the sequences that are within a Hamming distance of 3 to a true non-zero sequence.

Chapter 4

Simulation Results

4.1 Marginal Distribution Results

The marginal distribution results, overall, indicate that the Recombination Model produces estimates of the pairwise, threewise, or fourwise marginal distributions that have satisfactorily low bias and empirical standard error. However, we are able to observe an interaction between the recombination probability and a directional effect of the Markov chain.

4.1.1 Pairwise Estimation

Overall, the marginal pairwise estimation results show that the hierarchical estimation is able to maintain relatively low bias and empirical standard error even at this smallest Markov chain order. The bias was at most $|Bias(\hat{\phi}_s)| = 0.00932$ and the empirical standard error was at most $SE(\hat{\phi}_s) = 0.05521$. See Figure 4.1 for a plot of the mean squared error on $\hat{\phi}_s$ across the 100 replications. Observe that, overall, the values of mean squared error were small providing evidence that, on average, the estimates are close to the true values with rela-

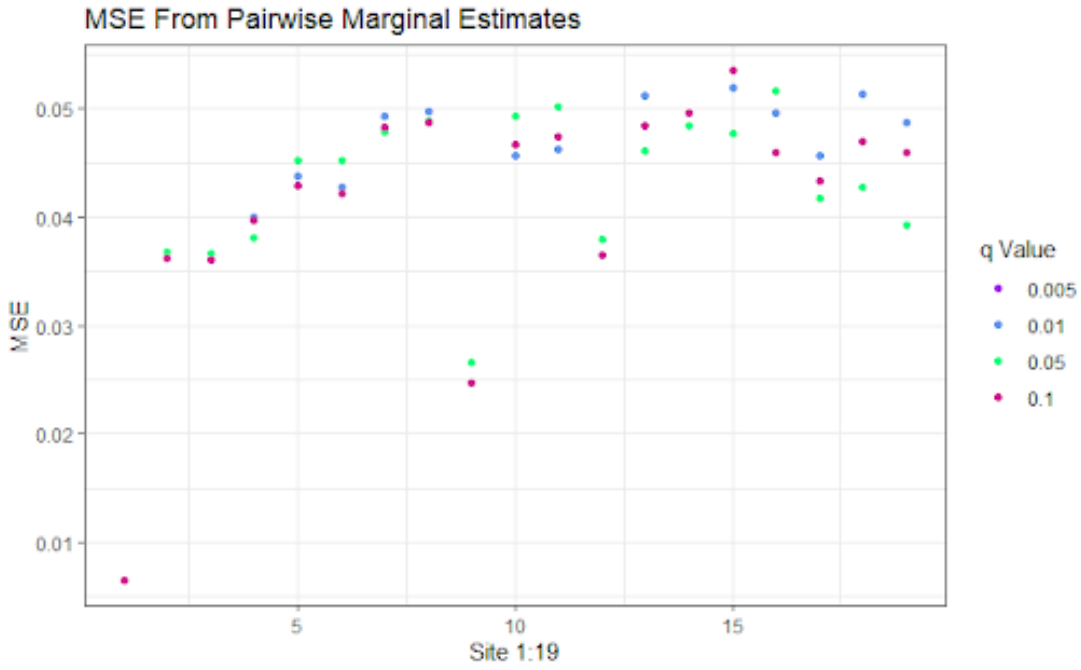

Figure 4.1: Mean Squared Error from $m = 1$ simulations for the free parameter ϕ_s. MSE is plotted for all four implemented q-values; purple represents the simulation where $q = 0.005$, blue represents the simulation where $q = 0.01$, green represents the simulation where $q = 0.05$, and red represents the simulation where $q = 0.1$. Observe that the MSE values increase from left to right, consistent with the directional effect, and that for the majority of sites the MSE is similar to its adjacent sites.

tively little variability. In addition, we can also observe that for the most part the mean squared error at a given site is similar to the mean squared error at the adjacent sites. The exceptions are site 9 and 12, whose mean squared errors are lower than their respective adjacent sites.

We interpret these bias and empirical standard error results to be satisfactory because, for all four implemented values of q across all of the 19 sites, each true probability was within the 95% bootstrapped confidence intervals. For example, Figure 4.2 shows the true marginal probabilities plotted in relation to the average estimates with 95% bootstrapped confidence intervals for

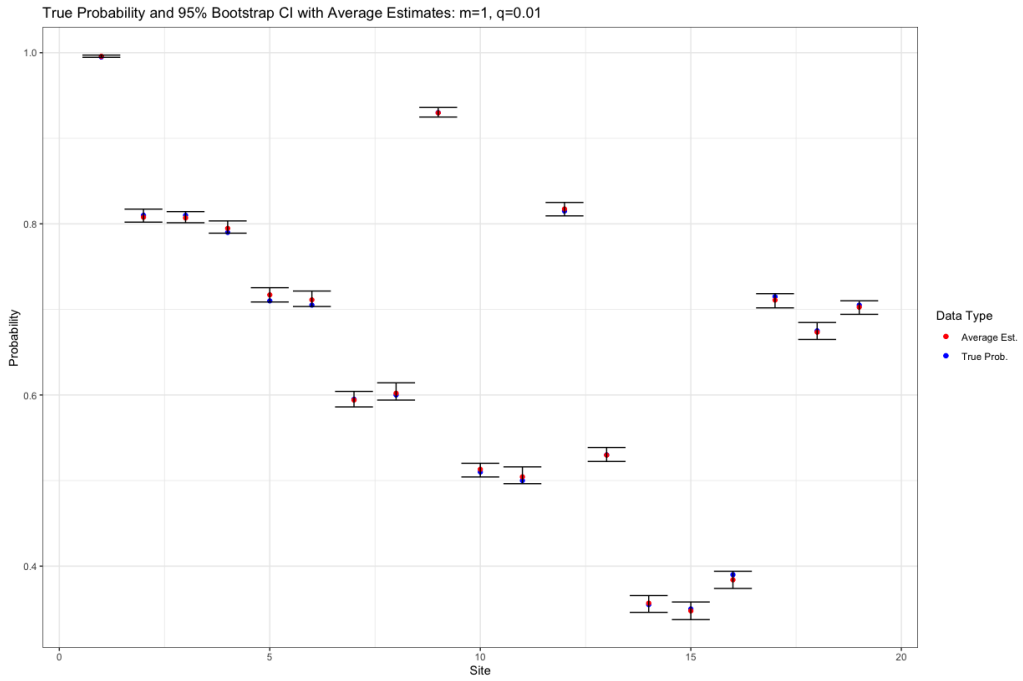

Figure 4.2: True Pairwise Marginal Estimates plotted with 95% bootstrapped confidence intervals and the average estimates. This plot is based on the pairwise simulation where $q = 0.01$. The blue points represent the true marginal probability and the red points represent the average estimated probability from the 100 replications. The error bars represent bootstrapped confidence intervals with a confidence level of $\alpha = 0.95$. Observe that for all 19 sites the true marginal probability falls within the 95% confidence intervals, suggesting that this estimation method consistently provides estimates of the pairwise marginal probabilities which are similar to the simulated true probabilities.

the pairwise simulation where $q = 0.01$. Observe that the true probability falls within the confidence intervals for all 19 sites, suggesting that the bias and standard error are small enough to consistently provide reasonable estimates of these marginal probabilities. This pattern was consistent across the other implemented values of q.

The results of our marginal pairwise estimates do, however, indicate a directional effect of the Markov chain conditioning left-to-right in the model formation. Bias was small for sites close to the start of the chain and increased

for sites further right on the chain. This effect also seems to be dependent on the chosen value of q. For smaller values of the recombination probability, $q = 0.005, 0.01, 0.05$, there is a tendency towards underestimation on the sites that are further right on the chain. In Figure 4.3, observe that for these three small q values, nearly all the sites past site 15 show a negative bias. However, when $q = 0.1$, we observe that the bias increases by as much as fourfold for sites located further right on the chain, with both positive and negative bias (See Figure 4.3). The increase was larger for the largest q-values. This suggests that the chain being defined from left-to-right leads to larger bias for the $\hat{\phi}_s$ where s is closer to L, but only if a relatively large probability is selected for q. The bias indicating this directional effect is a logical result of the dependency within the Markov chain. Because of the conditional probability structure, $\hat{\pi}_{9,10}$ will depend on $\hat{\pi}_{8,9}$ which will depend on $\hat{\pi}_{7,8}$ and so on and so forth. This means that, if $\hat{\pi}_{7,8}$ is a biased estimate then this bias will accumulate along the chain. However, while we can observe these patterns in the biases, it remains important to note that these biases are still low in magnitude.

A directional effect can also be observed in the empirical standard error, but unlike the bias it does not appear to be dependent on q. Across all pairwise simulations, we observe a small standard error for sites close to the start of the chain but the standard error increases for sites further from the start of the chain. For example, for the pairwise marginal estimates where $q = 0.05$, $\hat{\phi}_1$ has a standard error of approximately 0.0025 while $\hat{\phi}_{16}$ has a standard error of about 0.051 across the 100 replications. This suggests that the choice to have the chain be defined left-to-right may have the effect of creating increased variability for the $\hat{\phi}_s$ where s is closer to L regardless of the selected value of q. Similar to the argument given with respect to the bias, this is likely due to

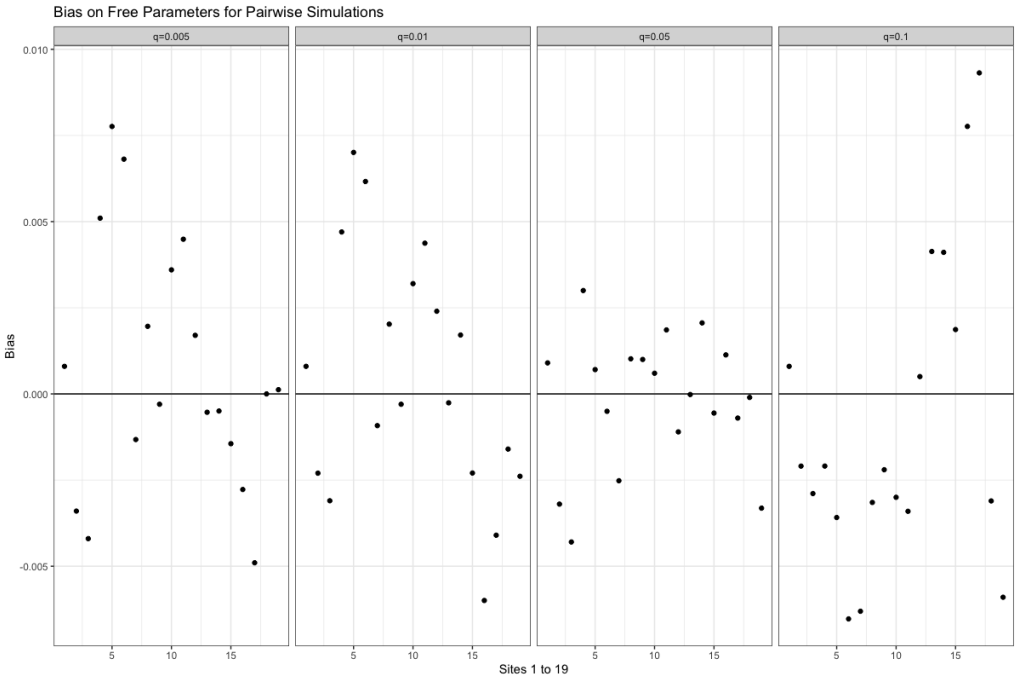

Figure 4.3: Bias for the free parameter across 100 replications where $m = 1$. Bias is calculated relative to the true values calculated from the simulated true ancestor distribution. Horizontal line plotted to show where bias is equal to 0, which would indicate the average estimate is equal to the true value. We are able to observe here that there is evidence of a direction effect of the Markov chain formation which is dependent on q. Bias is small for sites close to the start of the chain, but increases for sites further right. Note that for $q = 0.005, 0.01, 0.05$ there is a tendency toward underestimation for sites further right on the chain. For the largest value of $q = 0.1$, however, there is larger bias, both positive and negative, for sites further from the start of the chain.

an accumulation of variability going toward the right of the chain as a result of conditioning on previous sites.

4.1.2 Threewise Estimation

Marginal threewise estimation results similarly showed that the hierarchical estimation is able to obtain relatively low bias and empirical standard error on the two free parameters, $\hat{\phi}_s(0)$ and $\hat{\phi}_s(1)$. The bias was at most $|Bias(\hat{\phi}_s(0))| = 0.01151$, $|Bias(\hat{\phi}_s(1))| = 0.01458$. The empirical standard error was at most $SE(\hat{\phi}_s(0)) = 0.05521$, $SE(\hat{\phi}_s(1)) = 0.05218$.

See Figures 4.4 and 4.5 for plots of the mean squared error on the two free parameters. There was a notable observation in the MSE in relation to the free parameter being estimated ($\hat{\phi}_s(0)$ or $\hat{\phi}_s(1)$). Observe that, in Figure 4.4, the MSE for a given $s = 1, \ldots, 19$ on $\hat{\phi}_s(0)$ is relatively similar to the values of s that immediately precede and follow it with the exception of $s = 5$ and $s = 6$. For example, the MSE of $\hat{\phi}_8(0)$ is around 0.05 and the MSE of $\hat{\phi}_9(0)$ is around either 0.045 or 0.05. On the other hand, for the other free parameter $\phi_s(1)$, see in Figure 4.5 that the MSE is much less similar for adjacent sites. For example, $\hat{\phi}_1(1)$ has an MSE around 0.0075, $\hat{\phi}_2(1)$ has an MSE around 0.0375, and $\hat{\phi}_3(1)$ has an MSE around 0.0025. This suggests that the estimates for $\hat{\phi}_s(1)$ may be less predictable than the estimates for $\hat{\phi}_s(0)$. Such a difference may arise from an underlying property of SNP data: $\phi_s(1) = \pi_{s,s+1,s+1}(0, 1, 0)$ is a sequence containing the minor allele, which is by definition more rare in the data. Threewise estimates are built hierarchically from the onewise, which are sample proportions and thus potentially impacted by the rarity in the data. This potential relationship between the major/minor alleles and the

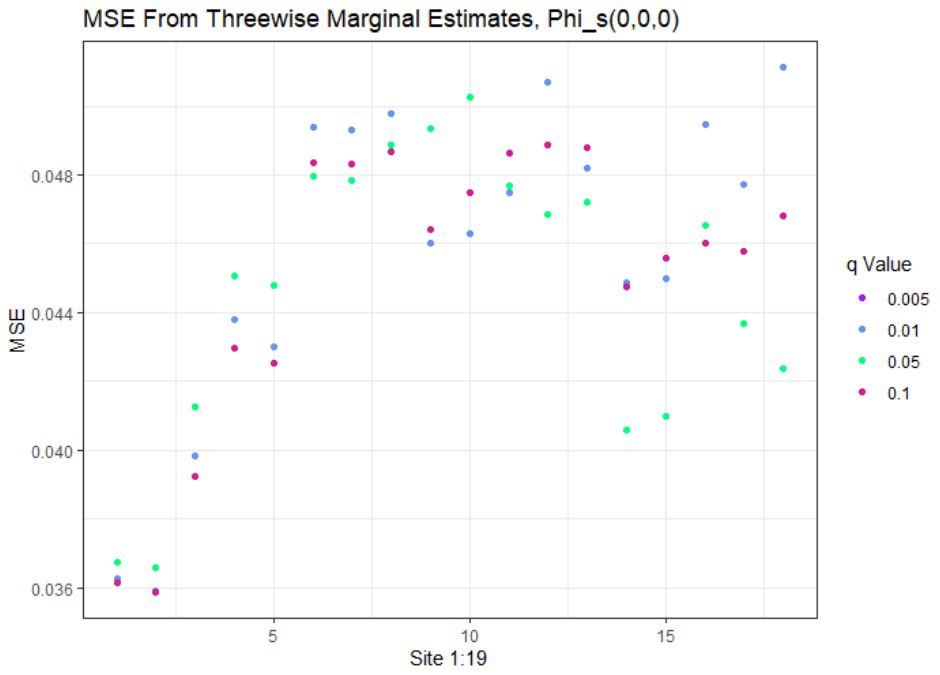

Figure 4.4: Mean Squared Error from $m = 2$ Simulations for the first free parameter, $\hat{\phi}_s(0)$. MSE is plotted for all four implemented q-values; purple points represent the simulation where $q = 0.005$, blue points represent the simulation where $q = 0.01$, green points represent the simulation where $q = 0.05$, and red points represent the simulation where $q = 0.1$. Observe that the MSE values increase from left to right, consistent with the directional effect, and that the MSE for a given site is similar to its adjacent sites.

mean squared error is a property of the data whose investigation is beyond the scope of the present project, but may be an important avenue for future research.

Nonetheless, the results for the bias and empirical standard error for the threewise marginal distribution estimates seem to be satisfactory. For the four implemented values for q and across all of the 18 sites on the two free parameters, the true probability was within the 95% bootstrapped confidence intervals for the majority of estimates. When the true probability was not within the interval, it was close to one of the end points of the interval. For an example

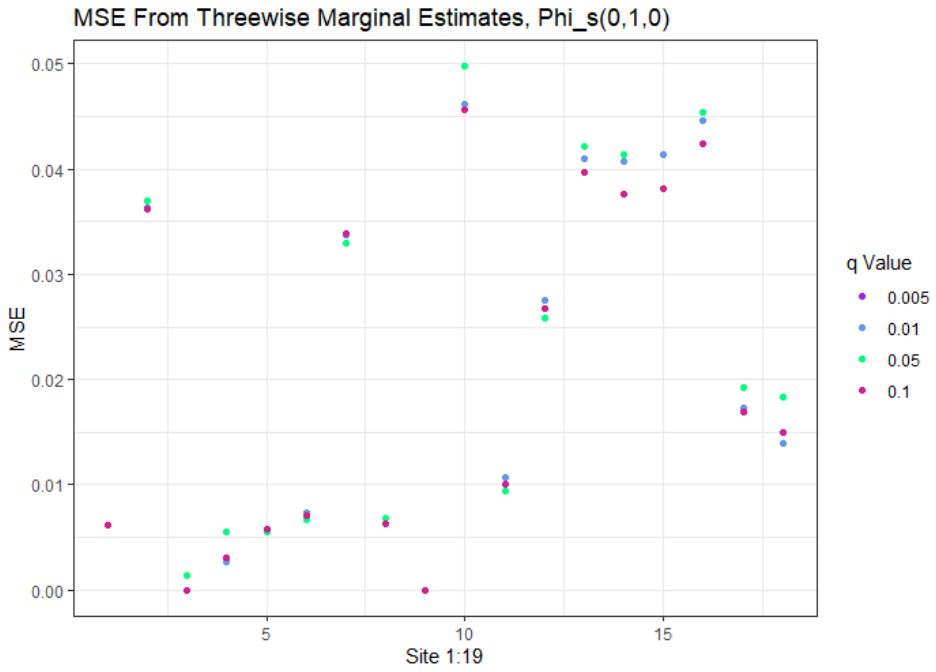

Figure 4.5: Mean Squared Error from $m = 2$ Simulations for the second free parameter, $\hat{\phi}_s(1)$. MSE is plotted for all four implemented q-values; purple points represent the simulation where $q = 0.005$, blue points represent the simulation where $q = 0.01$, green points represent the simulation where $q = 0.05$, and red points represent the simulation where $q = 0.1$. Observe that, unlike Figure 4.4, the MSE for a given site is not consistently similar to its adjacent sites.

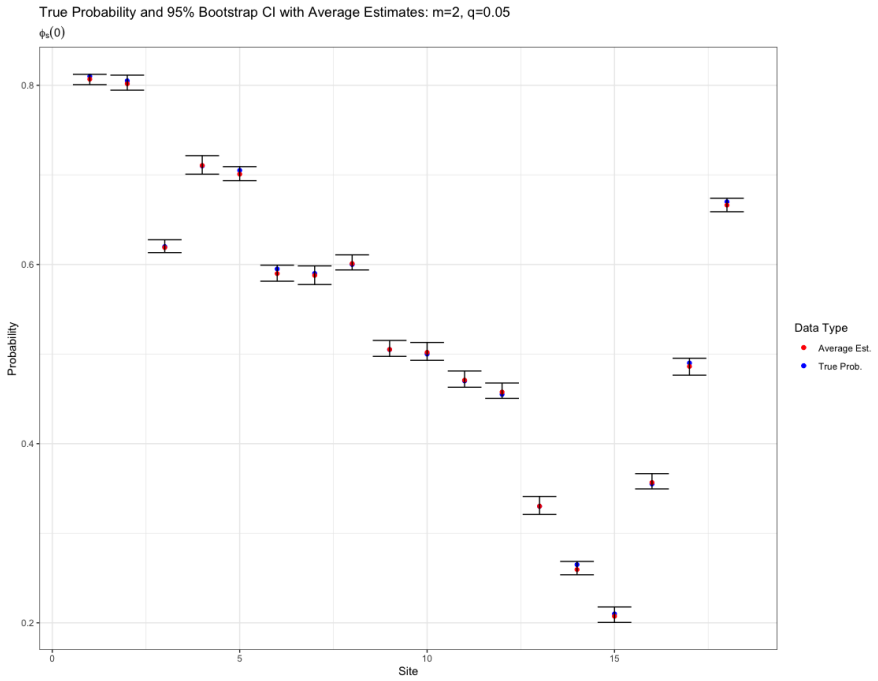

Figure 4.6: True threewise marginal estimates plotted with 95 % bootstrapped confidence intervals and the average estimates. This plot is based on the threewise simulation where $q = 0.05$ on the free parameter $\hat{\phi}_s(0)$. The blue points represent the true marginal probability and the red points represent the average estimated probability from the 100 replications. The error bars represent bootstrapped confidence intervals with a confidence level of $\alpha = 0.95$. Observe that, for this case, the true marginal probability falls within the 95% confidence intervals, suggesting that under certain conditions this estimation method consistently provides estimates of the threewise marginal probabilities which are similar to the simulated true probabilities.

of a case where the true marginal probability is within the confidence interval for all of the sites, see Figure 4.6 which shows the true marginal probabilities plotted in relation to the average estimates with 95% bootstrapped confidence intervals for the threewise simulation where $q = 0.05$ on the free parameter $\hat{\phi}_s(0)$. Observe that the true probability falls within the confidence interval for all 18 sites.

For an example where the true marginal probability is not within the confi-

dence interval for all of the sites, see Figure 4.7. This shows the true marginal probabilities plotted in relation to the average estimates with 95% bootstrapped confidence intervals for the threewise simulation where $q = 0.1$ on the free parameter $\hat{\phi}_s(1)$. Observe that the true probability falls within the confidence interval for the majority of sites, with the exception of sites 4, 16, and 18. For site 4, we observe that the true probability is $\phi_4(1) = 0$ while the average estimate was $\hat{\phi}_4(0) = 0.0046$ and the lower bound was $(\bar{x}^* - \delta_{0.975}) = 0.0036$. For site 16, we observe that the true probability is $\phi_{16}(0) = 0.245$ while the average estimate was $\hat{\phi}_{16}(0) = 0.2304$ and the upper bound was $(\bar{x}^* - \delta_{0.025}) = 0.2382$. For site 18, we observe that the true probability is $\phi_{18}(1) = 0.015$ while the average estimate was $\hat{\phi}_{18}(1) = 0.0258$ and the lower bound was $(\bar{x}^* - \delta_{0.975}) = 0.0228$. Therefore, from these three cases we can observe that, when the true probability falls outside of the bootstrapped confidence intervals, the difference between the true probability and the closest confidence interval bound is very small to negligibly small– in fact, the difference was at most 0.0078.

Interestingly, this second less desirable case where the true marginal probability is not always within the confidence interval points to a potential effect of the presence of the minor allele in the free parameter. For the free parameter $\hat{\phi}_s(0)$ which is dependent only on the major allele, we see that the true marginal probability is within the confidence interval for all 18 sites for all four values of q. In contrast, for the free parameter $\hat{\phi}_s(1)$ which is dependent in part on the minor allele, we see that the true probability is not always within the confidence interval. As discussed above, the difference between true probabilities and the interval bound is very small, but the fact that the true marginal probability only falls outside of the confidence interval when the free parameter contains the minor allele may provide further support for investigating the impact of

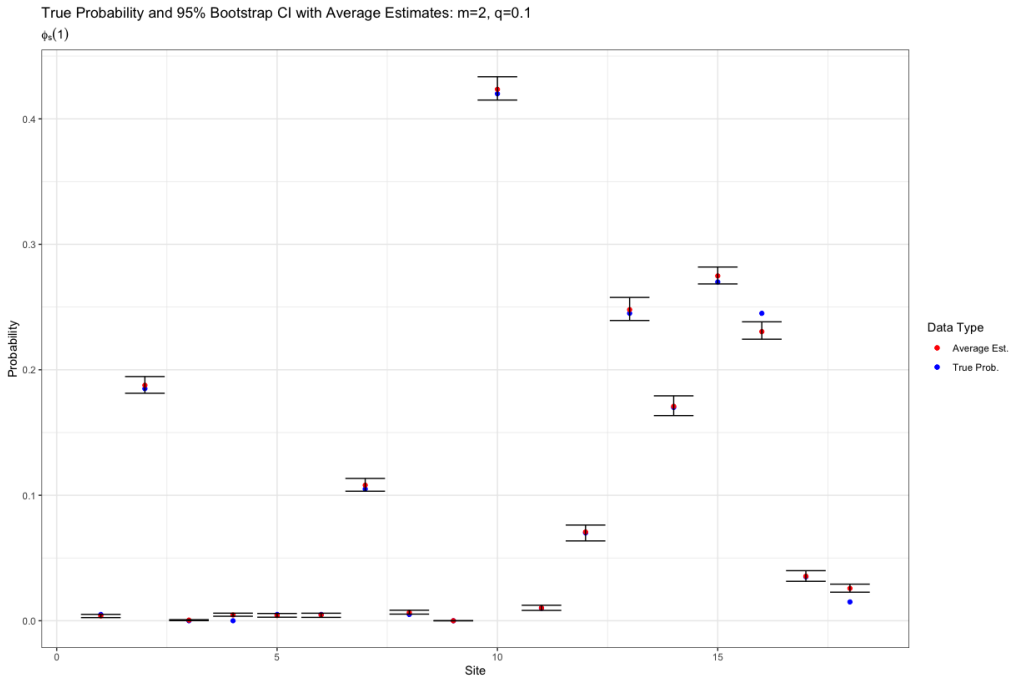

Figure 4.7: True threewise marginal estimates plotted with 95% bootstrapped confidence intervals and the average estimates. This plot is based on the threewise simulation where $q = 0.1$ on the free parameter $\hat{\phi}_s(1)$. The blue points represent the true marginal probability and the red points represent the average estimated probability from the 100 replications. The error bars represent bootstrapped confidence intervals with a confidence level of $\alpha = 0.95$. Observe that the true marginal probability falls within the 95% confidence intervals for the majority of sites, with the exceptions of $s = 4, 16, 18$. However, for these borderline cases we can observe that the difference between the true probability and the closest interval bound is very small to negligibly small.

the inherent sample size differences between major and minor alleles.

Similarly as with the pairwise simulation results, there is evidence of a directional effect of the Markov chain. This can be seen in the bias calculated relative to the true values of the free parameters. We observe very low bias (less than $|0.005|$) for the first one or two sites, but the magnitude increases by as much as threefold for sites located further right on the Markov chain. We see a similar pattern to the pairwise simulations in that the directional effect

interacts with the recombination probability– specifically that there is a more pronounced increase in bias for $q = 0.1$. However, this is also dependent on the free parameter that is being estimated (see Figure 4.8).

Consider first the free parameter dependent only on the major allele, $\hat{\phi}_s(0)$. This free parameter shows the same patterns as was observed for the pairwise marginal estimation results. When $q = 0.005, 0.01, 0.05$, the bias on later sites increases with a tendency towards negative bias. When $q = 0.01$, the bias on the later sites increases in bias toward both positive and negative values. Next, we consider the other free parameter, $\hat{\phi}_s(1)$, which is dependent on both the major and minor allele. There is a tendency towards overestimation on all four values of q.

These differential tendencies toward positive or negative bias may be an artifact of our data-generating mechanism. As previously mentioned, sequences containing the minor allele will, by definition, appear less frequently in the data. The fact that we simulate the true probabilities of the parameters based on sample proportions means that the probabilities assigned to parameters containing the minor allele will be zero or near zero. Thus, the only bias possible for the parameter $\hat{\phi}_s(1)$ is overestimation, which will also force $\hat{\phi}_s(0)$ to tend toward underestimation. Keeping this in mind, the patterns observed in relation to the free parameters are potentially not of great relevance. Therefore, we arrive at the same conclusion as was observed for the pairwise simulations: the increase in bias going right along the chain and indicating a directional effect of the Markov chain is dependent on the recombination probability, with the largest recombination probability, $q = 0.1$, showing a more pronounced increase in bias than the other implemented q values.

The directional effect of the Markov chain definition can also be seen in the

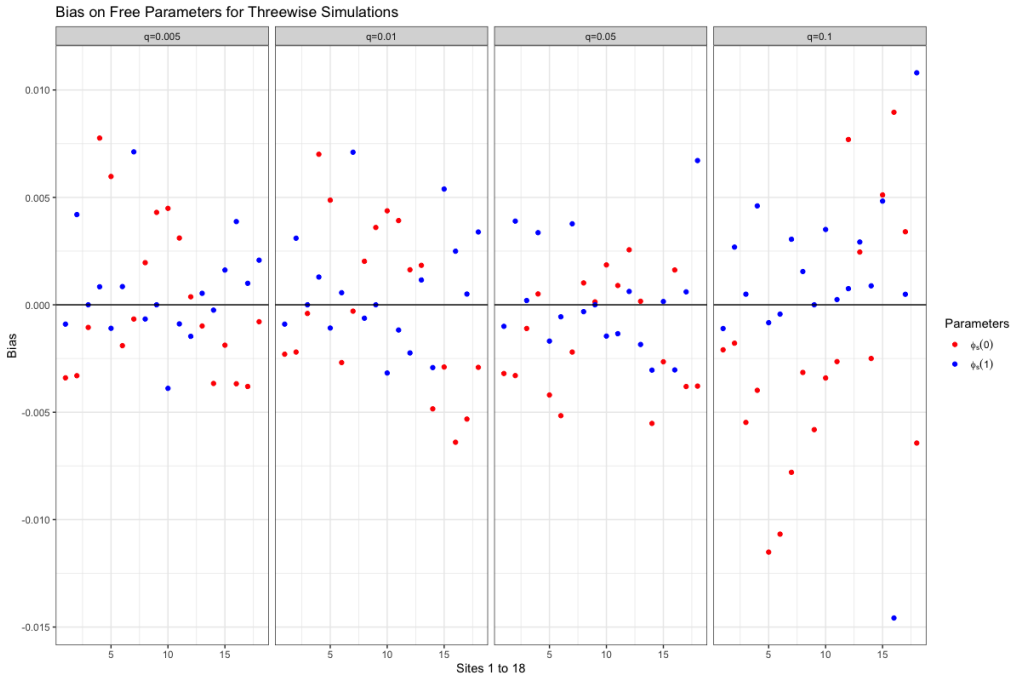

Figure 4.8: Bias for the free parameter across 100 replications where $m = 2$. Bias is calculated relative to the true values calculated from the simulated true ancestor distribution. Horizontal line plotted to show where bias is equal to 0, which would indicate the average estimate is equal to the true value. We are able to observe here that there is evidence of a direction effect of the Markov chain formation. Note that the bias is small for sites close to the start of the chain (site 1) but tends towards negative bias for sites further from the start of the chain (site 19) for $\hat{\phi}_s(0)$ and towards positive bias for further sites for $\hat{\phi}_s(1)$ for the smaller values of q. For the largest value of $q = 0.1$, however, there is the same pattern in bias but the increase in magnitude of bias is more pronounced.

empirical standard error. This did not seem to be dependent on the recombination probability, but we do observe differences between the free parameters. For $\hat{\phi}_s(0)$, we observe that the standard error increases as we go from left to right across the chain with a given site being similar to its adjacent sites. (*a plot for these results is available in the Appendix*, Figure 7.3). This makes sense as a result of the dependency structure implemented via the Markov chain which causes variability to accumulate down the chain. However, with the free parameter $\hat{\phi}_s(1)$ this pattern in standard error did not exist. For example, see Figure 4.9 which shows the standard error on the free parameter $\hat{\phi}_s(1)$ for the threewise simulations. While there is a slight, general increase in standard error, it is not a consistent increase as we observe with the other free parameter. This is likely the same phenomenon that we observed in the mean squared error plots, suggesting a possible influence of the rarity of the minor allele in the data.

4.1.3 Fourwise Estimation

Marginal fourwise estimation results showed that the hierarchical estimation is able to obtain relatively low bias and empirical standard error on the four free parameters, $\hat{\phi}_s(0,0)$, $\hat{\phi}_s(0,1)$, $\hat{\phi}_s(1,0)$, and $\hat{\phi}_s(1,1)$. The bias was at most $|Bias(\hat{\phi}_s(0,0))| = 0.01507897$, $|Bias(\hat{\phi}_s(0,1))| = 0.00843499$, $|Bias(\hat{\phi}_s(1,0))| = 0.007743742$, and $|Bias(\hat{\phi}_s(1,1))| = 0.006259105$. The empirical standard error was at most $SE(\hat{\phi}_s(0,0)) = 0.05615834$, $SE(\hat{\phi}_s(0,1)) = 0.0539674$, $SE(\hat{\phi}_s(1,0)) = 0.05069003$, and $SE(\hat{\phi}_s(1,1)) = 0.03201899$.

See Figure 4.10 for a plot of the mean squared error on the free parameter $\hat{\phi}_s(0,0)$ and Figure 4.11 for a plot of the mean squared error on the free

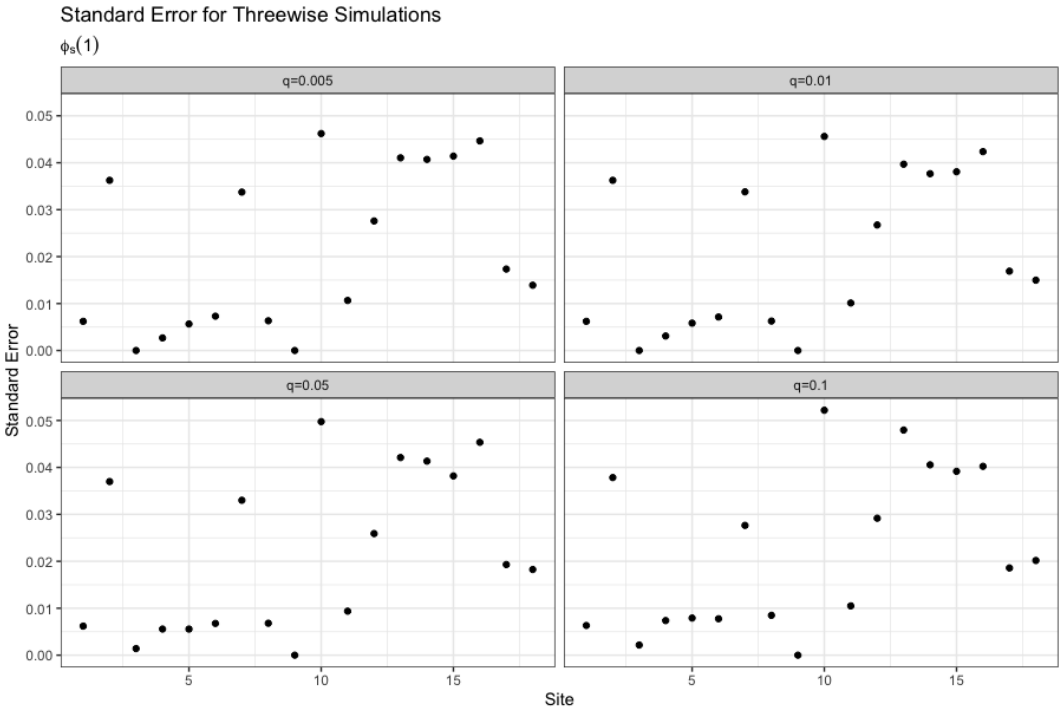

Figure 4.9: Standard error calculated across the 100 replications for $\hat{\phi}_s(1)$ in the threewise simulations. We observe a slight, general increase in standard error going toward the right of the chain, but this increase is less consistent than was observed on the other free parameter, $\hat{\phi}_s(0)$ and for the pairwise simulations.

parameter $\hat{\phi}_s(0,1)$. Plots for the remaining two free parameters, $\hat{\phi}_s(1,0)$ and $\hat{\phi}_s(1,1)$, can be found in the Appendix (7.1, 7.2). From these plots we observe a similar pattern which was observed in the threewise mean squared error plots in relation to the free parameters. Observe that, in Figure 4.10, the MSE for a given $s = 1, \ldots, 17$ on $\hat{\phi}_s(0,0)$ is relatively similar to the values of s which are adjacent to it with the exceptions of $s = 4$ and $s = 5$. For example, the MSE values for $\hat{\phi}_{11}(0,0)$ are around 0.043 and the MSE values for $\hat{\phi}_{12}(0,0)$ are around 0.045.

In contrast, we observe that, in Figure 4.11, the MSE for adjacent sites are not always similar. For example, the MSE values for $\hat{\phi}_8(0,1)$ are 0, but the MSE values for $\hat{\phi}_9(0,1)$ are around 0.05 and the MSE values for $\hat{\phi}_{10}(0,1)$ are around 0.01. We observe similar patterns for the plots of the MSE for the remaining two free parameters $\hat{\phi}_s(1,0)$ and $\hat{\phi}_s(1,1)$. The fact that, in these fourwise simulations, this pattern of lack of similarity in MSE is only appearing on the free parameters which contain the minor allele further suggests that there is a lack of predictability in estimates on the sequences that contain the minor allele. As discussed above, this feature of the estimator will be important to be further studied in the future.

Overall the results for the bias and the empirical standard error for the fourwise marginal distribution seem to be satisfactory because, for the four implemented values for q across the 17 sites on the free parameters, the true probability was within the 95% bootstrapped confidence intervals for the majority of estimates. When the true probability was not within the interval, it was close to the nearest interval bound. For an example of a case where the true marginal probability is within the confidence interval for all sites, see Figure 4.12 which shows the true marginal probabilities plotted in relation to

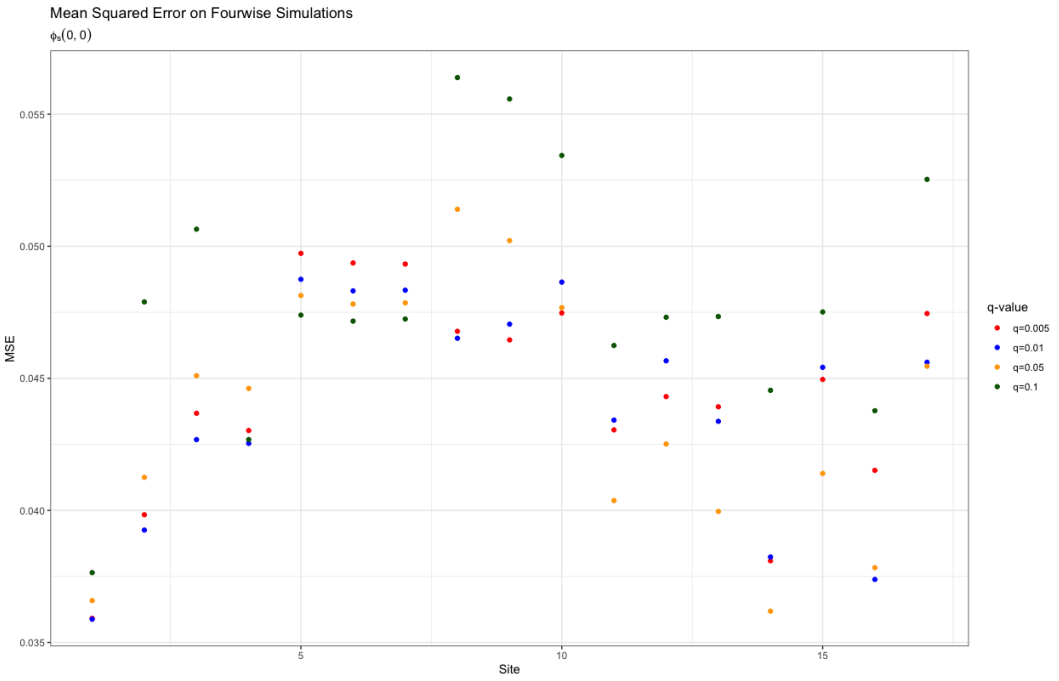

Figure 4.10: Mean Squared Error from $m = 3$ Simulations for the free parameter $\hat{\phi}_s(0,0)$. MSE is plotted for all four implemented q-values; red points represent the simulation where $q = 0.005$, blue points represent the simulation where $q = 0.01$, yellow points represent the simulation where $q = 0.05$, and green points represent the simulation where $q = 0.1$. Observe that the MSE for a given site is similar to the MSE for its adjacent sites. Furthermore, it is clear that, for sites further right on the chain, the recombination probability q has more of an influence on the MSE with $q = 0.1$ resulting in a larger MSE.

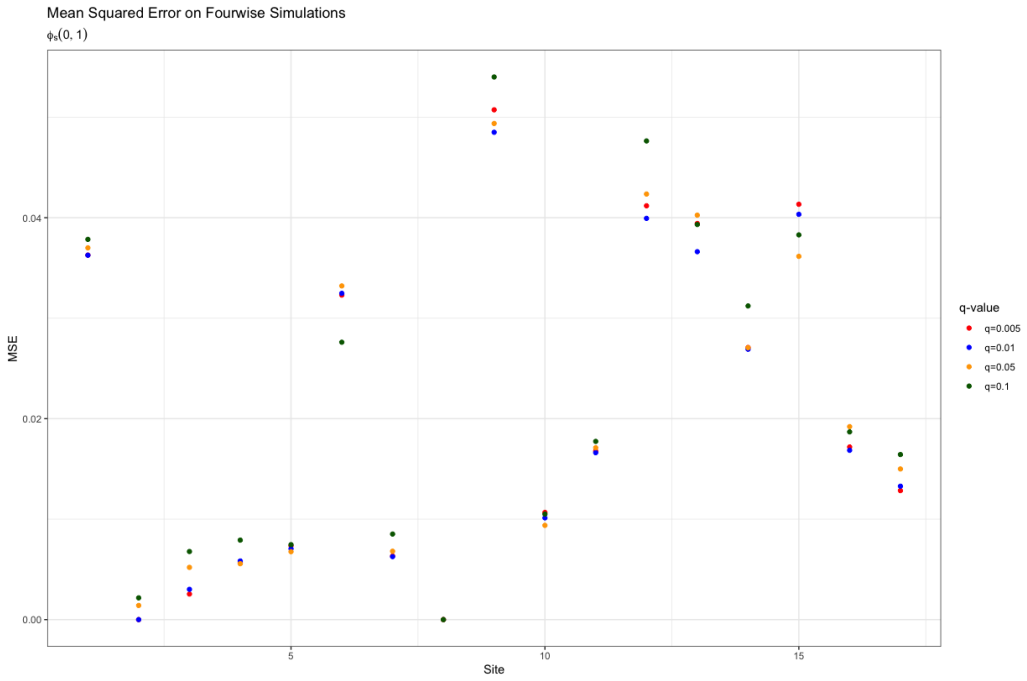

Figure 4.11: Mean Squared Error from $m = 3$ Simulations for the free parameter $\hat{\phi}_s(0,1)$. MSE is plotted for all four implemented q-values; red points represent the simulation where $q = 0.005$, blue points represent the simulation where $q = 0.01$, yellow points represent the simulation where $q = 0.05$, and green points represent the simulation where $q = 0.1$. Observe that the MSE for a given site is not consistently similar to the MSE for its adjacent sites, unlike in Figure 4.10.

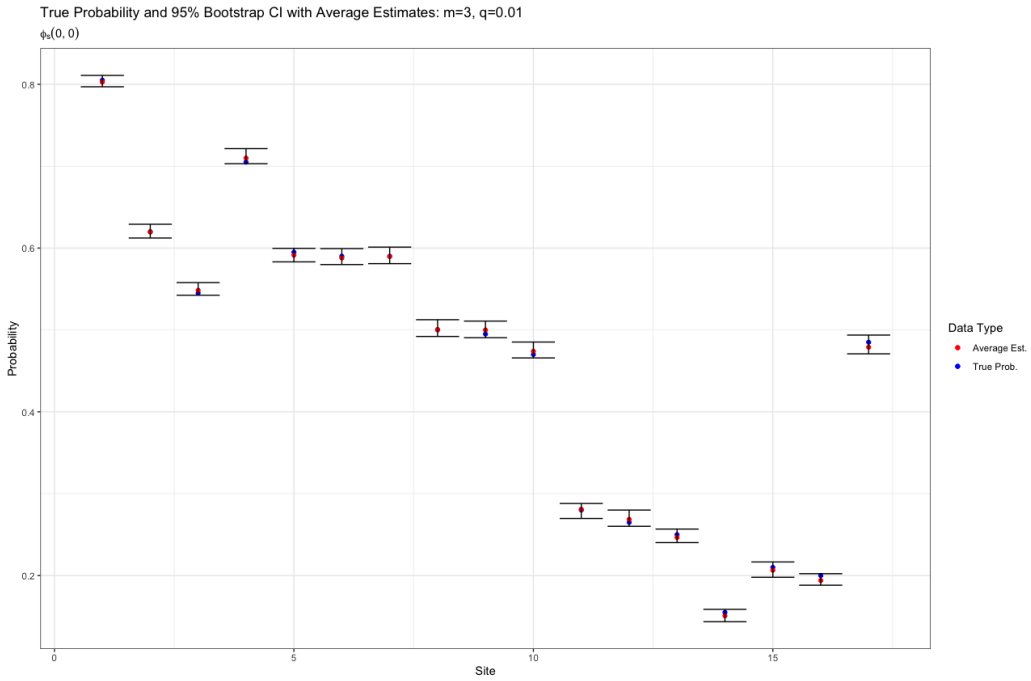

Figure 4.12: True fourwise marginal estimates plotted with 95 % bootstrapped confidence intervals and the average estimates. This plot is based on the fourwise simulation where $q = 0.01$ on the free parameter $\hat{\phi}_s(0, 0)$. The blue points represent the true marginal probability and the red points represent the average estimated probability from the 100 replications. The error bars represent bootstrapped confidence intervals with a confidence level of $\alpha = 0.95$. Observe that, for this case, the true marginal probability falls within in the 95% confidence intervals suggesting that, under certain conditions, this estimation method consistently provides estimates of the fourwise marginal probabilities which are similar to the simulated true probabilities.

the average estimates, including 95% bootstrapped confidence intervals, for the fourwise simulation where $q = 0.01$ on the free parameter $\hat{\phi}_s(0, 0)$. Observe that the true probability falls within the confidence interval for all 17 sites.

For an example where the true marginal probability is not within the confidence interval for all of the sites, see Figure 4.13. This figure shows the true marginal probabilities plotted in relation to the average estimates, including 95% bootstrapped confidence intervals, for the fourwise simulation where

$q = 0.1$ on the free parameter $\hat{\phi}_s(1,1)$. Observe that the true probability falls within the confidence intervals for the majority of the 17 sites, with the exceptions being sites 4, 6, and 16. For site 4, we observe that the true probability is $\phi_4(1,1) = 0$ while the average estimate was $\hat{\phi}_4(1,1) = 0.00426$ and the lower bound was $(\bar{x}^* - \delta_{0.975}) = 0.00236$. For site 6, we observe that the true probability is $\phi_6(1,1) = 0$ while the average estimate was $\hat{\phi}_6(1,1) = 0.00625$ and the lower bound was $(\bar{x}^* - \delta_{0.975}) = 0.00489$. For site 16, the true probability is $\phi_{16}(1,1) = 0$ while the average estimate was $\hat{\phi}_{16}(1,1) = 0.000816$ and the lower bound was $(\bar{x}^* - \delta_{0.975}) = 0.000211$. Therefore, from these three cases, we can observe that, when the true probability falls outside of the bootstrapped confidence intervals, the difference between the true probability and the closest confidence interval bound is very small to negligibly small– in fact, the difference was at most 0.00489.

Similarly to the pairwise and threewise marginal results, the bias on the fourwise marginal estimates show evidence of a directional effect of the Markov chain. In particular, we see low bias (less than $|0.005|$) for the sites close to the start of the chain, but the magnitude increases by as much as threefold for sites located further right on the Markov chain. This directional effect also appears to interact with the recombination probability; the increase in bias is more pronounced for the largest recombination probability value, $q = 0.1$, than the smaller values $q = 0.005, 0.01, 0.05$ (see Figure 4.14). This pattern was the case on all four of the free parameters. The free parameter being estimated also seems to mediate the behavior of the directional effect.

First consider the free parameter which is dependent only on the major allele, $\hat{\phi}_s(0,0)$. This free parameter shows the same patterns as was observed for the pairwise marginal estimation results. When $q = 0.005, 0.01, 0.05$, the

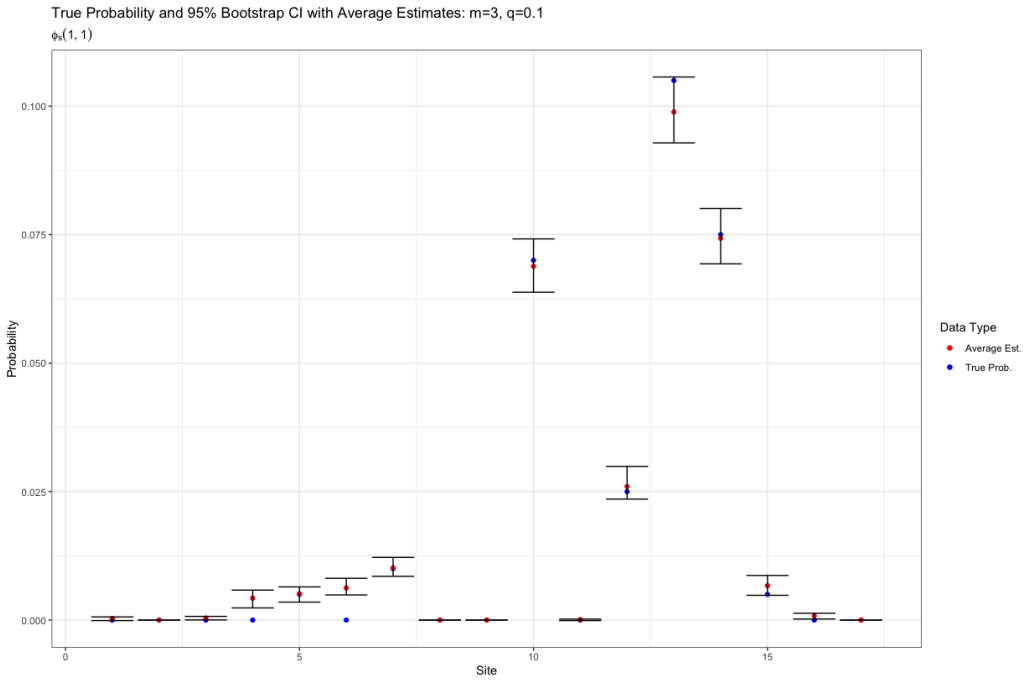

Figure 4.13: True fourwise marginal estimates plotted with 95% bootstrapped confidence intervals and the average estimates. This plot is based on the fourwise simulation where $q = 0.1$ on the free parameter $\hat{\phi}_s(0,1)$. The blue points represent the true marginal probability and the red points represent the average estimated probability from the 100 replications. The error bars represent bootstrapped confidence intervals with a confidence level of $\alpha = 0.95$. Observe that, for this case, the true marginal probability falls within in the 95% confidence intervals for the majority of sites with the exceptions of sites $s = 4, 6, 16$. For these borderline cases, however, we can observe that the difference between the true probability and the closest interval bound is very small to negligibly small.

bias increases for sites further right on the chain with a tendency towards negative bias. When $q = 0.1$, the bias increases for sites further right on the chain with both positive and negative bias.

Next, we consider the free parameters which are dependent on both the major and minor alleles: $\hat{\phi}_s(0,1)$ and $\hat{\phi}_s(1,0)$. On these free parameters, the bias increases for sites further right on the chain with both positive and negative bias for all four values of q.

Lastly, we consider the free parameter which is dependent only on the minor allele, $\hat{\phi}_s(1,1)$. On this free parameter, the bias increases for sites further right on the chain with majority positive bias for all for values of q. These patterns in relation to the free parameter are interesting to note and may warrant further investigation, but are also likely an artifact of characters of our data-generating mechanism as discussed in Section 4.1.2.

Overall, we conclude that the fourwise marginal estimation results also support the notion that the bias indicates a directional effect of the Markov chain which is dependent on the recombination probability, with the largest probability ($q = 0.1$) showing a more pronounced increase in bias than the other q values.

The directional effect of the Markov chain can also be seen in the empirical standard error due to the fact that the standard error increases for sites further right on the chain. As with the threewise marginal results, this did not seem to depend on the recombination probability but did depend on the free parameter being estimated. On the free parameter $\hat{\phi}_s(0,0)$, we see that the standard error increased for sites further right on the chain across all four values of q and the standard error for a given site was relatively similar to its adjacent sites (*a plot for these results is available in the Appendix*, Figure 7.4). On the free

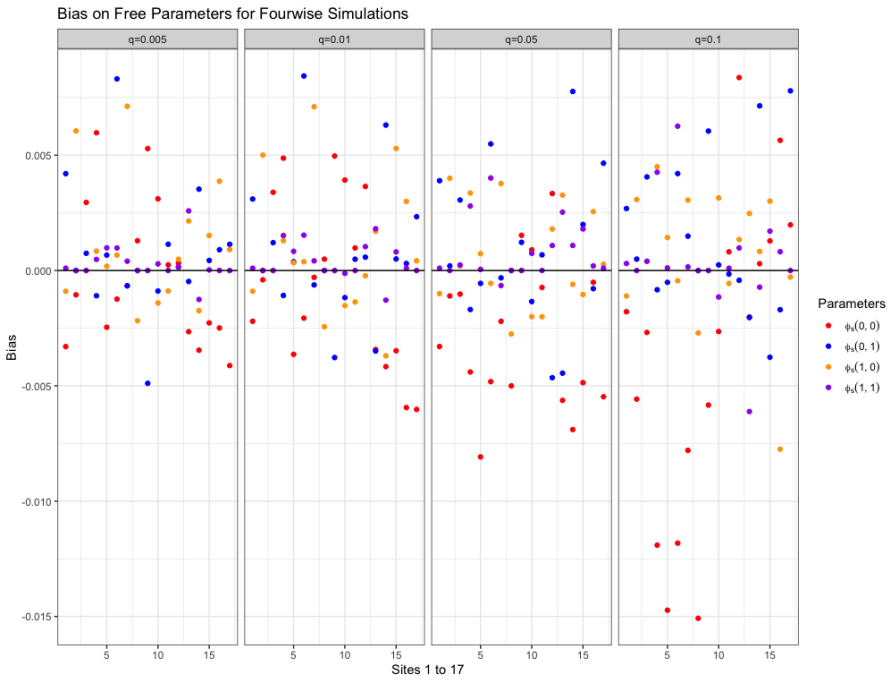

Figure 4.14: Bias for the free parameter across 100 replications where $m = 3$. Bias is calculated relative to the true values calculated from the simulated true ancestor distribution. The horizontal line indicates a bias of 0. This figure provides evidence of a directional effect of the Markov chain; note that the bias is small for sites close to the start of the Markov chain but increases in magnitude moving right on the chain. This increase in bias is most notable for $q = 0.1$. When the free parameter is $\hat{\phi}_s(0,0)$, the increased bias tends towards negative values. When the free parameter is $\hat{\phi}_s(0,1)$ or $\hat{\phi}_s(1,0)$, the increased bias tends towards both positive and negative values. When the free parameter is $\hat{\phi}_s(1,1)$, the increased bias tends towards positive values.

parameters $\hat{\phi}_s(0,1)$ and $\hat{\phi}_s(1,0)$, there was an increase in standard error for sites further right on the chain across all four values of q, but the standard error for a given site was not necessarily similar to its adjacent sites (*plots for these results are available in the Appendix*, Figure 7.5 and 7.6). On the last free parameter, $\hat{\phi}_s(1,1)$, we did not observe the same gradual increase in standard error moving right on the chain. Instead, the majority of the sites had very low standard error with some sites between $s = 10$ and $s = 15$ showing larger standard error. This tendency was observed on all four values of q. See Figure 4.15. This same pattern is observable in the 95% bootstrapped confidence intervals (Figure 4.13) given that the majority of sites had near-zero interval length. As has been touched upon previously, since this free parameter is dependent only on the minor allele, this may be an artifact of the data-generating mechanism but warrants further investigation.

4.2 Joint Distribution Estimates

The marginal distribution estimates that we have discussed thus far have shown that, while there is evidence of a directional effect which interacts with the recombination probability and evidence of an effect of major/minor alleles, the model is able to obtain reasonable estimates of these marginal probabilities as assessed by the relatively low bias and empirical standard error. However, the ultimate goal is to obtain estimates for the joint distribution. Recall that this is done via reconstruction using the marginal estimates according to Equation 2.4. Previous results looking at maximum Markov chain composite likelihood estimates show that the estimation process of the reconstructed joint will be more accurate for higher Markov chain orders with an notable trade-off of

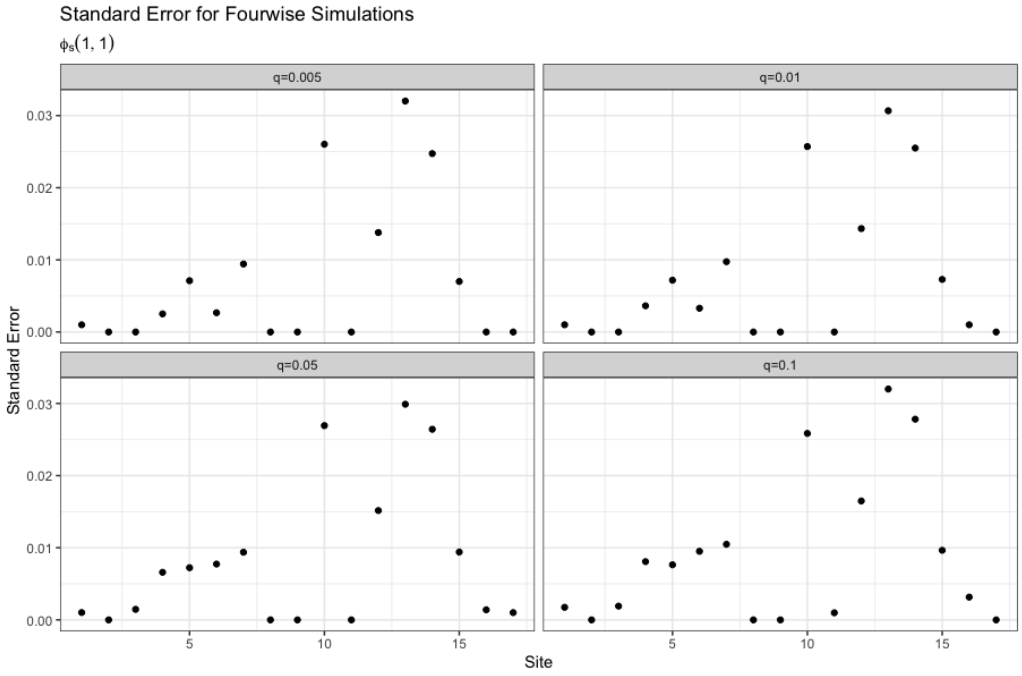

Figure 4.15: Standard error calculated across the 100 replications for $\hat{\phi}_s(1,1)$ in the fourwise simulations. Rather than an increase in standard error moving right on the chain, there did not seem to be any meaningful, predictive patterns to be observed in these standard errors.

increased complexity (Dillon and Lebanon 2010). Therefore we expect, in our simulations, that the bias will decrease as the Markov chain order increases. Since these simulations do, however, implement only $m = 1, 2, 3$, it is unclear whether we will encounter this trade-off for complexity.

Overall, in terms of the bias and empirical standard error results for the joint distribution estimates, we observe a trade-off between the bias and empirical standard error. If we increase the value of m, then we decrease the bias but increase the empirical standard error. This trade-off is apparent in Figures 4.16 and 4.17.

When $m = 1$, the largest observed bias was -0.06847766 and the largest observed standard error was 0.006185387. When $m = 2$, the largest observed bias was -0.05561639 and the largest observed standard error is 0.01032787. When $m = 3$, the largest observed bias was -0.0356277 and the largest observed standard error was 0.01859386. From these values, we can conclude that choosing a larger value of m allows us to obtain lower bias on the joint distribution estimates with the stipulation that this will also increase the variability.

4.2.1 Order-1 Markov Chain Reconstruction

As discussed above, for the order-1 Markov chain reconstruction of the joint distribution we observed the largest bias but the lowest empirical standard error on the 91 true non-zero sequences. In Figure 4.16, we can also observe that the majority of the bias on these 91 sequences was negative. This suggests that, in estimating these 91 sequences, the probability tends to be underestimated (by as much as 0.068), but the variability tends to be small.

Furthermore, we can observe the effect of this tendency towards underesti-

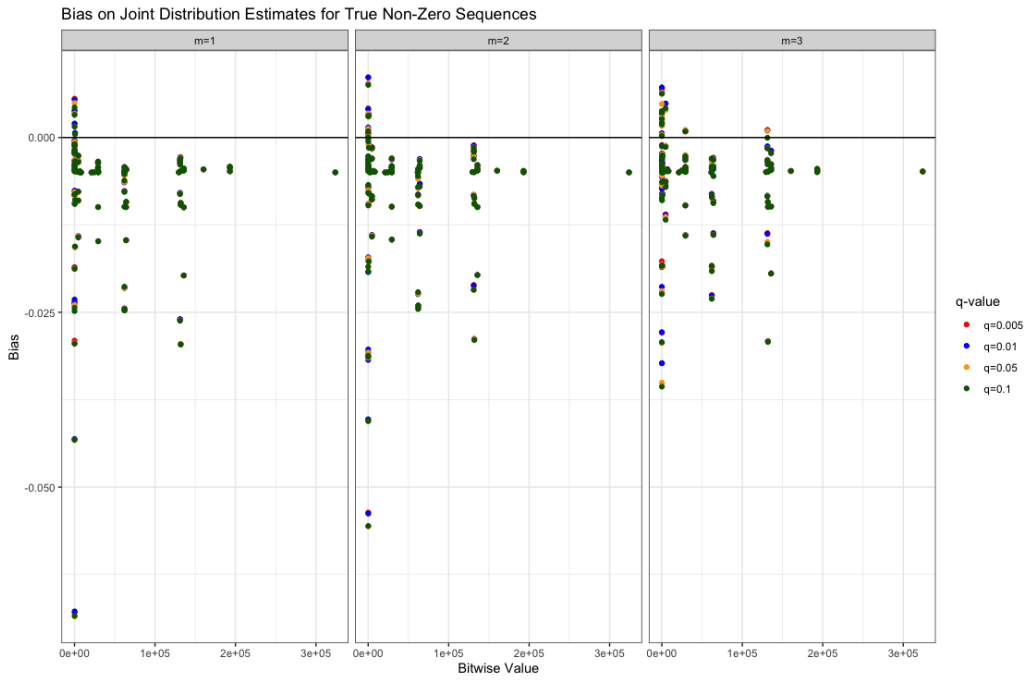

Figure 4.16: Bias for the true non-zero sequences across the 100 replications for the Order-1, Order-2, and Order-3 joint distribution reconstructions. Red points represent simulations where $q = 0.005$, blue points represent simulations where $q = 0.01$, yellow points represent simulations where $q = 0.05$, and green points represent simulations where $q = 0.1$. The recombination probability does not seem to be related to the bias on the true non-zero sequences, however we can observe that increasing m decreases the bias overall.

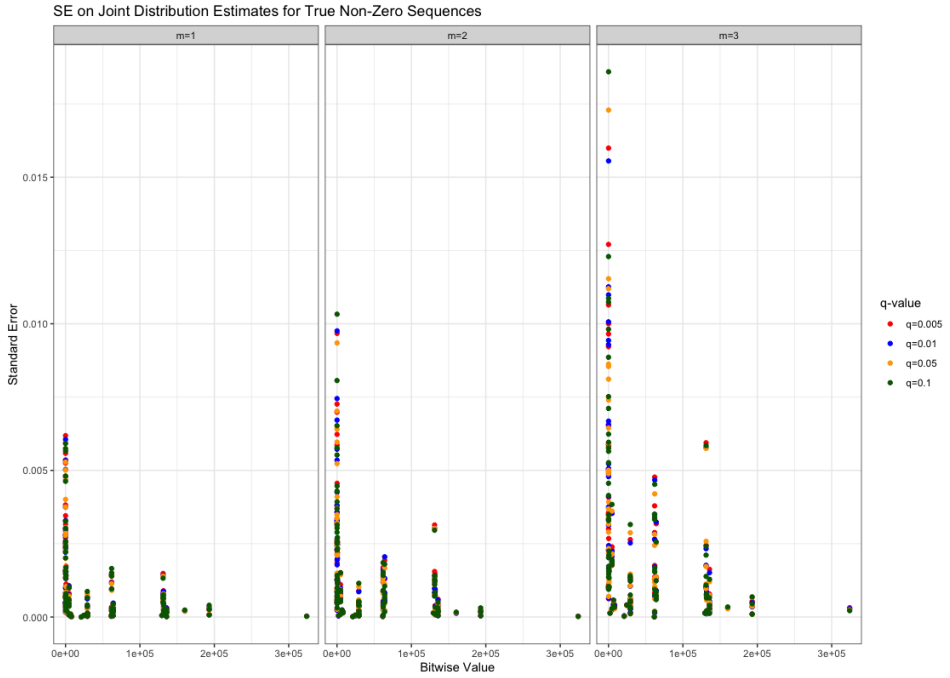

Figure 4.17: Empirical standard error for the true non-zero sequences across the 100 replications for the Order-1, Order-2, and Order-3 joint distribution reconstructions. Red points represent simulations where $q = 0.005$, blue points represent simulations where $q = 0.01$, yellow points represent simulations where $q = 0.05$, and green points represent simulations where $q = 0.1$. The recombination probability does not seem to be related to the standard error on the true non-zero sequences, however we can observe that increasing m increases the standard error overall.

q	Non-Zeros	95% Neighbors	90% Neighbors	85% Neighbors
0.005	$(m = 1)$ 0.2367 $(m = 2)$ 0.2711 $(m = 3)$ 0.43197	$(m = 1)$ 0.6919 $(m = 2)$ 0.7172 $(m = 3)$ 0.85563	$(m = 1)$ 0.94275 $(m = 2)$ 0.9608 $(m = 3)$ 0.985253	$(m = 3)$ 0.9954 $(m = 2)$ 0.9981 $(m = 3)$ 0.99945
0.01	$(m = 1)$ 0.2361 $(m = 2)$ 0.2700 $(m = 3)$ 0.42971	$(m = 1)$ 0.6906 $(m = 2)$ 0.7150 $(m = 3)$ 0.85231	$(m = 1)$ 0.9422 $(m = 2)$ 0.9599 $(m = 3)$ 0.98419	$(m = 3)$ 0.9954 $(m = 2)$ 0.9980 $(m = 3)$ 0.99942
0.05	$(m = 1)$ 0.2310 $(m = 2)$ 0.2620 $(m = 3)$ 0.41315	$(m = 1)$ 0.6825 $(m = 2)$ 0.7037 $(m = 3)$ 0.83816	$(m = 1)$ 0.9391 $(m = 2)$ 0.9551 $(m = 3)$ 0.97958	$(m = 3)$ 0.9949 $(m = 2)$ 0.9975 $(m = 3)$ 0.99903
0.10	$(m = 1)$ 0.2276 $(m = 2)$ 0.2579 $(m = 3)$ 0.4038	$(m = 1)$ 0.6768 $(m = 2)$ 0.6971 $(m = 3)$ 0.8258	$(m = 1)$ 0.9361 $(m = 2)$ 0.9514 $(m = 3)$ 0.97499	$(m = 3)$ 0.9943 $(m = 2)$ 0.9970 $(m = 3)$ 0.9986

Table 4.1: Total density sums from the joint distribution simulation results. See Table 3.2 for the definitions of the 95%, 90%, and 85% Neighbors. We can observe that increasing m increases the proportion of the density which is accounted for by the true non-zero sequences only. However, we can also observe that, even at the smallest m-value, nearly all of the density is accounted for by the true non-zero sequences and their 85% Neighbors. The recombination probability, q, does not seem to play a significant role in the proportion of the density that is assigned to these different categories of sequences.

mation on the sum of the density assigned to these 91 true non-zero sequences. On the true non-zero sequences alone, we observe that only about 23% of the joint distribution's density is assigned. However, this increases notably when we include neighbor sequences following the procedure described in Section 3.5.4. When we include the 95% neighbors, the proportion of the joint distribution's density accounted for increases more than twofold to about 69%. When we include the 85% neighbors, nearly all of the joint distribution's density is accounted for even at this smallest Markov chain value (about 99.5%). See Table 4.1. From this we conclude that, even though the tendency towards underestimation is most extreme for this smallest value of m, the misplaced density is assigned to sequences which differ by at most 3 sites from a true non-zero sequence.

4.2.2 Order-2 Markov Chain Reconstruction

For the order-2 Markov chain reconstruction of the joint distribution, we observed lower magnitude bias compared with the order-1 Markov chain reconstruction. However, the empirical standard error is higher. See Figure 4.16, Figure 4.17. Similarly as was observed with the order-1 Markov chain reconstruction, the majority of the bias observed on the 91 true non-zero sequences was negative. Therefore, in estimating these 91 sequences, there is a tendency towards underestimating the probability (by as much as 0.051). This underestimation is less extreme, but there is more variability.

The effect of this tendency towards underestimation can be observed in the sum of the density assigned to the 91 true non-zero sequences. On the true non-zero sequences alone, we observe that only about 26% of the joint distribution's density is accounted for. However, we can also note that this is a slight increase from what was observed for the order-1 Markov chain reconstruction. The proportion of the density assigned also increases when we include neighbor sequences following the procedure described in Section 3.5.4. When we include the 95% neighbors, we observe that the proportion of the density accounted for more than doubles to about 70%. This is similar to the increase observed for the order-1 Markov chain reconstruction. When we include the 85% neighbors, nearly all of the joint distribution's density is accounted for (about 99.75%). From this we conclude that, even though there is a tendency towards underestimation, the misplaced density is assigned to sequences which differ by at most 3 sites from a true non-zero sequence.

4.2.3 Order-3 Markov Chain Reconstruction

For the order-3 Markov chain reconstruction of the joint distribution, we observed the lowest magnitude bias compared with the order-1 and order-2 Markov chain reconstructions of the joint distribution (see Figure 4.16). However, we also observed the highest empirical standard error (see Figure 4.17). Similarly as was observed with the two lower order Markov chain reconstructions, the majority of the biases observed on the 91 true non-zero sequences were negative indicating a tendency towards underestimation of the probabilities assigned to those sequences. This underestimation was by as much as 0.036, which is an improvement over the two previous reconstructions, but these estimates contain the most variability.

We can observe the effect of this tendency towards underestimation on the sum of the density assigned to the 91 true non-zero sequences. On the true non-zero sequences alone, we observe that only about 40% of the joint distribution's density is assigned. However, there is a direct correspondence between the degree of underestimation and the proportion of the density assigned to these 91 sequences. The order-3 Markov chain reconstruction has the smallest magnitude bias compared with the order-2 and order-1 Markov chain reconstructions, and the order-3 Markov chain reconstruction also had the highest proportion of density assigned to the true non-zero sequences. In addition, when we include the 95% neighbors, the proportion of the joint distribution's density accounted for increases by more than twofold to about 83%. When we include the 85% neighbors, nearly all of the joint distribution's density is accounted for (about 99.9%). From this we conclude that, even though there is a tendency towards underestimation, the misplaced density is assigned to se-

quences which differ by at most 3 sites from a true non-zero sequence. Further, the success of the order-3 Markov chain in recovering nearly all of the density when we allow for 3 sites to be different provides preliminary evidence that, in the trade-off between bias and empirical standard error we observed in the selection of the m-value, the benefits in bias afforded by choosing a larger value of m may outweigh the downside of accepting a larger amount of variance in the estimates.

Chapter 5

R Package

To implement the Recombination Model, we proceed using hierarchical estimation. This means, computationally, that to obtain joint distribution estimates reconconstructed from an order-m Markov chain, we must

1. Convert the SNP data from the four nucelotide bases (A, G, T, C) to binary sequences,

2. Estimate the onewise marginal distribution, the pairwise marginal distribution, and all marginal distributions to and including the $(m+1)$-wise marginal distribution,

3. Use the $(m+1)$-wise marginal distribution and the m-wise marginal distribution to reconstruct the joint distribution.

The **recombinationMCCL** package therefore includes the four following elements,

1. Built-in binary SNP haplotype data from the YRI population TRIOs International HapMap Project data,

2. Functions that perform the marginal distribution estimation,

3. Functions that reconstruct the joint distribution estimation,

4. Functions that can be used to simulate ancestor distributions and descendant samples.

5.1 Built-in Data

The International HapMap Project has SNP haplotype data publicly available on their website. This data includes: (**1**) the variable `rsID` which is a unique ID given to the SNP site, (**2**) the variable `positionb36` which represents a distance on the chromosome measured in centromeres, and (**3**) the variables which are written in the format `NA12345_A` each of which represents a SNP haplotype, where the variable name indexes the family code and haplotype. For our purposes, the family code and haplotype aren't of interest– in this phased version of the data, only the parents' haplotypes are included so this data includes only unrelated individuals. Looking at the data below, we can see that for each SNP location on each haplotype the nucleotide base has been recorded:

```
rsID position_b36 NA19095_A NA19095_B NA19096_A
  NA19096_B NA18867_A NA18867_B NA18868_A NA18868_B
  NA18924_A NA18924_B NA18923_A NA18923_B NA18488_A
  NA18488_B NA18486_A NA18486_B
rs10458597 554484 C C C C C C C C C C C C C C C C
rs2185539 556738 C C C C C C C T C C C C C C C C
rs11240767 718814 C C C T C C C C T C C C C C C T
```

```
rs12564807 724325 A A A A A A A A A A A A A A A A
rs3131972 742584 A A A A A A A G A A A A A A A G
rs3131969 744045 A A G A A A A G A G A G G A G G
rs3131967 744197 T T C T T T T C T T T C C T C C
rs1048488 750775 C C T C C C C T C T T C T C T T
rs12562034 758311 G G G G G G G G G G G G G G G G
rs12124819 766409 A A A A A A A A A A A A A A A A
```

You will recall that, for the purposes of this model, we will convert this data into binary sequences where 0 is the major allele and 1 is the minor allele. Consider the `rs3131969` row of the data. We can see that `A` is the major allele, or the most frequent allele on this site. Therefore we recode each `A` with a 0. This makes `G` the minor allele, so we recode each `G` with a 1. Following such a process for each SNP site gives us data in the following form:

```
NA19095_A NA19095_B NA19096_A NA19096_B NA18867_A
   NA18867_B NA18868_A NA18868_B NA18924_A NA18924_B
   NA18923_A NA18923_B NA18488_A NA18488_B NA18486_A
   NA18486_B
rs10458597 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
rs2185539 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
rs11240767 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 1
rs12564807 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
rs3131972 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1
rs3131969 0 0 1 0 0 0 0 1 0 1 0 1 1 0 1 1
rs3131967 0 0 1 0 0 0 0 1 0 0 0 1 1 0 1 1
rs1048488 0 0 1 0 0 0 0 1 0 1 1 0 1 0 1 1
```

```
rs12562034 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
rs12124819 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Displayed here are the first 10 rows of the data. For the data to be properly formatted for use in the remainder of the functions available in package, we then take the transpose so that the rows correspond to one haplotype and the columns correspond to one SNP site.

```
rs10458597 rs2185539 rs11240767 rs12564807 rs3131972
   rs3131969 rs3131967 rs1048488 rs12562034 rs12124819
NA19095_A 0 0 0 0 0 0 0 0 0 0
NA19095_B 0 0 0 0 0 0 0 0 0 0
NA19096_A 0 0 0 0 0 1 1 1 0 0
NA19096_B 0 0 1 0 0 0 0 0 0 0
NA18867_A 0 0 0 0 0 0 0 0 0 0
NA18867_B 0 0 0 0 0 0 0 0 0 0
NA18868_A 0 0 0 0 0 0 0 0 0 0
NA18868_B 0 1 0 0 1 1 1 1 0 0
NA18924_A 0 0 1 0 0 0 0 0 0 0
NA18924_B 0 0 0 0 0 1 0 1 0 0
NA18923_A 0 0 0 0 0 0 0 1 0 0
NA18923_B 0 0 0 0 0 1 1 0 0 0
NA18488_A 0 0 0 0 0 1 1 1 0 0
NA18488_B 0 0 0 0 0 0 0 0 0 0
NA18486_A 0 0 0 0 0 1 1 1 0 0
NA18486_B 0 0 1 0 1 1 1 1 0 0
```

The full data frame is available in the package stored as `yri_trio_1`.

5.2 Marginal Estimation Functions

5.2.1 Onewise Estimates

The function `estimates_m0` is used to calculate onewise estimates. This function takes three parameters:

1. `n` is the number of descendants in the sample
2. `L` is the length of the SNP sequences in the sample
3. `descendents` is a data frame of descendant samples structured such that each row represents one descendent and each column represents on SNP site (the number of rows should be equal to `n` and the number of columns should be equal to `L`)

The onewise estimates are calculated following Equation 2.13. The output is a matrix with two rows, one for $\hat{\pi}_s(0)$ and one for $\hat{\pi}_s(1)$. It has L columns, one for each of the sites $s = 1, \ldots, L$.

For example, the output generated for the $L = 10$ subset of the `yri_trio_1` is

```
onewise <- estimates_m0(descendents = yri_trio_1[(1:16)
   ,(1:10)], L=10, n=16)
onewise
```

```
site_1 site_2 site_3 site_4 site_5 site_6 site_7 site_8
   site_9 site_10
pi_0 1 0.9375 0.8125 1 0.875 0.5625 0.625 0.5625 1 1
pi_1 0 0.0625 0.1875 0 0.125 0.4375 0.375 0.4375 0 0
```

This tells us that $\hat{\pi}_3(0) = 0.8125$, or that the estimated marginal probability that the ancestor has a 0 on site 3 is 81.25% for this subset of the data.

5.2.2 Pairwise Estimates

The function `estimates_m1` is used to calculate the pairwise marginal estimates. Since the pairwise estimates are dependent on the onewise marginal estimates, this function involves an unseen call to `estimates_m0`. This function takes four parameters:

1. `L` is the length of the SNP sequences in the sample
2. `q` is the chosen probability of recombination (value between 0 and 1)
3. `n` is the number of descendants in the sample
4. `d` is a data frame of descendant samples structured such that each row represents one descendent and each column represents one SNP (the number of rows should be equal to `n` and the number of columns should be equal to `L`)

The pairwise estimates are calculated following Equation 2.9. The output is a matrix with

1. four rows; $\hat{\pi}_{s,s+1}(0,0)$, $\hat{\pi}_{s,s+1}(0,1)$, $\hat{\pi}_{s,s+1}(1,0)$, and $\hat{\pi}_{s,s+1}(1,1)$
2. $L-1$ columns; each column represents one of the sites $s = 1, \ldots, L-1$

For example, the output generated for the $L = 10$ subset of the `yri_trio_1` with a recombination probability of $q = 0.01$ is

```
pairwise <- estimates_m1(L=10, q=0.01, n=16, d=yri_trio_
    1[(1:16),(1:10)])
pairwise
```

```
site_1 site_2 site_3 site_4 site_5 site_6 site_7
    site_8 site_9
pi_0,0 0.9375 0.75 0.8125 0.875 0.5625 0.5625
    0.5014993687 0.5625 1
pi_0,1 0.0625 0.1875 0 0.125 0.3125 0 0.1235006313 0 0
pi_1,0 0 0.0625 0.1875 0 0 0.0625 0.0610006313 0.4375 0
pi_1,1 0 0 0 0 0.125 0.375 0.3139993687 0 0
```

This tells us that $\hat{\pi}_{2,3}(0,0) = 0.75$, or that the estimated marginal probability that the ancestor has a 0 on site 2 and a 0 on site 3 is 75% for this subset of the data when the recombination probability is 0.01.

5.2.3 Threewise Estimates

The function `estimates_m2` is used to calculate the threewise marginal estimates. Since the threewise marginal estimates are dependent on the onewise and pairwise marginal estimates, this function includes unseen calls to `estimates_m0` and `estimates_m1`. This function takes four parameters:

1. `L` is the length of the SNP sequences in the sample
2. `q` is the chosen probability of recombination (a value between 0 and 1)
3. `n` is the number of descendants in the sample

4. `descend` is a data frame of descendant samples structured such that each row represents one descendant and each column represents one SNP site (the number of rows should be equal to `n` and the number of columns should be equal to `L`)

The threewise estimates are calculated following Equation 2.10. The calculation is done via ten helper functions which calculate the cubic equation coefficients, calculate the roots of the cubic equation, and select which of these roots are real. The output is a matrix with

1. eight rows; $\hat{\pi}_{s,s+1,s+2}(0,0,0)$, $\hat{\pi}_{s,s+1,s+2}(0,0,1)$, $\hat{\pi}_{s,s+1,s+2}(0,1,0)$, $\hat{\pi}_{s,s+1,s+2}(0,1,1)$, $\hat{\pi}_{s,s+1,s+2}(1,0,0)$,$\hat{\pi}_{s,s+1,s+2}(1,0,1)$, $\hat{\pi}_{s,s+1,s+2}(1,1,0)$, and $\hat{\pi}_{s,s+1,s+2}(1,1,1)$
2. $L-2$ columns; each column represents one of the sites $s = 1, \ldots, L-2$

For example, the output generated for the $L = 10$ subset of the `yri_trio_1` with a recombination probability of $q = 0.01$ is

```
threewise <- estimates_m2(L=10, q=0.01, n=16, descend =
   yri_trio_1[(1:16), (1:10)])
threewise
```

```
site_1 site_2 site_3 site_4 site_5 site_6 site_7 site_8
pi_000 0.75 0.75 0.750793127 0.5625 0.5625 0.5014993687
   0.5014993687 0.5625
pi_001 0.1875 0 0.061706873 0.3125 0 0.0610006313 0 0
pi_010 0.0625 0.1875 0 0 0.0625 0 0.1235006313 0
pi_011 0 0 0 0.125 0.25 0 0 0
pi_100 0 0.0625 0.124206873 0 0 0 0.0610006313 0.4375
```

```
pi_101 0 0 0.063293127 0 0 0.0625 0 0
pi_110 0 0 0 0 0 0.0610006313 0.3139993687 0
pi_111 0 0 0 0 0.125 0.3139993687 0 0
```

This output tells us that $\hat{\pi}_{2,3,4}(0,0,0) = 0.75$, or that the estimated marginal probability that the ancestor has a 0 on site 2, 0 on site 3, and 0 on site 4 is 75% for this subset of the data when the recombination probability is 0.01.

5.2.4 Fourwise Estimates

The function `estimates_m3` is used to calculate the fourwise marginal estimates. Since the fourwise marginal estimates are dependent on the onewise, pairwise, and threewise marginal estimates, this function includes unseen calls to `estimates_m0`, `estimates_m1`, and `estimates_m2`. This function takes four parameters:

1. `L` is the length of the SNP sequences in the sample
2. `q` is the chosen probability of recombination (a value between 0 and 1)
3. `n` is the number of descendants in the sample
4. `d` is a data frame of the descendant samples structured such that each row represents one descendant and each column represents one SNP site (the number of rows should be equal to `n` and the number of columns should be equal to `L`)

The fourwise estimates are calculated following Equation 2.17. The calculation is done via 14 helper functions which calculate the cubic equation coefficients, calculate the roots of those cubic equations, and selects which of the roots are real. The output is a matrix with

1. sixteen rows: $\hat{\pi}_{s,s+1,s+2,s+3}(0,0,0,0)$, $\hat{\pi}_{s,s+1,s+2,s+3}(0,0,0,1)$, $\hat{\pi}_{s,s+1,s+2,s+3}(0,0,1,0)$, $\hat{\pi}_{s,s+1,s+2,s+3}(0,0,1,1)$, $\hat{\pi}_{s,s+1,s+2,s+3}(0,1,0,0)$, $\hat{\pi}_{s,s+1,s+2,s+3}(0,1,0,1)$, $\hat{\pi}_{s,s+1,s+2,s+3}(0,1,1,0)$, $\hat{\pi}_{s,s+1,s+2,s+3}(0,1,1,1)$, $\hat{\pi}_{s,s+1,s+2,s+3}(1,0,0,0)$, $\hat{\pi}_{s,s+1,s+2,s+3}(1,0,0,1)$, $\hat{\pi}_{s,s+1,s+2,s+3}(1,0,1,0)$, $\hat{\pi}_{s,s+1,s+2,s+3}(1,0,1,1)$, $\hat{\pi}_{s,s+1,s+2,s+3}(1,1,0,0)$, $\hat{\pi}_{s,s+1,s+2,s+3}(1,1,0,1)$, $\hat{\pi}_{s,s+1,s+2,s+3}(1,1,1,0)$, and $\hat{\pi}_{s,s+1,s+2,s+3}(1,1,1,1)$
2. $L-3$ columns; each column represents one of the sites $s = 1, \ldots, L-3$

For example, the output generated for the $L = 10$ subset of the `yri_trio_1` with a recombination probability of $q = 0.01$ is

```
fourwise <- estimates_m3(L=10, q=0.01, n=16, d =
   yri_trio_t[(1:16), (1:10)])
fourwise
```

```
site_1 site_2 site_3 site_4 site_5 site_6 site_7
pi_0000 0.75 0.75 0.438293127 0.5625 0.5014993687
   0.5014993687 0.5014993687
pi_0001 0 0 0.3125 0 0.0610006313 0 0
pi_0010 0.1875 0 0 0.0625 0 0.0610006313 0
pi_0011 0 0 0.061706873 0.25 0 0 0
pi_0100 0.0625 0.124206873 0 0 0 0 0.1235006313
pi_0101 0 0.063293127 0 0 0.0625 0 0
pi_0110 0 0 0 0 0.0610006313 0 0
pi_0111 0 0 0 0.125 0.1889993687 0 0
```

```
pi_1000 0 0.000793127 0.124206873 0 0 0 0.0610006313
pi_1001 0 0.061706873 0 0 0 0 0
pi_1010 0 0 0 0 0 0.0625 0
pi_1011 0 0 0.063293127 0 0 0 0
pi_1100 0 0 0 0 0 0.0610006313 0.3139993687
pi_1101 0 0 0 0 0 0 0
pi_1110 0 0 0 0 0 0.3139993687 0
pi_1111 0 0 0 0 0.125 0 0
```

This output tells us that $\hat{\pi}_{2,3,4,5}(0,0,0,0) = 0.75$, or that the estimated marginal probability that the ancestor has 0 on site 2, 0 on site 3, 0 on site 4, and 0 on site 5 is 75% for this subset of the data when the recombination probability is 0.01.

5.3 Joint Estimation Functions

5.3.1 Order-1 Reconstruction

To estimate the joint distribution from the pairwise marginal distribution, use the function `ancestor_pair_estimation`. According to Equation (2.4), this will be based on the pairwise marginal estimates (from `estimates_m1`) in the numerator and the onewise marginal estimates (from `estimates_m0`) in the denominator. This function takes three parameters,

1. `L` is the length of the SNP sequences in the sample
2. `pairs_est` is a data frame of the pairwise marginal estimates, the output of `estimates_m1`

3. `ones_est` is a data frame of the onewise marginal estimates, the output of `estimates_m0`

The output of this function is a data frame with one column and 2^L rows. Each row represents one of the estimates $\pi_{1,\ldots,L}(x_1, \ldots, x_L)$. These estimates are ordered according to their bitwise value.

For example, a portion of the output generated for the $L = 10$ subset of the `yri_trio_1` with a recombination probability of $q = 0.01$ is

```
m1_joint <- ancestor_pair_estimation(L=10, m=1,
   pairs_est = pairwise, ones_est = onewise)
m1_joint_subset <- as.matrix(m1_joint[(1:10),])
m1_joint_subset
```

```
Probability
1  0.3385120738725
2  0
3  0
4  0
5  0.0833629261275
6  0
7  0
8  0
9  0
10  0
```

From this output, we read that $\hat{\pi}_{1,\ldots,10}(0,0,0,0,0,0,0,0,0,0) \approx 0.339$. Therefore, for this subset of the data when $q = 0.01$, the estimated population frequency with which the ancestor has 0 on all of $s = 1, \ldots, 10$ is about 33.9 %

when reconstructed from an $m = 1$ Markov chain.

5.3.2 Order-2 Reconstruction

To estimate the joint distribution from the threewise marginal distribution, use the function `ancestor_three_estimation`. According to Equation (2.4), this will be based on the threewise marginal estimates (from `estimates_m2`) in the numerator and the onewise marginal estimates (from `estimates_m1`) in the denominator. This function takes three parameters,

1. `L` is the length of the SNP sequences in the sample
2. `three_est` is a data frame of the threewise marginal estimates, the output of `estimates_m2`
3. `pairs_est` is a data frame of the pairwise marginal estimates, the output of `estimates_m1`

The output of this function is a data frame with one column and 2^L rows. Each row represents one of the estimates of $\pi_{1,\ldots,L}(x_1, \ldots, x_L)$. These estimates are ordered according to their bitwise value.

For example, a portion of the output generated for the $L = 10$ subset of the `yri_trio_1` with a recombination probability of $q = 0.01$ is

```
m2_joint <- ancestor_three_estimation(L=10, m=2,
   three_est = threewise, pairs_est = pairwise)
m2_joint_subset <- as.matrix(m2_joint[(1:10),])
m2_joint_subset
```

```
Probability
```

```
1  0.397210316534293
2  0
3  0
4  0
5  0.0483152753118605
6  0
7  0
8  0
9  0
10  0
```

From this output, we read that $\hat{\pi}_{1,\ldots,10}(0,0,0,0,0,0,0,0,0,0) \approx 0.397$. Therefore, for this subset of the data when $q = 0.01$, the estimated population frequency with which the ancestor has 0 on all of $s = 1, \ldots, 10$ is about 39.7 % when reconstructed from an $m = 2$ Markov chain.

5.3.3 Order-3 Reconstruction

To estimate the joint distribution from the fourwise marginal distribution, use the function `ancestor_four_estimation`. According to Equation (2.4), this will be based on the fourwise marginal estimates (from `estimates_m3`) in the numerator and the threewise marginal estimates (from `estimates_m2`) in the denominator. This function takes three parameters,

1. `L` is the length of the SNP sequences in the sample
2. `four_est` is a data frame of the fourwise marginal estimates, the output of `estimates_m3`

3. `three_est` is a data frame of the threewise marginal estimates, the output of `estimates_m2`

The output of this function is a data frame with one column and 2^L rows. Each row represents one of the estimates of $\pi_{1,\ldots,L}(x_1, \ldots, x_L)$. These estimates are ordered according to their bitwise value.

For example, a portion of the output generated for the $L = 10$ portion of `yri_trio_1` with a recombination probability of $q = 0.01$ is

```
m3_joint <- ancestor_four_estimation(L=10, m=3, four_est
    = fourwise, three_est = threewise)
m3_joint_subset <- as.matrix(m3_joint[(1:10),])
m3_joint_subset
```

```
Probability
1  0.390349384921934
2  0
3  0
4  0
5  0.0474807355581117
6  0
7  0
8  0
9  0
10  0
```

From this output, we read that $\hat{\pi}_{1,\ldots,10}(0,0,0,0,0,0,0,0,0,0) \approx 0.390$. Therefore, for this subset of the data when $q = 0.01$, the estimated population frequency with which the ancestor has 0 on all of $s = 1, \ldots, 10$ is about 39 %

when reconstructed from an $m = 3$ Markov chain.

5.4 Functions to Simulate Data

5.4.1 Simulate the Ancestor Distribution from Data

The function `ancestor` can be used to simulate the distribution of possible ancestor sequences. The simulated probabilities are based on a set of SNP haplotype data. This function takes two parameters,

1. `L` is the length of the ancestor sequences to be simulated
2. `hapmap_data` is a data frame structured such that each row corresponds to one SNP site and each column corresponds to one haplotype.

The output of this function is a data frame with one column and 2^L rows. Each row represents one of the possible ancestor sequences $\pi_{1,\ldots,L}(x_1, \ldots, x_L)$ and the value recorded for each sequence is the simulated probability that the ancestor has that particular sequence. The sequences are ordered by their bitwise value.

These probabilities are simulated such that the sample proportion is taken to be the probability. A subset of the data frame of SNP haplotype data passed to the function is taken so that the number of rows is equal to L. Then for each possible ancestor sequence,

$$\pi_{1,\ldots,L}(x_1, \ldots, x_L) = \frac{n_{1,\ldots,L}(x_1, \ldots, x_L)}{n}$$

where $n_{1,\ldots,L}(x_1, \ldots, x_L)$ is the number of haplotypes in the sample that has the particular sequence $(x_1, \ldots, x_L)$ and n is the total number of haplotypes in

the sequence (the number of columns in `hapmap_data`).

5.4.2 Simulate Descendant Sequences from Data

The function `descendant_sequences` can be used to simulate a sample of descendant sequences from a set of SNP haplotype data. This function takes five parameters,

1. `L` is the length of the descendant sequences to be simulated
2. `q` is the recombination probability to be used in the simulation
3. `n` is the number of descendant sequences to be simulated
4. `seed` is any numeric value to be used in the function for reproducibility of random results
5. `hapmap` is a data frame of SNP haplotype data structured such that each row corresponds to one SNP site and each column corresponds to one haplotype

The output of this function is a data frame with L columns, each representing a SNP site, and n rows, each representing a descendant.

These descendant sequences are simulated by first simulating an ancestral distribution returned from `ancestor`. To simulate a sequence, one of the possible 2^L sequences is randomly selected using the `sample` function such that the probability that each sequence is selected is the simulated probability returned by `ancestor` for that sequence. The selected sequence is used as the descendant sequence's "starting point". From this starting point, we simulate recombination events based on the q value passed to the function. There will

be $L-1$ possible locations for a recombination event (i.e. between sites 1 and 2, between sites 2 and 3, etc.). For each of these locations, we use the `rbinom` function to return either a `0` or `1` where the probability of a `1` is the recombination probability q. If a `1` is returned for location k (where $k = 1, \ldots, L-1$,) then for SNP sites $k+1, \ldots, L$ a different possible ancestor sequences is randomly selected and the new $x_{k+1}, \ldots, x_L$ replace those SNP sites. This is repeated for each of the n sequences to be simulated. For more details and examples, see Section 3.2.

Chapter 6

Conclusion

The Recombination Model proposed by Jianping Sun 2011 utilizes Markov chain composite likelihood to estimate the unknown distribution of possible ancestral binary SNP sequences from a sample of descendant binary SNP sequences while accounting for the probability of a recombination event occurring. This estimated unknown distribution of the possible ancestral binary SNP sequences provides estimated population frequencies with which the ancestor will have each possible binary sequence.

This model serves to address an important gap in genetics research. The current capabilities of DNA sequencing technologies allow researchers access to data which is both high quality and highly detailed. In particular, this data is specific enough to encode the nucleotide bases found at single nucleotide polymorphism (SNP) sites on an individual's respective haplotypes. Since genetic material is inherited, in simple terms, such that the offspring receives one haplotype each from each of the parents, being able to obtain data that separates genetic material into haplotypes provides important information about the inheritance of genetic material. Furthermore, SNPs are the sites on the

human genome where there is variability between individuals. Having data on these sites, therefore, provides researchers insight into how the variation in DNA, which results from processes such as mutation and recombination, contributes to the passing of genetic material. Understanding the passing of genetic material is crucial to understanding human evolutionary history, as well as identifying genes associated with increased likelihood of developing diseases such as diabetes. A current issue in answering these questions, however, is that high quality and highly detailed data is not available for past generations of humans. It is then necessary to be able to estimate backwards across generations, from the detailed data on descendants to the unknown ancestors, while accounting for biological complexities such as recombination events. The Recombination Model addresses this need (Jianping Sun 2011).

Simulation results for the Recombination Model show that this estimation process is able to provide useful estimates for the ancestral distribution of possible binary SNP sequences. One desirable property shown for the estimator is that, for the marginal distribution estimates, the estimates across 100 replications show both low bias and low empirical standard error. In particular, the true estimate consistently falls within the 95% bootstrapped confidence intervals for the average estimates. However, results also indicate potential pitfalls of this method which require further investigation. These include effects of sample size differences between the major and minor allele, and a directional effect which interacts with the recombination probability.

For the joint distribution, the estimates across the 100 replications also show relatively low bias and empirical standard error. However, as m increases, the bias decreases while the empirical standard error increases. An important conclusion from these results is that the estimator tends to underes-

timate the probabilities on the joint distribution. On all three of the order-1, order-2, and order-3 Markov chain reconstructions the majority of the true non-zero sequences show negative bias. As a result, less than half of the joint distribution's density is assigned to the true non-zero sequences. However, the misplaced density is assigned to sequences that are within a Hamming distance of 3 of a true non-zero sequences. This suggests that, even though there is a tendency towards underestimation, when the estimator "selects" the wrong sequence when assigning density, it tends to be a sequence that matches a true non-zero sequence on at least 85% of sites.

6.1 Simulation Conclusion in Genetic Context

Overall, the results of this simulation indicate that the Recombination Model proposed by Jianping Sun 2011 is an effective method for obtaining estimates of the ancestor's SNP sequence from their descendant's sequences. The results also suggest that the method is effective regardless of the recombination probability q that is used, or the use of small Markov chain orders.

Theoretical results relating to the use of Markov chain composite likelihood indicate that the joint distribution reconstruction should achieve the best estimates when the Markov chain order m approaches the SNP sequence length L (Xu and N. Reid 2011; Dillon and Lebanon 2010). The present simulation study, however, was able to show that the Recombination Model is able to obtain estimates with desirable properties even under the condition where the Markov chain order implemented is small ($m = 1, 2, 3$). In particular, these estimates were shown to have low bias, to have low empirical standard error, and to be able to assign density on the joint distribution to sequences with some

accuracy. We can therefore conclude that this estimator can provide useful estimates of the ancestor's possible SNP sequences even if it is not possible to implement a large Markov chain order.

Ultimately, the joint distribution results, and not the marginal distributions, are of practical relevance. We observed that there was a directional effect of the Markov chain definition when estimating the marginal distribution such that there was an interaction between the recombination probability and whether the free parameter being estimated included a minor allele ($x_s = 1$). Particularly, the largest bias on a free parameter when estimating the marginal distribution was observed for the free parameters containing the minor allele which are further right on the chain when $q = 0.1$. Then the smallest bias was observed for the free parameters containing only the major allele where s is close to 1 and $q = 0.005, 0.01,$ or 0.05. Therefore, we can see that for the marginal distributions the estimator's performance is dependent on the selected q value. However, it is also important to keep in mind that $q = 0.1$ was implemented as an extreme value of q. This is too large to be a realistic recombination probability in most applications, so it is unlikely that future researchers would choose such a high value. It is, though, something to take into account if using the method.

However, after the marginal distribution estimates were used to reconstruct the joint distribution, we observe no notable effect of the selected q value. The total density on the joint distribution assigned to the true non-zero sequences did not show much of a difference between the q-values– the difference in the total density was at most 0.03. (See Table 4.1). We can therefore tentatively conclude that, when the estimator is being used to obtain estimates of the SNP sequences that are most likely for the ancestor, the recombination probability

selected will not have a large effect on the quality of those estimates.

As we have already highlighted, the simulation results provide the important insight that this method tends to underestimate the probability associated with the true non-zero sequences on the joint distribution. The misplaced density is assigned to sequences which match a true non-zero sequence on 85% of sites. This has a practical interpretation in a genetic context.

The true non-zero sequences here represent the subset of binary sequences which are the actual possibilities for the ancestor. For example, it might be that the actual possibly sequences for the ancestor for $L = 5$ are

$$(0, 0, 0, 0, 0)$$

and,

$$(0, 1, 0, 0, 0).$$

Having about 85% of the sites match when $L = 5$ means that 4 of the sites must match. Then, in theory, the estimator may assign some probability to the following sequences as well:

$$(0, 0, 0, 0, 0),$$
$$(1, 0, 0, 0, 0),$$
$$(0, 1, 0, 0, 0),$$
$$(0, 0, 1, 0, 0),$$
$$(0, 0, 0, 1, 0),$$
$$(0, 0, 0, 0, 1).$$

While this does mean that the probabilities of the two actual possible sequences will be underestimated and some probability will be assigned to sequences which are not possible, only sequences which are very similar are being incorrectly included. Therefore, results can still be useful for drawing conclusions. In applications such as creating hierarchical trees of SNP inheritance, for example, it is evident that if any of the identified sequences were used as the ancestor's SNP sequence it would be accurate on the majority of the sites. This allows for important conclusions to still be drawn.

6.2 Limitations and Future Directions

The simulations implemented here have two major limitations, namely that we were limited in the length of SNP sequences L and the Markov chain orders m that we were able to investigate. In these simulations, we looked only at $L = 20$ and $m = 1, 2, 3$ due to programming restraints.

We were limited in the length of SNP sequences that we were able to consider because, as we increase L, the number of possible sequences increases exponentially (2^L sequences). Our method of programming requires us to store vectors and matrices with 2^L as a dimension. However, in `R`, a vector can store at most $2^{31} - 1$ elements (University 2022). We were also limited in RAM access for the present analysis. The sequence length of $L = 20$ was chosen to balance these two considerations. However, this is a limitation of the generalizability of the simulation conclusions. DNA sequences are in reality incredibly long, and thus, in most research applications, geneticists would likely be interested in estimating the ancestor SNP sequences for sequence lengths much longer than 20. So, while the simulation tells us that the estimator works well for $L = 20$,

we so far do not know how it would perform for longer L.

The limitations on the L values that could be run for the present simulations also leave an important question about the Hamming distance neighbors. We found that, with $L = 20$, nearly all of the joint distribution's density was recovered when we take the sum across the true non-zero sequences and their 85% neighbors, where the 85% neighbors are sequences within a Hamming distance of 3 with a true non-zero sequence. Since we do not have more than one sequence length to compare, it is unknown whether the density will be recovered for 85% neighbors at other sequence lengths. It could also be the case that a Hamming distance of 3 is an absolute threshold, and the percent similarity is irrelevant. This will be an important question to be addressed for this method in the future.

The Markov chain orders implemented were limited to $m = 1, 2,$ and 3 because of how programming intensive the method is as a result of the hierarchical nature. In order to estimate the $(m + 1)$-wise marginal distribution, we must be able to estimate the onewise marginal distribution, the pairwise marginal distribution, and so on up through the m-wise marginal distribution. In addition, as can be seen in Section 7.3, calculating the coefficients used in the cubic equation for a given m are quite complex and involve long summations as m increases. The process of creating functions that can calculate these coefficients, as well as the lower-order marginal estimates, is therefore very programming intensive.

Coding functions for m beyond three is important as our preliminary simulation results indicate that it plays an important role in the estimator's performance, but is beyond the scope of the present project. We found here that increasing m decreases the bias on the true non-zero sequences, with dramatic

improvements comparing $m = 1, 2, 3$. For example, about 20% of the joint density was assigned to the true non-zero sequences when $m = 1$ but this doubled to about 40% when $m = 3$. A question that could be investigated in the future would be whether the estimator's performance continues to improve this drastically for even larger m, or if the improvement levels off at some m.

In summary, the role of L and m, as well as their relationship, require further investigation in future simulation studies of this method. To do this, it is necessary to find a method for storing larger vectors in `R` and to program functions for larger m-values. It may also be possible to find a way to structure the code so that it is more generalizable and one function can implement multiple m values.

The results of our simulations also show a limitation of the Recombination Model. We found that, for the marginal distribution estimates, there was a notable increase in the bias for large q. An assumption for the Recombination Model is that there is a fixed probability of recombination such that the probability of recombination is equal for all sites on the sequence. However, this assumption is not necessarily biologically accurate. Recombination hotspots are sites on the human genome where the probability of recombination is higher than on the rest of the genome, and these hotspots are distributed non-uniformly on the genome (Paul, Nag, and Chakraborty 2016). Our finding that bias is larger for large q may indicate that this method of estimation wouldn't handle hotspots well, calling into question the validity of this simplifying model assumption. This is a question to be investigated further in future studies. Another possibility could be to allow q to be variable in the model, such as by assigning it to be a random variable with some specified distribution.

Another notable conclusion from our simulations was that there was a di-

rectional effect of defining the Markov chain left-to-right. Specifically, we found that both the bias and standard error accumulated for sites further right on the chain. It is important to note that we observed this on the marginal distribution results, and there is at this point not evidence that this property persists into the joint distribution estimates. Nonetheless, a potential future direction would be to explore reformulating the model to condition on sites to both the left and right within some physical distance.

Lastly, another conclusion of our simulations was that, on the marginal distribution results, the estimator seemed to perform differently for sequences which contained the minor allele. Specifically, their mean squared error values were less predictable in that the mean squared error was not similar for adjacent sites. As was discussed previously, this may be a result of the fact that the minor allele will by definition be more rare in the data and we defined the true probabilities with sample proportions. It is unclear at this point whether this yields any meaningful impacts on the estimator performance. A potential future direction to explore this preliminary finding would be to to alter the data-generating mechanism to include more sequences containing the minor allele. This could be done by starting with the International HapMap project data to get initial sequences, and then replace some values in the data with the minor allele through a randomization procedure.

Chapter 7

Appendix

7.1 Order-2 MC Cubic Coefficients

We show that $\frac{d}{d\phi_s(i)}\ell(\phi_s(i))$ can be rewritten as a cubic equation of $\phi_s(i)$, Equation (2.16). First not that for each $x_s, x_{s+1} \in \{0,1\}$ we can write

$$f(x_s, x_{s+1} = i, x_{s+2}) = a^s_{x_s,x_{s+2}}\phi_s(i) + b^s_{x_s,x_{s+2}}.$$

where

$$a^s_{00} = a^s_{11} = (1-q)^2, \quad a^s_{01} = a^s_{10} = -(1-q)^2,$$

$$b^s_{00} = q(1-q)\left[\hat{\pi}_s(0)\hat{\pi}_{s+1,s+2}(i,0) + \hat{\pi}_{s,s+1}(0,i)\hat{\pi}_{s+2}(0)\right] + q^2\hat{\pi}_s(0)\hat{\pi}_{s+1}(i)\hat{\pi}_{s+2}(0),$$

$$b^s_{01} = (1-q)^2\hat{\pi}_{s,s+1}(0,i) +$$
$$q(1-q)\left[\hat{\pi}_s(0)\hat{\pi}_{s+1,s+2}(i,1) + \hat{\pi}_{s,s+1}(0,i)\hat{\pi}_{s+2}(1)\right] + q^2\hat{\pi}_s(0)\hat{\pi}_{s+1}(i)\hat{\pi}_{s+2}(1)$$

$$b^s_{10} = (1-q)^2\hat{\pi}_{s+1,s+2}(i,0) +$$
$$q(1-q)\left[\hat{\pi}_s(1)\hat{\pi}_{s+1,s+2}(i,0) + \hat{\pi}_{s,s+1}(1,i)\hat{\pi}_{s+2}(0)\right] + q^2\hat{\pi}_s(1)\hat{\pi}_{s+1}(i)\hat{\pi}_{s+2}(0)$$

$$b^s_{11} = (1-q)^2\left[\hat{\pi}_{s+1}(i) - \hat{\pi}_{s,s+1}(0,i) - \hat{\pi}_{s+1,s+2}(i,0)\right]$$
$$q(1-q)\left[\hat{\pi}_s(1)\hat{\pi}_{s+1,s+2}(i,1) + \hat{\pi}_{s,s+1}(1,i)\hat{\pi}_{s+2}(1)\right] + q^2\hat{\pi}_s(1)\hat{\pi}_{s+1}(i)\hat{\pi}_{s+2}(1)$$

For example, consider $f(0,i,1)$:

$$\begin{aligned}
P(X=(0,i,1)) &= (1-q)^2\pi_{s,s+1,s+2}(0,i,1) \\
&\quad + q(1-q)\left[\pi_s(0)\pi_{s+1,s+2}(i,1) + \pi_{s,s+1}(0,i)\pi_{s+2}(1)\right] \\
&\quad + q^2\pi_s(0)\pi_{s+1}(i)\pi_{s+2}(1) \\
&= (1-q)^2\left(\pi_{s,s+1}(0,i) - \phi_s(i)\right) \\
&\quad + q(1-q)\left[\pi_s(0)\pi_{s+1,s+2}(i,1) + \pi_{s,s+1}(0,i)\pi_{s+2}(1)\right] \\
&\quad + q^2\pi_s(0)\pi_{s+1}(i)\pi_{s+2}(1) \\
&= -(1-q)^2\phi_s(i) + (1-q)^2\pi_{s,s+1}(0,i) \\
&\quad + q(1-q)\left[\pi_s(0)\pi_{s+1,s+2}(i,1) + \pi_{s,s+1}(0,i)\pi_{s+2}(1)\right] \\
&\quad + q^2\pi_s(0)\pi_{s+1}(i)\pi_{s+2}(1) \\
&= a^s_{01}\phi_s(i) + b^s_{01}.
\end{aligned}$$

Then,

$$\begin{aligned}\frac{d}{d\phi_s(i)}\ell(\phi_s(i)) &= \sum_{x_s=0}^{1}\sum_{x_{s+2}=0}^{1} n_{s,s+1,s+2}(x_s,i,x_{s+2})\log f(x_s,i,x_{s+2}) \\ &= \sum_{x_s=0}^{1}\sum_{x_{s+2}=0}^{1} n_{s,s+1,s+2}(x_s,i,x_{s+2})\frac{a^s_{x_s,x_{s+2}}}{f(x_s,i,x_{s+2}} \\ &= \frac{A_1^3\phi_s(i)^3 + A_2^3\phi_s(i)^2 + A_3^3\phi_s(i) + A_4^3}{\prod_{x_s=0}^{1}\prod_{x_{s+2}}^{1} a^s_{x_s,x_{s+2}}\phi_s(i) + b^s_{x_s,x_{s+2}}}\end{aligned}$$

where,

$$A_1^3 = \left(\sum_{x_s,x_{s+2}} n^s(x_s,i,x_{s+2})\right)\left(\prod_{x_s,x_{s+2}} a^s_{x_s,x_{s+2}}\right)$$

$$\begin{aligned}A_2^3 &= (n^s(0,i,0)+n^s(0,i,1)+n^s(1,i,0))a^s_{00}a^s_{01}a^s_{10}b^s_{11} \\ &+ (n^s(0,i,0)+n^s(0,i,1)+n^s(1,i,1))a^s_{00}a^s_{01}b^s_{10}a^s_{11} \\ &+ (n^s(0,i,0)+n^s(1,i,0)+n^s(1,i,1))a^s_{00}b^s_{01}a^s_{10}a^s_{11} \\ &+ (n^s(0,i,1)+n^s(1,i,0)+n^s(1,i,1))b^s_{00}a^s_{01}a^s_{10}a^s_{11}\end{aligned}$$

$$\begin{aligned}A_3^3 &= (n^s(0,i,0)+n^s(0,i,1))a^s_{00}a^s_{01}b^s_{10}b^s_{11} + (n^s(0,i,0)+n^s(1,i,0))a^s_{00}b^s_{01}a^s_{10}b^s_{11} \\ &+ (n^s(0,i,0)+n^s(1,i,1))a^s_{00}b^s_{01}b^s_{10}a^s_{11} + (n^s(0,i,1)+n^s(1,i,0))b^s_{00}a^s_{01}a^s_{10}b^s_{11} \\ &+ (n^s(0,i,1)+n^s(1,i,1))b^s_{00}a^s_{01}b^s_{10}a^s_{11} + (n^s(1,i,0)+n^s(1,i,1))b^s_{00}b^s_{01}a^s_{10}a^s_{11}\end{aligned}$$

$$\begin{aligned}A_4^3 &= n^s(0,i,0)a^s_{00}b^s_{01}b^s_{10}b^s_{11} + n^s(0,i,1)b^s_{00}a^s_{01}b^s_{01}b^s_{11} + n^s(1,i,0)b^s_{00}b^s_{01}a^s_{10}b^s_{11} \\ &+ n^s(1,i,1)b^s_{00}b^s_{01}b^s_{10}a^s_{11}\end{aligned}$$

and

$$n^s(x_s,i,x_{s+2}) = n_{s,s+1,s+2}(x_s,i,x_{s+2}).$$

7.2 Order-3 MC Cubic Coefficients

We show that $\frac{d}{d\phi_x(i,j)}\ell(\phi_s(i,j)$ can be equivalently solved as a cubic equation of $\phi_s(i,j)$, Equation (2.17). First note that for each $x_s, x_{s+3} \in \{0,1\}$ we can write

$$f(x_s, x_{s+1} = i, x_{s+2} = j, x_{s+3}) = a^s_{x_s,x_{s+3}}\phi_s(i,j) + b^s_{x_s,x_{s+3}}$$

where

$$a^s_{00} = a^s_{11} = (1-q)^3, \quad a^s_{01} = a^s_{10} = -(1-q)^3$$

$$\begin{aligned}
b^s_{00} &= q(1-q)^2\left[\hat{\pi}_s(0)\hat{\pi}_{s+1,s+2,s+3}(i,j,0) + \hat{\pi}_{s,s+1}(0,i)\hat{\pi}_{s+2,s+3}(j,0) + \hat{\pi}_{s,s+1,s+2}(0,i,j)\hat{\pi}_{s+3}(0)\right] \\
&\quad + q^2(1-q)[\hat{\pi}_s(0)\hat{\pi}_{s+1}(i)\hat{\pi}_{s+2,s+3}(j,0) + \hat{\pi}_{s,s+1}(0,i)\hat{\pi}_{s+2}(j)\hat{\pi}_{s+3}(0) \\
&\qquad\qquad + \hat{\pi}_s(0)\hat{\pi}_{s+1,s+2}(i,j)\hat{\pi}_{s+3}(0)] \\
&\quad + q^3\hat{\pi}_s(0)\hat{\pi}_{s+1}(i)\hat{\pi}_{s+2}(j)\hat{\pi}_{s+3}(0)
\end{aligned}$$

$$\begin{aligned}
b^s_{01} &= (1-q)^3\hat{\pi}_{s,s+1,s+2}(0,i,j) \\
&\quad + q(1-q)^2\left[\hat{\pi}_s(0)\hat{\pi}_{s+1,s+2,s+3}(i,j,1) + \hat{\pi}_{s,s+1}(0,i)\hat{\pi}_{s+2,s+3}(j,1) + \hat{\pi}_{s,s+1,s+2}(0,i,j)\hat{\pi}_{s+3}(1)\right] \\
&\quad + q^2(1-q)[\hat{\pi}_s(0)\hat{\pi}_{s+1}(i)\hat{\pi}_{s+2,s+3}(j,1) + \hat{\pi}_{s,s+1}(0,i)\hat{\pi}_{s+2}(j)\hat{\pi}_{s+3}(1) \\
&\qquad\qquad + \hat{\pi}_s(0)\hat{\pi}_{s+1,s+2}(i,j)\hat{\pi}_{s+3}(1)] \\
&\quad + q^3\hat{\pi}_s(0)\hat{\pi}_{s+1}(i)\hat{\pi}_{s+2}(j)\hat{\pi}_{s+3}(1)
\end{aligned}$$

$$\begin{aligned}
b^s_{10} &= (1-q)^3\hat{\pi}_{s+1,s+2,s+3}(i,j,0) \\
&\quad + q(1-q)^2\left[\hat{\pi}_s(1)\hat{\pi}_{s+1,s+2,s+3}(i,j,0) + \hat{\pi}_{s,s+1}(1,i)\hat{\pi}_{s+2,s+3}(j,0) + \hat{\pi}_{s,s+1,s+2}(1,i,j)\hat{\pi}_{s+3}(0)\right] \\
&\quad + q^2(1-q)[\hat{\pi}_s(1)\hat{\pi}_{s+1}(i)\hat{\pi}_{s+2,s+3}(j,0) + \hat{\pi}_{s,s+1}(1,i)\hat{\pi}_{s+2}(j)\hat{\pi}_{s+3}(0) \\
&\qquad\qquad + \hat{\pi}_s(1)\hat{\pi}_{s+1,s+2}(i,j)\hat{\pi}_{s+3}(0)] \\
&\quad + q^3\hat{\pi}_s(1)\hat{\pi}_{s+1}(i)\hat{\pi}_{s+2}(j)\hat{\pi}_{s+3}(0)
\end{aligned}$$

$$\begin{aligned}
b^s_{11} &= (1-q)^3\left[\hat{\pi}_{s+1,s+2}(i,j) - \hat{\pi}_{s,s+1,s+2}(0,i,j) - \hat{\pi}_{s+1,s+2,s+3}(i,j,0)\right] \\
&\quad + q(1-q)^2\left[\hat{\pi}_s(1)\hat{\pi}_{s+1,s+2,s+3}(i,j,1) + \hat{\pi}_{s,s+1}(1,i)\hat{\pi}_{s+2,s+3}(j,1) + \hat{\pi}_{s,s+1,s+2}(1,i,j)\hat{\pi}_{s+3}(1)\right] \\
&\quad + q^2(1-q)[\hat{\pi}_s(1)\hat{\pi}_{s+1}(i)\hat{\pi}_{s+2,s+3}(j,1) + \hat{\pi}_{s,s+1}(1,i)\hat{\pi}_{s+2}(j)\hat{\pi}_{s+3}(1) \\
&\qquad\qquad + \hat{\pi}_s(1)\hat{\pi}_{s+1,s+2}(i,j)\hat{\pi}_{s+3}(1)] \\
&\quad + q^3\hat{\pi}_s(1)\hat{\pi}_{s+1}(i)\hat{\pi}_{s+2}(j)\hat{\pi}_{s+3}(1)
\end{aligned}$$

Then,

$$\begin{aligned}\frac{d}{d\phi_s(i,j)}\ell(\phi_s(i,j)) &= \sum_{x_s=0}^{1}\sum_{x_{s+3}=0}^{1} n_{s,s+1,s+2,s+3}(x_s,i,j,x_{s+3})\log f(x_s,i,j,x_{s+3}) \\ &= \sum_{x_s}\sum_{x_{s+3}} n_{s,s+1,s+2,s+3}(x_s,i,j,x_{s+3})\frac{a^s_{x_s,x_{s+3}}}{f(x_s,i,j,x_{s+3})} \\ &= \frac{A_1^4\phi_s(i,j)^3 + A_2^4\phi_s(i,j)^2 + A_3^4\phi_s(i,j) + A_4^4}{\prod_{x_s=0}^{1}\prod_{x_{s+3}=0}^{1} a^s_{x_s,x_{s+3}}\phi_s(i,j) + b^s_{x_s,x_{s+3}}}\end{aligned}$$

where,

$$A_1^4 = \left(\sum_{x_s}\sum_{x_{s+3}} n^s(x_s,i,j,x_{s+3})\right)\left(\prod_{x_s}\prod_{x_{s+3}} a^s_{x_s,x_{s+3}}\right)$$

$$\begin{aligned}A_2^4 &= (n^s(0,i,j,0)+n^s(0,i,j,1)+n^s(1,i,j,0))a^s_{00}a^s_{01}a^s_{10}b^s_{11} \\ &+ (n^s(0,i,j,0)+n^s(0,i,j,1)+n^s(1,i,j,1))a^s_{00}a^s_{01}b^s_{10}a^s_{11} \\ &+ (n^s(0,i,j,0)+n^s(1,i,j,0)n^s(1,i,j,1))a^s_{00}b^s_{01}a^s_{10}a^s_{11} \\ &+ (n^s(0,i,j,1)+n^s(1,i,j,0)+n^s(1,i,j,1))b^s_{00}a^s_{01}a^s_{10}a^s_{11}\end{aligned}$$

$$\begin{aligned}A_3^4 &= (n^s(0,i,j,0)+n^s(0,i,j,1))a^s_{00}a^s_{01}b^s_{10}b^s_{11} + (n^s(0,i,j,0)+n^s(1,i,j,0))a^s_{00}b^s_{01}a^s_{10}b^s_{11} \\ &+ (n^s(0,i,j,0)+n^s(1,i,j,1))a^s_{00}b^s_{01}b^s_{10}a^s_{11} + (n^s(0,i,j,1)+n^s(1,i,j,0))b^s_{00}a^s_{01}a^s_{10}b^s_{11} \\ &+ (n^s(0,i,j,1)+n^s(1,i,j,1))b^s_{00}a^s_{01}b^s_{10}a^s_{11} + (n^s(1,i,j,0)+n^s(1,i,j,1))b^s_{00}b^s_{10}a^s_{10}a^s_{11}\end{aligned}$$

$$\begin{aligned}A_4^4 &= n^s(0,i,j,0)a^s_{00}b^s_{01}b^s_{10}b^s_{11} + n^s(0,i,j,1)b^s_{00}a^s_{01}b^s_{10}b^s_{11} \\ &+ n^s(1,i,j,0)b^s_{00}b^s_{01}a^s_{10}b^s_{11} + n^s(1,i,j,1)b^s_{00}b^s_{01}b^s_{10}a^s_{11}\end{aligned}$$

7.3 Order-m MC Cubic Coefficients ($m \geq 2$)

Solving the score equation $\frac{d}{d\phi_s(\underline{x})}$ to estimate the $(m+1)$-wise margins where $m \geq 2$ is equivalent to solving the cubic equation, Equation (2.18). For each

$x_s, x_{s+m} \in \{0, 1\}$,

$$f(x_s, \underline{x}, x_{s+m}) = a^s_{x_s, x_{s+m}} \phi_s(\underline{x}) + b^s_{x_s, x_{s+m}}$$

where

$$
\begin{aligned}
a^s_{00} &= a^s_{11} = (1-q)^m, \;\; a^s_{01} = a^s 10 = -(1-q)^m \\
b^s_{00} &= q(1-q)^{m-1}\left[\sum_{j=0}^{m-1} \hat{\pi}_{s,\ldots,s+j}(0,\ldots,s_{s+j})\hat{\pi}_{s+j+1,\ldots,s+m}(x_{s+j+1},\ldots,0)\right] \\
&\;\;\vdots \\
&\quad + q^m \hat{\pi}_s(0)\hat{\pi}_{s+1}(x_{s+1})\ldots\hat{\pi}_{s+m-1}(x_{s+m-1})\hat{\pi}_{s+m}(0) \\
&\quad = f(0,\underline{x},0) - a^s_{00}\phi_s(\underline{x}) \\
b^s_{01} &= (1-q)^m \hat{\pi}_{s,\ldots,s+m-1}(0,\underline{x}) \\
&\quad + q(1-q)^{m-1}\left[\sum_{j=0}^{m-1} \hat{\pi}_{s,\ldots,s+j}(0,\ldots,s_{s+j})\hat{\pi}_{s+j+1,\ldots,s+m}(x_{s+j+1},\ldots,1)\right] \\
&\;\;\vdots \\
&\quad + q^m \hat{\pi}_s(0)\hat{\pi}_{s+1}(x_{s+1})\ldots\hat{\pi}_{s+m-1}(x_{s+m-1})\hat{\pi}_{s+m}(1) \\
&\quad = f(0,\underline{x},1) - a^s_{01}\phi_s(\underline{x}) \\
b^s_{10} &= (1-q)^m \hat{\pi}_{s+1,\ldots,s+m}(\underline{x},0) \\
&\quad + q(1-q)^{m-1}\left[\sum_{j=0}^{m-1} \hat{\pi}_{s,\ldots,s+j}(1,\ldots,s_{s+j})\hat{\pi}_{s+j+1,\ldots,s+m}(x_{s+j+1},\ldots,0)\right] \\
&\;\;\vdots \\
&\quad + q^m \hat{\pi}_s(1)\hat{\pi}_{s+1}(x_{s+1})\ldots\hat{\pi}_{s+m-1}(x_{s+m-1})\hat{\pi}_{s+m}(0) \\
&\quad = f(1,\underline{x},0) - a^s_{10}\phi_s(\underline{x}) \\
b^s_{11} &= (1-q)^m\left[\hat{\pi}_{s+1,\ldots,s+m-1}(\underline{x}) - \hat{\pi}_{s,\ldots,s+m-1}(0,\underline{x}) - \hat{\pi}_{s+1,\ldots,s+m}(\underline{x},0)\right] \\
&\quad + q(1-q)^{m-1}\left[\sum_{j=0}^{m-1} \hat{\pi}_{s,\ldots,s+j}(1,\ldots,s_{s+j})\hat{\pi}_{s+j+1,\ldots,s+m}(x_{s+j+1},\ldots,1)\right] \\
&\;\;\vdots \\
&\quad + q^m \hat{\pi}_s(1)\hat{\pi}_{s+1}(x_{s+1})\ldots\hat{\pi}_{s+m-1}(x_{s+m-1})\hat{\pi}_{s+m}(1) \\
&\quad = f(1,\underline{x},1) - a^s_{11}\phi_s(\underline{x})
\end{aligned}
$$

Then,

$$\begin{aligned}\frac{d}{d\phi_s(\underline{x})}\ell(\phi_s(\underline{x})) &= \sum_{x_s=0}^{1}\sum_{x_{s+m}=0}^{1} n_{s,\ldots,s+m}(x_s,\underline{x},x_{s+m})\log f(x_s,\underline{x},x_{s+m}) \\ &= \sum_{x_s}\sum_{x_{s+m}} n_{s,\ldots,s+m}(x_s,\underline{x},x_{s+m})\frac{a^s_{x_s,x_{s+m}}}{f(x_s,\underline{x},x_{s+3})} \\ &= \frac{A^m_1\phi_s(\underline{x})^3 + A^m_2\phi_s(\underline{x})^2 + A^m_3\phi_s(\underline{x}) + A^m_4}{\prod_{x_s=0}^{1}\prod_{x_{s+m}=0}^{1} a^s_{x_s,x_{s+m}}\phi_s(\underline{x}) + b^s_{x_s,x_{s+m}}}\end{aligned}$$

where,

$$A^m_1 = \left(\sum_{x_s}\sum_{x_{s+m}} n^s(x_s,\underline{x},x_{s+m})\right)\left(\prod_{x_s}\prod_{x_{s+m}} a^s_{x_s,x_{s+m}}\right)$$

$$\begin{aligned}A^m_2 &= (n^s(0,\underline{x},0)+n^s(0,\underline{x},1)+n^s(1,\underline{x},0))a^s_{00}a^s_{01}a^s_{10}b^s_{11} \\ &+ (n^s(0,\underline{x},0)+n^s(0,\underline{x},1)+n^s(1,\underline{x},1))a^s_{00}a^s_{01}b^s_{10}a^s_{11} \\ &+ (n^s(0,\underline{x},0)+n^s(1,\underline{x},0)n^s(1,\underline{x},1))a^s_{00}b^s_{01}a^s_{10}a^s_{11} \\ &+ (n^s(0,\underline{x},1)+n^s(1,\underline{x},0)+n^s(1,\underline{x},1))b^s_{00}a^s_{01}a^s_{10}a^s_{11}\end{aligned}$$

$$\begin{aligned}A^m_3 &= (n^s(0,\underline{x},0)+n^s(0,\underline{x},1))a^s_{00}a^s_{01}b^s_{10}b^s_{11} + (n^s(0,\underline{x},0)+n^s(1,\underline{x},0))a^s_{00}b^s_{01}a^s_{10}b^s_{11} \\ &+ (n^s(0,\underline{x},0)+n^s(1,\underline{x},1))a^s_{00}b^s_{01}b^s_{10}a^s_{11} + (n^s(0,\underline{x},1)+n^s(1,\underline{x},0))b^s_{00}a^s_{01}a^s_{10}b^s_{11} \\ &+ (n^s(0,\underline{x},1)+n^s(1,\underline{x},1))b^s_{00}a^s_{01}b^s_{10}a^s_{11} + (n^s(1,\underline{x},0)+n^s(1,\underline{x},1))b^s_{00}b^s_{10}a^s_{10}a^s_{11}\end{aligned}$$

$$\begin{aligned}A^m_4 &= n^s(0,\underline{x},0)a^s_{00}b^s_{01}b^s_{10}b^s_{11} + n^s(0,\underline{x},1)b^s_{00}a^s_{01}b^s_{10}b^s_{11} \\ &+ n^s(1,\underline{x},0)b^s_{00}b^s_{01}a^s_{10}b^s_{11} + n^s(1,\underline{x},1)b^s_{00}b^s_{01}b^s_{10}a^s_{11}\end{aligned}$$

7.4 Simulated True Ancestor Distribution

As part of our simulation, we used the YRI population trios data from the International HapMap project to simulate a true ancestor distribution of possible binary sequences for length $L = 20$. This includes 1048576 possible sequences. Of this large number of total possible sequences, 91 of them were assigned a true non-zero probability. See Table 7.1 for these 91 sequences and their probability. Note that for these sequences, we identify them using their bitwise value such that

$$(x_1, x_2, \ldots, x_L) = \sum_{s=1}^{L} x_s 2^{L-s}.$$

Table 7.1: True non-zero probability sequences as simulated for the true joint distribution. The true joint distribution was simulated from the International HapMap Project YRI population Trios data such that the probability assigned to a given sequences is the sample proportion of that sequence. This resulted in 91 true sequences. Bitwise value is calculated as $(x_1, \ldots, x_L) = \sum_{s=1}^{L} x_s 2^{L-s}$.

Bitwise	**Probability**
0	0.025
3	0.05
8	0.035

Continued on next page

Table 7.1 – *Continued from previous page*

Bitwise	**Probability**
11	0.005
16	0.045
19	0.005
24	0.005
32	0.045
35	0.025
36	0.005
40	0.005
48	0.01
64	0.015
72	0.01
80	0.085
83	0.005
106	0.005
107	0.01
112	0.005
115	0.005
131	0.005
144	0.005
163	0.005
232	0.01
256	0.005
264	0.005

Continued on next page

Table 7.1 – *Continued from previous page*

Bitwise	**Probability**
579	0.02
595	0.005
771	0.01
776	0.005
2112	0.005
4610	0.005
4611	0.01
4616	0.005
4640	0.01
4688	0.005
4712	0.015
6659	0.005
6832	0.005
8195	0.005
21107	0.005
24579	0.005
29187	0.005
29216	0.005
29264	0.005
29280	0.005
29284	0.01
29291	0.005
29320	0.005

Continued on next page

Table 7.1 – *Continued from previous page*

Bitwise	**Probability**
29379	0.005
29387	0.005
29392	0.015
61586	0.005
61952	0.01
61968	0.01
61984	0.025
61985	0.005
61992	0.01
62027	0.005
62059	0.025
62088	0.005
62128	0.025
62131	0.01
64008	0.005
64036	0.01
64048	0.005
64072	0.005
64080	0.01
64083	0.015
129536	0.005
131091	0.01
131152	0.03

Continued on next page

Table 7.1 – *Continued from previous page*

Bitwise	**Probability**
131155	0.005
131168	0.005
131176	0.005
131592	0.005
131840	0.01
131848	0.03
131880	0.01
131944	0.005
135688	0.005
135728	0.02
135744	0.005
135760	0.005
135776	0.005
135780	0.01
160272	0.005
193056	0.005
193088	0.005
193099	0.005
324128	0.005

7.5 Simulated True Marginal Distributions

For the pairwise case, our simulated true values for the free parameter are specified in Table 7.2. For the threewise case, our simulated true values for the free parameters are specified in Table 7.3. For the fourwise case, our simulated true values for the free parameters are specified in Table 7.4.

s	ϕ_s
1	0.995
2	0.810
3	0.810
4	0.790
5	0.710
6	0.705
7	0.595
8	0.600
9	0.930
10	0.510
11	0.500
12	0.815
13	0.530
14	0.355
15	0.350
16	0.390
17	0.715
18	0.675
19	0.705

Table 7.2: Simulated True Marginal Pairwise Probabilities for the Free Parameter, ϕ_s. Recall that we assigned $\phi_s = \pi_{s,s+1}(0,0)$. Each of the marginal pairwise probabilities was calculated by taking the sum of the probabilities assigned to each joint distribution sequence where $x_s = 0, x_{s+1} = 0$. These values are used in the simulations to calculate the bias and mean squared error for the pairwise marginal results.

s	$\phi_s(0)$	$\phi_s(1)$
1	0.810	0.005
2	0.805	0.185
3	0.620	0.000
4	0.710	0.000
5	0.705	0.005
6	0.595	0.005
7	0.590	0.105
8	0.600	0.005
9	0.505	0.000
10	0.500	0.420
11	0.470	0.010
12	0.455	0.070
13	0.330	0.245
14	0.265	0.170
15	0.210	0.270
16	0.355	0.245
17	0.490	0.035
18	0.670	0.015

Table 7.3: Simulated True Marginal Threewise Probabilities for the Free Parameters, $\phi_s(0)$ and $\phi_s(1)$. Recall that we assigned $\phi_s(0) = \pi_{s,s+1,s+2}(0,0,0)$ and $\phi_s(1) = \pi_{s,s+1,s+2}(0,1,0)$. Each of the true marginal threewise probabilities was calculated by taking the sum of the probabilities assigned to each joint distribution sequence where $x_s = 0, x_{s+1} = 0, x_{s+2} = 0$ and $x_s = 0, x_{s+1} = 1, x_{s+2} = 0$ respectively. These values are used in the simulations to calculate the bias and mean squared error for the threewise marginal results.

s	$\phi_s(0,0)$	$\phi_s(0,1)$	$\phi_s(1,0)$	$\phi_s(1,1)$
1	0.805	0.185	0.005	0.000
2	0.620	0.000	0.170	0.000
3	0.545	0.000	0.000	0.000
4	0.705	0.005	0.000	0.000
5	0.595	0.005	0.000	0.005
6	0.590	0.100	0.005	0.000
7	0.590	0.005	0.105	0.010
8	0.500	0.000	0.005	0.000
9	0.495	0.355	0.000	0.000
10	0.470	0.010	0.345	0.070
11	0.280	0.020	0.010	0.000
12	0.265	0.245	0.025	0.025
13	0.250	0.165	0.075	0.105
14	0.155	0.085	0.145	0.075
15	0.210	0.140	0.270	0.005
16	0.200	0.035	0.180	0.000
17	0.485	0.010	0.035	0.000

Table 7.4: Simulated True Marginal Fourwise Probabilities for the Free Parameters: $\phi_s(0,0)$, $\phi_s(0,1)$, $\phi_s(1,0)$ and $\phi_s(1,1)$. Recall that we assigned $\phi_s(0,0) = \pi_{s,s+1,s+2,s+3}(0,0,0,0)$, $\phi_s(0,1) = \pi_{s,s+1,s+2,s+3}(0,0,1,0)$, $\phi_s(1,0) = \pi_{s,s+1,s+2,s+3}(0,1,0,0)$, and $\phi_s(1,1) = \pi_{s,s+1,s+2,s+3}(0,1,1,0)$. Each of the true marginal fourwise probabilities was calculated by taking the sum of the probabilities assigned to each joint distribution sequences where: (1) $x_s = 0$, $x_{s+1} = 0$, $x_{s+2} = 0$, and $x_{s+3} = 0$, (2) $x_s = 0$, $x_{s+1} = 0$, $x_{s+2} = 1$, and $x_{s+3} = 0$, (3) $x_s = 0$, $x_{s+1} = 1$, $x_{s+2} = 0$, and $x_{s+3} = 0$, (4) $x_s = 0$, $x_{s+1} = 1$, $x_{s+2} = 1$, and $x_{s+3} = 0$ respectively. These values are used in the simulations to calculate the bias and mean squared error for the fourwise marginal results.

7.6 Supplemental Simulation Results Plots

7.6.1 Mean Squared Error

7.6.2 Standard Error

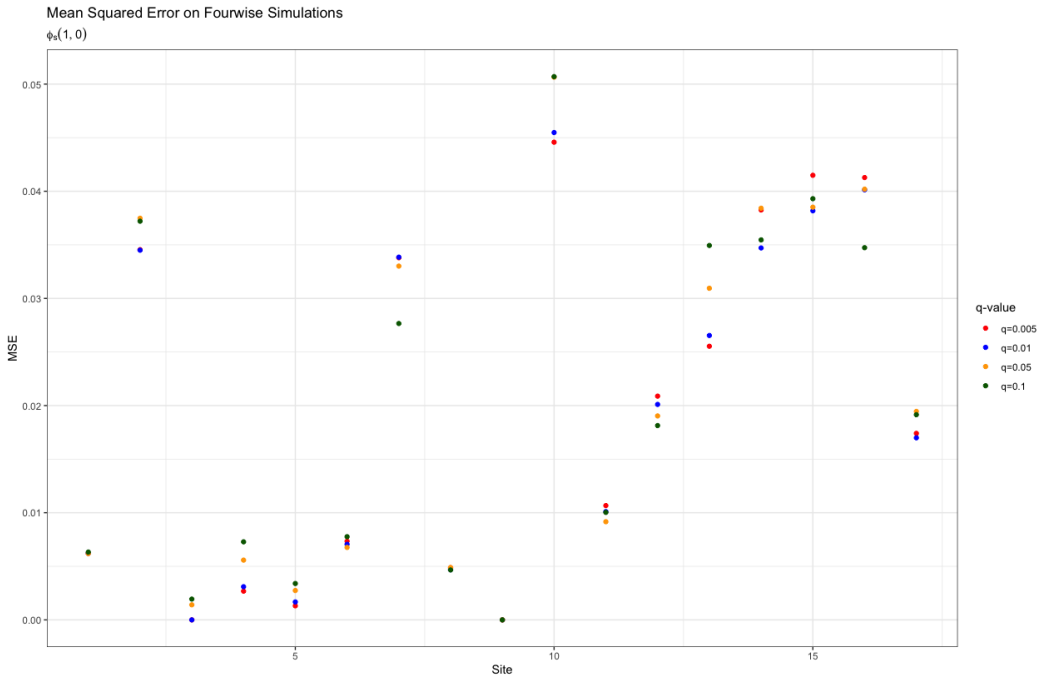

Figure 7.1: Mean Squared Error from the $m = 3$ Simulations for the free parameter $\phi_s(1,0)$. MSE is plotted for all four implemented q-values; red points represent the simulation where $q = 0.005$, blue points represent the simulation where $q = 0.01$, yellow points represent the simulation where $q = 0.05$, and green points represent the simulation where $q = 0.1$. Observe that, for a given site, the mean squared error for the adjacent sites is not necessarily similar but there is a general increase for sites further right on the chain. For example, consider sites 2 and 7.

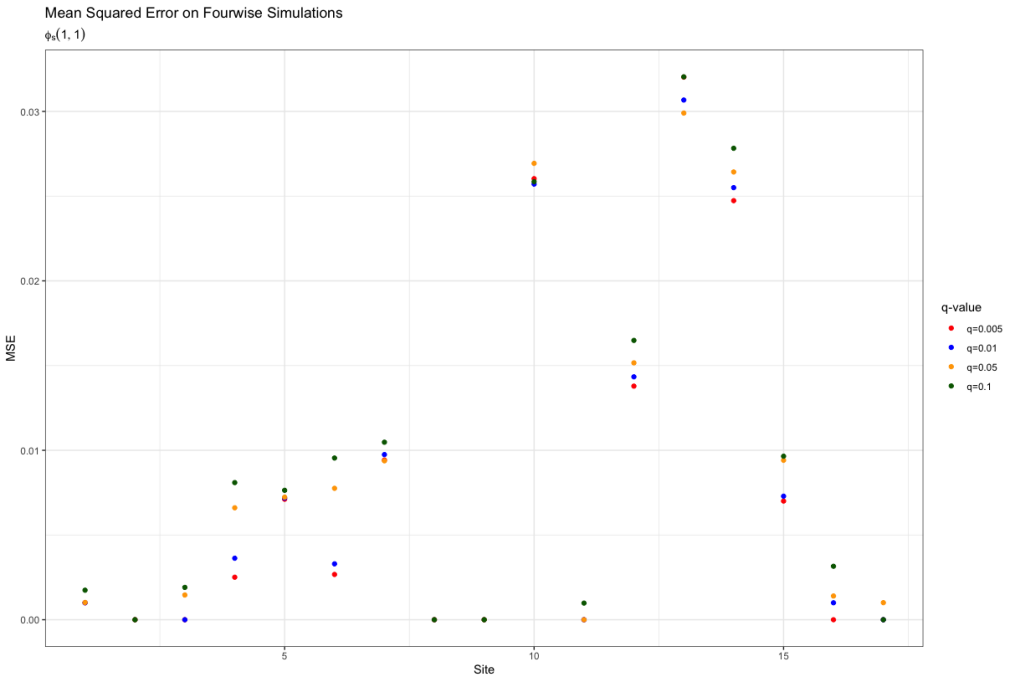

Figure 7.2: Mean Squared Error from the $m = 3$ Simulations for the free parameter $\phi_s(1,1)$. MSE is plotted for all four implemented q-values; red points represent the simulation where $q = 0.005$, blue points represent the simulation where $q = 0.01$, yellow points represent the simulation where $q = 0.05$, and green points represent the simulation where $q = 0.1$. Observe that, for a given site, the mean squared error for the adjacent sites is not necessarily similar. For example, consider sites 10, 13, and 14.

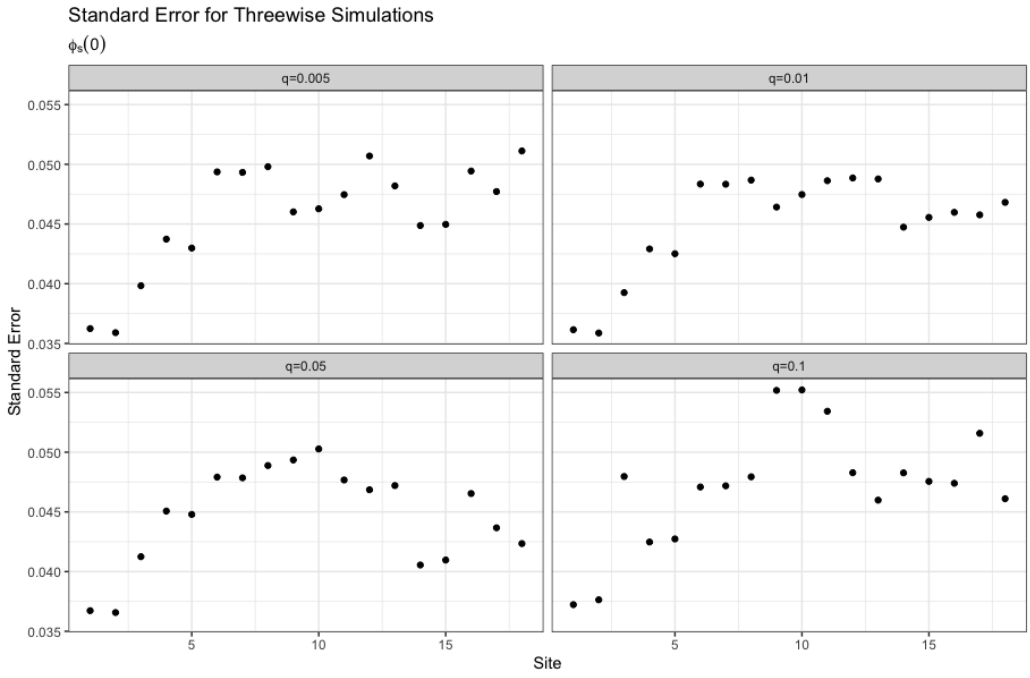

Figure 7.3: Standard error calculated across the 100 replications for $\phi_s(0)$ in the threewise simulations. Observe that there is an increase in standard error for sites further right on the chain, consistent with a directional effect of the Markov chain proceeding from left to right.

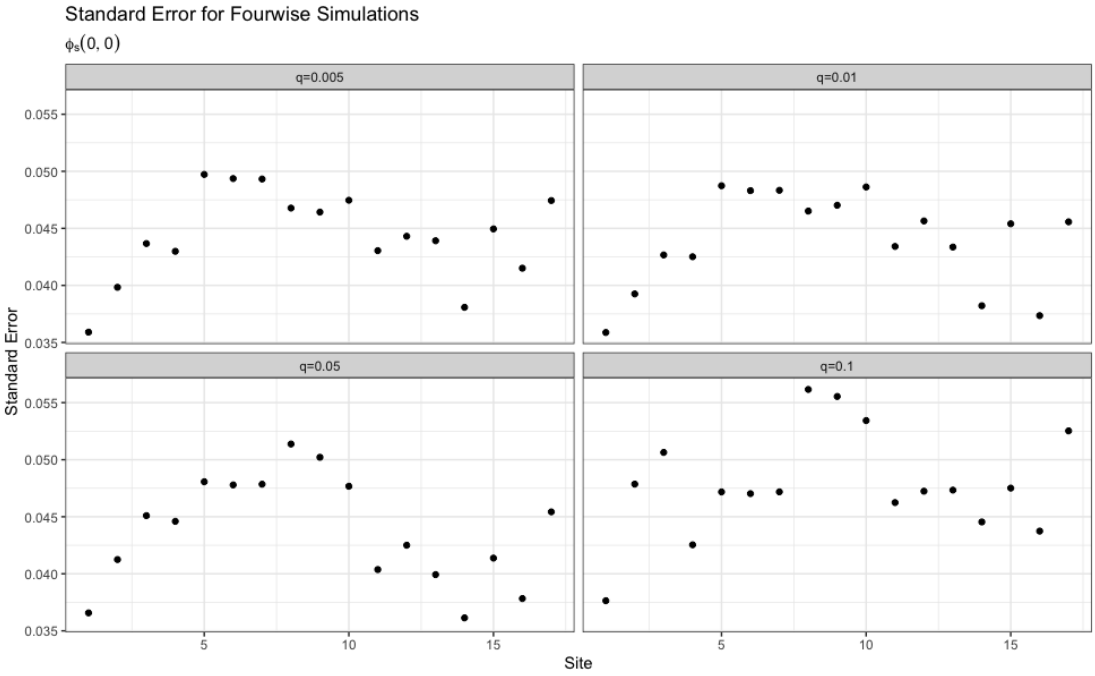

Figure 7.4: Standard error calculated across the 100 replications for $\phi_s(0, 0)$ in the fourwise simulations. We observe that the standard error increases for sites further right on the chain, consistent with a directional effect of the Markov chain proceeding from left to right. We also observe that the standard error for a given site is similar to its adjacent sites, consistent with the fact that this free parameter is dependent only on the major allele.

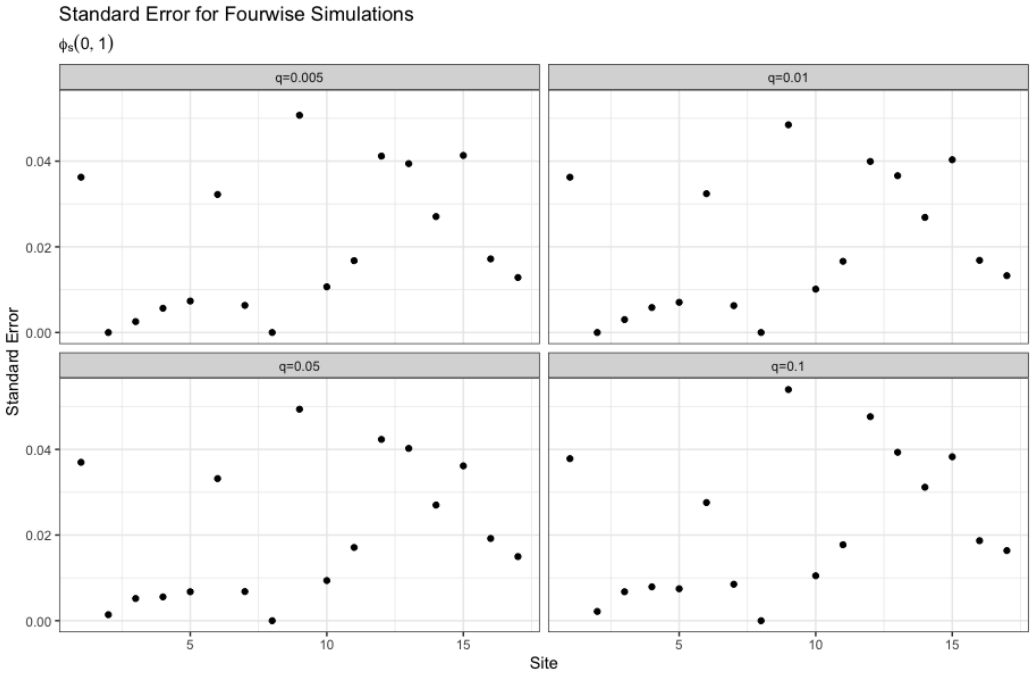

Figure 7.5: Standard error calculated across the 100 replications for $\phi_s(0, 1)$ in the fourwise simulations. We observe that the standard error increases for sites further right on the chain, consistent with a directional effect of the Markov chain proceeding from left to right. We also observe that the standard error for a given site is not necessarily similar to its adjacent sites, consistent with the fact that this free parameter is dependent on the major and minor allele.

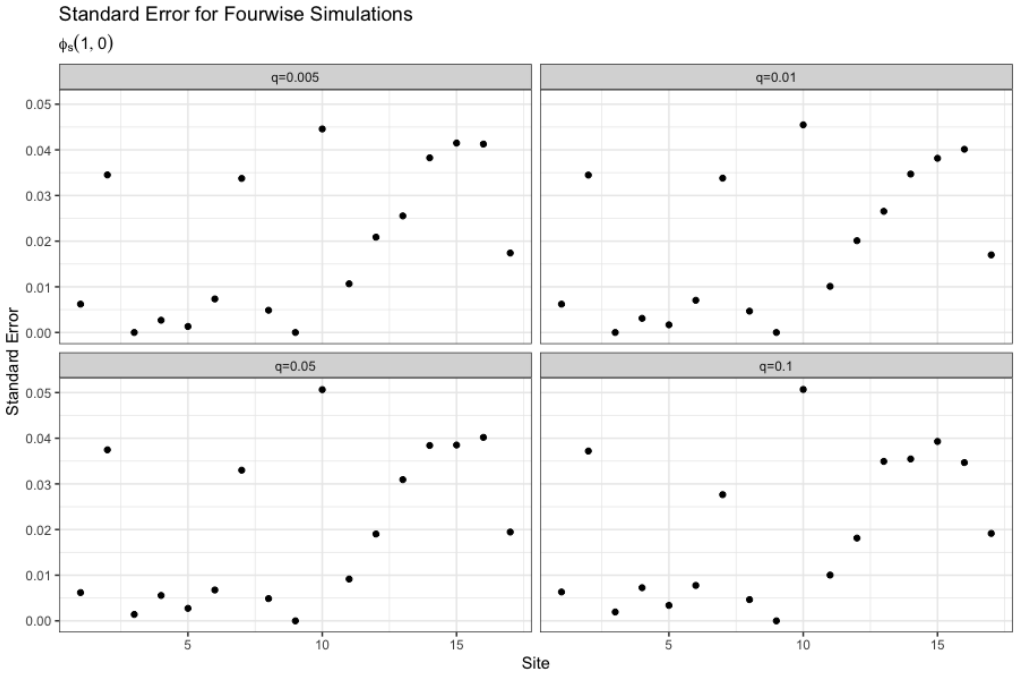

Figure 7.6: Standard error calculated across the 100 replications for $\phi_s(1, 0)$ in the fourwise simulations. We observe that the standard error increases for sites further right on the chain, consistent with a directional effect of the Markov chain proceeding from left to right. We also observe that the standard error for a given site is not necessarily similar to its adjacent sites, consistent with the fact that this free parameter is dependent on the major and minor allele.

Bibliography

[100] 1000 Genomes. *1000 Genomes — A Deep Catalog of Human Genetic Variation*. URL: https://www.internationalgenome.org/ (visited on 03/28/2022).

[Bes74] Julian Besag. "Spatial Interaction and the Statistical Analysis of Lattice Systems". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 36.2 (1974). Publisher: [Royal Statistical Society, Wiley], pp. 192–236. ISSN: 0035-9246. URL: https://www.jstor.org/stable/2984812 (visited on 04/15/2022).

[BSC17] Lisa Bartee, Walter Shriner, and Catherine Creech. *Principles of Biology*. en. Open Oregon Educational Resources, 2017. ISBN: 978-1-63635-040-0. URL: https://openoregon.pressbooks.pub/mhccmajorsbio/ (visited on 03/28/2022).

[CL06] Shu-Chuan Chen and Bruce G. Lindsay. "Building Mixture Trees from Binary Sequence Data". In: *Biometrika* 93.4 (2006). Publisher: [Oxford University Press, Biometrika Trust], pp. 843–860. ISSN: 0006-3444. URL: https://www.jstor.org/stable/20441331 (visited on 10/15/2021).

[Cor16] B Cornell. *Crossing Over — BioNinja*. 2016. URL: https://ib.bioninja.com.au/standard-level/topic-3-genetics/33-meiosis/crossing-over.html (visited on 04/23/2022).

[DL10] Joshua V Dillon and Guy Lebanon. "Stochastic Composite Likelihood". en. In: *Journal of Machine Learning Research* 11 (2010), p. 37.

[Dob16] Robert P. Dobrow. *Introduction to Stochastic Processes with R*. Hoboken, UNITED STATES: John Wiley & Sons, Incorporated, 2016. ISBN: 978-1-118-74070-5. URL: http://ebookcentral.proquest.com/lib/mtholyoke/detail.action?docID=4462510 (visited on 10/21/2021).

[FMR13] Marta Farré, Diego Micheletti, and Aurora Ruiz-Herrera. "Recombination Rates and Genomic Shuffling in Human and Chimpanzee—A New Twist in the Chromosomal Speciation Theory". In: *Molecular Biology and Evolution* 30.4 (Apr. 2013), pp. 853–864. ISSN: 0737-4038. DOI: 10.1093/molbev/mss272. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3603309/ (visited on 02/07/2022).

[Har82] Harold L.K. Whitehouse. *Genetic Recombination: Understanding the Mechanisms*. John Wiley & Sons Ltd., 1982. ISBN: 0-471-10205-9.

[Int09] International HapMap Project. *Index of /hapmap/phasing/2009-02_phaseIII/HapMap3_r2*. 2009. URL: https://ftp.ncbi.nlm.nih.gov/hapmap/phasing/2009-02_phaseIII/HapMap3_r2/ (visited on 03/28/2022).

[Jia11] Jianping Sun. "Composite Likelihood in Long Sequence Data". Dissertation in Statistics for the Degree of Doctor of Philosophy. The Graduate School Department of Statistics: The Pennsylvania State University, 2011. URL: https://etda.libraries.psu.edu/files/final_submissions/3204.

[KC09] P.P. Khil and R.D. Camerini-Otero. "Variation in Patterns of Human Meiotic Recombination". In: *Genome dynamics* 5 (2009), pp. 117–127. ISSN: 1660-9263. DOI: 10.1159/000166623. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3105470/ (visited on 02/07/2022).

[Lee16] Chang-Yong Lee. "A model for the clustered distribution of SNPs in the human genome". en. In: *Computational Biology and Chemistry* 64 (Oct. 2016), pp. 94–98. ISSN: 1476-9271. DOI: 10.1016/j.compbiolchem.2016.06.003. URL: https://www.sciencedirect.com/science/article/pii/S1476927116300287 (visited on 10/21/2021).

[LF11] F. Larribe and P. Fearnhead. "ON COMPOSITE LIKELIHOODS IN STATISTICAL GENETICS". In: *Statistica Sinica* 21.1 (2011). Publisher: Institute of Statistical Science, Academia Sinica, pp. 43–69. ISSN: 1017-0405. URL: https://www.jstor.org/stable/24309262 (visited on 04/15/2022).

[Lin88] Bruce Lindsay. "Composite Likelihood". In: *Contemporary Mathematics* 80 (Jan. 1988), pp. 221–239. ISSN: 9780821850879. DOI: 10.1090/conm/080/999014.

[MWC19] Tim Morris, Ian White, and Michael Crowther. *Using simulation studies to evaluate statistical methods - Morris - 2019 - Statistics*

in Medicine - Wiley Online Library. Jan. 2019. URL: https://onlinelibrary.wiley.com/doi/10.1002/sim.8086 (visited on 10/15/2021).

[Nata] National Human Genome Research Institute. *About the International HapMap Project*. en. URL: https://www.genome.gov/11511175/about-the-international-hapmap-project-fact-sheet (visited on 03/28/2022).

[Natb] National Human Genome Research Institute. *Haplotype*. en. URL: https://www.genome.gov/genetics-glossary/haplotype (visited on 03/28/2022).

[Natc] Scitable by Nature Education. *haplotype / haplotypes — Learn Science at Scitable*. en. Cg_cat: haplotype / haplotypes. URL: https://www.nature.com/scitable/definition/haplotype-haplotypes-142/ (visited on 03/28/2022).

[Nat14] Scitable by Nature Education. *single nucleotide polymorphism / SNP — Learn Science at Scitable*. en. Cg_cat: single nucleotide polymorphism / SNP. 2014. URL: http://www.nature.com/scitable/definition/single-nucleotide-polymorphism-snp-295 (visited on 10/15/2021).

[PM15] Ron Philips and Ron Milo. » *What is the rate of recombination?* en. 2015. URL: http://book.bionumbers.org/what-is-the-rate-of-recombination/ (visited on 10/15/2021).

[PNC16] Prosenjit Paul, Debjyoti Nag, and Supriyo Chakraborty. "Recombination hotspots: Models and tools for detection". en. In: *DNA Repair* 40 (Apr. 2016), pp. 47–56. ISSN: 1568-7864. DOI: 10.1016/j.

dnarep.2016.02.005. URL: https://www.sciencedirect.com/science/article/pii/S1568786416300258 (visited on 03/27/2022).

[Tsy+18] Viachaslau Tsyvina et al. "Fast estimation of genetic relatedness between members of heterogeneous populations of closely related genomic variants". In: *BMC BIOINFORMATICS* 19 (Oct. 2018), p. 360. ISSN: 14712105. DOI: 10.1186/s12859-018-2333-9. URL: http://proxy.mtholyoke.edu:2048/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edswsc&AN=000447941700004&site=eds-live&scope=site (visited on 02/08/2022).

[Uni] Portland State University. *R: Memory Limits in R*. URL: https://astrostatistics.psu.edu/su07/R/html/base/html/Memory-limits.html (visited on 03/25/2022).

[VRF11] Cristiano Varin, Nancy Reid, and David Firth. "An Overview of Composite Likelihood Methods". In: *Statistica Sinica* 21 (Jan. 2011), pp. 0–0.

[Wik21a] Wikipedia. *Genetic linkage*. en. Page Version ID: 1052657545. Oct. 2021. URL: https://en.wikipedia.org/w/index.php?title=Genetic_linkage&oldid=1052657545 (visited on 02/03/2022).

[Wik21b] Wikipedia. *Mixture model*. en. Page Version ID: 1054346507. Nov. 2021. URL: https://en.wikipedia.org/w/index.php?title=Mixture_model&oldid=1054346507 (visited on 03/28/2022).

[WKH15] Charlotte Wang, Wen-Hsin Kao, and Chuhsing Kate Hsiao. "Using Hamming Distance as Information for SNP-Sets Clustering and Testing in Disease Association Studies". In: *PLoS ONE* 10.8 (Aug.

2015), pp. 1–24. ISSN: 19326203. URL: http://proxy.mtholyoke.edu:2048/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edb&AN=109066732&site=eds-live&scope=site (visited on 02/08/2022).

[XR11] Ximing Xu and N. Reid. "On the robustness of maximum composite likelihood estimate". en. In: *Journal of Statistical Planning and Inference* 141.9 (Sept. 2011), pp. 3047–3054. ISSN: 0378-3758. DOI: 10.1016/j.jspi.2011.03.026. URL: https://www.sciencedirect.com/science/article/pii/S0378375811001236 (visited on 10/15/2021).