MARKOV CHAIN MONTE CARLO SAMPLING OF GENE GENEALOGIES CONDITIONAL ON OBSERVED GENETIC DATA

by

Kelly M. Burkett

M.Sc., Simon Fraser University, 2003

B.Sc., University of Guelph, 2000

THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN THE

DEPARTMENT OF STATISTICS AND ACTUARIAL SCIENCE

FACULTY OF SCIENCE

SIMON FRASER UNIVERSITY
Fall 2011

APPROVAL

Name:	Kelly M. Burkett
Degree:	Doctor of Philosophy
Title of Thesis:	Markov chain Monte Carlo sampling of gene genealogies conditional on observed genetic data
Examining Committee:	Dr. Richard Lockhart (Chair)

Dr. Jinko Graham, Senior Supervisor

Dr. Brad McNeney, Senior Supervisor

Dr. Tim Swartz

Dr. David Campbell

Dr. Fabrice Larribe, External Examiner,
Université du Québec à Montréal

Date Approved: November 25, 2011

Declaration of Partial Copyright Licence

Abstract

The gene genealogy is a tree describing the ancestral relationships among genes sampled from unrelated individuals. Knowledge of the tree is useful for inference of population-genetic parameters such as the mutation or recombination rate. It also has potential application in genomic mapping, as individuals with similar trait values will tend to be more closely related genetically at the location of a trait-influencing mutation.

One way to incorporate genealogical trees in genetic applications is to sample them conditional on genetic data observed at present. In this thesis, we describe our Markov chain Monte Carlo (MCMC) based genealogy sampler. First, we describe the sampler that conditions on haplotype data. Our implementation is based on the sampler described in Zöllner and Pritchard (2005). However, we have made several changes to increase the efficiency of sampling. We illustrate the use of our sampler on haplotype data from a publicly-available dataset, where we examine statistics summarizing the degree to which case haplotypes are more related to each other than to control haplotypes.

Most genealogy samplers condition on the haplotype data of present day sequences being available. However, commonly used genotyping technology measures genotypes at single loci rather than haplotypes and therefore the haplotype data needs to be imputed. To avoid single imputation, we then describe how the original sampler was extended to handle the case of only genotype data being available. We apply the sampler to simulated data to evaluate how well it estimates genetic parameters and predicts haplotypes.

Adequate mixing of the sampler was a concern for some of the test datasets. The mixing difficulties were attributed to substantial dependence between the tree structure and the latent variables introduced to facilitate sampling of the trees. We describe our experiences with using simulated tempering in order to improve the mixing of the sampler. Our heated distributions were chosen so that the dependencies between the latent variables and the tree structure were gradually reduced.

We apply this approach to a simulated dataset to illustrate how simulated tempering can improve mixing over the haplotype configurations.

Keywords: Gene genealogy; coalescent model; Markov chain Monte Carlo; genetic association studies; population genetics

Acknowledgements

First and foremost, I would like to thank my senior supervisors, Jinko Graham and Brad McNeney for their encouragement and ongoing guidance throughout the degree. Thank you especially for your understanding and support during the challenging past year and a half.

I would also like to thank all of my fellow students in the Department of Statistics and Actuarial Science at SFU. Thank you Simon, Jean, Carolyn, and Elizabeth for your friendship throughout the years. And thank you Flavia for encouraging me to think about life outside of (or after) school.

Last but not least, I would like to thank my parents and family for their encouragement and patience throughout my many years at school. And I would like to extend a special thank you to Mike for providing endless support and encouragement throughout the years. Your help and expertise has been invaluable. From suggesting applications and tools for programming, to dropping your own work to fix my computer when it was acting up, I could not have done this without you.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The variation observed in genes in the human genome is a result of stochastic evolutionary processes such as mutation and recombination acting over time. The gene genealogy for a sample of genes from unrelated individuals is a tree describing these ancestral events and relationships. Individuals who are more closely related would be expected to share copies of genes that are similar to each other. Knowledge of the tree is useful for population genetics, where it can be used in inference of parameters like the mutation or recombination rate. The genealogical tree may also be useful in assessing association between a trait or outcome and a genomic location since those with a similar trait value will tend to also be more closely related genetically if they share a mutation that influences the value of the trait. However, the time scale for genealogical trees is on the order of tens of thousands of years, and there is therefore no way to know the true underlying tree for a random sample of genes from a population.

In order to incorporate genealogical trees in genetic applications, it is therefore necessary to model the distribution for the tree conditional on genetic data observed at present. A model for gene genealogies unconditional on observed data, called the coalescent model, has been well studied and can be used to simulate sequence data. However, it is not as straightforward to model genealogical trees that must have given rise to a particular sample. As will be described, for even a relatively small sample of sequences, the space of such trees is too large to enumerate and most trees will have negligible probability of producing a given sample of sequences. Markov Chain Monte Carlo (MCMC) techniques have therefore been used to concentrate sampling on the trees that are likely given the observed data.

In this thesis, we will be presenting an MCMC-based sampler that samples genealogical trees at given genomic locations conditional on observed data. The initial sampler that we implemented was based on an approach outlined in Zöllner and Pritchard (2005). Although it is beyond the scope of this work, such a sampler can be used in conjunction with a tree-based association statistic to localize variants that are associated with a trait of interest. In this chapter, we give background information that is relevant to understanding our sampler. We describe the coalescent process, as it provides a prior distribution of the genealogy for our approach. We summarize how ancestral trees have been incorporated in applications in population genetics and gene mapping. We also provide brief background information on the use of MCMC in our context. We assume a background understanding of basic genetic concepts, however a brief summary of the relevant concepts is provided in Appendix A.1.

1.1 Introduction to the gene genealogy

As mentioned, the gene genealogy describes the ancestral relationships among sequences from unrelated individuals. An example genealogy for five sequences is depicted in Figure 1.1. The variation observed in the haplotypes of the sequences, shown as sequences of 0's and 1's in the figure, reflects the mutation and recombination events that occurred over time in ancestors of the present day sample. Sequences that are more closely related on the genealogy would be expected to be more similar to each other than they would be to a sequence that is less closely related. At some point back in time, all sequences will have a common ancestor.

Although so far the genealogy of a gene or sequence segment has been referred to and drawn as a tree, if recombination occurs in that segment there are potentially multiple trees that describe the history of the region. As defined in Appendix A.1, recombination is a genetic process that merges two haplotypes of different parental origins. The resulting sequence is therefore a combination of the maternally and paternally derived haplotypes. For example, the ancestor of sequence before a recombination breakpoint might be derived from the maternal lineage, and the ancestor after the breakpoint might be derived from the paternal lineage. Since the mother and father have different ancestors, their sequences will also have different ancestral histories. Therefore the genealogy changes at different locations along the sequence.

The genealogy across multiple loci can be represented with the ancestral recombination graph (ARG). An example ARG for five sequences is shown in Figure 1.2(A). In the ARG, recombination results in a fork as there are now two ancestral lineages for that ancestor. If attention is restricted to

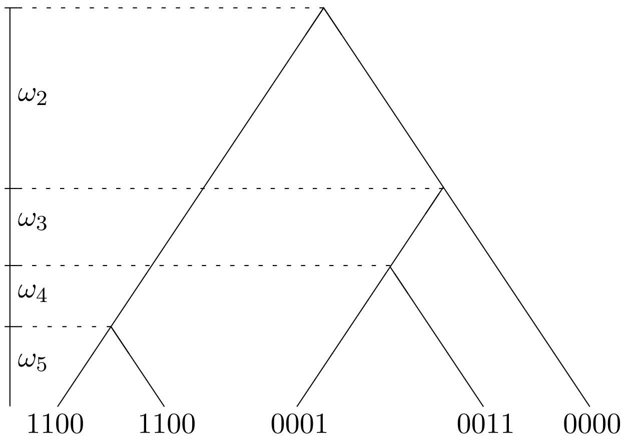


FIGURE 1.1: Genealogy of a point in the genome for five samples. ω_j denotes the times between coalescence events. The haplotypes of the sequences observed at present are shown.

a single locus in the region however, the genealogy is represented as a tree. In addition, if attention is restricted to a set of loci between two recombination events, the genealogy for that set can also be drawn as a tree. The ARG can therefore be described by the set of marginal trees depicting the ancestral relationships for loci between recombination break points. The two marginal trees for the example ARG are provided in Figure 1.2(B) and (C). Note that the common ancestor for the first three loci is much younger than the common ancestor for the last two loci.

Our sampler is used to sample marginal trees at different focal points across a genomic region. We therefore restrict our attention now to tree-shaped genealogies rather than graphs. In the next section, we give background on the coalescent process without recombination, which is a retrospective probability model for a genealogy of samples. This model is used to obtain the prior distribution for the genealogy in our sampler, as described in Chapter 2.

1.1.1 The coalescent model without recombination

The coalescent (Kingman, 1982; Hudson, 1983) is a stochastic process that models the gene genealogy of a random sample of DNA sequences at selectively neutral loci. Reviews describing the coalescent process can be found in Hudson (1990) and Nordborg (2003), and in the recent book by Hein et al. (2005).

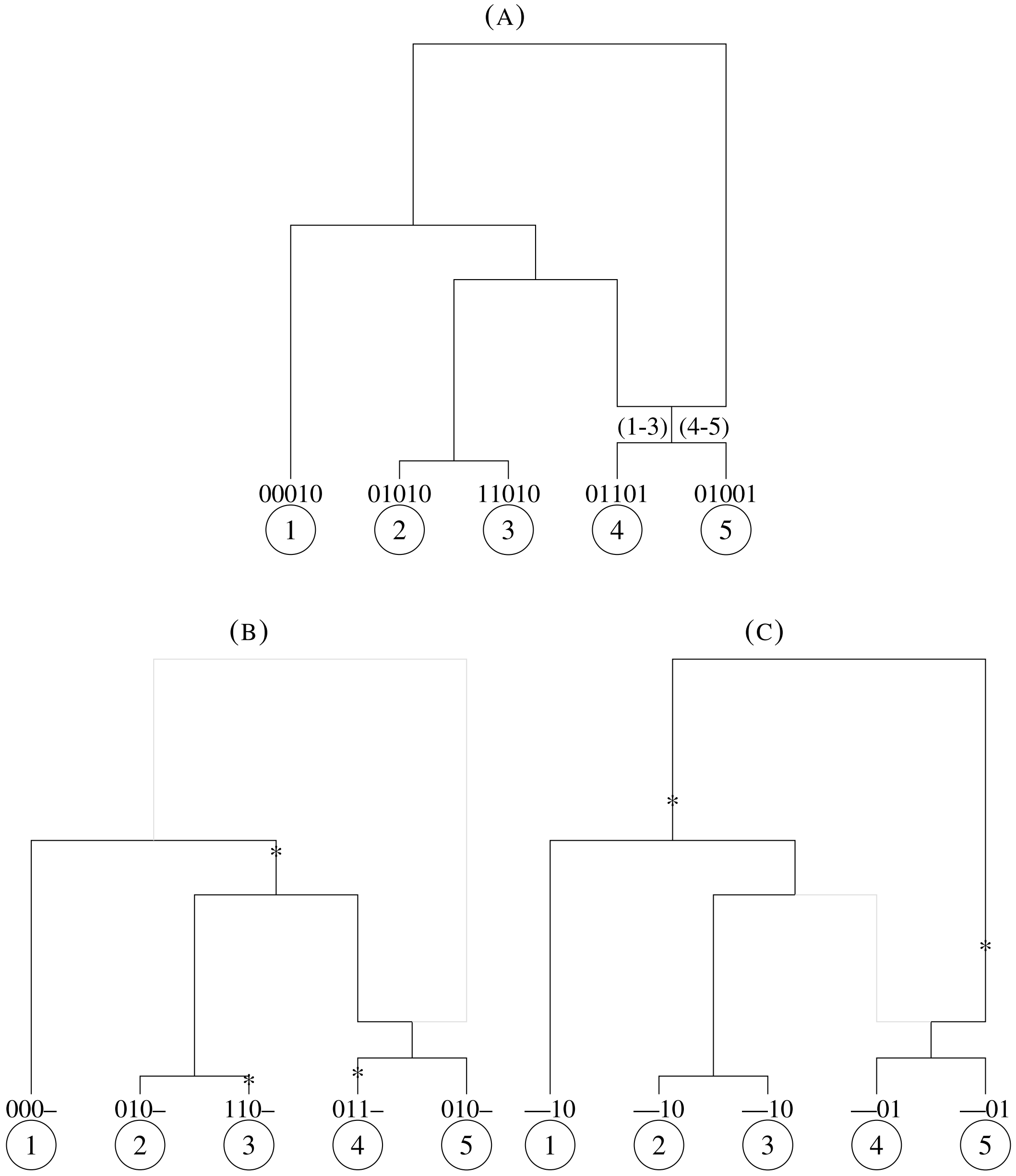


Figure 1.2: An example ARG and marginal trees. (A) The ARG relating five tip sequences. A recombination occurred in an ancestor of tips 4 and 5 between the third and fourth marker locus. (B) The marginal tree for markers 1-3 is extracted from the ARG by following the left path from the ancestor of 4 and 5 until all sequences have coalesced. Mutation events are marked with stars. (C) The marginal tree for markers 4-5 is extracted from the ARG by following the right path from the ancestor of 4 and 5 until all sequences have coalesced. Mutation events are marked with stars.

Assume that a sample of n DNA sequences is drawn from a population. We assume that these sequences are selectively neutral, meaning that they do not affect the reproductive success of individuals having these sequences. We are interested in the ancestry of this sample and we will assume that recombination can not occur in this region. The genealogy is therefore represented by a tree topology, with each internal node indicating the common ancestor of the sequences descending from that node and branches indicating the relationships between nodes. Two lineages are said to coalesce when their branches merge. The node at the merge point represents the common ancestor of sequences in the two lineages. At some point back in time, all lineages will have coalesced with each other. The single ancestor of all sequences is called the most recent common ancestor (MRCA) of the sample.

Over time, mutations occur on the sequences along the lineages of the topology. However, since this is a selectively neutral locus, these mutations do not affect the fertility or survival of sequences carrying mutations and therefore the mutations do not affect the shape of the genealogical tree. Thus a model for the genealogical process can be separated from a model for the mutation process. Mutations are typically assumed to be Poisson distributed with rate proportional to the branch lengths of the tree. More information about the mutation model will be given later in this section.

The coalescent gives a model for the shape of the genealogical tree topology and for the times of coalescence events. The model is retrospective in that, starting at the present, it provides the distribution of times back until sequences find ancestors in the population. The model is also very general in that it can be derived as the large population limit of the ancestral structure for a broad class of neutral models of reproduction in which individuals have exchangeable offspring numbers (i.e., the distribution of the number of offspring is the same for all individuals) (Kingman, 1982). For illustration, we review the case of a Wright-Fisher model of reproduction (Fisher, 1930; Wright, 1931). Assumptions of the Wright-Fisher model are: discrete and non-overlapping generations; constant population size; all individuals are equally fit with respect to offspring number; and mating is at random. Under this model, each new generation is formed by randomly sampling parents with replacement from the current generation. Conversely, at a point in time, each child sequence can be thought of as randomly choosing its parents in the previous generation.

Let N be the effective population size from which the sample is drawn. In the basic coalescent model, and the model that we use as our prior, N is assumed to be constant over time and large relative to n. In a population size of N humans there are twice as many chromosomes, so the population size of sequences is $2N$. Considering the n sequences at present, the number of generations until two of the n have a common ancestor or coalesce has been shown to be approximated by an exponential distribution with rate $\binom{n}{2}/2N$. Time is usually scaled in units of $2N$ generations, in which case the rate is $\binom{n}{2}$ until the time T_n when the two coalesce. By the Wright-Fisher model, offspring pick their parents randomly and independently of each other, so each of the $\binom{n}{2}$ pairs are equally likely to coalesce with each other at time T_n. After the first pair coalesces, there are now $n-1$ lineages remaining. In general, the scaled time during which there are j lineages remaining, T_j, is exponential with rate $\binom{j}{2}$ and all j lineages are equally like to coalesce. All intercoalescence time intervals are independent of each other and the probability of two coalescence events occurring at the same time is negligible.

Although we apply only the basic coalescent described above, the coalescent has been generalized to handle recombination, selection, non-constant population size and structured populations. More information about extensions to the basic coalescent can be found in, for example, Hein et al. (2005).

The coalescent model is particularly efficient for simulating DNA sequence data for neutral loci since the history is only for those sequences having descendants in the current sample. Using the coalescent model, a genealogy with n tip sequences is simulated with intercoalescence times drawn from independent exponential distributions and topology chosen randomly. Mutation events are then added to the genealogy by sampling from a Poisson process over the branches of the tree. The rate of mutations is usually assumed to be $\theta/2$, where $\theta = 4N\mu$ and μ is the per generation mutation rate across the region. Since time is scaled in units of $2N$ generations, θ is the rate for the number of mutations occurring between two nodes having a common ancestor $2N$ generations ago and $\theta/2$ is the rate for the number of mutations occurring on the branch between one of the two nodes and this same ancestor.

Although the process for simulating genealogies and sequence data using the coalescent is straightforward, the reverse problem of sampling genealogies given observed sequence data at present is not. As mentioned, there are two applications where knowledge of this distribution is handy, estimation of population genetic parameters and mapping genes through association of genetic markers with a trait of interest. A brief summary of both is now provided.

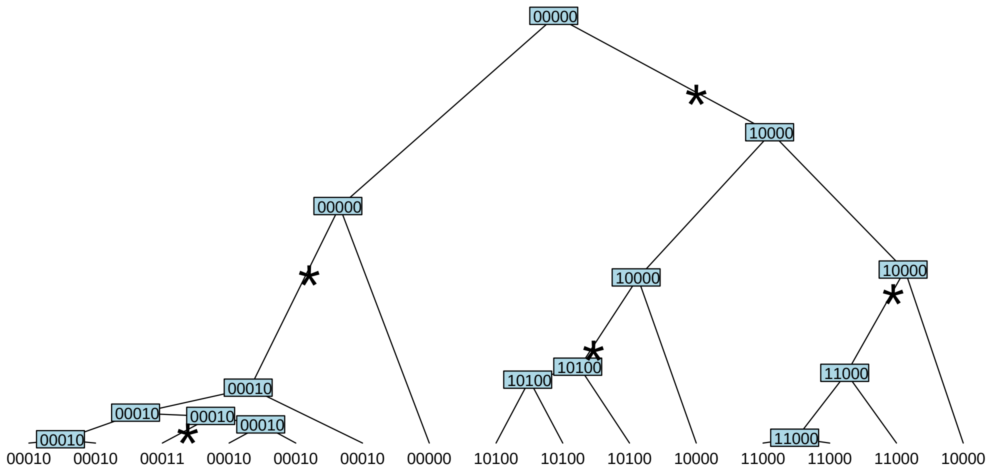


FIGURE 1.3: A hypothetical genealogical tree with sequence at internal nodes. The five historical mutation events are marked with stars.

1.1.2 The genealogical tree in population genetic inference

Although population-genetic parameter estimation is not directly related to the anticipated use of our sampler, these approaches also require sampling latent trees or graphs using MCMC. For this reason, it is useful to briefly outline why MCMC is required and the implementations that have been developed in this context. In our review, we focus on comparing the proposal distributions for tree topology changes used in these population-genetic applications and those outlined by Zöllner and Pritchard (2005), which we adopt in our sampler.

If the true ancestral tree for a set of sequences were known, it would be straightforward to estimate population genetic parameters of interest. For example, assume that we are interested in estimating the mutation rate θ. An example tree, with mutation events superimposed, is provided in Figure 1.3. Given the tree, we could estimate θ with the number of mutations observed divided by the total scaled tree time.

However, it is not possible to know the true tree. Letting τ be the tree topology and node times and D be the sequence data, the tree is introduced as a latent variable and the likelihood is written as

$$L(\theta) = \int_{\tau} \Pr(D|\tau, \theta) \Pr(\tau|\theta) d\tau$$

where the integral is over all topologies and all intercoalescence times. A naive approach might first enumerate all trees and compute $\Pr(D|\tau, \theta)$ for each tree and for a given mutation model. Considering only topologies, for n sequences there are $\frac{n!(n-1)!}{2^{n-1}}$ different tree topologies. Enumerating all is impossible except when n is small. Rather than enumerating all, one might choose to sample trees from $\Pr(\tau|\theta)$ which can be modeled with the coalescent process described above. However, since the space of trees is so large, the majority of these samples would have a negligible probability of producing the data so that $\Pr(D|\tau, \theta) \approx 0$ for most trees.

An approach to concentrate sampling on the most plausible trees is therefore required. MCMC sampling forms the basis for most of the genealogy-based population genetic estimation procedures. Even for likelihood-based estimation, MCMC techniques have been used as a means to compute the likelihood. A review of these genealogy-based approaches is given in Stephens (2003) and a recent review of the software implementing these approaches is given in Kuhner (2009).

An MCMC-based approach to sample genealogies was described in a number of papers by the same group (Kuhner et al., 1995, 1998; Beerli and Felsenstein, 1999; Felsenstein et al., 1999; Kuhner et al., 2000). This approach uses a topology update that involves randomly dropping a branch from the tree and reconnecting it to a random lineage that exists at a time chosen based on the coalescent model. Since they do not include the variables corresponding to the sequence of the internal nodes, the new lineage that the node connects to is not guaranteed to have a similar sequence and the proposed tree may be inconsistent with the observed data (Stephens, 2003). The proposed tree would likely be rejected and the sampler could be slow to mix, particularly for a large number of sequences. A small number of sequences may be sufficient to estimate population-genetic parameters. Felsenstein (2006) found that after 20 individuals the efficiency of estimation does not increase linearly with sample size and that given genotyping costs at the time the optimal sample size was 8 individuals. Kuhner et al. (2000) illustrate their approach to estimate the recombination rate with a dataset of only 10 sequences. However, a genealogy sampler to be used in association mapping applications must handle larger sample sizes.

To improve the mixing of the sampler for larger sample sizes, one could take an approach similar to Wilson and Balding (1998) and introduce latent variables for the sequence at internal nodes. These authors suggest a topology update similar to the "major topology change" of Zöllner and

Pritchard (2005) described in Chapter 2. By including internal node sequences, sequence similarity can be used to guide the selection of the location of the new node in the topology. This should in turn improve the acceptance probability of the update. However, an important limitation of their approach is that they do not consider recombination. Since recombination causes the genealogy to change along the sequence, it is an important evolutionary process to include when sampling genealogies of a genomic region.

1.1.3 The genealogical tree in association mapping

As summarized in Appendix A.1, an association study assesses the correlation between marker loci, for which we have genotypes, and trait or disease outcome for a large set of markers. If a significant correlation is found at one of the loci, the marker may be a risk factor, or more likely, it is close to a disease-predisposing locus.

The concept of the genealogy is also useful in mapping trait loci using association methodology, and it was for this purpose that the sampler in Zöllner and Pritchard (2005) was developed for. An example genealogy of a sample of sequences from unrelated individuals is shown in Figure 1.4. At some point back in time a disease-predisposing mutation occurred on a lineage; this is marked with a star on the tree. All descendants of this branch will carry the disease-predisposing mutation and will therefore also carry an increased risk of developing the disease relative to those which do not carry the mutation. The location of the disease-predisposing mutation is not known, but the mutation occurred on a particular haplotype. Descendants will inherit copies of that haplotype, although mutation and recombination events will limit the similarity between the original and descendant sequences. For example, in Figure 1.4, all sequences with haplotype 10100 carry the disease mutation and therefore the 10100 haplotype may be associated with disease.

As mentioned, the ancestral tree is not known but there has been much interest in incorporating the concept of an underlying tree of sampled sequences into association methodology. First, if a disease-predisposing locus is not included in the dataset, the ancestral tree of the locus provides the best information about its location as it describes which sequences are related to each other. Second, the concept of an ancestral tree is also useful as a data reduction strategy for handling multiple markers simultaneously. When multiple markers or haplotypes are assessed simultaneously, there can often be too many categories, which results in reduced power of the association test. By clustering similar haplotypes, the degrees of freedom are reduced. Finally, grouping similar haplotypes

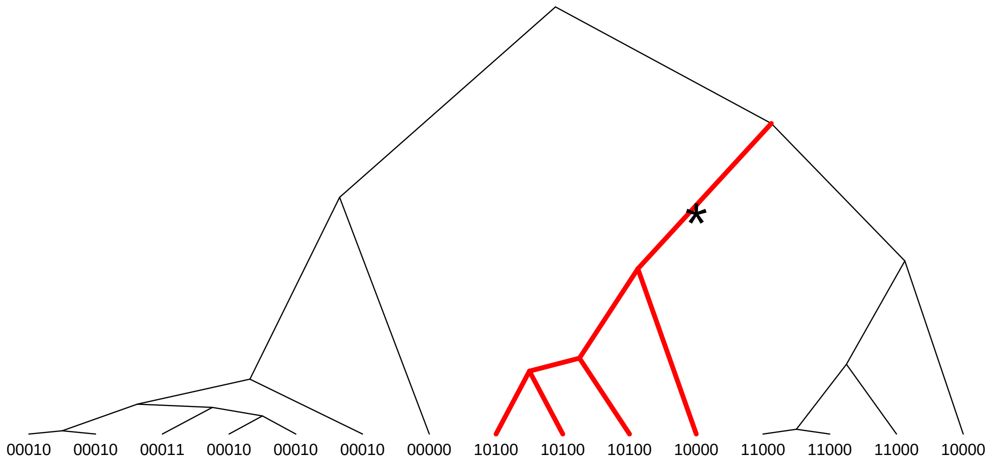


FIGURE 1.4: A hypothetical genealogical tree with a disease-predisposing mutation marked by a star. All nodes that descend from the branch with the mutation will also carry the disease-predisposing mutation.

also provides a more coherent approach for dealing with rare haplotypes than the more common approach of binning rare haplotypes together into one category when computing the association statistic.

Multiple approaches have been proposed and implemented to incorporate the ancestral tree into association methods. These have primarily been based on statistical and phylogenetic clustering techniques. There are many variations, but the common procedure can be summarized as:

1. Construct a tree or graph from the data;

2. Use the tree or graph to define clusters or clades;

3. Compute an association statistic based on membership in clusters.

Our sampler provides a means to construct a tree from the data and so addresses step 1 above. A non-exhaustive summary of some of the other methods to construct trees and graphs is given below. There is some flexibility in how 2 and 3 could be accomplished using our sampler. We therefore also provide a summary of the different association statistics below as it gives an idea of what can be done with our sampled genealogies. However, a detailed evaluation of the properties of different tree-based association statistics based on genealogies sampled from our sampler is beyond the scope of this work.

Constructing a tree or graph from the data

The first step for incorporating an ancestral tree into association methodology is to choose an approach for estimating the tree. A number of approaches construct a tree or graph from haplotype data using clustering techniques also used in phylogenetics. For example, Templeton et al. (1987), Seltman et al. (2003) and Bardel et al. (2005) use parsimony to group the haplotypes into a "cladogram" or into a tree. Tachmazidou et al. (2007), Kimmel et al. (2008) and Mailund et al. (2006) construct a tree assuming a "perfect phylogeny" and Durrant et al. (2004) uses hierarchical clustering. Molitor et al. (2003) first finds cluster centres then places each haplotype in a cluster based on which centre is closest. Many of the clustering based approaches require a haplotype similarity measure to group similar haplotypes together. There is also some variation in how similarity is measured between different methods although this is not reviewed here. These approaches typically only use a single tree or graph to summarize the relationships on the tree. Most also assume that the haplotypes are known; as will be discussed in Chapter 3, the genotypes are typically observed and the haplotypes are often not known. Finally, recombination is generally assumed to have not occurred in the region of interest, although some approaches will explicitly limit the clustering to markers that are compatible with no recombination having occurred in the past.

Minichiello and Durbin (2006) use an approach that samples multiple histories and handles recombination. A heuristic approach is used to sample a set of plausible ancestral recombination graphs (ARGs). Starting at the present and working back until a single ancestor is found, a set of rules determines if the next event should be a mutation, coalescence or recombination event. For example, if two sequences are identical they can coalesce; otherwise a mutation event must occur. A recombination is performed only if no mutation or coalescence is possible. Since at any stage there may be many choices for the next event, a random choice is made. This random choice is then used to create an ensemble of 100 different ARGs.

Tree or clade-based association statistics

Conditional on a tree, there are many different ways to assess association with a disease or trait of interest. Durrant et al. (2004) use logistic regression by classifying individuals into clusters and regressing cluster membership on disease status. Since the number of clusters is not known in advance, all potential clusterings induced by the tree are tested and the one with the minimum p-value is selected as the best clustering. Seltman et al. (2003) also tests for association with the partitions induced by the graph however they generalize this approach to different types of outcomes by using generalized linear models. Bardel et al. (2005) use a nested clustering approach like that of Durrant et al. (2004); however, they test for correlation between disease status and cluster membership at each clustering induced by the tree and stop when a significant p-value is found. Mailund et al. (2006) take a different approach by using the imputed tree as a decision tree for the classification of cases and controls.

Since a set of ARGs are created in the approach due to Minichiello and Durbin (2006), a different means of assessing association is required. First, for each marker, the marginal tree of that marker is extracted from the ARG for all 100 graphs. Then, for each tree, the partition induced by cutting each branch (or, placing a mutation on that branch and classifying the descendent tips according to mutation state) is tested for association using a chi-squared test of non-independence between disease state and allelic state at the mutation. There are $n - 3$ partitions and therefore the maximum chi-square statistic across partitions is retained. The average maximum chi-square over all 100 trees is the final score for the marker and permutation of disease status is used to determine p-values for each marker.

There are three potential criticisms which apply to these approaches. The first is that many do not incorporate the uncertainty due to tree imputation. The majority construct a single tree and base their clusterings on that tree. Since the tree is reconstructed based on haplotype similarity, identical haplotypes will tend to appear in the same cluster and will share the same disease risk. However, due to the stochastic nature of mutation and the coalescent process, identical haplotypes which would not have the same risk could be grouped together. A simple example of this effect can be seen in Figure 1.4, where one tip node with sequence 10000 has the disease mutation and the other does not (the mutations that gave rise to the tip sequence are drawn in Figure 1.3). This could result in a misclassification of the haplotype to the wrong cluster. Second, most also do not incorporate haplotype uncertainty. Although Seltman et al. (2003), Mailund et al. (2006) and Minichiello and Durbin (2006) do handle missing haplotype phase, the majority of tree-based association approaches mentioned do not. Finally, phylogenetic clustering methods are typically used to cluster different species and therefore recombination is not a factor. These methods will therefore assume that the same single tree can be constructed for the entire region of interest or will restrict markers to a subset between putative recombination breakpoints. The former could result in misclassification of cluster membership; the latter results in less genotype data being used to construct the tree.

Although these cluster-based methods can capture the structural similarity of haplotypes, they do not incorporate knowledge of genetic processes such as mutation and recombination in the clustering algorithm. On the surface Minichiello and Durbin (2006) seem to include mutation and recombination events; however, these events are not sampled in proportion to their population-genetic rates but rather uniformly based on which events can be performed given the data.

We may instead be interested in inferring trees or ARGs based on a population genetic model that captures the evolutionary processes. In fine-mapping approaches, which seek to find the most likely location for a disease-predisposing mutation in a region known to harbour such a mutation, modelling of the evolutionary processes that gave rise to the sample has been more common. Typically only the genealogy of cases is constructed. A summary of these approaches is given in Morris et al. (2002).

Finally, Zöllner and Pritchard (2005) provide an approach that both tests for association of a region with a trait of interest and estimates the location of a mutation, as in fine-mapping, given known haplotype data. Unlike the fine-mapping approaches mentioned above, the ancestry of all individuals, not just cases, is sampled. In order to make the sampling computationally feasible with reasonable sample sizes, they use a local approximation to the ARG by sampling marginal ancestral trees at multiple focal points. Sampling is done with an MCMC algorithm that they briefly outline in the Appendix of their article; our sampler is based on this description. Their approach incorporates the uncertainty of the underlying tree but by assuming known haplotypes they do not incorporate haplotype uncertainty.

1.2 Background on Markov Chain Monte Carlo

The approach that we have chosen uses Markov Chain Monte Carlo to sample genealogies from their distribution and estimate quantities of interest based on the genealogies. There are many thorough reviews (Besag et al., 1995; Brooks, 1998; Besag, 2001) and books (Gilks et al., 1998) on the topic, so this section is not comprehensive and instead focuses on background that aids in understanding our MCMC algorithm. For ease of explanation, generic notation will be used rather than notation specific to the genealogy sampler. In addition, this description assumes discrete state spaces. The state space for our sampler is not discrete, however the results described here have been shown to apply to general state-spaces (Tierney, 1994).

Let $X \in \mathcal{X}$ be a random variable or vector with distribution $\pi(x) = f(x)/C$ where the normalizing constant C may be unknown. We are interested in estimating the mean of $g(X)$, $E[g(X)]$, but the distribution of X is so complex that the mean can not be computed analytically. If we could generate n independent samples $x_1, x_2, \ldots, x_n$ from $\pi(x)$, Monte Carlo integration could be used to estimate the mean with

$$\frac{\sum_{i=1}^{n} g(x_i)}{n}.$$

However, due to the complexity of the distribution, we are also not able to do so with standard techniques for sampling from probability distributions. Instead, a Markov chain is constructed such that the stationary distribution of the chain is our target distribution $\pi(x)$ and the chain is guaranteed to converge to the target distribution.

The Metropolis-Hastings method is an approach to construct such a Markov chain. Let $Q(y|x) = \Pr(X_1 = y|X_0 = x)$ be a transition matrix for an ergodic Markov chain. Define a second transition matrix $\mathcal{P}$ having off-diagonal elements $\mathcal{P}(y|x) = Q(y|x)\alpha(x,y)$ and diagonal elements obtained to ensure that $\mathcal{P}$ has unit row sums. Define

$$\alpha(x,y) = \begin{cases} \min\left\{1, \frac{\pi(y)Q(x|y)}{\pi(x)Q(y|x)}\right\} = \min\left\{1, \frac{f(y)Q(x|y)}{f(x)Q(y|x)}\right\} & Q(y|x) \neq 0 \\ 0 & Q(y|x) = 0 \end{cases}. \tag{1.1}$$

Q is called the proposal distribution and $\alpha(x,y)$ is called the acceptance probability. Since C cancels in the acceptance probability this approach can be used when the normalizing constant is not known. The stationary distribution of $\mathcal{P}$ is $\pi(x)$, which can be verified by showing that the detailed balance condition holds (Gilks and Roberts, 1996). Provided the chain is ergodic (irreducible and aperiodic), it converges to the stationary distribution regardless of the initial state.

Since the states sampled using this scheme consist of a Markov chain, they are not independent. However, provided the Markov chain is ergodic, the ergodic theorem guarantees that for a given function $g(X)$ the sample average of the states of the Markov chain evaluated at the function converges to $E[g(X)]$. Therefore integrals can be approximated by sampling using Metropolis-Hastings and computing the mean over the sampled states.

In practice, the chain is constructed by first determining an initial state x_0. Then, for $t > 0$ a proposal y is generated from the proposal distribution $Q(y|x_{t-1}) = \Pr(X_1 = y|X_0 = x_{t-1})$ and a uniform(0,1) random variable u is also generated. The state of X_t is

$$x_t = \begin{cases} y & u < \alpha(x_{t-1}, y) \\ x_{t-1} & \text{otherwise} \end{cases} .$$

Although not theoretically justified, common practice has been to discard initial samples as "burn-in" as they are thought to depend too heavily on the initial condition and therefore bias the estimate of the mean. If the chain is run long enough, burn-in shouldn't be required; however, the time to complete the required number of iterations may be infeasibly long. "Thinning", that is storing only every k^{th} iterate, is also a practice occasionally used to reduce the output stored and therefore memory requirements.

1.2.1 Updating components or blocks of X

In practice, it is common for MCMC to be used to sample from the distribution of vector-valued X since under these circumstances it can be difficult to sample from the target distribution. To update all components of X, a vector-valued proposal distribution would be required. Rather than find a proposal distribution that simultaneously updates all components of X, one strategy is to construct an MCMC algorithm by combining several transition matrices (see for example Besag (2001)) each having stationary distribution $\pi(x)$ and with each updating a subset of X. Letting S_i contain indices of the elements of X in the i^{th} subset, the Metropolis-Hastings approach is used to identify the collection of transition matrices $\mathcal{P}_1, \ldots, \mathcal{P}_n$ that update subsets $S_1, \ldots, S_n$ of X. These subsets are often either individual components of X, X_i, or are disjoint or overlapping blocks of highly correlated components of X. Each transition matrix might correspond to a chain that is not irreducible but the transition matrices are combined in such a way that irreducibility is achieved.

If each transition matrix $\mathcal{P}_1, \ldots, \mathcal{P}_n$ has stationary distribution π then combinations will also have stationary distribution π. For example, $\mathcal{P} = \mathcal{P}_1\mathcal{P}_2 \cdots \mathcal{P}_n$, the transition matrix corresponding to applying each proposal Q_i in turn has stationary distribution π since

$$\pi\mathcal{P} = \pi\mathcal{P}_1\mathcal{P}_2 \cdots \mathcal{P}_n = \pi\mathcal{P}_2 \cdots \mathcal{P}_n = \cdots = \pi.$$

Similarly, if the updates are made by randomly sampling a proposal distribution Q_i with probability

p_i where $\sum_i p_i = 1$, then the transition matrix for the chain is $\mathcal{P} = \sum_i p_i \mathcal{P}_i$ and

$$\pi\mathcal{P} = \pi \sum_i p_i \mathcal{P}_i = \sum_i p_i \pi \mathcal{P}_i = \pi \sum p_i = \pi.$$

Let X_S denote the subset of elements of X defined by the set of indices S, and X_{-S} denote the subset of all elements of X except those in X_S. If Q_S updates only elements in X_S then the acceptance probability can be written as

$$\alpha_S(x, \tilde{x}) = \min\{1, \frac{\pi(\tilde{x}_S | x_{-S}) Q_S(x_S | \tilde{x_S}, x_{-S})}{\pi(x_S | x_{-S}) Q_S(\tilde{x}_S | x_S, x_{-S})}\}.$$

Updating the block X_S can lead to simplification of the Metropolis-Hastings acceptance probability for each of the chains. Many π are typically formed as products, including our target distribution. Any term in the product that does not depend on the set X_S cancels from the fraction in the acceptance probability. This saves some computation time in computing α_S.

The MCMC sampler that we implemented has a total of five proposal distributions, each updating only a subset of the variables, that are combined by sampling the proposal distribution at each step according to probabilities p_i. Due to the tree structure, there is a complex dependency between the variables defined at each of the nodes of the tree; it is difficult to change the value of one variable without also changing the value of another variable. Therefore, three of the proposal distributions modify somewhat overlapping subsets of variables.

1.3 Overview of the thesis

In the next three chapters, we describe our MCMC based genealogy sampler and present examples on how it can be used to estimate means of tree-based statistics of interest.

First, we describe the sampler that assumes that haplotype data are available. Our implementation is based on the sampler described in Zöllner and Pritchard (2005). Initially, the sampler was simply to be implemented as described but with the intention of extending it and using it to study tree-based association statistics. However, during implementation we made changes to the proposal distributions in order to ensure compatibility of the tree. The description of each of the different proposal types used has mostly remained the same, but the individual proposal distributions and similarity scores that we have chosen are not necessarily the same. The latent variables and the joint model over the latent variables and genealogy is the same as was proposed by Zöllner and Pritchard (2005); however, we have chosen to modify a prior distribution for one of the parameters. Our version is summarized in Chapter 2 and we use our sampler on haplotype data from a publicly available dataset. The data consists of genotypes from parents and affected children and we examine statistics summarizing the degree to which the "case" haplotypes are more related to each other than the "control" haplotypes.

Most genealogy samplers condition on the haplotype data of present day sequences being available. The current genotyping technology that is most widely used does not provide haplotypes, however, and therefore typically haplotypes are first imputed. As single imputation has the potential to bias results, it is desirable to jointly estimate the haplotypes and the genealogy from the actual observed data, especially since it is the genealogy itself which gave rise to the sample of haplotypes. In Chapter 3, we describe how the original sampler was extended to handle only genotype data being available. This involved adding latent variables corresponding to the haplotypes of the tip sequences, extending the model to include the extra latent variables, and evaluating different proposal distributions for sampling haplotype configurations. We apply the sampler to simulated data to evaluate how well the sampler estimates genetic parameters and the haplotypes.

There is significant dependence between the tree structure and the latent variables introduced at nodes of the tree and so adequate mixing of the sampler may be of concern for some datasets. In Chapter 4, we describe our experiences with using simulated tempering in order to improve the mixing of both the haplotype and genotype-based samplers. Our form of heating was motivated by independent sampling of the topologies, and our heated distributions were therefore chosen to gradually reduce the dependencies between the latent variables at internal nodes so that they could be moved more freely. We also attempted to vary the heating scheme in order to improve performance.

The scope of this thesis has been restricted to the sampler itself, but the ultimate goal is to use the sampler in genetic association studies. In the final chapter, we discuss this and other potential directions for future work.

Chapter 2

Sampling genealogies conditional on haplotype data

In this chapter, we describe our implementation of a genealogy sampler that samples ancestral trees conditional on observed haplotype data. It is based on a sampler outlined in Zöllner and Pritchard (2005) that was implemented for use with their fine-mapping approach. A version of the approach described in Zöllner and Pritchard (2005) was implemented by the authors in a program called LATAG and is included in their mapping software, TreeLD (see Zöllner (2005) for documentation and information about downloading the program). As we were not interested in mapping *per se*, but rather in developing a stand-alone sampling algorithm that handled missing haplotype phase, we used the description of their approach as a guide for developing our own MCMC sampling algorithm that could later be extended. During implementation, we modified the proposal distributions from those outlined in the original paper and there are therefore some important differences between our haplotype-based sampler and the description provided. In the following chapter, we give background on Zöllner and Pritchard (2005)'s sampler and highlight where our implementation differs. We then illustrate its use by computing tree-based clustering statistics on genealogies that have been sampled from their conditional distribution given imputed haplotypes from a publicly-available dataset.

2.1 Outline of the haplotype-based sampler

Recall from the previous chapter that the ancestral history of a genomic region is described by the ancestral recombination graph (ARG) and that the marginal history of a single point in the region is described by a genealogical tree. Rather than sample from the ARG conditional on observed data, Zöllner and Pritchard (2005) proposed sampling from an approximation to the ARG by sampling marginal trees at a set of focal points in the region. This method is a simplification of sampling from the full ARG as only genetic material that is passed along with the focal point to the tip sequences is tracked. The purpose of the original sampler was to determine if a location in a region was associated with a trait of interest. Therefore a sample of trees is collected for each focal point and association is assessed at each focal point.

Let x be the focal point, possibly a trait-influencing locus, and $\mathcal{T}_x$ the marginal genealogy of that focal point. The genealogy consists of τ_x, the topology of the tree, and $\mathbf{\Omega}$ the coalescence times of the nodes of the tree. We sample $\mathcal{T}_x$ from its distribution conditional on the observed haplotype data $\mathbf{H}$, $f(\mathcal{T}_x|\mathbf{H})$, using MCMC methods. Zöllner and Pritchard (2005) introduced latent variables corresponding to recombination events and internal node sequences. Adding variables is a common technique employed in MCMC approaches as it can simplify specification of the posterior (Gilks and Roberts, 1996). In this case, the latent variables represent interpretable but unobservable biological quantities. Marginalization to get the distribution of the genealogy is achieved by ignoring the latent variables that are ultimately not of interest in the final output (Besag et al., 1995). The mutation and recombination rates are also assumed to be random; we use studies from population genetics to inform their distributions. Thus the augmented data $\mathbf{A}$ consists of the mutation rate θ, recombination rate ρ, sequence at nodes in the genealogy $\mathbf{S}$, recombination-related variables $\mathbf{R}$, node times $\mathbf{\Omega}$ and topology τ_x of the tree. We use MCMC methods to generate a sample from $f(\mathbf{A}|\mathbf{H})$. The notation and variable definitions are described in more detail in Section 2.2.

Background information on sampling using MCMC was provided in Section 1.2 but a short summary is given here using notation specific to this context. We desire a sample from target distribution $f(\mathbf{A}|\mathbf{H})$. At the t^{th} step, a candidate value for the augmented data, $\tilde{\mathbf{A}}$, is sampled from proposal density $Q(\tilde{\mathbf{A}}|\mathbf{A})$, which depends on the current value of the Markov chain, $\mathbf{A}$. $\tilde{\mathbf{A}}$ is then accepted as the t^{th} value of the Markov chain with probability

$$\alpha(\mathbf{A}, \tilde{\mathbf{A}}) = \min\left\{1, \frac{f(\tilde{\mathbf{A}}|\mathbf{H})Q(\mathbf{A}|\tilde{\mathbf{A}})}{f(\mathbf{A}|\mathbf{H})Q(\tilde{\mathbf{A}}|\mathbf{A})}\right\}. \tag{2.1}$$

TABLE 2.1: The six proposal schemes suggested by Zöllner and Pritchard (2005) to update the augmented variables

Q_1	update θ
Q_2	update ρ
Q_3	Local Updates (update of the time since the present t_i, the recombination variables r_i, and the sequence $\mathbf{s}_i$ for each node in turn)
Q_4	Major Topology Rearrangement (modifies topology τ_x by moving one of the nodes of the tree; also requires modification of some r_i and $\mathbf{s}_i$ to accommodate the new topology)
Q_5	Minor Topology Rearrangement (modifies τ_x by swapping an aunt/niece relationship; also requires modification of some r_i and $\mathbf{s}_i$ to accommodate the new topology)
Q_6	Time Swap (reordering of two coalescence events by swapping t's associated with two nodes) but does not modify τ_x

Otherwise the old value $\mathbf{A}$ is retained. $\alpha(\mathbf{A}, \tilde{\mathbf{A}})$ is called the Metropolis-Hastings (MH) acceptance probability and the corresponding Markov chain of sampled values has $f(\mathbf{A}|\mathbf{H})$ as its stationary distribution provided the chain is ergodic. The description of the target density $f(\mathbf{A}|\mathbf{H})$ is given in Section 2.3 but a quick summary of the approach to generate proposal values is now given.

A proposal distribution for this sampler must involve changes not only to the sequence, time and recombination variables at internal nodes, but also to the topology of the tree. Determining good proposal distributions is especially challenging as the variables at nodes on the tree are dependent on the values at surrounding nodes and the latent data have both continuous and discrete components. This dependence means that an update to one variable in one node can make the value of the same variable at another node either extremely unlikely or even impossible. When variables are highly correlated, sets or blocks of variables can be updated together; Section 1.2.1 gives background on updating blocks of variables in MCMC methods.

The approach suggested in Zöllner and Pritchard (2005) is to sample from six different proposal distributions, $Q_1, \ldots, Q_6$, each updating only a subset of the augmented variables. The suggested proposal schemes are summarized in Table 2.1 and will be described in detail in Section 2.5. For each, the candidate value is accepted with probability

$$\alpha_i(\mathbf{A}, \tilde{\mathbf{A}}) = \min\left\{1, \frac{f(\tilde{\mathbf{A}}|\mathbf{H})Q_i(\mathbf{A}|\tilde{\mathbf{A}})}{f(\mathbf{A}|\mathbf{H})Q_i(\tilde{\mathbf{A}}|\mathbf{A})}\right\}. \tag{2.2}$$

Zöllner and Pritchard (2005) also suggested combining these six proposal distributions so that certain updates were completed in sequence. For example, a topology change update would be immediately followed by the local updates. Since the major and minor topology rearrangements involve changes to some of the same variables as are changed by the local updates (see Table 2.1 for brief descriptions of these proposals), we assumed that each update type is proposed separately, with the subset of changes accepted or rejected before the next update type is applied to avoid duplicate changes to the same variables. Although the motivation for combining proposals was not given in Zöllner and Pritchard (2005), they may have observed that it aided mixing and convergence to the stationary distribution.

By default, we chose a simpler approach. In our implementation, a step of the chain consists of sampling one of the proposal distributions, rather than some combination of the six. The probabilities associated with selecting the proposal type are chosen to ensure that certain types of updates, like the local updates, are done more often. Although for our sampler we have not yet observed a benefit to combining proposals, for larger datasets the more complicated sequential may offer some improvement. Therefore, our implementation does allow for sets of updates to be completed during a step of the chain. In the step, each update type is proposed and accepted separately; the combination simply controls the order of updates being applied. For example, it may be more efficient to do local updates after a topology change in order to realign the tree so that it becomes more compatible with the new topology, as implemented by Zöllner and Pritchard (2005). Such a set may consist of performing (Q_5, Q_3) with some probability p. Completing the local updates after a topology change will not increase the chance of the topology change being accepted, but it may generally make the data at all internal nodes more consistent with the topology change if the topology change was accepted.

Although the proposal schemes summarized in Table 2.1 were suggested in Zöllner and Pritchard (2005), the actual distributions for sampling new r, s and ω values were typically not provided. We therefore used their outlined updates as a guide but determined our own proposal distributions. In addition, for a number of the proposals, we have chosen to update different variables than was suggested in order to ensure compatibility of the topology and associated variables. The proposal distributions and corresponding Metropolis-Hastings acceptance probabilities for our implementation are described in detail in Section 2.5.

Although we initially implemented update type 6, the time swap, it causes only a very minor change. It is unclear why such an update was necessary, but it is possible that the proposal distribution that updated node times did not change the order of the coalescence event associated with

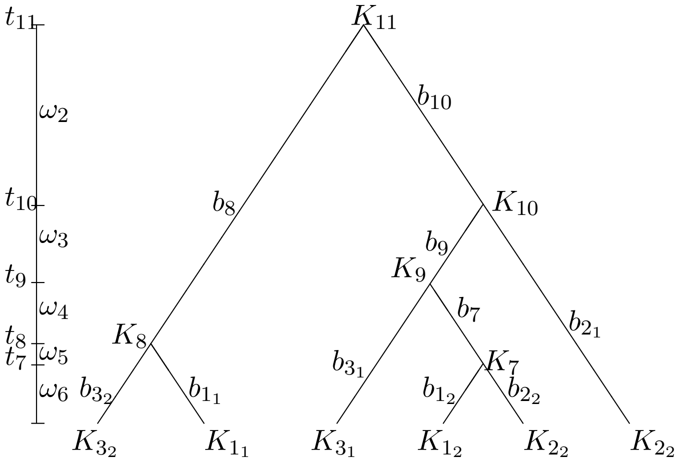


FIGURE 2.1: Illustration of the notation used to describe the genealogy $\mathcal{T}_x$. Nodes are labelled $K_{i_h},\ i = 1, \ldots, n,\ h = 1, 2$ for tip nodes and $K_i,\ i = 2n + 1, \ldots, 4n - 1$ for internal nodes of the topology τ_x. $\Omega = (\omega_6, \ldots, \omega_2)$ are the intercoalescence times, $\mathbf{T} = (t_7, \ldots, t_{11})$ are the times since the present and b_i is the branch length between node K_i and its parent.

the node. This would be the case if the proposal distribution for the node times was, for example, uniform in the interval between adjacent coalescence times. Our update to the node times can cause a re-ordering of the coalescence events and so a separate time swap update is redundant. As our results were unchanged when the time swap update was omitted, we do not consider it further.

2.2 Notation and Definitions

As mentioned in the overview in the last section, the goal of the algorithm is to sample from $f(\mathcal{T}_x|\mathbf{H})$, where $\mathcal{T}_x$ is the gene genealogy and $\mathbf{H}$ is the haplotype data, using an MCMC approach. $\mathcal{T}_x$ consists of the labelled topology, τ_x, of the genealogical tree and the coalescence times, $\mathbf{\Omega}$. We now summarize the latent variables and notation that will be used throughout this document. They are essentially the same as given in Zöllner and Pritchard (2005); however, we provide additional detail here as needed. For an illustration of the notation, refer to Figure 2.1. Table 2.2 also provides a quick summary of the notation. Background genetic definitions are provided in Appendix A.1.

TABLE 2.2: Notation and latent variables for the sampler

n	Number of individuals
$2n$	Number of observed sequences
L	Number of SNP markers
$\mathbf{s}_{i_h}$	The observed h^{th} sequence for the i^{th} individual. The haplotype of this sequence consists of the alleles at each of the L loci so that $\mathbf{s}_{i_h}$ is a string of 0's and 1's. The vector of haplotypes is given by $\mathbf{H}$.
K_{i_1} and K_{i_2}	Tip nodes corresponding to individual i's two sequences
K_i	Internal node corresponding to a coalescence event between 2 lineages
b_i	Branch length between node K_i and its parent K_a
t_i	Time since the present for node K_i, in units of coalescence time. The vector of times since the present is given by $\mathbf{T}$
ω_i	The i^{th} intercoalescence time or the time during which there are i lineages. The vector of intercoalescence times is given by $\mathbf{\Omega}$
$\theta/2$	Mutation rate per unit of coalescence time for the observed SNP markers
$\rho/2$	Recombination rate per unit of coalescence time per pair of adjacent base pairs
d_i	Number of base pairs between the i^{th} marker and the focal point x
$\mathbf{s}_i$	The latent sequence at internal node K_i. The vector of internal node haplotypes is given by $\mathbf{S} - \mathbf{H}$
y_i	The first marker after the closest recombination event to the focal point that occurred on the branch from K_i to its parent (see Figure 2.2).
z_i	The closest marker to the focal point that is not passed down to the present along any of the lines of descent from node K_i to the present (see Figure 2.2).
r_i	The first marker after the closest recombination event to the focal point that is not passed to the present down the lineage through node K_i (see Figure 2.2). That is $r_i = \min(y_i, z_i)$ and $z_i = \max(r_{c_1}, r_{c_2})$.

TABLE 2.3: Example dataset consisting of the observed haplotypes for a sample of 15 sequences

Haplotype	Count
00010	5
00011	1
00000	1
10100	3
10000	2
11000	3

2.2.1 The observed data

Assume that the sample consists of $2n$ sequences from n individuals. Table 2.3 gives an example of the sequence data that might be observed. A total of fifteen sequences were sampled having six distinct haplotypes. Here, the term sequence should be thought of as the sequence of alleles that were genotyped rather than the actual sequence of all nucleotides as the sequence of alleles used in the sampler need not consist of actual sequence data genotyped using sequencing technology. We will restrict the loci to be single nucleotide polymorphism (SNP) markers having only two alleles labelled 0 and 1. The two sequences for individual i are labelled $\mathbf{s}_{i_1}$ and $\mathbf{s}_{i_2}$, where the haplotype of each sequence $\mathbf{s}_{i_h} = (s_{i_h,1}, s_{i_h,2}, \ldots, s_{i_h,L})$ is composed of a vector of alleles at L loci. The observed sequence data therefore consists of the haplotypes $\mathbf{H} = (\mathbf{s}_{1_1}, \mathbf{s}_{1_2}, \ldots, \mathbf{s}_{n_1}, \mathbf{s}_{n_2})$.

Let d_i denote the physical location of marker i, for markers 1 to L. We assume that the distance is measured in base pairs rather than the genetic distance (cM). If genetic distance is the desired unit of distance, the recombination rate would need to be scaled accordingly so that the probability of recombination events in intervals is correctly computed.

2.2.2 The genealogy

The topology τ_x at focal point x consists of a binary tree of nodes and branches connecting the nodes. If there are n individuals there are $2n$ tip nodes, one for each individual's two sequences. The tip nodes of the tree are labelled according to the individual and sequence that they represent. For individual i, the two tip nodes corresponding to this individual are K_{i_1} for i's first sequence and K_{i_2} for i's second sequence. With $2n$ tip nodes there are $2n - 1$ internal nodes of the tree, each representing a coalescence event between two branches in the tree or equivalently a common ancestor of the tips that descend from these branches. There are a total of $4n - 1$ nodes, $2n$ tips and

$2n-1$ internal nodes. The internal nodes of the tree are labelled K_i for i from $2n+1$ to $4n-1$, with K_{2n+1} corresponding to the first coalescence event and K_{4n-1} corresponding to the root of the tree or the *most recent common ancestor* (*MRCA*) of all tips. By labelling the internal nodes from $2n+1$ to $4n-1$, the $2n$ tip nodes can also be referred to by the labels $i = 1 \dots 2n$, when convenient, without duplicating node labels. We will generally use K_i when referring to a selected node in the tree, K_a when referring to parent of K_i and K_{c_1} and K_{c_2} when referring to K_i's children.

As mentioned, the coalescence times are denoted $\mathbf{\Omega}$, where ω_k is the time interval, in coalescent time units, when there are k lineages in the tree. A unit of coalescent time is $2N$ generations, where N is the historical effective population size. It will generally be more convenient to consider the corresponding set of ordered times since the present $\mathbf{T} = (t_{2n+1}, t_{2n+2}, \dots, t_{4n-1})$ for the internal nodes. The times of the $2n$ tip nodes are all 0 since they are observed at present. To convert from $\mathbf{\Omega}$ to $\mathbf{T}$, we use $\mathbf{\Omega} = (\omega_{2n} = t_{2n+1}, \omega_{2n-1} = t_{2n+2} - t_{2n+1}, \dots, \omega_2 = t_{4n-1} - t_{4n-2})$.

Let b_i be the branch length between node K_i and its parent K_a. The branch lengths can be written as differences of the appropriate times since the present; for example $b_i = t_a - t_i$ is the branch length between nodes K_i and parent K_a.

2.2.3 Latent variables corresponding to mutation and recombination events

Following Zöllner and Pritchard (2005), we assume that recombination and mutation events occur independently on the sequence segments to the right and left of the focal point x. This allows the description to be simplified as only sampling with respect to variables on one side of x is needed. In the following definitions of the variables, we assume that the focal point is to the left of all the markers and the locations are specified relative to the focal point. However, since we are usually interested in focal points at all locations in a dataset, we give a description of the full sampling scheme involving both sides of the focal point in Section 2.5.6.

Let $\theta/2$ be the mutation rate of each SNP marker per unit coalescence time, where $\theta = 4N\mu$ and μ is the mutation rate per generation for the marker. Although $\theta/2$ typically denotes the biological mutation rate for a random base pair, this is not the case under the mutation model we consider. The mutation model, which will be described in Section 2.3, allows for a "mutation" event where half of the time the SNP does not change. Under this model, SNP markers will have a lower biological mutation rate of $\theta/4$. Additionally, the data will typically not be full DNA sequences with all variation ascertained; rather, they will be haplotypes of SNPs ascertained to be relatively common in the population. As such, the mutation rate is expected to be higher than the mutation rate of a random base pair. Hence our mutation rates are expected to be at least twice the biological mutation rates of a random base pair. More information about the definition of the mutation rate is given in Section 1.1.1.

Let $\rho/2$ be the recombination rate per pair of consecutive base pairs per unit of coalescence time. Unlike the mutation rate, this parameter can be interpreted as the standard population genetic recombination rate which is usually written as $\rho = 4Nr$, where N is the effective population size and r is the recombination rate per generation.

As discussed, in order to handle recombination, Zöllner and Pritchard (2005) proposed sampling marginal trees for the focal point x. Therefore only sequence that is passed along with that focal point to at least one present-day descendant is tracked at the internal nodes. For this reason, at each node, latent variables, r_i and z_i, are introduced that store which markers are co-inherited with the focal point. Latent variables corresponding to internal node sequences $\mathbf{s}_i$ are also introduced.

The definition of the r_i and z_i variables is complex, so to make the definition more clear the intermediate variable y_i is introduced. For each node except the MRCA consider one side of the focal point only, and let y_i be the first marker after the closest recombination event to the focal point that occurs on the branch b_i. In other words, imagine a recombination event taking place on branch b_i between markers m and $m+1$ and let the intermediate variable $y_i = m+1$ denote the index of the first marker after the location of the recombination. The sequence from y_i to L recombines in from an unknown ancestral origin and, since we are only sampling sequence that is co-inherited with the focal point, we have no information about the original sequence that was lost due to recombination. As we traverse from the tips, where we have sequence information, to the root of the tree, some of these y_i will take place in the sequence that has already been lost due to recombination events closer to the tips of the tree. As the sequence we would like to store at each node must be passed to at least one descendant in the tree, we are not interested in these recombination events occurring in sequence that is not passed to the present. Therefore, the variables r_i and z_i are introduced. They also track recombination event locations, but are defined in a recursive way so that the r_i and z_i at K_i depend on recombination events occurring on all branches of the subtree from K_i to the present.

For each node, let z_i be the closest marker to x on K_i's sequence that leaves no descendants in the present sample. For two tip nodes K_{c_1} and K_{c_2} that coalesce at K_i, $z_i = \max(y_{c_1}, y_{c_2})$. For other internal nodes, z_i will depend on the y_i in all the branches that descend from K_i. Let r_i be the closest marker to the focal point that leaves no descendants in the present sample along the path

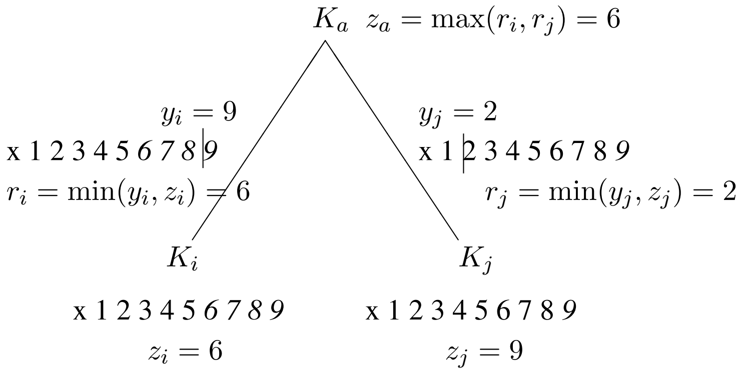


FIGURE 2.2: Illustration of the definition of r and z. The sequence data is given as "x 1 2 ..." with loci after historical recombination breakpoints italicized and current recombination breakpoints indicated by a vertical line.

through K_i to the present. That is $r_i = \min(z_i, y_i)$ and $z_i = \max(r_{c_1}, r_{c_2})$. For the tip nodes define $z = L + 1$ and for the root node (MRCA) define $r_{4n-1} = 0$.

Figure 2.2 provides an example illustrating this complex definition. Since $z_i = 6$, the sequence at node K_i between markers 6 and 9 inclusive has no descendants in the present day sample because recombination events on the subtree below have removed this sequence. On the branch from K_i to its parent a recombination event has taken place between marker 8 and 9, so $y_i = 9$. However, since it is located in the sequence that is not passed to K_i's descendants, we are not interested in this event and $r_i = 6$. On the other hand, $z_j = 9$ so that the sequence between 1 and 8 is passed to at least one of K_j's descendants. Since $y_j = 2$, a recombination event breaks up the sequence that is passed from K_a to K_j so $r_j = 2$ and only marker 1 is passed to the present along with the focal point down this subtree.

We are only interested in sequence that has descendants in the sample at present. We therefore store sequence from the first locus after the focal point to locus $z_i - 1$. Let $\mathbf{S} = (\mathbf{s}_{1_1}, \mathbf{s}_{1_2}, \ldots, \mathbf{s}_{n_1}, \mathbf{s}_{n_2}, \mathbf{s}_{2n+1}, \ldots, \mathbf{s}_{4n-1})$ be the combined vector of haplotype data on all tip and internal nodes. It will occasionally be useful to label the sequence as $\mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{4n-1})$ when the individual label is not necessary. The set $\mathbf{S} - \mathbf{H}$ consists of the latent sequence data at the internal nodes that is passed to the present via the subtree that descends from each node.

2.3 The target distribution for the haplotype sampler

The augmented data consists of $\mathbf{A} = (\boldsymbol{\Omega}, \tau_x, \theta, \rho, \mathbf{R}, \mathbf{S} - \mathbf{H})$ and we sample $\boldsymbol{\Omega}$ and τ_x from $f(\boldsymbol{\Omega}, \tau_x|\mathbf{H})$ by sampling the augmented variables from $f(\mathbf{A}|\mathbf{H})$ using Metropolis-Hastings and storing only the variables of interest. The augmented target distribution $f(\mathbf{A}|\mathbf{H})$ can be written as

$$\begin{aligned} f(\mathbf{A}|\mathbf{H}) &= f(\boldsymbol{\Omega}, \tau_x, \theta, \rho, \mathbf{R}, \mathbf{S} - \mathbf{H}|\mathbf{H}) \\ &= \frac{f(\boldsymbol{\Omega}, \tau_x, \theta, \rho, \mathbf{R}, \mathbf{s}_{1_1}, \mathbf{s}_{1_2} \ldots, \mathbf{s}_{n_1}, \mathbf{s}_{n_2}, \mathbf{s}_{2n+1}, \ldots, \mathbf{s}_{4n-1})}{\Pr(\mathbf{s}_{1_1}, \mathbf{s}_{1_2}, \ldots, \mathbf{s}_{n_1}, \mathbf{s}_{n_2})} \\ &= \frac{\Pr(\mathbf{S}|\mathbf{R}, \boldsymbol{\Omega}, \tau_x, \theta) \Pr(\mathbf{R}|\boldsymbol{\Omega}, \tau_x, \rho) h(\boldsymbol{\Omega}, \tau_x, \theta, \rho)}{\Pr(\mathbf{s}_{1_1}, \mathbf{s}_{1_2}, \ldots, \mathbf{s}_{n_1}, \mathbf{s}_{n_2})} \end{aligned} \tag{2.3}$$

Written as a product of conditionals, population genetic theory provides a model for each of the components.

2.3.1 Prior distributions for the genealogy, mutation and recombination rates: $h(\boldsymbol{\Omega}, \tau_x, \theta, \rho)$

It is reasonable to assume independence between the prior distributions for the mutation rate, recombination rate and for the gene genealogy. Therefore

$$h(\boldsymbol{\Omega}, \tau_x, \theta, \rho) = h(\boldsymbol{\Omega}, \tau_x) h(\theta) h(\rho)$$

Zöllner and Pritchard (2005) suggest uniform prior distributions for the mutation rate θ and recombination rate ρ, however they do not motivate this choice. Both rates are known to vary, both across the genome and due to non-genetic factors like gender, and some work has been done to estimate these quantities from genomic data. In particular, McVean et al. (2004) found that recombination rates estimated from a 10 Mb region on chromosome 20 had a roughly exponential distribution with values that varied two orders of magnitude above and below 1 cM/Mb. As 1cM/Mb corresponds to roughly one recombination event every 100 meioses in a 1Mb region, 1 cM/Mb corresponds to a rate of $r = 10^{-8}$ per generation for a pair of adjacent base pairs. Assuming $N = 10,000$ (see, for example, Takahata (1993)), we obtain a range of roughly $\rho = 10^{-6}$ to 10^{-2} per adjacent bases per unit of scaled time.

We therefore chose an exponential/gamma rather than a uniform prior distribution for ρ. By default, the parameters of the prior are set to exponential distribution values. Figure 2.3 plots the

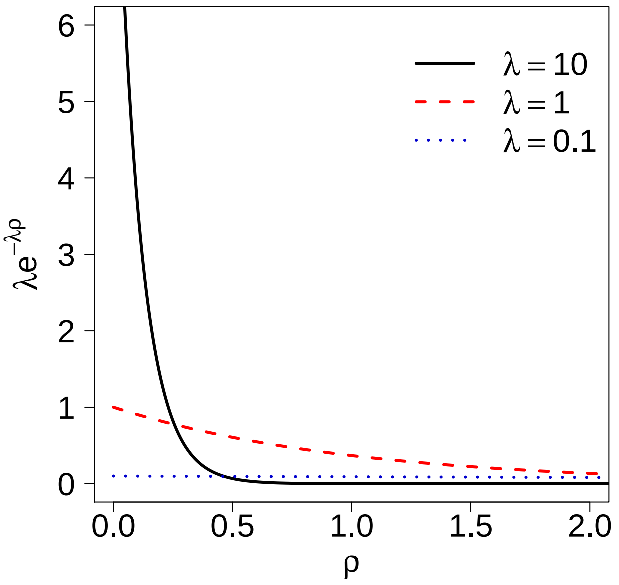


FIGURE 2.3: Exponential density functions for different values of the rate parameter λ

density for different values of the rate λ. Values of $\lambda > 10$ would result in density functions having most values of ρ that are similar to those observed in McVean et al. (2004). The value for λ should be modifiable by the user since for some datasets higher values of ρ might be expected in advance, particularly if the focal point is located in a recombination hotspot.

We have implemented both an exponential/gamma prior and a uniform prior for the mutation rate θ, however the uniform prior is currently in use. Point estimates of the biological mutation rate on the order of 10^{-8} per base pair per generation (θ on the order of 10^{-4} assuming $N = 10,000$) have been determined for the human genome (Awadalla et al., 2010; The 1000 Genomes Project Consortium, 2010); however, there is little information about the distribution of these values across the genome. Therefore, we think the uniform prior on a suitable range (l, u) is more appropriate than a gamma prior. Since the mutation rate in the mutation model is at least twice the biological rate and is for ascertained SNPs, the range for the prior should not be set based on the biological mutation rate and should be wide enough to include plausible values given the number of SNPs and length of the region under consideration.

We let the prior for $(\mathbf{\Omega}, \tau_x)$ be the standard neutral coalescent model (Kingman, 1982; Hudson, 1983), which is a model for ancestral trees of selectively neutral loci that is motivated by population genetic theory. Background information about this model is given in Section 1.1.1. Recall that $\mathbf{\Omega}$ denotes the times between successive coalescence events and τ_x the labelled topology. By the neutral coalescent model, the topology and coalescence times are independent, so $h(\mathbf{\Omega}, \tau_x) = h(\mathbf{\Omega})h(\tau_x)$. The time interval when there are l distinct lineages, ω_l, is exponentially distributed with rate $\binom{l}{2}$. Since these intercoalescence times are independent,

$$h(\mathbf{\Omega}) = \prod_{l=2}^{n} \binom{l}{2} \exp(-\binom{l}{2}\omega_l). \tag{2.4}$$

The topology τ_x describes which nodes coalesce and in which order. Under the neutral coalescent model, when there are l lineages there are $\binom{l}{2}$ different pairs that are equally likely to coalesce with each other. The probability that any particular pair coalesces is then $\binom{l}{2}^{-1}$ so that $\Pr(\tau_x)$ is

$$\prod_{i=2}^{n} \binom{i}{2}^{-1} = \frac{2^{n-1}}{n!(n-1)!}.$$

All topologies are equiprobable and therefore this term is constant.

2.3.2 Recombination Model: $\Pr(\mathbf{R}|\mathbf{\Omega}, \tau_x, \rho)$

The probability of recombination events between nodes on the tree depends on the recombination rate ρ and the branch lengths of the tree, which are determined by $\mathbf{\Omega}$ and τ_x. Due to the recursive definition of the recombination variables, the r's and z's defined on page 26, a natural model for the recombination-related variables is backward-in-time, so that the location of the closest recombination breakpoint to the focal point at a node depends on the recombination information in the node's descendants. Therefore, $\Pr(\mathbf{R}|\mathbf{\Omega}, \tau_x, \rho)$, or equivalently $\Pr(\mathbf{R}|\mathbf{T}, \tau_x, \rho)$, is found by successively conditioning on descendants. That is, the distribution for r_i depends on the r values on the subtree descending from K_i, and

$$\begin{aligned}\Pr(\mathbf{R}|\mathbf{\Omega}, \tau_x, \rho) &= \Pr(r_{1_1}|\mathbf{\Omega}, \tau_x, \rho)\Pr(r_{1_2}|r_{1_1}, \mathbf{\Omega}, \tau_x, \rho)\cdots\Pr(r_{4n-2}|r_1, \ldots, r_{4n-3}, \mathbf{\Omega}, \tau_x, \rho) \\ &= \prod_{i=1}^{n} \Pr(r_{i_1} \mid z_{i_1} = L+1, b_{i_1}, \rho)\Pr(r_{i_2} \mid z_{i_2} = L+1, b_{i_2}, \rho) \\ &\quad \times \prod_{i=2n+1}^{4n-2} \Pr(r_i \mid z_i, b_i, \rho),\end{aligned} \tag{2.5}$$

where r_i has support $\{1, 2, \ldots, z_i\}$, $z_i \in \{1, 2, \ldots, L+1\}$ and $z_i = L+1$ for the tip nodes. To determine $\Pr(r_i|z_i, b_i, \rho)$ we use the intermediate variable y_i, defined on page 26 and illustrated in Figure 2.2, which stores the marker index immediately after the closest recombination event to x (to one side of the focal point) on the branch from K_i to K_a. Then

$$\Pr(r_i = c|z_i, b_i, \rho) \quad = \quad \begin{cases} \Pr(y_i = c|b_i, \rho) & 1 \leq c < z_i \\ \Pr(y_i \geq c|b_i, \rho) = \sum_{c=z_i}^{L+1} \Pr(y_i = c|b_i, \rho) & c = z_i \\ 0 & c > z_i \end{cases} . \quad (2.6)$$

That is, $\Pr(r_i = c|z_i, b_i, \rho)$ is either the probability that the closest recombination event to x on the branch from K_i to K_a is between locus $c-1$ and c for $c \neq z_i$ or the probability that the first recombination event occurs after the c^{th} locus for $c = z_i$.

Recombination events along a lineage (time) and along a chromosome (space) are modelled as a two-dimensional Poisson process with rate $\lambda = \rho/2$. Given the branch length b_i and the marker locations relative to the focal point d_i, the probability that $y_i = c$ is the probability that no recombination events occur in the region $A = \{(x, y) : 0 < x < d_{c-1}, 0 < y < b_i\}$ and at least one event occurs in $B = \{(x, y) : d_{c-1} < x < d_c, 0 < y < b_i\}$. Letting $N(A)$ be the number of recombination events that occur in the set A and similarly for $N(B)$

$$\begin{aligned} \Pr(N(A) = 0) &= \frac{\exp(-\lambda|A|)(\lambda|A|)^0}{0!} = \exp(-\lambda|A|) \\ \Pr(N(B) \geq 1) &= 1 - \Pr(N(B) = 0) = 1 - \exp(-\lambda|B|), \end{aligned}$$

where $|A| = d_{c-1}b_i$ and $|B| = (d_c - d_{c-1})b_i$ are the areas of the corresponding regions. Since $\lambda = \rho/2$,

$$\begin{aligned} \Pr(y_i = c|b_i) &= \exp\Big(\frac{-d_{c-1}b_i\rho}{2}\Big)\Bigg(1 - \exp\Big(\frac{-(d_c - d_{c-1})b_i\rho}{2}\Big)\Bigg) \\ &= \exp\Big(\frac{-d_{c-1}b_i\rho}{2}\Big) - \exp\Big(\frac{-d_c b_i\rho}{2}\Big) \\ &= \int_{d_{c-1}}^{d_c} \frac{b_i\rho}{2} \exp\Big(\frac{b_i\rho}{2}s\Big) ds. \end{aligned}$$

The distribution of locations of recombination events is exponential with rate $b_i\rho/2$. Letting $d_0 = 0$ and $d_{L+1} = \infty$ and substituting (2.7) into (2.6) gives the model provided by Zöllner and Pritchard

(2005):

$$\Pr(r_i = c|z_i, b_i, \rho) = \begin{cases} 0 & c > z_i \\ \int_{d_{c-1}}^{d_c} \frac{b_i\rho}{2}\exp(-\frac{b_i\rho}{2}t)dt & 1 < c < z_i \\ \int_{d_{c-1}}^{\infty} \frac{b_i\rho}{2}\exp(-\frac{b_i\rho}{2}t)dt & c = z_i \end{cases} \tag{2.7}$$

2.3.3 Sequence Model: $\Pr(\mathbf{S}|\mathbf{R}, \mathbf{\Omega}, \tau_x, \theta)$

At each marker, we make the standard assumption that mutation occurs according to a Poisson process with rate $\theta/2$ per unit of coalescent time. As mentioned, a unit of coalescent time is $2N$ where N is the effective population size (see Section 1.1.1 for more detail about the coalescent model). The mutation rate $\theta/2$ is that of a pre-ascertained SNP and not the usual rate for a random base pair since the data consists of the L SNPs that were selected for genotyping and not all bases in the sequence. We are only interested in the mutation process at the polymorphic SNPs for which we have data. Assuming a mutation event has occurred, the new allele is drawn randomly from the set of two possible alleles at the SNP so the locus can remain as the original allele. This model does not exclude multiple mutation events occurring at a marker more than once over the topology. This might be unrealistic for SNP data showing only two allelic types. However, trees with recurrent mutation events should be accepted with low probability since the mutation rate is low.

The model for sequence evolution is forward-in-time, so that the sequence at a node depends on the sequence at its parent's node. Therefore, $\Pr(\mathbf{S}|\mathbf{R}, \mathbf{\Omega}, \tau_x, \theta)$, or equivalently $\Pr(\mathbf{S}|\mathbf{R}, \mathbf{T}, \tau_x, \theta)$ can be rewritten by successively conditioning on ancestors:

$$\begin{aligned} \Pr(\mathbf{S}|\mathbf{R}, \mathbf{\Omega}, \tau_x, \theta) &= \Pr(\mathbf{s}_{4n-1}|\mathbf{R}, \mathbf{\Omega}, \tau_x, \theta) \cdots \Pr(\mathbf{s}_{1_1}|\mathbf{s}_{1_2}, \ldots, \mathbf{s}_{4n-1}, \mathbf{R}, \mathbf{\Omega}, \tau_x, \theta) \\ &= \Pr(\mathbf{s}_{4n-1}|z_{4n-1}, b_{4n-1}, \theta) \\ &\quad \times \prod_{i \in \{1_1, 1_2, \ldots n_1, n_2, 2n+1, \ldots 4n-2\}} \Pr(\mathbf{s}_i|\mathbf{s}_a, r_i, z_i, b_i, \theta) \end{aligned} \tag{2.8}$$

where recall $\mathbf{s}_i$ is the sequence at node K_i and subscript a is used to indicate the ancestor of K_i. At a given node we only store the sequence that is passed to present in at least one descendant, the sequence from 1 to $z_i - 1$. Conditional on the parental sequence, all loci up to r_i are independent of each other and of the sequence between r_i and $z_i - 1$. Therefore we can write each term in (2.8) as a product of terms corresponding to the alleles before and after the recombination breakpoint between $r_i - 1$ and r_i. For $i \neq 4n-1$

$$\Pr(\mathbf{s}_i | \mathbf{s}_a, r_i, z_i, b_i, \theta) = \left[\prod_{j=1}^{r_i - 1} \Pr(s_{i,j} | s_{a,j}, b_i, \theta) \right] \Pr(h_{r_i -> z_i - 1}) \tag{2.9}$$

where $h_{r_i -> z_i - 1}$ represents the haplotype of the sequence segment between r_i and $z_i - 1$. For example, if a sample's haplotype is (1,0,0,1,1,0) then $h_{3->6} = (0,1,1,0)$. For the MRCA ($i = 4n-1$),

$$\Pr(\mathbf{s}_{4n-1} | z_{4n-1}, b_{4n-1}, \theta) = \Pr(h_{1->z_{4n-1}}).$$

We have assumed that mutation events are also Poisson distributed with rate $\theta/2$. Given that a mutation event has occurred, the type of the allele is then chosen randomly from the two allelic types. This leads to the following expression for each component of the first term in (2.9):

$$\Pr(s_{i,j} = a_1 | s_{a,j} = a_2, b_i, \theta) = \begin{cases} \frac{1}{2}(1 - e^{-\theta b_i/2}) & \text{if } a_1 \neq a_2 \\ \frac{1}{2}(1 - e^{-\theta b_i/2}) + e^{-\theta b_i/2} & \text{if } a_1 = a_2 \end{cases}, \tag{2.10}$$

where a_1 and a_2 are the allelic types at the j^{th} marker. This formula is explained as follows:

Case I ($a_1 \neq a_2$): Since the parental and offspring alleles are different, at least one mutation event has occurred on the branch from K_i to K_a and at the most recent event the new allele chosen is different from the previous allele. The probability of no mutations occurring during time b_i is $e^{-\theta b_i/2}$ and the probability of at least one mutation event in this time period is $1 - e^{-\theta b_i/2}$. The probability that the last allele sampled is a_1 is 1/2. The required probability is therefore $\frac{1}{2}(1 - e^{-\theta b_i/2})$.

Case II ($a_1 = a_2$): If the parental and offspring alleles are the same either no mutation has occurred or a mutation has occurred but the most recent new allele sampled is the same as the old allele. For the latter explanation, the probability is the same as given in Case I. For the former explanation, the probability of no mutations occurring during time b_i is $e^{-\theta b_i/2}$. Summing the probability of the two disjoint events gives $\frac{1}{2}(1 - e^{-\theta b_i/2}) + e^{-\theta b_i/2}$.

For $j \geq r_i$ the allelic state at locus j does not depend on K_i's parent with respect to the tree τ_x since this sequence will have recombined in from another ancestral source. That is, the ancestral nodes do not provide any information about their allelic type. For these markers, modelling the probability of their allelic state requires a haplotype probability model that corresponds to the probability of sampling a particular chromosomal segment from the general population. We will assume that these probabilities are constant over time so that any sequence at a node that has recombined in at any point back in time will have the sequence drawn from the same frequency distribution.

Let $q < v \in \{1, 2, \ldots, L+1\}$ be two marker positions. Zöllner and Pritchard (2005) assumed that $\Pr(h_{q->v})$ could be modelled as a first-order Markov process. That is, the allele at a marker depends only on the allele of the marker immediately preceding it. Thus, the probability of haplotype $h_{q->v}$ is

$$\begin{aligned}
\Pr(h_{q->v}) &= \Pr(a_q)\Pr(a_{q+1}|a_q)\cdots\Pr(a_v|a_{v-1}) \\
&= \Pr(a_q)\frac{\Pr(h_{a_{q+1}->a_q})}{\Pr(a_q)}\cdots\frac{\Pr(h_{a_{v-1}->a_v})}{\Pr(a_{v-1})}. \qquad (2.11)
\end{aligned}$$

The general approach is described in McPeek and Strahs (1999), where they mention that a k^{th} order Markov process can be assumed but that in general there will not be enough information in the data to estimate such a model.

We must still specify values for these two-locus probabilities. Since the haplotype probability model is also important when only genotype data are observed, more detail about this model can be found in Chapter 3. For now, note that Zöllner and Pritchard (2005) chose to fix these probabilities with estimates based on counting the alleles and two-locus haplotypes in the observed haplotype data then adding one to each of the counts. The addition of one to the counts could be justified by assuming that the haplotype/allele proportions $\boldsymbol{p}$ have a Dirichlet$(\mathbf{1})$ prior distribution (see for example Weir (1996) p83-86). This is the conjugate prior for the multinomial distribution and, if all Dirichlet parameters are equal to 1, the prior distribution is uniform over the unit simplex.

Letting the vector of allele or haplotype counts be $\boldsymbol{n} = (n_1, n_2, \ldots, n_k)$, the posterior distribution of $\boldsymbol{p}$ is also Dirichlet with parameter vector $(n_1+1, n_2+1, \ldots, n_k+1)$. The posterior mean can be used as an estimator for p_j. The estimate of the population proportion of allele j is

$$\mathrm{E}(p_j|n = (n_0, n_1)) = \frac{n_j + 1}{\sum_{i=0}^{1}(n_i+1)}$$

for $j = 0, 1$. The estimate of the population proportion of haplotype j is

$$\mathrm{E}(p_j|n = (n_0, n_1, n_2, n_3)) = \frac{n_j + 1}{\sum_{i=0}^{3}(n_i+1)}$$

for $j = 0, 1, 2, 3$ corresponding to the two-locus haplotypes $00, 01, 10$, and 11. A benefit of the Bayesian approach to estimating the allelic and haplotypic proportions with the posterior mean is that the frequency estimates will not be 0, even if the count is actually 0.

2.4 Proposal distributions for sampling components of $\mathbf{T}$, $\mathbf{R}$ and $\mathbf{S}$

In Section 2.5, five of the six proposal schemes that were summarized in Table 2.1 will be described in more detail. Three of the schemes, the local updates, major and minor topology rearrangements, will include updates to the time t_i, recombination variable r_i, and sequence variable $\mathbf{s}_i$ of a single node. Even though the local updates differ from the topology changes, the proposal distributions for updating the components are the same. Therefore, in this section, we introduce the proposal distributions that are used to update these single components of $\mathbf{A}$.

As mentioned in Section 2.2.2 and summarized in Table 2.2, the times on the tree can be equivalently represented by either the times since the present $\mathbf{T}$ or the intercoalescence times $\boldsymbol{\Omega}$. In this section, it is more convenient to work with the times since the present, $\mathbf{T}$, and we therefore provide a proposal distribution for the component t_i.

The proposal distributions for t_i, r_i and $\mathbf{s}_i$ are all motivated based on the decomposition of $f(\mathbf{A}|\mathbf{H})$ given in equation (2.3):

$$f(\mathbf{A}|\mathbf{H}) \propto \Pr(\mathbf{S}|\mathbf{R}, \mathbf{T}, \tau_x, \theta) \Pr(\mathbf{R}|\mathbf{T}, \tau_x, \rho) h(\mathbf{T}) \Pr(\tau_x) h(\theta) h(\rho).$$

We sample each individual component t_i, r_i and $\mathbf{s}_i$ from a distribution motivated by the corresponding conditional distribution. That is, t_i is sampled from an approximation to $h(t_i|T_{-i})$, r_i from an approximation to $\Pr(r_i|\mathbf{R}_{-i}, \mathbf{T}, \tau_x, \rho)$ and $\mathbf{s}_i$ from $\Pr(\mathbf{s}_i|\mathbf{S}_{-i}, \mathbf{R}, \mathbf{T}, \tau_x, \theta)$.

2.4.1 Sampling new t_i in the proposals that update $\mathbf{T}$

The proposal distribution for component t_i is motivated based on $h(t_i|\mathbf{T}_{-i})$; however, rather than condition on all elements of $\mathbf{T}_{-i}$ in the proposal distribution, we condition only on the times of the nodes adjacent to K_i. Letting K_{c_1} and K_{c_2} be K_i's two children, and K_a be K_i's parent (see Figure

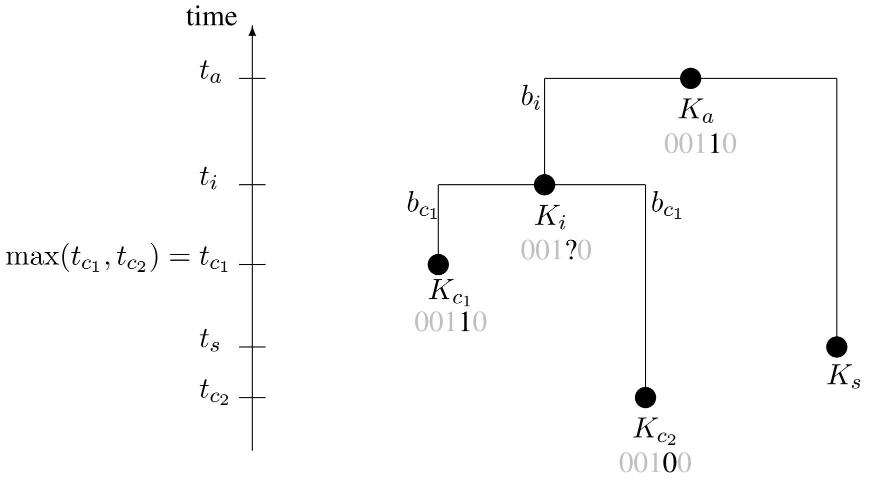


FIGURE 2.4: Notation for the proposal distributions for t_i, r_i and $\mathbf{s}_i$ and for the local updates

2.4 for the nodes involved in sampling these values), the proposal distribution for t_i is

$$q(t_i|t_{c_1}, t_{c_2}, t_a) = \begin{cases} U(t_{c_1}, t_a) & i \neq 4n-1 \\ \exp(-(t_i - t_{c_1}))I[t_i > t_{c_1}] & i = 4n-1 \end{cases} \tag{2.12}$$

The explanation for this proposal distribution is now given.

Case I: $i \neq 4n-1$; K_i is not the MRCA

Assume for the moment that K_i's sibling, K_s, is younger than K_i's oldest child; that is, $t_s < t_{c_1}$. In that case, considering only the times of the nodes in this cluster and ignoring the nodes in the rest of the tree, K_i corresponds to the only coalescence event between nodes K_{c_1} and K_a. The intercoalescence time when there are three lineages is exponential(3), but conditional on the coalescence time being between t_{c_1} and t_a, it has a uniform distribution on (t_{c_1}, t_a). In general, t_s may be greater than t_{c_1} and the tree typically contains more than four nodes in total, so using a uniform distribution is an approximation to $h(t_i|\mathbf{T}_{-i})$.

Case II: $i = 4n-1$; K_i is the MRCA

If K_i corresponds to the MRCA, t_i is the time of the final coalescence event. The intercoalescence time from K_{c_1} to K_i is ω_2 and from the coalescent model, ω_2 has an exponential(1) distribution. Since t_i is the time since the present, it can be written as $t_i = \omega_2 + t_{c_1}$ which has density

$$\exp(-(t_i - t_{c_1}))I[t_i > t_{c_1}].$$

This value is generated by sampling from the standard exponential distribution and adding t_{c_1}.

2.4.2 Sampling new r_i in the proposals that update R

Zöllner and Pritchard (2005) allowed r_i values to be sampled that were incompatible with the other r values on the tree. These incompatible values would then be rejected automatically. In our updates to the r_i variables, we restrict the support of the proposed r_i to be compatible with all other r variables on the tree. The proposed value for r_i is constrained by the length of sequence that is passed to the present through node K_i, that is, by z_i. Since the r's are defined recursively, the value r_a of the parent, K_a, of K_i is also restricted by the value r_i. Therefore, when only r_i is changed, we must propose an r value that does not make the current value of r_a impossible. For this reason, the support for r_i, $\mathcal{S}(\mathbf{R}_{-i})$, depends on the r and z values of nodes in the vicinity of K_i. The nodes that restrict r_i depend on the update type, therefore the support for each update is given in the sections describing the local updates, major and minor topology rearrangements.

With the support $\mathcal{S}(\mathbf{R}_{-i})$ giving the set of values for r_i that are compatible with the data at surrounding nodes, the conditional distribution for component r_i can be written

$$\Pr(r_i|\mathcal{S}(\mathbf{R}_{-i}), \mathbf{T}, \tau_x, \rho) = \frac{\Pr(r_i|z_i, b_i, \rho)}{\sum_{r^* \in \mathcal{S}(\mathbf{R}_i)} \Pr(r^*|z_i, b_i, \rho)} 1[r_i \in \mathcal{S}_i(\mathbf{R}_{-i})].$$

We thus take as our proposal distribution for r_i

$$q(r_i|\mathcal{S}(\mathbf{R}_{-i}), \mathbf{T}, \tau_x, \rho) = \Pr(r_i|\mathcal{S}(\mathbf{R}_{-i}), \mathbf{T}, \tau_x, \rho). \tag{2.13}$$

Under the model for r_i described in Section 2.3.2 that assumes recombination events have a Poisson distribution, each term is calculated according to equation (2.7).

2.4.3 Sampling new s_i in the proposals that update S

The full conditional for component $\mathbf{s}_i$ is

$$\Pr(\mathbf{s}_i|\mathbf{S}_{-i}, \mathbf{R}, \mathbf{T}, \tau_x, \theta) \quad = \quad \frac{\Pr(\mathbf{s}_i, \mathbf{S}_{-i}|\mathbf{R}, \mathbf{T}, \tau_x, \theta)}{\sum_{\mathbf{s}^* \in \mathcal{S}(\mathbf{S}_{-i})} \Pr(\mathbf{s}^*, \mathbf{S}_{-i}|\mathbf{R}, \tilde{t}_i, \mathbf{T}_{-i}, \theta)} \tag{2.14}$$

where $\mathcal{S}(\mathbf{S}_{-i})$ is the support for s_i and from (2.8)

$$\Pr(\mathbf{S} \mid \mathbf{R}, \mathbf{\Omega}, \theta) = \Pr(\mathrm{s}_{4n-1}) \prod_{i=1}^{4n-2} \Pr(\mathrm{s}_i | \mathrm{s}_a, b_i, z_i, r_i, \tau_x, \theta).$$

Substituting (2.8) in to Equation (2.14) and cancelling terms that do involve s_i gives

$$\frac{\Pr(\mathrm{s}_i|\mathrm{s}_a, b_i, r_i, z_i, \theta) \Pr(\mathrm{s}_{c_1}|\mathrm{s}_i, b_{c_1}, r_{c_1}, z_{c_1}, \theta) \Pr(\mathrm{s}_{c_2}|\mathrm{s}_i, b_{c_2}, r_{c_2}, z_{c_2}, \theta)}{\sum_{\mathrm{s}^* \in \mathcal{S}(\mathbf{S}_{-i})} \Pr(\mathrm{s}^*|\mathrm{s}_a, b_i, r_i, z_i, \theta) \Pr(\mathrm{s}_{c_1}|\mathrm{s}^*, b_{c_1}, r_{c_1}, z_{c_1}, \theta) \Pr(\mathrm{s}_{c_2}|\mathrm{s}^*, b_{c_2}, r_{c_2}, z_{c_2}, \theta)}, \quad (2.15)$$

where K_a is the parent of K_i, and the children of K_i are K_{c_1} and K_{c_2}.

The support $\mathcal{S}(\mathbf{S}_{-i})$ is the set of s^* compatible with the sequences at other nodes. Note that any constraint on s_i acts through s_a, s_{c_1} and s_{c_2}. However, any sequence s_i is possible; s_a, s_{c_1} and s_{c_2} only make certain sequence values of s_i more or less probable. Theoretically, any number of mutations can occur on the branches to and from K_i. Therefore, the sum in the denominator is over all possible sequences for nodes one to $z_i - 1$, many of which will be improbable given the surrounding sequences. An efficient way to sample is therefore needed.

Recall that the sequence s_i is itself a vector, with L elements corresponding to the alleles at each locus. We can therefore sample a new sequence, $\tilde{\mathrm{s}}_i$, by sampling a new allele at each locus $s_{i,j}$ starting at the first locus, $j = 1$, and finishing at $j = z_i - 1$. Since we are only sampling alleles at loci that are passed to at least one descendant in the tree, all alleles at markers at z_i and above are given the value '-' with probability one.

To motivate a proposal distribution for sampling a new allele at the j^{th} locus, we consider the alleles at the j^{th} locus in the surrounding nodes. Example sequences are given below the node labels in Figure 2.4. When sampling a new value for $s_{i,4}$, which is marked by a question mark in the figure, the proposal distribution should account for the fact that the parent and one child have a '1' in the fourth position and the second child has a '0' in that position. If $s_{i,j} = 0$ and no recombination events occurred before this locus on any of the branches between these nodes, then a mutation from allele '1' to '0' would have had to occur on branch b_i, a mutation from '0' to '1' would have had to occur on b_{c_1}, and no mutation would have occurred on b_{c_2}. The events on each branch are independent of each other so we let the probability of sampling a '0' be the probability of each event, which is given by the mutation model in Section 2.3.3, multiplied together. In this example, a '1' is probably a much better choice since only a single mutation event is required to produce the set of alleles.

The allele at $s_{i,j}$ also depends on whether this locus is inherited from K_a or not, and whether it is passed to K_{c_1} and K_{c_2}. For example, if the j^{th} locus is not passed to K_{c_1}, due to a recombination event occurring on b_{c_1}, then the j^{th} allele on K_{c_1}'s sequence gives us no information about $s_{i,j}$ because the allele at this locus is from a different parent than K_i. If in Figure 2.4 $r_{c_1} = 2$, then a mutation from '1' to '0' could have occurred either on branch b_i or b_{c_2} and therefore $s_{i,j}$ is much more likely to be a '0' than in the previous example. Similarly, if $r_i < 4$ then the j^{th} locus was not passed from K_a to K_i and instead this portion of the sequence recombined in from an unknown ancestor. For this case, the probability for the locus is taken from the haplotype probability model also given in Section 2.3.3. We therefore use the allele at the previous locus, $s_{i,j-1}$, in the probability distribution for sampling $\tilde{s}_{i,j}$

To summarize, our proposal distribution is motivated by combining the inheritance of the j^{th} locus from K_a through nodes K_i, K_{c_1} and K_{c_2}. Let the proposal distribution for the j^{th} locus be

$$\begin{aligned}
& q(s_{i,j}|s_{i,j-1}, s_{c_1,j}, s_{c_2,j}, s_{a,j}, \mathbf{R}, \mathbf{T}, \tau, \theta) \\
= \quad & \Pr(s_{i,j}|s_{i,j-1}, s_{c_1,j}, s_{c_2,j}, s_{a,j}, \mathbf{R}, \mathbf{T}, \tau_x, \theta) \\
\propto \quad & \Pr(s_{i,j}, s_{c_1,j}, s_{c_2,j}|s_{i,j-1}, s_{a,j}, \mathbf{R}, \mathbf{T}, \tau_x, \theta) \\
= \quad & \Pr(s_{i,j}|s_{i,j-1}, s_{a,j}, \mathbf{R}, \mathbf{T}, \theta) \Pr(\mathbf{s}_{c_1,j}|s_{i,j}, \mathbf{R}, \mathbf{T}, \theta) \Pr(\mathbf{s}_{c_1,j}|s_{i,j}, \mathbf{R}, \mathbf{T}, \theta). \qquad (2.16)
\end{aligned}$$

The last line follows from the conditional independence of $s_{c_1,j}$ and $s_{c_2,j}$ given $s_{i,j}$, their parent's allele at the j^{th} locus.

Each term in equation (2.16) is specified by either the mutation model or the haplotype frequency model, depending on whether the j^{th} locus is inherited from the parent in τ_x or not. If K_i is not the MRCA, that is $i \neq 4n - 1$,

$$\Pr(s_{i,j}|s_{i,j-1}, s_{a,j}, \mathbf{R}, \mathbf{T}, \theta) \quad = \quad \begin{cases} \Pr(s_{i,j}|s_{a,j}, b_i, \theta) & j < r_i \\ \Pr(s_{i,j}) & j = r_i \\ \Pr(s_{i,j}|s_{i,j-1}) & r_i < j < z_i - 1 \end{cases} \qquad (2.17)$$

with $\Pr(s_{i,j}|s_{a,j}, b_i, \theta)$ given in equation (2.10), and $\Pr(s_{i,j})$ and $\Pr(s_{i,j}|s_{i,j-1})$ given by the haplotype probability model. If K_i is the MRCA, then K_i does not have a parent in the tree and therefore the probability of the j^{th} locus is based on the haplotype probability model:

$$\Pr(s_{i,j}|s_{i,j-1},\mathbf{R},\mathbf{T},\theta) = \begin{cases} \Pr(s_{4n-1,j}) & j=1 \\ \Pr(s_{4n-1,j}|s_{4n-1,j-1}) & 1<j\leq z_i-1 \end{cases}.$$

Each of $\Pr(\mathbf{s}_{c_1,j}|s_{i,j},\mathbf{R},\mathbf{T},\theta)$ and $\Pr(\mathbf{s}_{c_2,j}|s_{i,j},\mathbf{R},\mathbf{T},\theta)$ can be specified based on the mutation or haplotype probability models. If for example $j \geq r_{c_1}$, then this locus will not be passed from node K_a to K_i and the probability of its allele depends on the haplotype probability model. Since the terms in the haplotype frequency model do not depend on the parental allele at the j^{th} locus, this term would cancel from equation (2.16). A similar argument can be made if $j \geq r_{c_2}$. Therefore

$$\begin{aligned} \Pr(\mathbf{s}_{c_1,j}|s_{i,j},\mathbf{R},\mathbf{T},\theta) &= \begin{cases} \Pr(s_{c_1,j}|s_{i,j},b_{c_1},\theta) & j<r_{c_1} \\ 1 & j\geq r_{c_1} \end{cases} \\ \Pr(\mathbf{s}_{c_2,j}|s_{i,j},\mathbf{R},\mathbf{T},\theta) &= \begin{cases} \Pr(s_{c_2,j}|s_{i,j},b_{c_2},\theta) & j<r_{c_2} \\ 1 & j\geq r_{c_2} \end{cases} \end{aligned}$$

The proposal distribution for the full sequence $\mathbf{s}_i$ is found by multiplying the proposal probabilities for each locus and will be proportional to

$$\begin{aligned} & q(\mathbf{s}_i|\mathbf{S}_{-i},\mathbf{R},\mathbf{T},\tau_x,\theta) \\ = \; & \Pr(s_{i,r_i})\Pr(s_{i,r_i+1}|s_{i,r_i})\cdots\Pr(s_{i,z_i-1}|s_{i,z_i-2}) \\ & \times \prod_{j=1}^{r_i-1}\Pr(s_{i,j}|s_{a,j},b_i,\theta)) \prod_{j=1}^{j=r_{c_1}-1}\Pr(s_{c_1,j}|s_{i,j},b_{c_1},\theta) \prod_{j=1}^{j=r_{c_2}-1}\Pr(s_{c_2,j}|s_{i,j},b_{c_1},\theta) \\ = \; & \Pr(h_{r_i->z_i-1}) \prod_{j=1}^{r_i-1}\Pr(s_{i,j}|s_{a,j},b_i,\theta)) \prod_{j=1}^{j=r_{c_1}-1}\Pr(s_{c_1,j}|s_{i,j},b_{c_1},\theta) \\ & \times \prod_{j=1}^{j=r_{c_2}-1}\Pr(s_{c_2,j}|s_{i,j},b_{c_1},\theta). \end{aligned} \tag{2.18}$$

Note that these are the same as equation (2.15) after substitution of terms from the sequence model and haplotype probability model; we are just sampling locus-by-locus.

2.5 Description of our implementation of the five proposal schemes

The six proposal distributions proposed by Zöllner and Pritchard (2005) were briefly summarized in Table 2.1. In this section, we provide more information on the implementation of five of these proposal types: the update to θ, the update to ρ, the local updates and the major and minor topology updates. Although the "time-swap" update was implemented, results are similar when it is not used and it is therefore not described here.

2.5.1 Updating the mutation rate θ

Zöllner and Pritchard (2005) suggested proposing candidate values $\tilde{\theta}$ from a uniform distribution on $(\theta^{(t)}/c, c\theta^{(t)})$, where $\theta^{(t)}$ is the value of θ at the t^{th} iteration, $c = 2$, and all other variables are kept constant. However, with this proposal distribution it is possible to propose a $\tilde{\theta}$ that is outside the prior support of θ; $f(\tilde{\theta}, \mathbf{A}_{-\theta} \mid \mathbf{H})$ will be 0 and the proposal will automatically be rejected.

We instead propose θ from a uniform proposal distribution with support restricted to lie within the support of the uniform(l, u) prior. This is achieved with a uniform proposal distribution on $(\max(l, \theta/2), \min(u, 2\theta))$:

$$Q_1(\tilde{\mathbf{A}} \mid \mathbf{A}) = \begin{cases} \frac{1}{\min(u,2\theta)-\max(l,\theta/2)} 1[\max(l, \theta/2) < \tilde{\theta} < \min(u, 2\theta)] & \text{if } \tilde{\mathbf{A}} = (\tilde{\theta}, \mathbf{A}_{-\theta}) \\ 0 & \text{otherwise} \end{cases} .$$

From equation (2.2), the Metropolis-Hastings acceptance probability for this update is

$$\alpha_1(\mathbf{A}, \tilde{\mathbf{A}}) = \min\left\{\frac{f(\tilde{\mathbf{A}}|\mathbf{H})Q_1(\mathbf{A} \mid \tilde{\mathbf{A}})}{f(\mathbf{A}|\mathbf{H})Q_1(\tilde{\mathbf{A}} \mid \mathbf{A})}, 1\right\} \quad = \quad \min\left\{\frac{f(\tilde{\theta}, \mathbf{A}_{-\theta} \mid \mathbf{H})\frac{1}{\min(u,2\tilde{\theta})-\max(l,\tilde{\theta}/2)}}{f(\theta, \mathbf{A}_{-\theta} \mid \mathbf{H})\frac{1}{\min(u,2\theta)-\max(l,\theta/2)}}, 1\right\}$$

Note that $1[\theta/2 < \tilde{\theta} < 2\theta] = 1$ because of the uniform proposal and therefore $1[\tilde{\theta}/2 < \theta < 2\tilde{\theta}] = 1$ so these terms do not need to be included in the acceptance probability. Since only θ is updated this expression can be simplified. Recall the factorization of $f(\mathbf{A}|\mathbf{H})$ in equation (2.3)

$$f(\mathbf{A}|\mathbf{H}) \propto \Pr(\mathbf{S} \mid \mathbf{R}, \mathbf{\Omega}, \tau_x, \theta) \Pr(\mathbf{R} \mid \mathbf{\Omega}, \tau_x, \rho) h(\mathbf{\Omega}) \Pr(\tau_x) h(\theta) h(\rho).$$

Terms in this product that are not functions of θ will cancel from the numerator and denominator of the acceptance probability, leaving only $\Pr(\mathbf{S} \mid \mathbf{R}, \mathbf{\Omega}, \tau_x, \theta)$ and $h(\theta)$. Since the prior for θ,

$h(\theta)$, is uniform and our proposal distribution only samples candidate θ in the uniform range, $h(\theta)$ is constant and also cancels from the fraction.

Now letting tip nodes be labelled from 1 to $2n$, recall from (2.8) that $\Pr(\mathbf{S} \mid \mathbf{R}, \mathbf{\Omega}, \tau_x, \theta)$ can be written as

$$\Pr(\mathbf{S}|\mathbf{R}, \mathbf{\Omega}, \tau_x, \theta) \quad = \quad \Pr(\mathbf{s}_{4n-1}|z_{4n-1}, b_{4n-1}, \theta) \prod_{i=1}^{4n-2} \Pr(\mathbf{s}_i|\mathbf{s}_a, r_i, z_i, b_i, \theta),$$

where from (2.9)

$$\Pr(\mathbf{s}_i \mid \mathbf{s}^a, b_i, z_i, r_i, \theta) = \Pr(h_{r_i->z_i-1}) \prod_{j=1}^{r_i-1} \Pr(s_{i,j} \mid s_{a,j}, b_i, \theta),$$

r_i, z_i, $s_{i,j}$ are as defined in Section 2.2 and K_a is K_i's parent. Since the haplotype probability $\Pr(h_{r_i->z_i-1})$ does not depend on θ it will also cancel from the acceptance probability

To summarize, at the $t + 1^{st}$ step, we propose a new value $\tilde{\mathbf{A}} = (\tilde{\theta}, \mathbf{A}_{-\theta})$ by sampling $\tilde{\theta}$ from a uniform distribution conditional on the current value $\theta^{(t)}$ and leaving $\mathbf{A}_{-\theta}$ unchanged. We accept this value with probability

$$\alpha_1(\mathbf{A}, \tilde{\mathbf{A}}) = \min\left\{ \frac{\left[\prod_{i=1}^{4n-1} \prod_{j=1}^{r_i-1} \Pr(s_{i,j} \mid s_{a,j}, b_i, \tilde{\theta})\right] \frac{1}{\min(u,2\tilde{\theta})-\max(l,\tilde{\theta}/2)}}{\left[\prod_{i=1}^{4n-1} \prod_{j=1}^{r_i-1} \Pr(s_{i,j} \mid s_{a,j}, b_i, \theta)\right] \frac{1}{\min(u,2\theta)-\max(l,\theta/2)}}, 1 \right\}. \tag{2.19}$$

2.5.2 Updating the recombination rate ρ

Proposals for ρ are sampled from $U(\rho^{(t)}/c, c\rho^{(t)})$ with $c = 2$.

The Metropolis-Hastings acceptance probability is

$$\alpha_2(\mathbf{A}, \tilde{\mathbf{A}}) = \min\left\{ \frac{f(\tilde{\mathbf{A}} \mid \mathbf{H}) Q_2(\mathbf{A} \mid \tilde{\mathbf{A}})}{f(\mathbf{A} \mid \mathbf{H}) Q_2(\tilde{\mathbf{A}} \mid \mathbf{A})}, 1 \right\} = \min\left\{ \frac{f(\tilde{\rho}, \mathbf{A}_{-\rho}|\mathbf{H}) \frac{1}{c\tilde{\rho}-\tilde{\rho}/c}}{f(\rho, \mathbf{A}_{-\rho}|\mathbf{H}) \frac{1}{c\rho-\rho/c}}, 1 \right\}.$$

This expression can be further simplified since it includes terms that are constant in the numerator and denominator. Recall from equation (2.3) that

$$f(\mathbf{A}|\mathbf{H}) \propto \Pr(\mathbf{S} \mid \mathbf{R}, \mathbf{\Omega}, \tau_x, \theta) \Pr(\mathbf{R} \mid \mathbf{\Omega}, \tau_x, \rho) h(\mathbf{\Omega}) \Pr(\tau_x) h(\theta) h(\rho).$$

Terms in this product that are not functions of ρ will cancel, leaving

$$\alpha_2(\mathbf{A}, \tilde{\mathbf{A}}) = \min\left\{ \frac{\Pr(\mathbf{R} \mid \boldsymbol{\Omega}, \tau_x, \tilde{\rho}) h(\tilde{\rho}) \frac{1}{c\tilde{\rho} - \tilde{\rho}/c}}{\Pr(\mathbf{R} \mid \boldsymbol{\Omega}, \tau_x, \rho) h(\rho) \frac{1}{c\rho - \rho/c}}, 1 \right\}. \tag{2.20}$$

The prior distribution for ρ, $h(\rho)$, is given in Section 2.3.1 and $\Pr(\mathbf{R} \mid \boldsymbol{\Omega}, \tau_x, \rho)$ is given in equation (2.5).

2.5.3 The local updates: updating the node times, recombination variables and sequence at internal nodes

The proposal type called the local updates systematically updates the recombination information $\mathbf{R}$, sequence $\mathbf{S}$ and times since the present $\mathbf{T}$ for each node, holding the topology τ_x, mutation rate θ, and recombination rate ρ constant. Our local updates differ from those described in Zöllner and Pritchard (2005). They proposed updating r_i, time t_i and sequence $\mathbf{s}_i$ for node K_i, although it is unclear whether the acceptance/rejection step is applied to the set of all nodes or to each node in turn. We assume the latter as it will result in much less drastic changes and more of the changes are likely to be accepted.

The tree structure of the data imposes a dependence of values among neighbouring nodes. The proposed update strategy can lead to inconsistencies for the data stored at other nodes but not yet updated. For example modifying a node's r_i value may lead to an automatic change of z_a for K_a, the parent of K_i, since $z_a = \max(r_i, r_s)$ (see Figure 2.4 for the relevant notation). If z_a were to increase due to an update to r_i, then the parental sequence $\mathbf{s}_a$ would also have to be extended since we only store sequence information at a node up to z_a. If only the triplet $(r_i, t_i, \mathbf{s}_i)$ is updated, then updates that affect z_a are not possible and so cannot be considered. One could restrict the proposal distribution so that z_a does not change. However, since one purpose of the local updates is to propose new recombination break points, this may be too restrictive. For example if $r_i = z_a$ and $r_s < z_a$ then the support for $\tilde{r}_i$ is z_a and no change to r_i can be made. We will limit this type of restriction to the topology changes described in Sections 2.5.4 and 2.5.5 since updates to r's for topology changes are done solely to ensure compatibility of the new topology.

Our implementation updates each internal node K_i in turn, from first coalescence event to last, by sampling new values for $(t_i, r_{c_1}, r_{c_2}, \mathbf{s}_i)$, where node K_{c_1} is the older child of K_i and K_{c_2} is the younger child of K_i. After all four new values for node K_i are proposed, the set is either accepted or rejected as a block. Thus, the local updates is comprised of $2n-1$ sub-steps each updating $(t_i, r_{c_1}, r_{c_2}, \mathbf{s}_i)$ for the node labelled K_i.

Referring to Figure 2.4, the local updates for node K_i are proposed as follows:

1. Sample $\tilde{t}_i$ from $q(t_i|t_{c_1}, t_{c_2}, t_a)$ as described in Section 2.4.1.
2. Sample $\tilde{r}_{c_1}$ from $q(r_{c_1}|\mathcal{S}(\mathbf{R}_{-c_1}), \tilde{\mathbf{T}}, \tau_x, \rho)$ as described in Section 2.4.2 and support $\mathcal{S}(\mathbf{R}_{-c_1}) = \mathcal{S}_1(z_{c_1}, z_{c_2}, r_i)$ given below.
3. Sample $\tilde{r}_{c_2}$ from $q(r_{c_2}|\mathcal{S}(\tilde{\mathbf{R}}_{-c_2}), \tilde{\mathbf{T}}, \tau_x, \rho)$ as described in Section 2.4.2 and support $\mathcal{S}(\tilde{\mathbf{R}}_{-c_2}) = \mathcal{S}_2(z_{c_2}, \tilde{r}_{c_1}, r_i)$ given below.
4. Sample $\tilde{\mathbf{s}}_i$ from $q(\mathbf{s}_i|\mathbf{S}_i, \tilde{\mathbf{R}}, \tilde{\mathbf{T}}, \tau_x, \theta)$ as described in Section 2.4.3.
5. Compute the acceptance probability to determine whether the updates to node K_i are accepted or rejected.

Each node from K_{2n+1} to K_{4n-1} is updated in turn using this procedure.

The support for r_{c_1}, $\mathcal{S}_1(z_{c_1}, z_{c_2}, r_i)$, is determined by how r_{c_1} is constrained by z_{c_1}, r_i and z_{c_2}. By definition of r_{c_1} on page 26, we know that $r_{c_1} \le z_{c_1}$. For the lower bound, recall that r_i tells us that K_i's ancestor's genetic material between x and marker $r_i - 1$ inclusive flows through K_i to the present. Therefore, at least one of K_i's two descendants, K_{c_1} and K_{c_2}, has to pass this material to the present. Since r_{c_2} will be updated after r_{c_1}, consider z_{c_2}. If $z_{c_2} \ge r_i$ its possible for the sequence between x and $r_i - 1$ to descend via K_{c_2} to the present and r_{c_1} must only be greater than or equal to 1. However, if $z_{c_2} < r_i$ the sequence between x and $r_i - 1$ can not descend to the present through K_{c_2} so it must do so through K_{c_1}. Hence, if $z_{c_2} < r_i$, it is required that $r_{c_1} \ge r_i$. Figure 2.5 gives two examples illustrating how to determine the lower bound. To summarize,

$$\mathcal{S}_1(z_{c_1}, z_{c_2}, r_i) = \begin{cases} \{r_i, \ldots, z_{c_1}\} & \text{if } z_{c_2} < r_i \\ \{1, \ldots, z_{c_1}\} & \text{if } z_{c_2} \ge r_i \end{cases}. \tag{2.21}$$

The support $\mathcal{S}_2(z_{c_2}, \tilde{r}_{c_1}, r_i)$ for r_{c_2} also ensures that r_i is compatible with K_{c_1} and K_{c_2}. Since $\tilde{r}_{c_1}$ has already been proposed, the value for $\tilde{r}_{c_2}$ is constrained by z_{c_2}, $\tilde{r}_{c_1}$ and r_i. The maximum value for r_{c_2} is z_{c_2}. The minimum value is determined by whether the material that is passed to the present from K_i can do so through K_{c_1} ($\tilde{r}_{c_1} \ge r_i$) or not ($\tilde{r}_{c_1} < r_i$). If $\tilde{r}_{c_1} < r_i$, then $\tilde{r}_{c_2}$ must be greater than or equal to r_i since K_{c_2} must allow the genetic material to pass to the present. On the

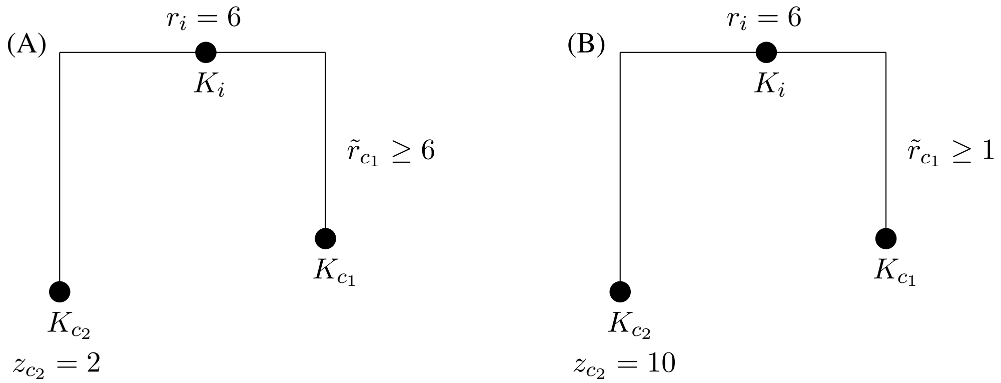


FIGURE 2.5: Two examples for determining the lower bound for $\tilde{r}_{c_1}$. (A) Since $r_i = 6$, either $\tilde{r}_{c_1}$ or $\tilde{r}_{c_2}$ must be greater than or equal to 6. However, $z_{c_2} = 2$ so it is not possible to propose $\tilde{r}_{c_2} \geq 6$. Therefore, we must propose $\tilde{r}_{c_1} \geq 6$. (B) Since $r_i = 6$, either $\tilde{r}_{c_1}$ or $\tilde{r}_{c_2}$ must be greater than or equal to 6. Since $z_{c_2} = 10$, it is possible to sample $\tilde{r}_{c_1} \geq 6$ so there is no constraint on the lower bound for $\tilde{r}_{c_1}$.

other hand, if $\tilde{r}_{c_1} \geq r_i$, then there is no constraint on the lower bound except that it is greater than or equal to 1. Therefore

$$\mathcal{S}_2(z_{c_2}, \tilde{r}_{c_1}, r_i) = \begin{cases} \{r_i, \ldots, z_{c_2}\} & \text{if } \tilde{r}_{c_1} < r_i \\ \{1, \ldots, z_{c_2}\} & \text{if } \tilde{r}_{c_1} \geq r_i \end{cases}. \tag{2.22}$$

The acceptance probability for the update to node K_i is given by

$$\alpha_{3i}(\mathbf{A}, \tilde{\mathbf{A}}) = \min\left\{\frac{f(\tilde{\mathbf{A}} \mid \mathbf{H})Q_{3i}(\mathbf{A} \mid \tilde{\mathbf{A}})}{f(\mathbf{A} \mid \mathbf{H})Q_{3i}(\tilde{\mathbf{A}} \mid \mathbf{A})}, 1\right\}.$$

where

$$\begin{aligned} Q_{3i}(\tilde{\mathbf{A}}|\mathbf{A}) &= q(\tilde{t}_i|t_{c_1}, t_{c_2}, t_a)q(\tilde{r}_{c_1}|\mathcal{S}(\mathbf{R}_{-c_1}), \tilde{\mathbf{T}}, \tau_x, \rho)q(\tilde{r}_{c_2}|\mathcal{S}(\tilde{\mathbf{R}}_{-c_2}), \tilde{\mathbf{T}}, \tau_x, \rho) \\ &\quad q(\tilde{\mathbf{s}}_i|\mathbf{S}_i, \tilde{\mathbf{R}}, \tilde{\mathbf{T}}, \tau_x, \theta) \end{aligned}$$

The first term is given in equation (2.12); the second and third terms in equation (2.13) and the last term in (2.18). As with the updates to θ and ρ however, the fraction simplifies due to constant terms in the numerator and denominator. The ratio of the densities $\frac{f(\tilde{\mathbf{A}}|\mathbf{H})}{f(\mathbf{A}|\mathbf{H})}$ will consist of terms that are functions of $t_i, r_{c_1}, r_{c_2}, z_i,$ and $\mathbf{s}_i$. Since the rate of the distribution for the intercoalescence times depends on the number of lineages, the ratio can also include terms where the intercoalescence time has not changed but the rate has changed due to a reordering of coalescence events. Coalescence events having times in (t_{c_1}, t_a) may be reordered by the update of t_i to $\tilde{t}_i$.

2.5.4 The major topology rearrangement

The fourth update outlined in Zöllner and Pritchard (2005) results in a modification to the topology of the genealogy. A node is selected and removed from its current location and made to coalesce with another node. Figure 2.6 illustrates the topology change, and a brief outline of our implementation of the update is now given.

1. Randomly select a target node K_i that can be removed from its location without causing incompatibility of the variables at surrounding nodes. Referring to Figure 2.6 for notation and Figure 2.7 for an illustration of this condition, such a node K_i must have $z_o \leq \max(r_q, z_j)$. Here z_o summarizes the sequence in K_o that makes it to present, r_q summarizes the sequence in K_o that flows through K_q's line of descent and makes it to present and z_j summarizes the sequence in K_j that makes it to present. To move K_i, K_p must be dissolved and a new line of descent must be established directly from the grand-parental node K_o to the sister node K_j. Consequently, a new r_j must be drawn. The condition, $z_o \leq \max(r_q, z_j)$, checks whether the sequence that makes it to present from K_i's grand-parental node can do so either through K_q's line of descent (r_q) or through K_j's line of descent in the new tree. If not, the resulting tree would have incompatible r values and would be rejected automatically.

2. A second node K_c is randomly selected based on $\mathbf{s}_c$'s similarity to $\mathbf{s}_i$. Here we first find all nodes with parents older than t_i and that are eligible to become K_i's sibling. A sequence similarity-based weight is computed for each node and a node is sampled based on the weight. After the node is selected, the new topology $\tilde{\tau}_x$ has K_i and K_c coalescing at node $\tilde{K}_p$.

3. A time for the node $\tilde{K}_p$ is sampled from the proposal distribution $q(t_p|t_i, t_c, t_k)$ given in Section 2.4.1.

4. The change of topology from τ_x to $\tilde{\tau}_x$ results in new lines of descent for the sequence from nodes K_o and K_k to their children and new branch lengths between these nodes. In general, when lines of descent are deleted and moved, new values for the recombination and sequence variables for related nodes must be sampled in order to ensure that these new paths are both

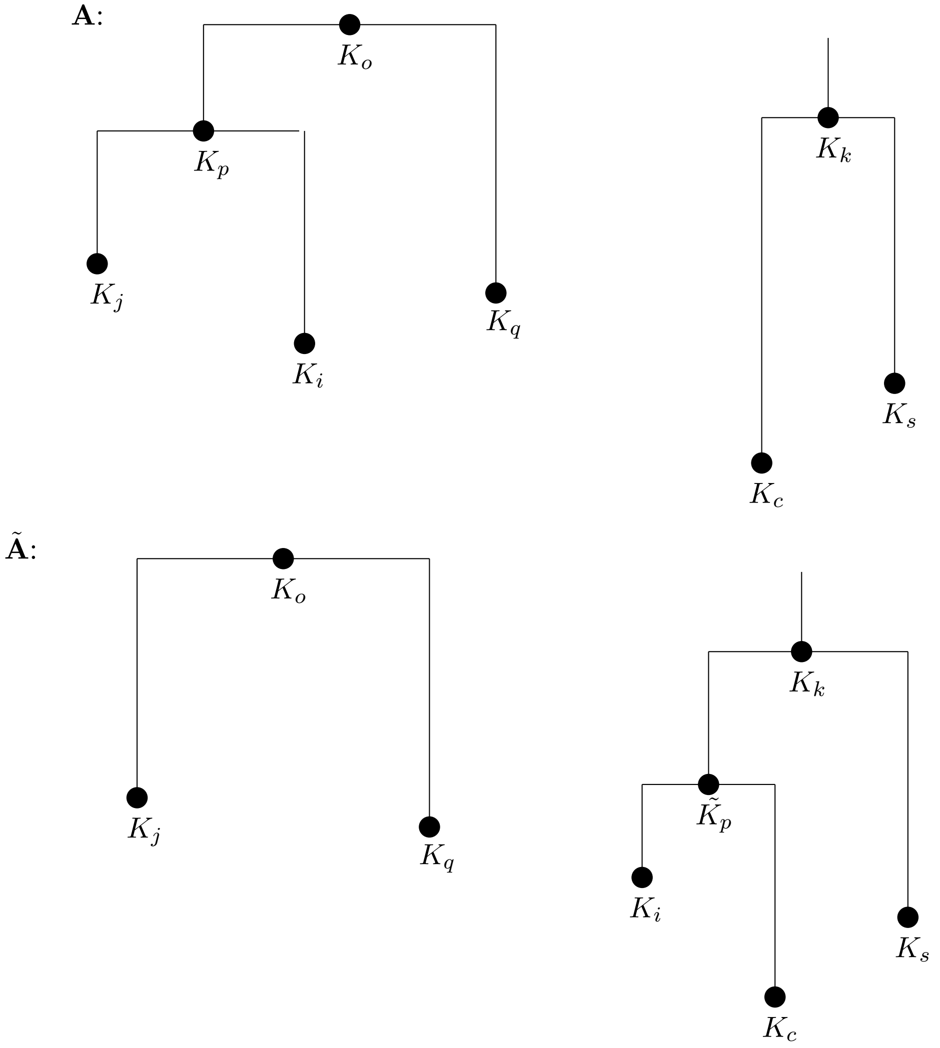


FIGURE 2.6: Illustration of the major topology rearrangement. The topology is initially as shown in **A**. Node K_i is randomly sampled from the set of nodes that are eligible to be moved. Node K_c is sampled based on its sequence similarity to K_i from the set of nodes that are eligible to be K_i's sibling. In $\tilde{\mathbf{A}}$, K_p is dissolved and K_i and K_c now coalesce at a new parent $\tilde{K_p}$. K_j now coalesces directly with node K_o. A new coalescence time is drawn for node $\tilde{K_p}$.

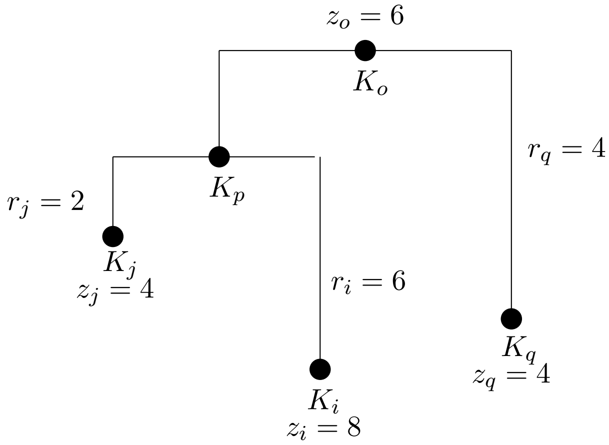


FIGURE 2.7: Example of incompatibility in the tree due to a node being moved. K_i can not be moved since moving it will cause an incompatible z value of its grandparent K_o. Without K_i, it is not possible for $z_o = 6$ since $z_j = 4$, $r_q = 4$ and $z_o = \max(r_j, r_q) \leq \max(z_j, r_q)$

compatible with the sequence known to pass to the present through this lineage and likely given the new branch lengths. Therefore for this particular topological rearrangement, we must draw $\tilde{r}_j$, $\tilde{r}_c$, $\tilde{r}_i$, $\tilde{r}_p$ and $\tilde{\mathrm{s}}_p$. The r values are sampled from the distribution given in Section 2.4.2 with the support described below and $\tilde{\mathrm{s}}_p$ is sampled from the distribution given in Section 2.4.3

More detailed information about the major topology rearrangement is now given. The first component of this update is sampling the node K_i from the set of nodes that can be moved. Requiring that $z_o \leq \max(r_q, z_j)$ ensures that the sequence that descends from z_o does not need to descend through node K_i in order for the tree to remain compatible. An illustration of a node that can't be moved is given in Figure 2.7.

Sampling a node K_i to move requires enumerating all nodes in τ_x that are able to be moved and randomly choosing one, so that the probability of choosing K_i is

$$\Pr(\text{choose } K_i|\mathbf{A}) = \frac{1}{m}, \tag{2.23}$$

where m is the size of the set of nodes meeting the condition given above.

For the second component of the update, a node K_c is chosen based on its sequence similarity to s_i. No detail on the sequence similarity measure was given in Zöllner and Pritchard (2005), so we describe our measure here. In addition, we found that more conditions for selecting a second node were required to ensure that the resulting tree was valid and to differentiate this update from a minor topology update, which will be described in Section 2.5.5.

We choose a second node K_c from the set of nodes with parents older than t_i and excluding:

- K_i itself
- the sibling of K_i
- the parent of K_i
- the aunt, niece, or grandparent of K_i, which would turn this update into a minor topology update. Doing so causes increased dependence among the r values since the nodes are more closely related, and therefore requires different proposal distributions that take this dependence into account.

For two sequences, $\mathbf{s}_i$ and $\mathbf{s}_c$, define a sequence similarity score as

$$p_{ic} = \begin{cases} \frac{\sum_{k=1}^{\min(z_i-1,z_c-1)} 1(s_{i,k}=s_{c,k})}{\min(z_i-1,z_c-1)} & \text{node } K_c \text{ is eligible to coalesce with } K_i \\ 0 & \text{node } K_c \text{ is not eligible to coalesce with } K_i \end{cases}.$$

This score counts the alleles that $\mathbf{s}_i$ and $\mathbf{s}_c$ share in common. Ideally, we would only like to compare loci that both sequences inherited from their shared parent. However, when selecting the new sibling, we don't have this information as $\tilde{r}_i$ and $\tilde{r}_c$ haven't yet been sampled. We therefore only compare the loci that both would have inherited from their parent if no recombination events occur on the branches $\tilde{b}_i$ and $\tilde{b}_c$. This sequence is between locus 1 and $\min(z_i - 1, z_c - 1)$. By dividing by $\min(z_i - 1, z_c - 1)$ we account for the different sizes of sequences being compared. Let the probability of choosing node K_c to coalesce with K_i be

$$w_{ic} = \frac{p_{ic}}{\sum_{k=1}^{4n-1} p_{ik}}. \tag{2.24}$$

This measure as written does not account for the case when one or both of the nodes passes no sequence to the present due to either $z_i = 1$ or $z_c = 1$, since we would get $p_{ic} = 0/0$. We could let $p_{ic} = 0$ but if there is only one eligible node, and it has 0 weight, the value for w_{ic} would be undefined. Even if there are multiple eligible nodes, we will also want to make it possible, though unlikely, to choose a node with no sequence similarity. Therefore, a small value of $\epsilon = 0.0001$ is added simply if the node is eligible to be the new sibling. The ϵ value controls how likely it will be for two dissimilar nodes to coalesce. Large values of ϵ make these events more likely and smaller values make it less likely. In the case of one eligible node this value of p_{ic} will ensure that $w_{ic} = 1$.

This similarity score is written assuming that the focal point is to the left of the markers, which isn't the case in general. The score should reflect the similarity of alleles to both the left and right of the focal point. This is discussed in more detail in Section 2.5.6.

After the topology change is made, that is node K_i is made to coalesce with K_c at node $\tilde{K}_p$, a time is sampled for $\tilde{K}_p$ from proposal distribution $q(t_p|t_i, t_c, t_k)$ given in Section 2.4.1. The next step of the update involves sampling r values for the surrounding nodes from $q(r|\mathcal{S}(\mathbf{R}_{-}), \mathbf{T}, \tau_x, \rho)$ given in Section 2.4.2, where $r \in \{\tilde{r}_j, \tilde{r}_c, \tilde{r}_i, \tilde{r}_p\}$ and the support is given below, and a sequence for the new parent $\tilde{K}_p$ from proposal distribution $q(\mathbf{s}_p|\mathbf{S}_p, \tilde{\mathbf{R}}, \tilde{\mathbf{T}}, \tau_x, \theta)$ given in Section 2.4.3. These ensure that the r values remain compatible with the tree and that K_p's sequence is similar to both its new offspring and its new parent.

As described in Section 2.4.2, we sample the r values based on the recombination probability model, but conditional on the r values being possible given the nodes whose r's and z's do not change. This alters the support for each of the nodes whose r's are modified. Referring to Figure 2.6 for relevant notation, new r_j, r_i, r_c and r_p are sampled. The support for each of the r updates is given below and is based on ensuring that z_o and z_k remain the same in $\tilde{\mathbf{A}}$.

$\tilde{r}_j$ Restrict the support so that z_o does not change. If z_o were allowed to change to a $\tilde{z_o}$ such that $\tilde{z_o} < z_o$, the new value may not permit the r_o value, which remains unchanged by this update. If $\tilde{z_o} > z_o$, we would need to sample the sequence $\mathbf{s}_o$ for loci $z_o \ldots \tilde{z_o} - 1$, as illustrated in Figure 2.8.

After the topology change, the two children of K_o are K_j and K_q. Since node K_i can be moved without causing inconsistencies, that is, $z_o \leq \max(r_q, z_j)$, we know that either node K_j or K_q carries the sequence of loci 1 to $z_0 - 1$ from node K_o to the present. The support for $\tilde{r}_j$ therefore depends on the value of r_q, which is not modified by this update. The support is

$$\mathcal{S}(\mathbf{R}_{-j}) = \mathcal{S}_3(z_j, r_q, z_o) = \begin{cases} \{1, \ldots, \min(z_j, z_o)\} & \text{if } r_q = z_o \\ z_o & r_q < z_o \end{cases},$$

which is explained as follows:

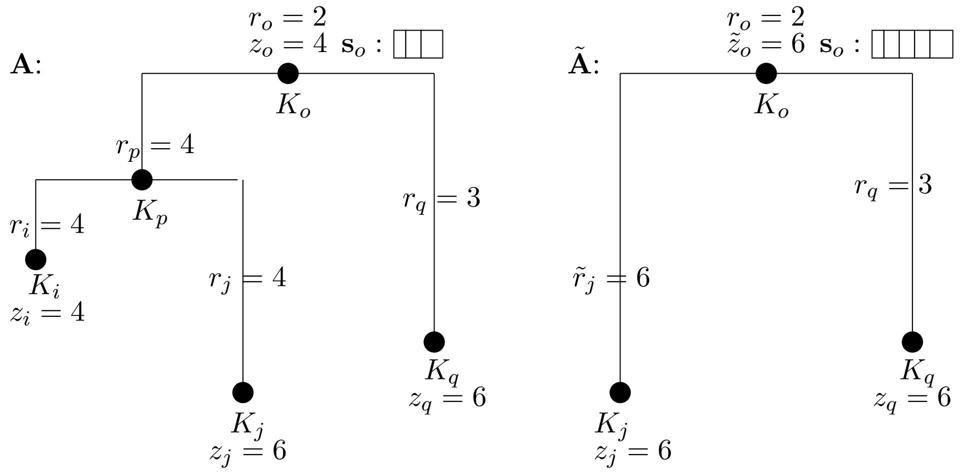


FIGURE 2.8: Illustration of the effect of allowing an $\tilde{r_j}$ that leads to $\tilde{z}_0 > z_0$. In $\mathbf{A}$, the sequence $\mathbf{s}_o$ has length 3 since $z_o = 4$. In $\tilde{\mathbf{A}}$, if the same support for $\tilde{r_j}$ is used as in the local updates, $\tilde{r}_j \in \{1, 2, 3, 4, 5, 6\}$ since $r_q > r_o$. In this example, $\tilde{r}_j = 6$ thereby increasing the length of $\mathbf{s}_o$ to 5. The sequence at loci 4 and 5, $s_{o,4}$ and $s_{o,5}$, would therefore have to be sampled to complete the update.

If $r_q < z_o$, the sequence in K_o that makes it to present has to go through K_j, since by definition $z_o = \max(r_q, \tilde{r}_j)$ in the new topology. Therefore we must propose $\tilde{r_j} = z_0$.

If $r_q = z_o$ (it cannot be greater than z_o since $z_o = \max(r_q, r_p)$) then node K_j need not carry the sequence from its parent to the present and there is no minimum restriction on its value other than that it be greater than or equal to 1. The only additional restriction to its maximum value is that $\tilde{r}_j \leq z_o$ since in the new topology $z_o = \max(\tilde{r}_j, r_q)$.

$\tilde{r}_i$ Referring to Figure 2.6, we see that the sequence in K_k that makes it to present can do so through the lines of descent going through K_c and K_s. Thus there are no additional restrictions on $\tilde{r}_i$ and $\mathcal{S}(\mathbf{R}_{-i}) = \mathcal{S}_4(z_i) = \{1, \ldots, z_i\}$.

$\tilde{r}_c$ We again restrict the support, in this case so that z_k doesn't change. To ensure that z_k does not change, the support will depend on the value r_s, which is not changed by this update. Prior to the topology change, we have $z_k = \max(r_s, r_c) \leq \max(r_s, z_c)$ because $z_c \geq r_c$. After the change, we require $\tilde{z}_k = z_k = \max(r_s, \tilde{r}_p)$.

If $z_k = r_s$, the sequence in K_k that makes it to present can do so through the line of descent that goes through K_s. There are therefore no restrictions on the support for $\tilde{r}_c$ and

$\tilde{r_c} \in \{1, \ldots, z_c\}$.

If $z_k > r_s$ then in the update to r_p we will require $z_k = \tilde{r}_p$ and therefore we must sample an $\tilde{r}_c$ that ensures that $z_k \le \tilde{z}_p = \max(\tilde{r}_i, \tilde{r}_c)$. We require that the sequence from K_k passes through K_c's line of descent in the new $\tilde{\mathbf{A}}$ since this ensures that K_i is eligible for a major topology update to take $\tilde{\mathbf{A}}$ back to $\mathbf{A}$. We therefore choose $\tilde{r_c} \ge z_k$ and set the support for $\tilde{r_c}$ to be $\{z_k, \ldots, z_c\}$.

To summarize

$$\mathcal{S}(\mathbf{R}_{-c}) = \mathcal{S}_5(z_c, r_s, z_k) = \begin{cases} \{1, \ldots, z_c\} & \text{if } z_k = r_s \\ \{z_k, \ldots, z_c\} & z_k > r_s \end{cases} .$$

$\tilde{r}_p$ The above proposals will set $\tilde{z}_p = \max(\tilde{r}_i, \tilde{r}_c)$. The value for $\tilde{r}_p$ should have support

$$\mathcal{S}(\mathbf{R}_{-p}) = \mathcal{S}_3(\tilde{z}_p, r_s, z_k) = \begin{cases} \{1, \ldots, \min(z_k, \tilde{z}_p)\} & \text{if } r_s = z_k \\ z_k & \text{if } r_s < z_k \end{cases}$$

to ensure that z_k does not change. Note that this is the same support as for the $\tilde{r}_j$ update and is explained similarly.

The acceptance probability for the major topology update is given by

$$\alpha_4(\mathbf{A}, \tilde{\mathbf{A}}) = \min\left\{\frac{f(\tilde{\mathbf{A}} \mid \mathbf{H})Q_4(\mathbf{A} \mid \tilde{\mathbf{A}})}{f(\mathbf{A} \mid \mathbf{H})Q_4(\tilde{\mathbf{A}} \mid \mathbf{A})}, 1\right\},$$

where

$$\begin{aligned} Q_4(\tilde{\mathbf{A}} \mid \mathbf{A}) \quad = \quad & \frac{1}{m} \times \frac{p_{ic}}{\sum_{k=1}^{4n-1} p_{ik}} \times q(\tilde{t}_p | t_i, t_c, t_k) \times q(\tilde{r}_j | \mathcal{S}(\mathbf{R}_{-j}), \tilde{\mathbf{T}}, \tilde{\tau}_x, \rho) \\ & \times q(\tilde{r}_i | \mathcal{S}(\tilde{\mathbf{R}}_{-i}), \tilde{\mathbf{T}}, \tilde{\tau}_x, \rho) \times q(\tilde{r}_c | \mathcal{S}(\tilde{\mathbf{R}}_{-c}), \tilde{\mathbf{T}}, \tilde{\tau}_x, \rho) \\ & \times q(\tilde{r}_p | \mathcal{S}(\tilde{\mathbf{R}}_{-p}), \tilde{\mathbf{T}}, \tilde{\tau}_x, \rho) \times q(\tilde{\mathbf{s}}_p | \mathbf{S}_{-p}, \tilde{\mathbf{R}}, \tilde{\mathbf{T}}, \tilde{\tau}_x) \end{aligned}$$

The first three terms are given in equations (2.23), (2.24) and (2.12), respectively; the fourth through seventh terms in equation (2.13) and the last term in (2.18).

As with the previous updates, the fraction simplifies due to constant terms in $\frac{f(\tilde{\mathbf{A}}|\mathbf{H})}{f(\mathbf{A}|\mathbf{H})}$. This update changes τ_x, t_p, r_i, r_j, r_q, r_p and $\mathbf{s}_p$. All terms directly involving any of these variables will not

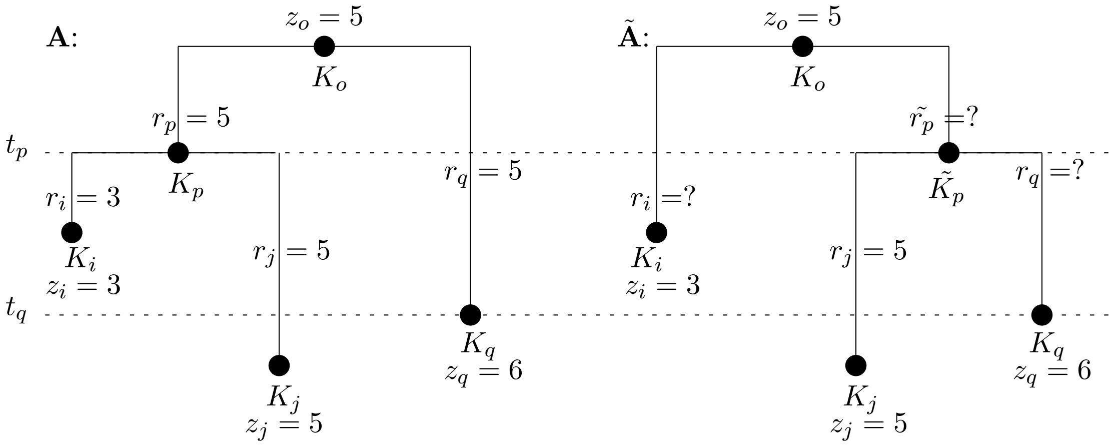


FIGURE 2.9: Illustration of a minor topology rearrangement. **A**: Before any changes. Node K_i is chosen at random from the internal nodes of the tree. Node K_j is K_i's sibling. **Ã**: The tree after a minor rearrangement. Node K_i now coalesces directly with K_o and K_j is now K_i's niece.

cancel. As with the local updates, since the update to t_p can alter the order of coalescence events, coalescence times that have had their rate changed due to a re-ordering of coalescence events will also not cancel. Finally, recall from Section 2.3 that $\Pr(\tau_x)$ is constant for all τ_x and therefore cancels from the ratio. However, the change in topology means that some nodes have new parents. For example, in Figure 2.6, K_j has parent K_p before the topology change and parent K_o after the topology change. Even if neither parent nor child sequence has changed, the terms in $\Pr(\mathbf{S}|\mathbf{T}, \mathbf{R}, \tau_x, \theta)$ will have changed due to the new parental relationship.

2.5.5 The minor topology rearrangement

The major topology rearrangement is a fairly drastic change. Not only does it find a completely new location for a node, but many new other variables must be sampled in order to make the tree compatible after the change. Zöllner and Pritchard (2005) also outlined a minor topology change, which they probably introduced in order to improve topology mixing. In our test datasets, the acceptance rate of the major topology change has been around 5% and this less drastic minor topology change has had better acceptance rates at or above 30%. The minor topology involves a swap of aunt and niece nodes, so that a node's sibling becomes its aunt and a node's aunt becomes its sibling. This topology rearrangement and the associated node labels are illustrated in Figure 2.9.

Our implementation of the minor topology change includes updating additional variables than were originally suggested. We also found that additional restrictions were required in order to ensure compatibility after the topology change. These changes are now summarized:

1. Even though the time of node $\tilde{K}_p$ is the same in $\tilde{\tau}_x$ as that of K_p in τ_x, the branch lengths b_i and b_q change since their parental node changes in $\tilde{\tau}_x$. For this reason, the swap could make r_i unlikely given $\tilde{b}_i$ and similarly for r_q. We therefore also update r_i and r_q.

2. Imposing the condition that $t_p > t_q$, or K_p must be older than its sibling K_q, ensures that the topology change produces a valid tree. After the rearrangement K_p will become the parent of K_q. Since t_p and t_q are not altered by this update, as shown in Figure 2.9, without this condition the parent could be younger than the child and the update would automatically be rejected.

3. Although Zöllner and Pritchard (2005) imposed a constraint on randomly sampling an internal node to be moved, this may lead to inefficient sampling and instead we allow sampling of any node except the MRCA. The problem is illustrated in Figure 2.10. For certain rearrangements, if terminal nodes are not sampled then even though $Q_5(\tilde{\mathbf{A}}|\mathbf{A}) \neq 0$, it is not possible to move from $\tilde{\mathbf{A}}$ to $\mathbf{A}$ so $Q_5(\mathbf{A}|\tilde{\mathbf{A}}) = 0$. Allowing tip nodes to be sampled ensures that the reverse rearrangement of $\tilde{\mathbf{A}}$ to $\mathbf{A}$ always has non-zero probability. If the topology change had been defined so that K_p was selected and became its sibling's parent, this constraint would not be required.

4. As with the major topology change described in Section 2.5.4, we will require $\tilde{z}_o = z_o$ so that a new sequence $\mathbf{s}_o$ need not be sampled. This results in a restricted support for $\tilde{r}_i$, $\tilde{r}_q$ and $\tilde{r}_p$.

More detail about the implementation of this update is now given.

The first step of this update is selecting the node to apply the topology change to. Referring to Figure 2.9, a node is selected from those whose parent's time, t_p, is greater than the time of the parent's sibling t_q. That is, the parent K_p is the older child of the grandparent K_o. The set of nodes in τ_x that can be moved is enumerated and a choice is randomly made. If the number of nodes that can be moved is m, then the probability of choosing K_i is

$$\Pr(\text{choose } K_i|\mathbf{A}) = \frac{1}{m}. \tag{2.25}$$

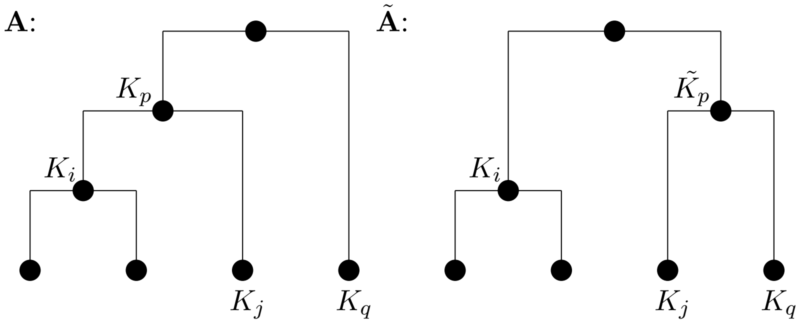


FIGURE 2.10: Illustration of why tip nodes should be eligible to be selected for the minor topology change. **A**: Node K_i is randomly chosen for this topology rearrangement. Its parent will become K_j and K_q's parent. $\tilde{\mathbf{A}}$: The resulting tree after the swap. Note that now if we wanted to return to the initial tree, we would have to first randomly select K_q. However, K_q is not an internal node and would be ineligible unless we allow the selection of tip nodes.

The resulting topology $\tilde{\tau}_x$ is illustrated in Figure 2.9. K_i now has sibling $\tilde{K_p}$ and K_j and K_q also become siblings.

After the topology change is made, certain r values are changed to generally increase compatibility of the variables with the new topology. As with the major topology change described in Section 2.5.4, we sample new r values so that the proposed values are compatible with the unchanged r and z values at nodes in the neighbourhood of the topology change. This is slightly trickier to accomplish with the minor topology change however, since the r values correspond to immediate relatives. This is perhaps why Zöllner and Pritchard (2005) chose not to update r_i and r_q.

As with all other updates to components of $\mathbf{R}$, new r values are proposed from the recombination probability model $q(r_i|\mathcal{S}(\mathbf{R}_{-i}), \mathbf{T}, \tau_x, \rho)$ given in Section 2.4.2 but conditional on r values being in a restricted support in order to ensure compatibility with the rest of the latent data on the tree. As with the major topology rearrangement, to determine the supports for $\tilde{r_i}, \tilde{r_q}$ and $\tilde{r_p}$, we must determine the constraints imposed by the surrounding nodes, which is now outlined. Referring to **A** in Figure 2.9, the sequence from 1 to $z_o - 1$ descends to the present from node K_o to one or both of K_p or K_q. If it descends through K_p, then it must have descended through either K_i or K_j. Therefore at least one of r_i, r_j or r_q must be greater than or equal to z_o. By definition of the z variables, at least one of z_i, z_j and z_q also must be greater than or equal to z_o. In $\tilde{\mathbf{A}}$ in Figure 2.9, the sequence now descends to the present through either $\tilde{K_p}$ or K_i. If it descends through $\tilde{K_p}$, then it must have descended to the present through either K_q or K_j, so that again at least one of

$\tilde{r}_i < z_i$, $r_j < z_j$ or $\tilde{r}_q < z_q$ must be greater than or equal to z_o. The support for each of $\tilde{r}_i, \tilde{r}_q$ and $\tilde{r}_p$ then depends on the values of z_i, r_j and z_q, which of these are greater than or equal to z_o, and therefore which can allow the genetic material to be passed to the present. With this brief outline of the constraints imposed by the surrounding nodes, more technical details can be given.

$\tilde{r}_i$ There are two cases to consider for determining the support for $\tilde{r}_i$ under the new topology $\tilde{\tau}_x$: whether the $z_o - 1$ bases could or could not have descended to the present through $\tilde{K_p}$. They could have descended to the present through $\tilde{K_p}$ when $\max(r_j, z_q) \geq z_o$ and could not have when $\max(r_j, z_q) < z_o$.

If $\max(r_j, z_q) \geq z_o$, the $z_o - 1$ bases could have descended to the present through $\tilde{K_p}$. Therefore r_i is constrained only by z_i ($r_i \leq z_i$ for all nodes) and z_o ($z_o = \max(r_i, r_q)$ so $r_i \leq z_o$). Therefore, $\tilde{r}_i \in \{1, \ldots, \min(z_i, z_o)\}$.

If $\max(r_j, z_q) < z_o$, the $z_o - 1$ bases could not have descended to the present through $\tilde{K_p}$, and therefore must have descended through K_i. We therefore must have $\tilde{r}_i = z_o$ to ensure $\tilde{z}_o = z_o$.

To summarize, the support is

$$\mathcal{S}(\mathbf{R}_{-i}) = \mathcal{S}_6(z_i, r_j, z_q, z_o) = \begin{cases} z_o & \max(r_j, z_q) < z_o \\ 1, \ldots, \min(z_i, z_o) & \text{otherwise} \end{cases}.$$

$\tilde{r}_q$ For the support $\mathcal{S}(\mathbf{R}_{-q}) = \mathcal{S}_7(z_q, r_j, \tilde{r}_i, z_o)$ we separate into two cases depending on the value of $\tilde{r}_i$ chosen in the previous step. Either $\tilde{r}_i = z_o$ or $\tilde{r}_i < z_o$.

$\tilde{r}_i = z_o$:

For the lower bound on $\tilde{r}_q$, since $\tilde{r}_i = z_o$, we know that the sequence that descends to the present does so through $\tilde{K_p}$ and therefore r_i only has lower bound of 1.

For the upper bound on $\tilde{r}_q$, $\tilde{r}_q$ is not forced to be smaller than z_o because K_q is closer to the tips of the tree than K_o and is not a daughter node of K_o. Therefore, the only upper bound is z_q.

Therefore, the support for $\tilde{r}_q$ when $\tilde{r}_i = z_o$ is $\mathcal{S}(\mathbf{R}_{-q})\{1, \ldots, z_q\}$.

$\tilde{r}_i < z_o$:

This condition implies that $\tilde{r}_p = z_o$ so that the $z_o - 1$ loci from K_o pass through $\tilde{K_p}$ to the present. This requires that at least one of r_j or $\tilde{r}_q$ be greater than or equal to z_o.

For the lower bound on $\tilde{r}_q$, if $r_j \geq z_o$, then $\tilde{r}_q$ does not have to be greater than z_o and the lower bound is 1. On the other hand, if $r_j < z_o$ then $\tilde{r}_q$ must be greater than or equal to z_o.

For the upper bound on $\tilde{r}_q$, as in the first case, the only upper bound on $\tilde{r}_q$ is z_q.

$\tilde{r}_p$ Note that the support $\mathcal{S}(\mathbf{R}_{-p}) = \mathcal{S}_3$ is the same as described in the major topology update section, and the explanation is similar to what was already presented. Therefore

$$\mathcal{S}(\mathbf{R}_{-p}) = \mathcal{S}_3(\tilde{z}_p, \tilde{r}_i, z_o) = \begin{cases} z_o & z_o > \tilde{r}_i \\ 1, \ldots, \min(\tilde{z}_p, z_o) & \text{otherwise} \end{cases} .$$

After the updates to the r values, a new value for $\mathbf{s}_p$ is proposed from the distribution given in Section 2.4.3. The proposal probability $Q_5(\tilde{\mathbf{A}}|\mathbf{A})$ is found by multiplying the individual proposal probabilities together. The acceptance probability for the minor topology update is given by

$$\alpha_5(\mathbf{A}, \tilde{\mathbf{A}}) = \min\left\{\frac{f(\tilde{\mathbf{A}} \mid \mathbf{H})Q_5(\mathbf{A} \mid \tilde{\mathbf{A}})}{f(\mathbf{A} \mid \mathbf{H})Q_5(\tilde{\mathbf{A}} \mid \mathbf{A})}, 1\right\},$$

where

$$\begin{aligned} Q_5(\tilde{\mathbf{A}} \mid \mathbf{A}) &= \frac{1}{m} \times q(\tilde{r}_i|\mathcal{S}(\mathbf{R}_{-i}), \mathbf{T}, \tilde{\tau}_x, \rho) \times q(\tilde{r}_j|\mathcal{S}(\tilde{\mathbf{R}}_{-j}), \mathbf{T}, \tilde{\tau}_x, \rho) \\ &\quad \times q(\tilde{r}_q|\mathcal{S}(\tilde{\mathbf{R}}_{-q}), \mathbf{T}, \tilde{\tau}_x, \rho) \times q(\tilde{\mathbf{s}}_p|\mathbf{S}_{-p}, \tilde{\mathbf{R}}, \mathbf{T}, \tilde{\tau}_x). \end{aligned}$$

The first term is given in equation (2.25), the second through fourth are given by equation (2.13) and the last term in (2.18). This update changes τ_x, r_i, r_q, r_p, z_p and $\mathbf{s}_p$, so terms in $\frac{f(\tilde{\mathbf{A}}|\mathbf{H})}{f(\mathbf{A}|\mathbf{H})}$ that do not involve these updated values in will cancel, leaving a simplified acceptance probability.

2.5.6 Sampling with markers to the left and right of the focal point

The outline of the sampler in Zöllner and Pritchard (2005) and our version have been written assuming that all markers were to the right of the focal point. This allowed a simpler description of the algorithm as detail for only one side was needed. However, the actual implementation of the algorithm requires sampling variables on both sides of the focal point. We therefore must introduce latent variables corresponding to the left and right hand side recombination processes. However, since it is assumed that recombination occurs independently on the left and right hand side of the focal point x given its ancestry, the updates for each side can be proposed independently. Although the sequence will now correspond to loci to the left and right of the focal point, the variables already defined can account for this by assuming that the first marker is the left-most marker, the L^{th} marker is the right-most marker and the focal point x is between the k^{th} and $(k+1)^{th}$ marker.

To indicate left or right hand variables, the superscript l and r is used. For example, r_i^l and r_i^r are the index of the closest marker to the left and right of x respectively that was not inherited from K_i's parent (see Figure 2.11 for an example) and let $\mathbf{R}^l = (r_1^l, r_2^l, \ldots r_{4n-2}^l)$ and $\mathbf{R}^r = (r_1^r, r_2^r, \ldots r_{4n-2}^r)$. Similar definitions are made for the z variables at each node.

The target distribution $f(\mathbf{A}|\mathbf{H})$, where we now have $\mathbf{A} = (\mathbf{\Omega}, \tau_x, \theta, \rho, \mathbf{R}^r, \mathbf{R}^l, \{\mathbf{S} - \mathbf{H}\})$ can be written as

$$f(\mathbf{A}|\mathbf{H}) = \Pr(\mathbf{S}|\mathbf{R}, \mathbf{\Omega}, \tau_x, \theta) \Pr(\mathbf{R}^r|\mathbf{\Omega}, \tau_x, \rho) \Pr(\mathbf{R}^l|\mathbf{\Omega}, \tau_x, \rho) f(\mathbf{\Omega}, \tau_x, \theta, \rho).$$

This follows from the assumption of conditional independence of the left and right hand side mutation and recombination processes.

Some alterations are also required for the five update schemes if they include updates to components of $\mathbf{R}$. Specifically

- For computing the distributions for the recombination variables, the distance between markers is required. All necessary distances are measured relative to the focal point. Therefore, the vector of distances is now given by

$$d^* = (x - d_1, x - d_2, \ldots, x - d_j, d_{j+1} - x, \ldots, d_L - x),$$

  where the focal point is between the k^{th} and $(k+1)^{th}$ marker.

- When an update is made to r_i, this actually involves proposing both r_i^l and r_i^r, independently of each other, from their corresponding proposal distributions. These proposal distributions will depend on the corresponding left or right-hand versions of the z and r values at surrounding nodes, and of the distances between markers given above. In addition, if a condition is based on z and r, for example the condition $z_o \leq \max(r_q, z_j)$ from the major topology change, it must be true for both the left and right hand versions of the corresponding variables.

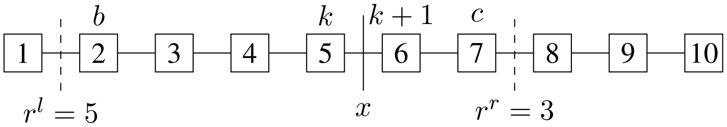


FIGURE 2.11: Illustration of the notation when the focal point is between SNP markers. The SNP markers, labelled 1 to 10, are denoted by boxes. The focal point, marked by x, is between the fifth and sixth markers. The location of the first recombination to the left of the focal point is marked by a dashed line between markers 1 and 2. Since there are four markers between the focal point and this recombination breakpoint, $r^l = 5$ and markers above $b = 2$ are co-inherited with the focal point. Similarly, the location of the first recombination to the right of the focal point is marked by the dashed line between markers 7 and 8. This gives $r^r = 3$ and $c = 7$.

- The sequence stored at a node depends on the r and z values for that node. Letting the r^l value correspond to a recombination event just before the $b \geq 1$ marker and the r^r value correspond to a recombination event just after the $c \leq L$ marker (see Figure 2.11 for an example), the probability from the mutation model for the portion of this sequence is given by $\prod_{j=b}^{j=c} \Pr(s_{i,j} \mid s_{a,j}, b_i, \tilde{\theta})$.

- For the major topology change, the new node is selected based on its sequence similarity to the sequence at the node that is moving. The similarity measure must be computed for the full sequence. Letting a correspond to the marker location for $\min(z_i^l, z_c^l)$ and b correspond to the marker location for $\min(z_i^r, z_c^r)$, the score is given by

$$p_{ic} = \frac{\sum_{j=a+1}^{j=b-1} 1[s_{i,j} = s_{c,j}] + \epsilon}{\min(z_i^l - 1, z_c^l - 1) + \min(z_i^r - 1, z_c^r - 1) + \epsilon},$$

2.6 Using the haplotype-based sampler

This section provides practical information for using the haplotype-based sampler. This includes how initial values are set and how they can be provided to the program. Details about how the sampler was programmed are also given.

2.6.1 Initial Values

In the MCMC approach, a new value $\tilde{\mathbf{A}}$ is sampled conditional on the current value $\mathbf{A}$. To begin the chain, an initial state, $\mathbf{A}_0$, must be specified. Recall that $\mathbf{A} = (\boldsymbol{\Omega}, \tau_x, \theta, \rho, \mathbf{R}, \mathbf{S} - \mathbf{H})$, where $\boldsymbol{\Omega}$ is the vector of intercoalescence times, τ_x is the topology, θ is the mutation rate, ρ is the recombination rate, $\mathbf{R}$ stores the recombination related variables and $\mathbf{S}$ the sequences at all nodes. The haplotype sequence at the tip nodes is our observed data $\mathbf{H}$. We now describe how each of these is initialized and we give the parameters of the prior distributions for ρ and θ.

ρ The prior for ρ is a gamma distribution. Based on the discussion in Section 2.3.1, our default parameter values for the gamma prior have shape set to 1 and scale set to 0.1, so that the prior is actually exponential with rate 10. However non-default values can and should be passed to the program since there is evidence of variation in recombination rates across the genome and it might be desirable to incorporate this information in the sampler. The default initial value for ρ is the mean of its prior, however a different initial value can be provided as well.

θ The current prior for θ is a uniform distribution with initial and range values that can be specified by the user. The default initial value for theta is $4N\mu D$ with $N = 10,000$ (Takahata, 1993), $\mu = 10^{-8}$ (see discussion in Section 2.3.1 and Awadalla et al. (2010)) and D the number of base pairs in the region. The default range for the uniform is two orders of magnitude above and below the initial value. As with the prior distribution for ρ, for some datasets the default values may not be adequate and the user should provide alternate values.

τ_x The initial topology is determined using UPGMA (Unweighted Pair Group Method with Arithmetic mean), which is a phylogeny reconstruction method that uses average linkage hierarchical clustering (see for example the books by Weir (1996) or Felsenstein (2004) for information on phylogenetic methods). A distance matrix is first created, with distance being defined as the number of different alleles between two sequences, and the pair with smallest distance is merged into the first cluster. A new distance matrix is determined, with the distance between two clusters now being the average of all the individual pairs between the clusters, and the closest pairs in the new matrix are merged. This process is continued until all pairs of clusters are merged. If there is a tie in the distance matrix, the pair that is merged is randomly chosen from the tied pairs. We have also added the option of a randomly generated initial topology as this may be helpful for determining whether the results are similar when the sampler is started with different initial values.

Ω After an initial topology is set, the intercoalescence times are generated using the neutral coalescent model (Kingman, 1982; Hudson, 1983). That is, the intercoalescence time when there are l lineages is simulated from an exponential distribution with rate $\binom{l}{2}$. Background information on the coalescent is provided in Section 1.1.1.

R The initial state for all r values on the tree is simulated from the model described in Section 2.3.2 (see equations (2.5) and (2.7)). This is necessary to ensure that the initial values of $\mathbf{R}$ are likely given the prior for ρ.

S The initial sequence for an internal node is set to the most frequent sequence among the descendants of that node.

Finally it is also possible to provide initial values for all the latent data at the nodes and for the topology. This can be useful for re-starting a run where a previous one finished.

2.6.2 Software implementation

The haplotype-based sampler was coded in C++. It uses the Gnu Scientific Library (GSL; `http://www.gnu.org/software/gsl/`) for random number generation and for sampling from standard probability distributions. It has successfully been compiled on Unix/Linux based operating systems and it runs at the command line. Input options are provided to the program in a separate file, which is also specified on the command line. Input options include the file names for the relevant datasets, a run name, chain length, burn-in, thinning, as well as prior parameters and initial values for the latent variables. Other than the file names which must be provided, most options include sensible defaults.

The default weights for sampling the five different update schemes are (0.1,0.1,0.5,0.15,0.15) for Q_1 to Q_5 respectively. By default, we do not perform certain updates in a sequential order as we did not observe improved mixing when the order was fixed. However, we do provide the option of completing certain updates sequentially. This is done in a separate input file, where each line of the file corresponds to a weight and either a single number or a set of numbers indicating the order to perform a subset of the five updates. For example, if a step involved completing the minor topology update followed by the local updates 10% of the time, a line of the file should read `'0.1 5 3'`

Multiple files are output by the program, including the latent data at the first and last iteration; however, the two that are useful for computing summary statistics are the file with the scalar-valued output (the update performed, an acceptance indicator and the ρ and θ values) and the file with the sampled trees. The trees are written in Newick format with branch lengths included in the output (for a description of the format see, for example, `http://evolution.genetics.washington.edu/phylip/newicktree.html`). Both files can be read in to `R` to compute summary statistics and plot output. The `R` phylogenetic package, `ape` (Paradis et al., 2004) is particularly useful for processing the trees. There are plans to import the C++ code into `R` to allow more efficient processing of the sampled trees.

2.7 Example - Sampling trees in the 5q31 region on a publicly-available dataset

To illustrate the use of the haplotype sampler, we applied it to a publicly-available dataset consisting of genotypes at 103 SNP markers on 258 trios consisting of a father, mother and a child affected with Crohn's disease (Rioux et al., 2001; Daly et al., 2001). The dataset is available either in the `R` `gap` package (`http://cran.r-project.org/web/packages/gap/index.html`) or at the author's website `http://www.broadinstitute.org/archive/humgen/IBD5/haplodata.html`).

Since the genotype data is derived from trio data, we define case haplotypes to be the set of transmitted haplotypes and control haplotypes to be the set of untransmitted haplotypes.

2.7.1 Background on the dataset and the 5q31 region

The 5q31 region was selected for association studies due to a peak found in a linkage study (Rioux et al., 2000). Rioux et al. (2001) genotyped SNPs in this region initially chosen from known genes and later from SNP discovery across a 500kb region on eight individuals. Common SNPs having a minor allele frequency greater than 5% were selected for further genotyping. Significant associations with p-values less than 0.0002 were found at 11 loci spanning 200kb of the 500kb region. They considered the 11 SNPs as a risk haplotype, called the IBD5 risk haplotype, and found that it could explain the linkage results. However, the causative mutation(s) were not identified due to the risk haplotype covering a number of different genes (labelled on the top panel of Figure 2.16).

Because of the complex linkage disequilibrium (LD) in this region the 11 SNPs give essentially equivalent information. In fact, a separate paper describing the haplotype structure and postulating the presence of haplotype blocks (middle panel of Figure 2.16 gives the block boundaries) in the genome was published based on this data (Daly et al., 2001).

Since the original publication, many other candidate gene association studies and genomewide association studies (GWAS) have investigated the region for association with Crohn's/IBD phenotypes in multiple populations and with different study designs. The IBD5 risk haplotype, or SNPs used as proxies for the haplotype, has replicated in multiple studies (see Waller et al. (2006) for example where five SNPs in the risk haplotype have about the same odds ratio, 95% CI, and allele frequencies). Peltekova et al. (2004) found that two SNPs in the OCTN1 and OCTN2 genes were associated with the IBD outcome independent of the risk haplotype, however subsequent studies did not replicate this finding and the location of the disease-predisposing mutation(s) remains uncertain. At least 10 subsequent GWAS have also replicated the association to the IBD5 region (see Cooney and Jewell (2009) for a review); however, the GWAS do not provide further localization of the disease-predisposing variant(s).

To conclude, there are probably one or more disease-predisposing mutations in this region. In general, many association results fail to replicate; however, for this region the results have been replicated in multiple studies having different study designs and samples from different populations. The complicated LD structure has so far made it difficult to attribute the significant results to specific mutations and/or genes.

2.7.2 Methods

There is more haplotype certainty with family data, however the haplotypes will still be unknown with trio data. To run the haplotype-based sampler, Beagle (Browning and Browning, 2009) was used to impute haplotypes based on the trio data. It returns an estimate for the haplotypes that each parent passed to their affected child (transmitted) and for the haplotypes that were not passed to their child (untransmitted). The transmitted haplotypes are considered the case haplotypes and the untransmitted the control haplotypes.

Recall that genealogies are sampled for a particular focal point. We chose 100 focal points spaced evenly throughout the 500kb region. For each focal point, a subset of the 103 SNPs was chosen for the analysis. All SNPs within a window size of 100kb around the focal point were

TABLE 2.4: Number of SNPs for 100 focal points

X	Number of focal points having X SNPs in their window
20	59
21-25	9
26-30	17
31-35	14
>35	1

TABLE 2.5: Initial conditions for running the sampler on the 5q31 datasets

Run option or Initial condition	Value
Chain length	8 million
Thinning	10,000
Burn-in	4 million
θ	Initial=0.1; Prior U(0.0001,10)
ρ	Initial=0.0004; Prior Gamma(Shape=1,Scale=0.1)
Update type weights	(0.1,0.1,0.5,0.15,0.15)

included in the dataset for that focal point. If fewer than 20 SNPs were available in the window then the dataset was made to include the closest 20 SNPs so that each dataset had a minimum of 20 SNPs. If there was less than 100kb between the focal point and the lower or upper edge of the region, the window size remained the same but the focal point was not centered in each subset. Due to the uneven spacing of SNPs in the region, the majority of datasets had 20 SNPs (Table 2.4).

A separate set of genealogical trees was sampled for each focal point, however other than the focal point and dataset settings, the run options and initial conditions were the same for each dataset. These are summarized in Table 2.5. Each focal point was run on a separate processor in a cluster computing environment. The median time across the 100 focal points to complete one million iterations was 49 hours. The maximum time for 100 focal points was 64 hours. Since the file sizes of sampled trees can become quite large, only every $10,000^{th}$ sample was saved.

After two and four million iterations, traceplots of θ and ρ were examined. The sampled θ values were all well below the maximum of the prior. For many of the 100 focal points, the θ and ρ values had ceased to either increase or decrease between two and four million iterations; however, for a few there was still sign of a trend even near the end of the four million iterations. Therefore, the run was restarted and a further four million iterations were completed, for a maximum run length of eight million MCMC iterations. Given a maximum time of 64 hours for one million iterations, the total time to complete all eight million iterations on all focal points was on the order of three weeks.

2.7.3 Results

Examining traceplots for sampler convergence

An *ad hoc* approach to examining whether the chain has converged or is mixing properly is to examine traceplots, or plots of sampled variables or summary statistics over time. This does not guarantee that the chain has converged or is mixing, but obvious violations can often be seen. Typically the traceplots of each variable of interest would be evaluated for increasing/decreasing trend or the sampled values not changing quickly. Both would indicate the the sampler should be run longer.

Our application involves running separate chains for each focal point. As detailed tuning and diagnostics for each of the 100 chains is not feasible, we chose to run all focal points under the same MCMC settings: eight million iterations and, for storage reasons, thinning and storing every $10,000^{th}$ sample. Traceplots of summary statistics that capture information about the latent variables as well as the tree shape were computed for each focal point and were examined for concerning trends. These plots consisted of:

(A) Acceptance proportion for each of the proposals.

This is summarized with a bar chart. For the local updates, the plotted bar shows the average over all MCMC samples of the proportion of nodes that have the updates accepted.

(B) Sampled θ values

(C) Sampled ρ values

(D) t_{Total}, which is the sum of the branch lengths of the tree. This summary captures information about whether node times are changing.

(E) Symmetric distance between two trees (Robinson and Foulds, 1981); computed with the `dist.topo` function in the `ape` package.

A bipartition of a binary tree cuts the tree at an internal branch into two sets. The tree can be described by its set of bipartitions generated for all internal nodes. The symmetric distance between two trees counts the number of bipartitions that are contained in the bipartition set of only one of the two trees. It can also be interpreted as twice the minimum number of coalescence event swaps to convert one tree to another. For a rooted binary tree this distance is always even and the maximum is $2(n-2)$ since there are $n-2$ possible bipartitions for a tree with n tips. We compare the current tree to the previous tree that was saved so it gives a measure like an autocorrelation between trees after thinning. We scale the symmetric distance by its maximum.

(F) Rand Index, which assesses the similarity between clusterings induced by the tree.

Rather than comparing sets of clusterings based on the bipartition, this measure compares cluster membership of tip labels. Clusters are defined by cutting the tree at a point back in time. The time is chosen based on the desired number of clusters, k; there will be k clusters if the tree is cut when there are only k lineages left. A smaller k value would compare clusters based on older branches and a larger k would compare clusters based on younger branches. We would expect cluster membership for very old branches to be fairly stable even if the tree is mixing, therefore the value of k should favour younger branches. However, since many sequences will be identical, if the k is set too small, the differences between cluster membership will simply be due to re-orderings of coalescence events between identical sequences. Therefore, k should be neither too big nor too small and we chose $k = 20$, which corresponds to approximately the number of sequences at the tips divided by 25 or a cluster size of 25 if all were equally sized.

The two clusterings induced by the trees are compared by looking at all pairs of labels and determining whether they are either both in the same cluster or in different clusters in the two trees (concordant) or whether they are grouped together in only one of the two trees (discordant). Various indices were proposed to measure similarity based on linear combinations of the number of concordant and discordant pairs (see Albatineh et al. (2006) for a summary). Since our purpose is to ensure that the topology is changing, we chose the Rand index (Hubert and Arabie, 1985), which is available in the `fpc` package for `R` and looks at the proportion of concordant pairs relative to the total number of pairs.

Although all 100 sets of traceplots cannot be included here, example traceplots across the eight million samples are provided in Figures 2.12 and 2.13. As mentioned in the previous section, we can see from the traceplots of θ and ρ that even up to close to four million iterations there is sign of increasing/decreasing trend. This prompted the longer runs for each focal point and therefore, the recombination, mutation and association results described in the next two sections are based only on the final four million iterations. No obvious increasing or decreasing trend is observable in the traceplots of the tree summary statistics. The average of the scaled symmetric distance is around 0.7, indicating that, from one sampled tree to the next, approximately 70% of the bipartitions are different. The average Rand index is around 0.5, indicating that, from one sampled tree to the next, approximately 50% of all pairs occur in the same cluster in both trees. Both indicate that the topology is changing.

Mutation and Recombination rates

The mutation and recombination rates, θ and ρ, are also estimated at each focal point by the ancestry sampling procedure. The estimate for each focal point is based on window sizes of varying numbers of markers (see Table 2.4) and of variably-spaced markers (spacing ranges from 38 to 133,517 kb). These will not generally be of interest for the applications we anticipate the sampler being used for; however, we would hope that the sampled values are biologically plausible. Figure 2.14 gives the average of the values sampled for θ and ρ across the MCMC samples at each focal point. The mutation rate and the recombination rate are both estimated to fluctuate in this region, in particular the mutation rate seems to be higher at focal points between 131.4 to 131.7 Mbp. However, the estimates for these focal points may be less reliable since the marker spacing is sparse and since there are fewer markers to the left of the focal point near the edge of the region. This region also contains a number of genes and therefore the mutation rate may vary due to selection against mutations in genes. In general, the mutation rate is also known to vary throughout the genome (see, for example, Duret (2009) for a review). Even adjacent nucleotides can have different mutation rates; for example, a C nucleotide followed by a G nucleotide has a much higher rate of mutation than a C followed by any other nucleotide.

Recombination rates across the genome have been estimated so we can compare our estimates of the recombination rates to those available in public databases. Recombination rate estimates computed by Peter Donnelly, Gil McVean and Simon Myers using the coalescent approach in McVean et al. (2004) are available with the Phase I HapMap data (release 16a) (International HapMap Consortium, 2005). This data was downloaded as part of the bulk data download of chromosome five from `http://hapmap.ncbi.nlm.nih.gov/`. The recombination rate was provided in

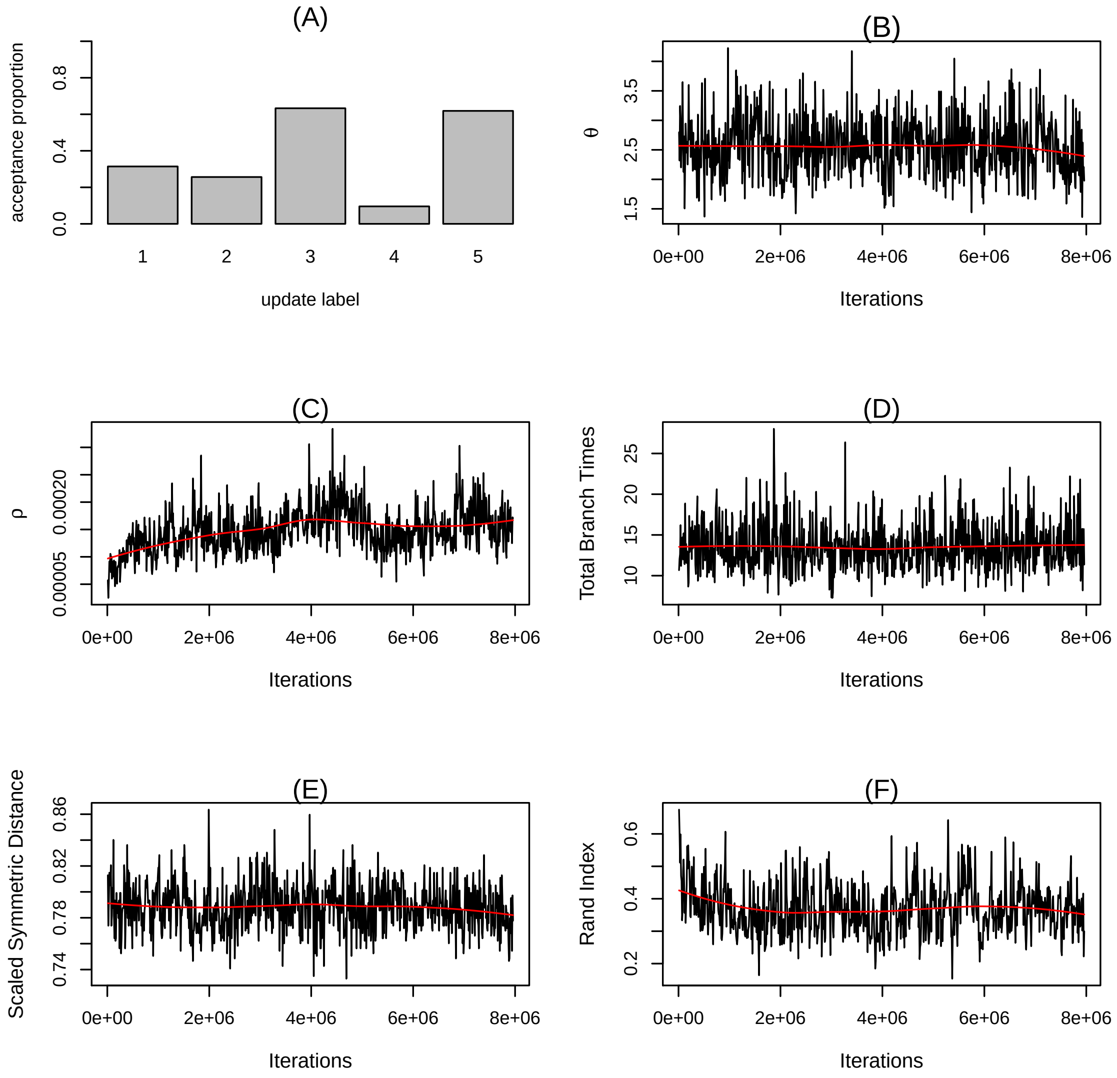


FIGURE 2.12: Acceptance proportions and traceplots of summary statistics from sampled genealogies for focal point 24 (position 131,819,075 on chromosome 5). (A) acceptance proportions for the five updates: 1- mutation rate; 2- recombination rate; 3- local updates; 4-major topology change; 5-minor topology change (B) traceplot of θ (C) traceplot of ρ (D) traceplot of Total time of all branches (E) traceplot of Symmetric distance scaled by the maximum value (F) traceplot of Rand Index

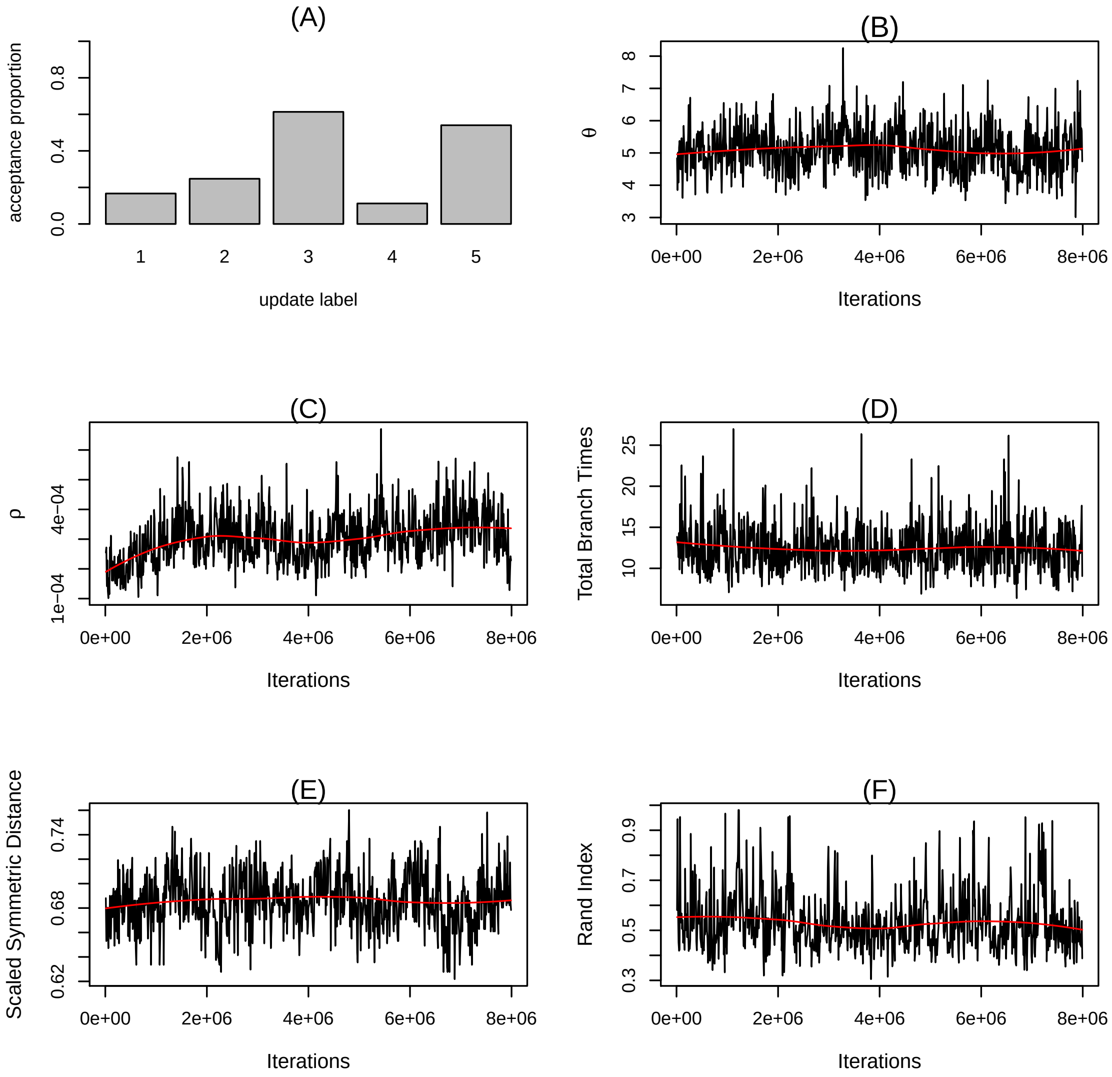


FIGURE 2.13: Acceptance proportions and traceplots of summary statistics from sampled genealogies for focal point 59 (position 131,605,400 on chromosome 5). (A) acceptance proportions for the five updates: 1- mutation rate; 2- recombination rate; 3- local updates; 4-major topology change; 5-minor topology change (B) traceplot of θ (C) traceplot of ρ (D) traceplot of Total time of all branches (E) traceplot of Symmetric distance scaled by the maximum value (F) traceplot of Rand Index

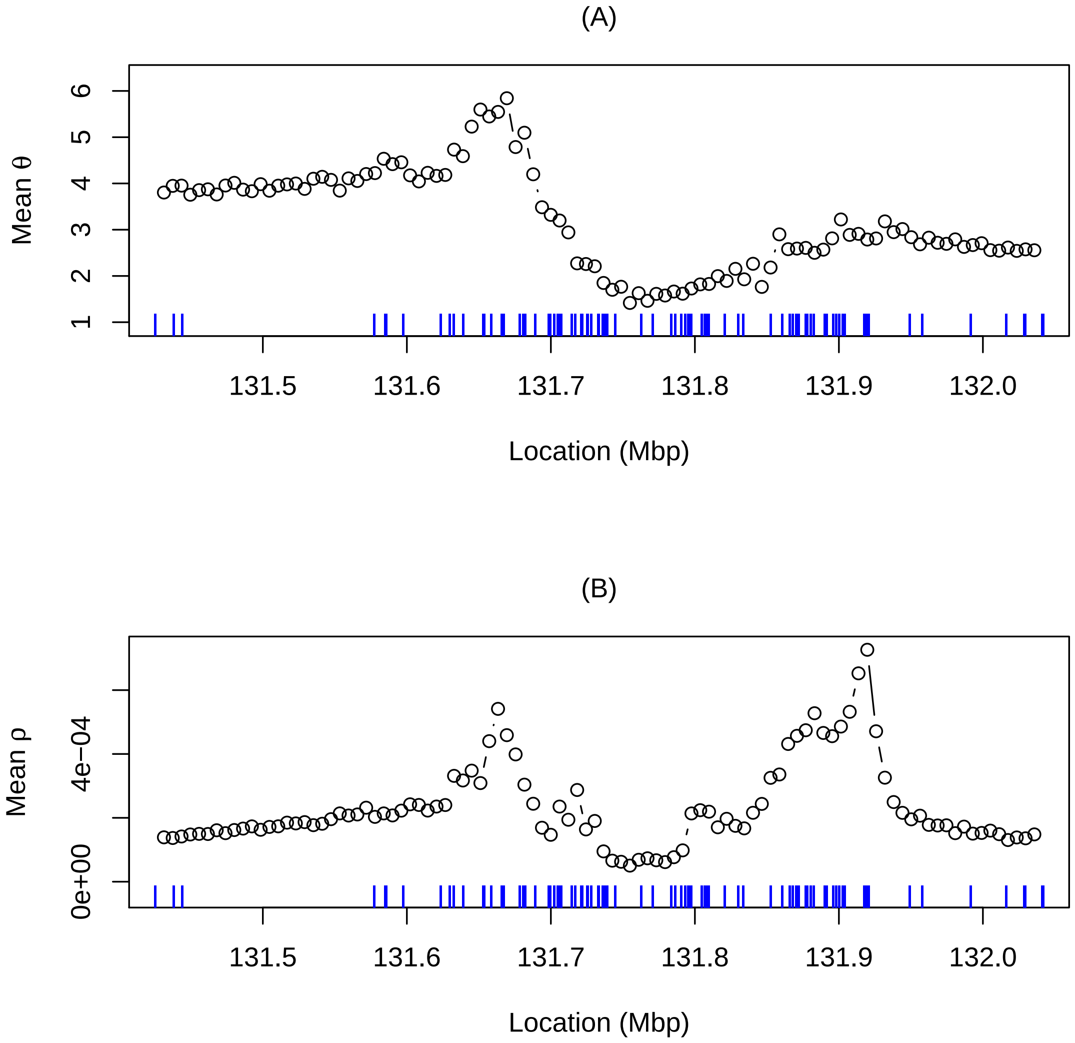


FIGURE 2.14: (A) average θ and (B) average ρ values across MCMC samples for each focal point. The dashed lines below the curve give the SNP locations.

cM/Mb and so it was converted to the per pair of base pairs per unit of coalescent time rate by noting that for the per generation rate $1cM/Mb \approx 10^{-8}$/bp. Although both sets of data cover the same region, the SNP positions provided with the Crohn's dataset were relative to the SNP discovery region and not the genomic positions. Therefore, the two sets of results could not immediately be compared as a mapping between the two sets of positions first needed to be found. The RS numbers for the SNPs were not provided with the dataset, but a literature search provided RS numbers for some of the SNPs. The genomic positions for two of the SNPs with RS numbers were determined, relative to the NCBI Build 34 human reference sequence, using the UCSC Genome Browser (`http://genome.ucsc.edu/`). Although this reference sequence dates to 2003, it was chosen as the markers from HapMap Phase 1 (release 16a) are relative to this build. However, the distance between the two SNPs was different between the provided positions and the genomic positions from UCSC. The order of the SNPs was also reversed in the two sets of positions. Therefore the conversion between the two sets of positions may not be completely accurate.

Figure 2.15 shows the average ρ over MCMC samples by a solid curve, as in Figure 2.14. The dashed curve gives the recombination rates estimated from the HapMap data. The solid curve in Figure 2.15 should be viewed as a smoothed version of the HapMap estimates (the dashed curve) since our estimates are based on fewer, less equally spaced SNPs than the HapMap estimates. It is therefore not surprising that the peaks are lower. The shifts in peaks are possibly due to the difficulties in aligning the HapMap positions with the Crohn's dataset positions. Nevertheless, subject to these shifts, it is satisfying to know that the variation in recombination rate estimated by our algorithm is not inconsistent with the variation estimated by others using different data and different algorithms. We do not pick up the peak near 131.5. However, in our data there were no markers genotyped in this region, and so there may not be enough genotype information to detect the increase in recombination rate indicated by the HapMap estimates.

Association of transmitted/untransmitted chromosomes

Close to a disease-predisposing mutation the cases will tend to be more closely related to each other than the controls and will therefore tend to share marker alleles in common. Typical association statistics compare the distribution of allele or haplotype frequencies of markers in cases versus controls using contingency tables or logistic regression. If a marker is associated with disease status then it may be the causative locus, or more likely it is linked to the causal locus.

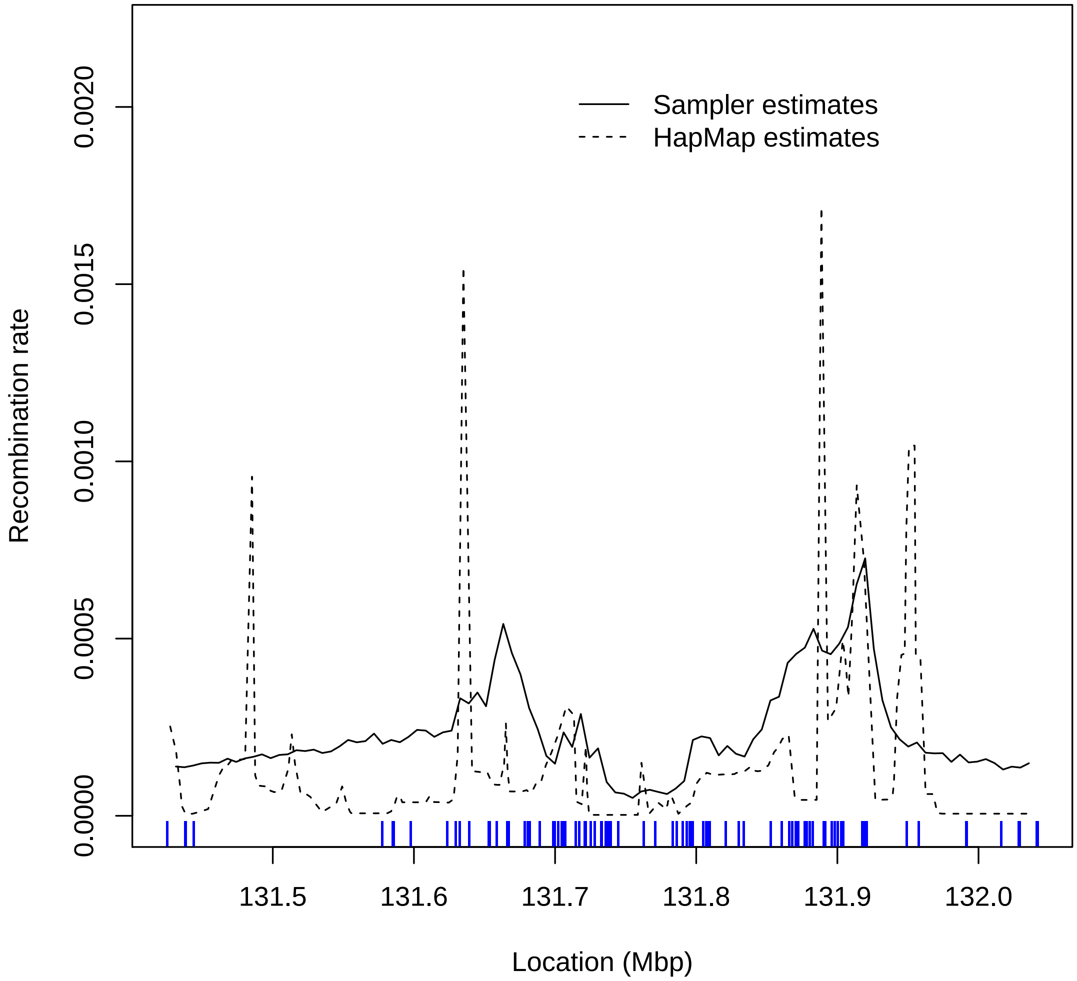


FIGURE 2.15: Plot of ρ values estimated by our sampler and by HapMap. Solid curve: average ρ values across MCMC samples for each focal point; Dashed curve: rescaled recombination rates estimated from Phase I HapMap data (release 16a) (International HapMap Consortium, 2005)

Figure 2.16 shows the single-locus association results for these data. At each locus, Fisher's exact test was used to determine whether there was an association between allelic state and case/control status, where "cases" were the transmitted chromosomes and "controls" the untransmitted chromosomes. This figure also shows the haplotype block boundaries estimated in Daly et al. (2001) and the locations of genes in the region. As expected from the published results on this region, although there are a few peaks, the signal in this region is not distinct and spans a large region. Many SNPs pass the 0.05 (uncorrected) threshold of significance; however there is significant LD in the region, as indicated by the long blocks with low haplotype diversity, so these tests are not independent.

With respect to the genealogy of the disease-mutation, the cases are more closely related if they tend to preferentially coalesce with each other rather than with controls. Ideally this would be reflected in a clustering of cases on the tree, or with shorter branch lengths among cases versus controls. In the future, we would like to assess which tree-based statistics tend to discriminate cases and controls better, however for the moment we will simply illustrate the use of the haplotype-based sampler by showing results with one such statistic.

Let T_i be the maximum absolute correlation between disease status and cluster membership, where cluster membership is derived from all bipartitions of the tips induced by the i^{th} tree. Recall from the description of symmetric distance that each internal branch of the tree induces a bipartition of the tree. If a hypothetical mutation occurred on that branch then all tips descending from this branch will carry the mutation and all other tips will not. The correlation then assesses the non-independence between the cluster descending from this branch and disease status. If there are fewer than $m = 25$ tips descending from a particular branch, the correlation is not assessed to avoid computing association of the cluster when the cluster size is too small. Even if all sequences in a cluster of size 25 belonged to case individuals, this cluster would represent only approximately 10% of the total number of case sequences. The choice of 25 is arbitrary, but it avoids the need to compute the association on clusters that are too small to be interesting. This statistic is similar to the association statistic described in Minichiello and Durbin (2006), but they do not put a lower bound on the number of tips in a cluster.

Although a formal evaluation of the properties of this statistic, or other tree-based association statistics, is beyond the scope of this work, we can use fuzzy p-values (Thompson and Geyer, 2007) to evaluate whether this region is associated with case/control status when a genealogy-based statistic is used. The fuzzy p-value concept is applicable when the statistic of interest is a function of a latent variable. This approach compares the distributions of two sets of test statistics based on

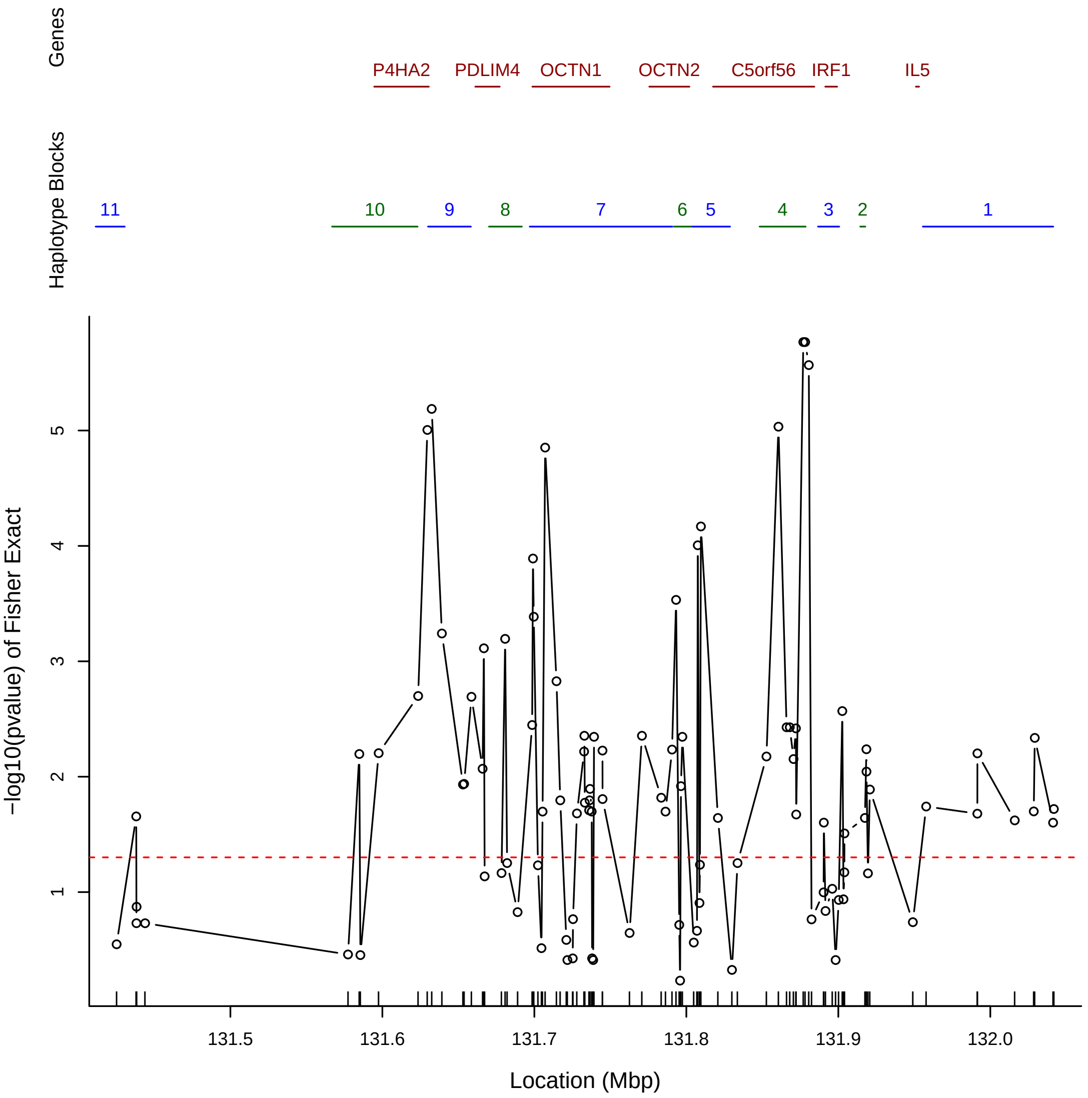


FIGURE 2.16: Single-locus association results in the 5q31 region. Plot shows -log10(pvalue) from Fisher exact test of association between allelic state and case/control status. Vertical dashes show locations of SNPs. Top panels give gene locations and haplotype blocks defined in Daly et al. (2001). Horizontal dashed line indicates a pvalue of 0.05 (uncorrected for multiple testing).

latent variables and sampled under the null hypothesis of no association between the trait and the genealogy, the first set conditional on the observed marker data and the second set unconditional of the observed marker data. The test statistics from the unconditional sample, T_i^u, form a reference distribution for computing the p-values of the conditional set of statistics, T_i^c. This leads to a distribution of latent p-values (the fuzzy p-value) that expresses both the strength of evidence against the null hypothesis and the uncertainty associated with the latent variables.

The reference distribution was computed by sampling 35,000 genealogies from the neutral coalescent model using the `ape` package and calculating T_i^u for the i^{th} genealogy. For each conditional genealogy j of a focal point, T_j^c was compared to the reference distribution and the latent p-value

$$\frac{\sum_{i=1}^{i=35,000} 1[T_i^u \geq T_j^c]}{35,000}$$

was computed. The $-\log_{10}$ of the median of the latent p-value distribution is given for each focal point in Figure 2.17. The signal from the median of the latent p-values can be compared to the single-locus results in Figure 2.16. The cluster-based statistic yields a smoother association curve than the single-locus results, with more distinct peaks.

The peak correlation between disease status and clade status, occurring near 131.9 Mbp in block four is close to a peak of the single-locus results (as indicated by the triangles). A second area of high signal from the cluster-based results is between 131.6 and 131.7 Mbp, in block eight of Figure 2.17. The corresponding peaks in the single-locus results are close to the second peak with the cluster-based statistic, however the second peak of the cluster-based statistic occurs between two peaks of the corresponding single-locus results. It is unclear which of the two sets of results is closer to a true-disease predisposing mutation, but it is possible that the two peaks in the single-locus results are due to LD at these SNPs with a single disease-predisposing mutation that is located closer to the peak of the cluster-based statistic.

Finally, Figure 2.18 summarizes the distribution of the latent p-values for each focal point and can be used to evaluate the uncertainty associated with the latent genealogy. The degree of uncertainty is not the same at each focal point, as indicated by the width of the inter-quartile range (IQR). In general, the width is larger when there are fewer SNPs as can be seen, for example, when comparing the widths near 131.9 Mbp to 132.0 Mbp. The width of the interval is smaller at the peak of the cluster-based results but this may be due to an inadequate number of null distribution samples to estimate the low p-values in this region. In particular, any latent p-values of zero have been set to

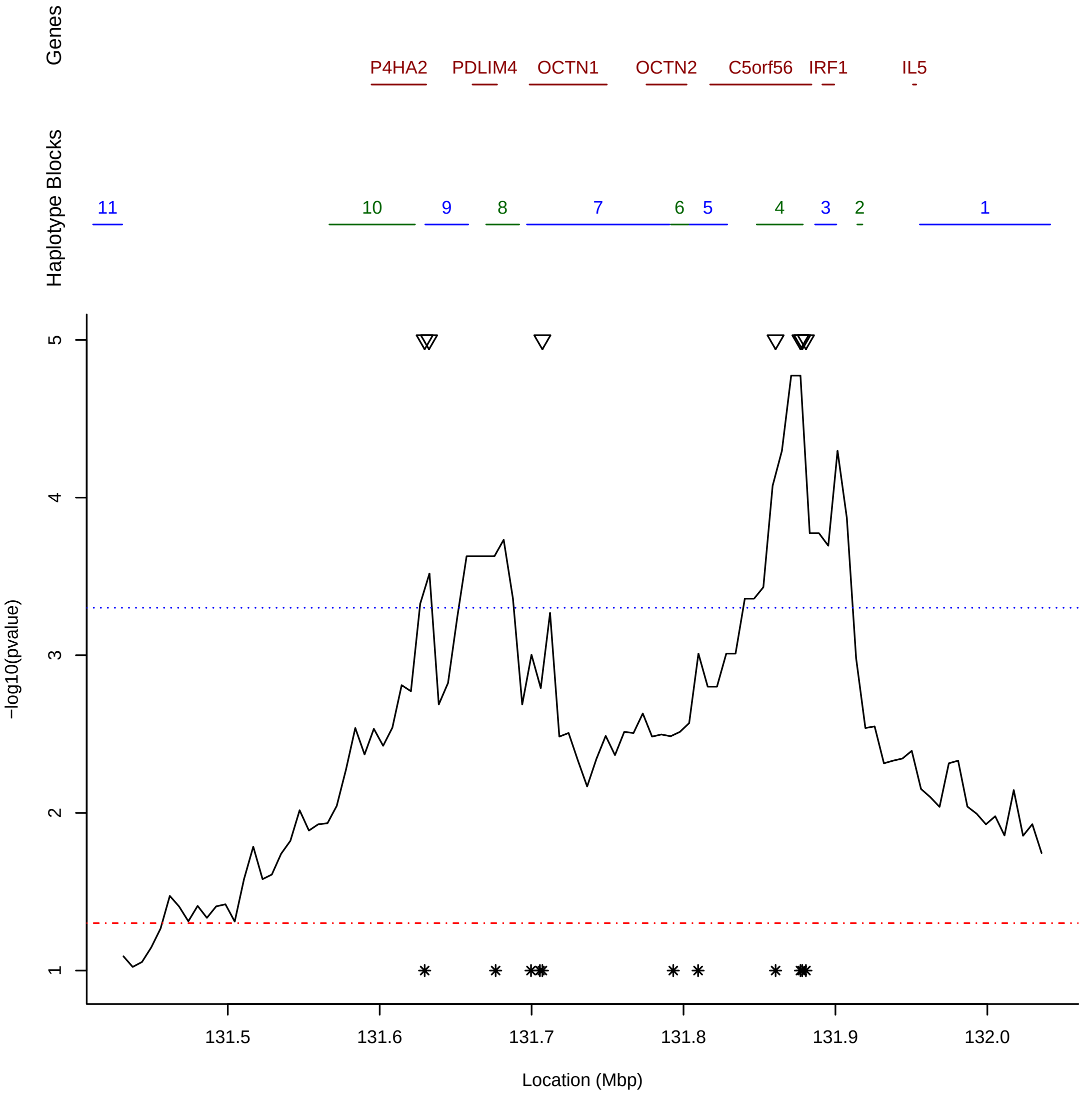


FIGURE 2.17: Plot of $-\log_{10}$ of the median of the fuzzy p-value by focal point. The red dashed horizontal line shows a p-value cutoff of 0.05 and the blue dotted horizontal line shows a p-value cutoff of 0.05 Bonferroni-corrected for 100 focal points. SNPs in the risk haplotype defined in Rioux et al. (2001) are marked with an asterisk. Triangles indicate single-locus results of -log10(pvalue)>4.5 in Figure 2.16. Top panels give gene locations and haplotype block boundaries as defined in Daly et al. (2001).

$\frac{1}{35000}$ to enable plotting on the log scale.

2.8 Discussion

In this section, a summary of our implementation of the haplotype-based sampler outlined in Zöllner and Pritchard (2005) was provided. Although few details were given in the original paper, we think there are some important differences in our implementation. In particular, all our proposal distributions for the recombination variable r lead to proposed values that do not cause incompatibilities with other variables on the tree and hence have non-zero probabilities under the target distribution. In contrast the proposal distributions of Zöllner and Pritchard (2005) allow r values with zero probability under the target distribution. This is inefficient as it will lead to the proposal being rejected. A related change is that we also restructured the local updates so that sequence at an ancestral node doesn't become incompatible with the proposed values. It is not clear how this was handled in the original paper. In further efforts to improve efficiency we have also modified their proposal distribution for the minor topology rearrangement so that transitions from $\tilde{\mathbf{A}}$ back to $\mathbf{A}$ are always possible. We have eliminated the need for the time swap, which reorders the nodes associated with coalescence events. Finally, by default, we also do not perform sets of updates in a sequential order, for example a major topology rearrangement followed by the local updates; so far we have not found that performance of the sampler has been affected by this simplification.

We used the publicly-available 5q31 trio dataset to illustrate the use of the haplotype-based sampler. We first imputed haplotype phase using the family information and determined the transmitted and untransmitted haplotypes from parents to affected offspring. We then sampled genealogical trees corresponding to 100 focal points across the 500kb region. We used traceplots of sampled values and tree summaries to assess mixing and convergence. Finally, we also measured whether transmitted chromosomes were more closely related to each other than control haplotypes with a cluster-based statistic. The plot across focal points of the fuzzy p-value showed that the maximum peak locations were close to the single-locus association results previously published; however, the cluster-based results appear smoother and the peaks more distinct than the single-locus results. Unfortunately since the location of the disease-predisposing locus is not known for certain, it is unclear whether our results provide additional information for localization of the true disease-predisposing mutation(s).

There are some drawbacks to using this sampler. First, as mentioned in the introduction in

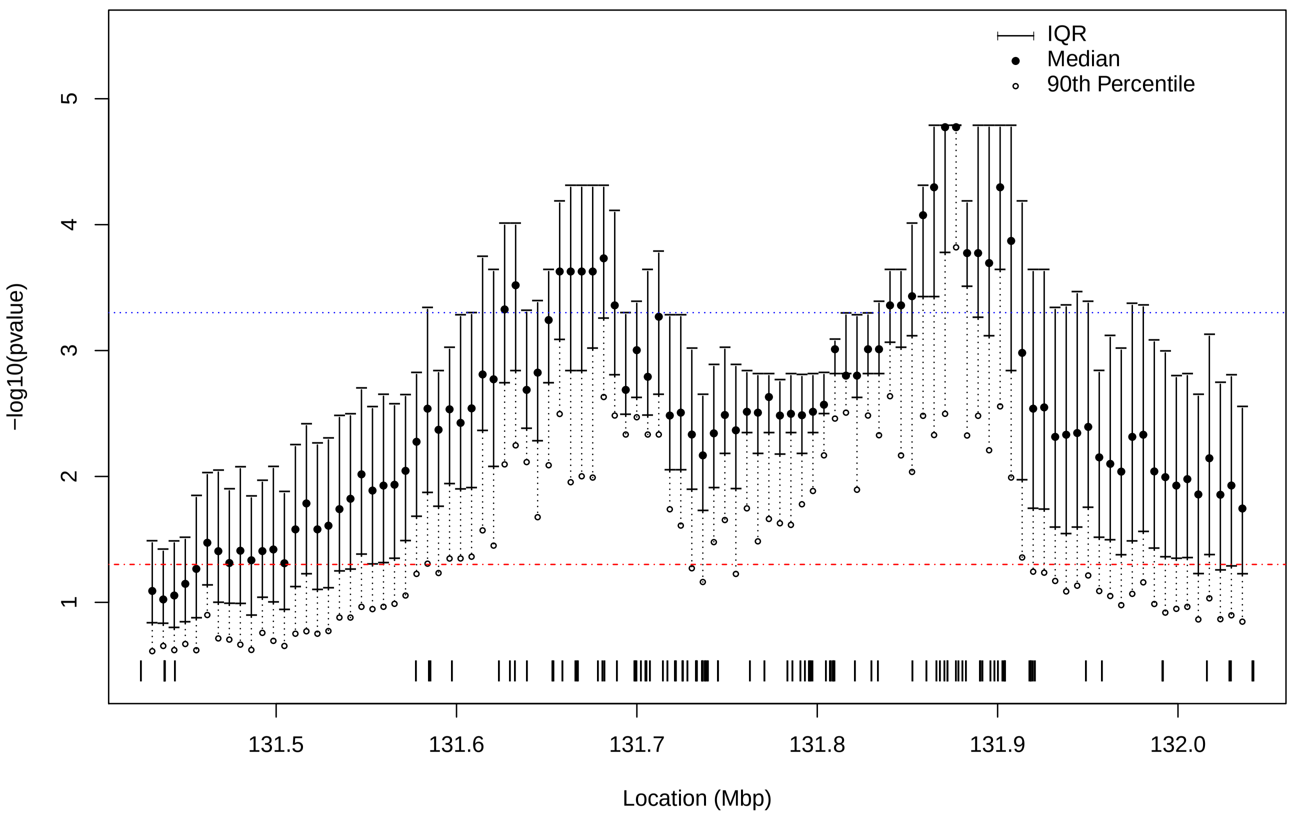


FIGURE 2.18: Plot summarizing the distribution of the latent p-values by focal point. The inter-quartile range (IQR) of the latent p-values at each focal point is indicated by the solid vertical line. The filled in circle is the median and the open circle is the 90th percentile of the distribution. The dashed vertical line therefore indicates the range from the 75th to 90th percentile. The red horizontal dashed line indicates a p-value cutoff of 0.05 and the blue dotted horizontal line shows a p-value cutoff of 0.05 Bonferroni-corrected for 100 focal points. SNP locations are marked by vertical dashes at the base of the plot.

Chapter 1, genotype data is typically observed and haplotypes are unknown. Therefore, the haplotypes are also latent data in this context. The suggestion in Zöllner and Pritchard (2005) was to first use a program to statistically impute phase. In our 5q31 example, we used Beagle (Browning and Browning, 2009) to impute phase. However, in the association study setting, single imputation of haplotypes has been shown to cause bias in resulting estimates (Mensah et al., 2007; Lin and Huang, 2007). Since we are sampling trees based on a single imputation of the haplotypes, the imputation clearly limits the space of trees that we sample from and similar bias in estimates may be observed. Multiple imputation would be better, however since the run times are very long and there are already multiple focal points, this would be infeasible for all but a small number of additional imputed datasets. Since we are already sampling over the latent sequence at internal nodes, it seems natural to also sample over the latent sequence at the tip nodes and condition on the observed data. This is described in the next section, which summarizes the extension of the sampler to genotype-based data.

Even assuming known haplotypes, the sampler is slow mixing and therefore long run lengths are required. For the 5q31 example, which is a relatively large dataset for sampling genealogies, we chose eight million MCMC samples since after four million there were some focal points where longer run lengths seemed appropriate. If this analysis was not an example, even longer run lengths would probably be recommended. In order to improve mixing, we tried to incorporate simulated tempering into the sampler. Our experiences with simulated tempering are described in Chapter 4.

Chapter 3

Sampling genealogies conditional on genotype data

Recall from Appendix A.1 that a haplotype is a set of alleles inherited together on the same chromosome from a parent. In Chapter 2, we described the algorithm to sample genealogies conditional on haplotype data from a sample of unrelated individuals. However, commonly used SNP genotyping technology measures the genotype at multiple loci rather than the haplotypes. If the genotypes at all loci are homozygous (that is, either 0/0 or 1/1), or if there is only one heterozygous 0/1 locus, the two haplotypes that were inherited can be inferred from the observed genotypes. In this case the *haplotype phase*, that is the pair of haplotypes, is known. Otherwise, the haplotype phase is missing. For example, if an individual's genotypes at three loci were (0/0, 0/1, 0/1) the two haplotype configurations that could have produced these genotypes are $\mathbf{s}_{i_1}/\mathbf{s}_{i_2} = 000/011$ and $\mathbf{s}_{i_1}/\mathbf{s}_{i_2} = 001/010$.

In a sample of unrelated individuals, haplotype phase will not be known for all individuals. Therefore, there has been extensive work on haplotype reconstruction and on estimation of haplotype frequencies based on observed genotype data (Clark, 1990; Hawley and Kidd, 1995; Excoffier and Slatkin, 1995; Long et al., 1995; Stephens et al., 2001; Niu et al., 2002; Scheet and Stephens, 2006). These methods have been based on parsimony, maximum likelihood (the EM algorithm) or Bayesian approaches requiring MCMC. However, an estimate of the haplotypes is typically not of direct interest. For example, haplotypes may be desired as covariates in a regression model to determine if a genomic region is associated with a trait of interest. Due to bias that may result from treating a haplotype reconstruction as if it were observed data, work has also been done on approaches that jointly estimate haplotype and regression parameters in order to account for the haplotype uncertainty (see Epstein and Kwee (2009) for a review).

Other than LAMARC (available at `http://evolution.genetics.washington.edu /lamarc/documentation/index.html`), the population genetic inference programs do not incorporate haplotype uncertainty. Instead, either experimentally determined haplotypes are used or the haplotypes are first statistically imputed then assumed to be observed in subsequent analyses. Kuhner and Felsenstein (2000) describe the extension to LAMARC that allows missing haplotype phase to be handled by the program. They found that by proposing new haplotypes 50% of the time or by using heating via Metropolis-coupled MCMC (Geyer, 1991), their sampler without recombination provided an estimate of the mutation rate parameter θ that was close to the true value. However, they did not see similar results when recombination was included and at the time were still working on improving mixing of the sampler with recombination by incorporating heating. In addition, they did not compare the sampled haplotypes to estimates from other programs. Instead, they rely on the sampler's ability to estimate the mutation and recombination rates to assess performance. In a later article describing LAMARC (Kuhner, 2006), these authors mention that their algorithm can be used with missing phase but suggest that the resulting estimates are more accurate if phase is known. It is unclear whether they were able to improve the performance sufficiently to handle recombination rate estimation.

Most of the cluster-based approaches summarized in Section 1.1.3 also assume known haplotypes when constructing a graph or tree to summarize the genetic variation. The approaches of Minichiello and Durbin (2006) and Seltman et al. (2003) do not assume that phase is known. However, as mentioned in Chapter 1, they only approximate the genealogy and therefore do not capture the uncertainty due to the evolutionary process.

Although Zöllner and Pritchard (2005) recognized that missing phase could be incorporated in their algorithm, they felt that the added computational burden of sampling over haplotype configurations did not make it worthwhile. They proposed using a two-stage approach that involved first imputing missing haplotypes with the PHASE algorithm (Stephens et al., 2001), and then treating the best reconstruction as known in the genealogy sampler.

There are some criticisms that can be made about the decision not to handle missing phase. First, the imputed haplotypes may not be accurate for all individuals and the resulting trees will be sampled conditional on inaccurate data. Andrés et al. (2007) examined the accuracy of phase imputation using experimentally determined haplotypes from real data. They found that the proportion of imputed haplotypes with at least one incorrectly phased locus was close to 1 for the methods that they compared, which included PHASE. The error rates were much smaller when defined as the proportions of loci with incorrect phase, but were still between 0.086 and 0.37.

Second, bias, inflated type I error rates and low power have been observed in association studies with imputed haplotypes. Mensah et al. (2007) observed bias of the estimated odds ratio of the true risk haplotype relative to all others and inadequate coverage of the associated confidence interval when single imputation was compared to a phase-known analysis. In Lin and Huang (2007), single imputation of haplotypes resulted in bias of the haplotypic odds ratio, an inflated type I error rate and lower power relative to a method that properly accounted for missing phase.

These studies did not assess the effects of haplotype imputation on tree reconstruction or on cluster-based association statistics. However, it is reasonable to expect that the problems of bias, inflated type I error rates and lower power could equally apply. For example, Seltman et al. (2001) examined the impact of basing inference on a tree or graph structure that was incorrect due to recombination events having occurred. They observed a reduction of power and interpretability of results, however there was no increase in type I error rates. Similar results might be expected if the modelling error was due to incorrect phase rather than ignoring recombination.

Finally, the true haplotypes can be thought of as a sample from an underlying coalescent process with mutation and recombination. The Gibbs sampler in PHASE assumes a model for the haplotypes that is based on an approximation to the coalescent process (Stephens et al., 2001). An alternative approach to haplotype estimation might simply introduce the genealogy as a latent variable rather than using the approximation. However if reconstruction of haplotype phase was all that was required, this approach would be too computationally intensive to pursue. However, since we are actually interested in the genealogy, the natural approach would be sampling from the the joint distribution of haplotypes and genealogy.

We extended our haplotype-based genealogy sampler to handle missing phase by including latent variables for the haplotypes of the tip sequences. The new model that includes missing haplotypes at the tips is described in the next section. In order to sample new haplotype configurations we introduce two new proposal distributions that shuffle haplotypes of the tip sequences. The allele-swap proposal distribution swaps the allele at a single-locus for a single individual. The dictionary rephaser samples a new configuration based on its similarity to the current tip sequences. We demonstrate the use of the sampler by applying it to simulated data and we assess the sensitivity of the sampler to the prior distributions for θ, ρ and the haplotype frequency model.

3.1 Target distribution with genotype data

Assume that we have a sample of n unrelated individuals with genotype data at L SNP loci. The observed data are denoted $\mathbf{G} = (\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_n)$, where $\mathbf{g}_i = (g_{i,1}, g_{i,2}, \ldots, g_{i,L})$ is the set of L genotypes for the i^{th} sample. The two alleles at the SNP locus are coded as either 0 or 1 and the genotype $g_{i,j} \in \{0,1,2\}$ counts the number of 1 alleles at the j^{th} locus. As in Chapter 2, let the haplotypes of individual i's two sequences be denoted by $\mathbf{s}_{i_1}$ and $\mathbf{s}_{i_2}$, where $\mathbf{s}_{i_k} = (s_{i_k,1}, s_{i_k,2}, \ldots s_{i_k,L})$ is the set of alleles at the L loci for haplotype k, $k \in \{1,2\}$ and $s_{i_k,j} \in \{0,1\}$.

Recall that $(\tau_x, \mathbf{\Omega})$ is the latent gene genealogy at a genomic location x for the sequences corresponding to the n sampled individuals. As with the haplotype-based sampler, there are two tip nodes for each individual in the sample corresponding to each of their two haplotypes. The $2n$ tip nodes are labelled $1_1, 1_2, \ldots n_1, n_2$ in order to keep track of both the individual and sequence that they correspond to. There are $2n-1$ internal nodes, which will be labelled 1 to $2n-1$. We are now interested in sampling genealogies conditional on the genotype data from $f(\tau_x, \mathbf{\Omega}|\mathbf{G}) \propto f(\tau_x, \mathbf{\Omega}, \mathbf{G})$.

The original purpose of our implementation of the haplotype-based sampler was ultimately to extend it to handle genotype data rather than haplotype data. Assume now that all components of $\mathbf{S} = (s_{1_1}, s_{1_2}, \ldots, s_{n_1}, s_{n_2}, s_1, \ldots, s_{2n-1})$ are latent. Letting $\boldsymbol{\gamma}$ be the parameter for the haplotype frequency model, equation (2.3) can now be written

$$f(\mathbf{A}, \mathbf{G}) \propto \Pr(\mathbf{G}|\mathbf{S}) \Pr(\mathbf{S}|\mathbf{R}, \mathbf{\Omega}, \tau_x, \theta, \boldsymbol{\gamma}) \Pr(\mathbf{R}|\mathbf{\Omega}, \tau_x, \rho) \Pr(\tau_x) f(\mathbf{\Omega}) f(\theta) f(\rho). \tag{3.1}$$

Note that although we now write $\Pr(\mathbf{S}|\mathbf{R}, \mathbf{\Omega}, \tau_x, \theta, \boldsymbol{\gamma})$ as explicitly dependent on the haplotype frequency parameter $\boldsymbol{\gamma}$, the model for this term is identical to what was given in equation (2.3). More information about $\boldsymbol{\gamma}$ is given in Section 3.1.1.

The term $\Pr(\mathbf{G}|\mathbf{S}) = \Pr(\mathbf{G}|\mathbf{H})$, where $\mathbf{H}$ consists of the vector of unobserved tip haplotypes, is new and is proportional to an indicator function 1 if the two haplotypes at the tip nodes are consistent with the observed genotype for each individual and 0 otherwise. As long as we sample haplotypes in such a way that they are consistent with the observed genotypes, this term will always be 1.

3.1.1 The haplotype frequency parameter $\boldsymbol{\gamma}$

In both the haplotype-based sampler and the genotype-based sampler, the term $\Pr(\mathbf{S}|\mathbf{R}, \boldsymbol{\Omega}, \tau_x, \theta, \boldsymbol{\gamma})$ depends on the haplotype frequency parameter $\boldsymbol{\gamma}$ through the model for sequence that recombines in to the tree. It is worthwhile to briefly review the model for haplotypes that recombine in, as described in Section 2.3, as we use the same model for the genotype-based sampler. A first order Markov model was assumed, so that the allele at a locus depends only on the allele at the previous locus. Thus, if $h_{q->v}$ is the haplotype between loci q and v inclusive, and $s_j = 0$ or 1 is the allele at locus j, the haplotype probability is modeled as

$$\Pr(h_{q->v}) \quad = \quad \Pr(s_q)\Pr(s_{q+1}|s_q)\cdots\Pr(s_v|s_{v-1}). \tag{3.2}$$

We have used fixed estimates for the terms of the form $\Pr(s_q)$ and $\Pr(s_{q+1}|s_q)$ in our phase-known sampler. These estimates are computed based on the observed sequence data, with the frequency estimation handled separately from the sampling algorithm. Briefly, two-locus haplotype probabilities and allele frequencies are estimated by counting alleles and two-locus haplotypes in the sequences observed at present and adding one to the counts.

Let $\boldsymbol{p} = (p_1, \ldots, p_{L-1})$ be the probabilities of the '1' allele at loci 1 through $L-1$ (i.e., $\Pr(s_j = 1)$) and

$$\boldsymbol{\gamma} = \begin{bmatrix} \gamma_{0,1} & \gamma_{0,2} & \cdots & \gamma_{0,L-1} \\ \gamma_{1,1} & \gamma_{1,2} & \cdots & \gamma_{1,L-1} \end{bmatrix}$$

where $\gamma_{a,j} = \Pr(s_{j+1} = 1|s_j = a)$ for $j = 1 \ldots L-1$. For example $\Pr(s_5 = 1|s_4 = 0) = \gamma_{0,4}$ and $\Pr(s_5 = 0|s_4 = 0) = 1 - \gamma_{0,4}$. The model suggested in Zöllner and Pritchard (2005) then is the same as assuming that the alleles of sequences that recombine into the tree are drawn from conditionally independent Bernoulli distributions, with parameters either from $\boldsymbol{p}$ or $\boldsymbol{\gamma}$ depending on the term being computed. That is, for the j^{th} locus $\Pr(s_j = a) = p_j^a(1-p_j)^{1-a}$ and $\Pr(s_{j+1} = b|s_j = a) = \gamma_{a,j}^b(1-\gamma_{a,j})^{1-b}$.

It is often more convenient to use an equivalent parametrization of this model in terms of the two-locus haplotype probabilities rather than the conditional haplotype probabilities. With this parametrization,

$$\boldsymbol{\gamma} = \begin{bmatrix} \gamma_{00,1} & \gamma_{00,2} & \cdots & \gamma_{00,L-1} \\ \gamma_{01,1} & \gamma_{01,2} & \cdots & \gamma_{01,L-1} \\ \gamma_{10,1} & \gamma_{10,2} & \cdots & \gamma_{10,L-1} \end{bmatrix}$$

where $\gamma_{ab,i}$ is the probability of haplotype ab at loci i and $i+1$. All other necessary probabilities can be determined from this matrix. For example, for any $k \in 1 \ldots L-1$

$$\gamma_{11,k} = 1 - \gamma_{00,k} - \gamma_{01,k} - \gamma_{10,k},$$

$$\Pr(s_k = a) = \gamma_{a0,k} + \gamma_{a1,k} = \gamma_{0a,k-1} + \gamma_{1a,k-1}$$

and

$$\Pr(s_{k+1} = b | s_k = a) = \frac{\Pr(s_{k+1} = b, s_k = a)}{\Pr(s_k = a)} = \frac{\gamma_{ab,k}}{\gamma_{a0,k} + \gamma_{a1,k}}.$$

Using fixed values for the unknown haplotype frequencies, as done in Zöllner and Pritchard (2005), rather than using the sampling algorithm to jointly estimate the frequencies and genealogy, seems inconsistent with the Bayesian approach to estimation. For both the haplotype and genotype-based samplers, we could instead have chosen to treat $\boldsymbol{\gamma}$ as unknown and set a prior that modelled evolutionary processes acting on ancestral haplotypes that recombine in and which are not modeled by the tree itself. This could perform better than assuming fixed values but at the cost of increased complexity and more terms to estimate. To simplify our implementation of the genotype-based sampler, we therefore continue to assume that haplotype probabilities are known and fixed.

We could also choose not to assume a first order Markov model on haplotype frequencies so that each element of $\boldsymbol{\gamma}$ is the frequency for an L-locus haplotype. For L loci, the size of $\boldsymbol{\gamma}$ would be $2^L - 1$. For $L > 2$, this is larger than $3(L-1)$, the number of elements of $\boldsymbol{p}$ and $\boldsymbol{\gamma}$ when the first order Markov assumption is made. It is also likely that many of these components would have near zero value. If we made this assumption, then

$$\Pr(h_{q->v}) \quad = \quad \sum_{j \in S} \gamma_j \qquad (3.3)$$

where S is the set of haplotypes that have sequence $h_{q->v}$ for their loci q through v. In this approach, we would be assuming that the sequence that recombines in at a node is drawn from a multinomial distribution with parameter $\boldsymbol{p}$ that is a function of $\boldsymbol{\gamma}$. At the present time, we have chosen not to use this haplotype parametrization and instead assume the first order Markov model. A drawback to this decision is that this model does not account for allelic association existing beyond pairs of loci. In genomic data it is known that allelic association tends to extend over longer distances than just the previous locus, but this information is not used and the resulting haplotype segments sampled from this model may not be biologically reasonable. However, this may be sufficient for the haplotype frequency model since most of the information about haplotype phase will be derived from the tree.

As we don't observe the haplotypes of the tip sequences we need to choose a method to set values for the γ that doesn't rely on known phase. Our only information about these values is either in the observed genotype data or from a public database such as HapMap (`http://hapmap.ncbi.nlm.nih.gov/`). If no external information is available, we can use one of the many proposed methods for haplotype frequency estimation and reconstruction, including the EM algorithm (Excoffier and Slatkin, 1995; Hawley and Kidd, 1995; Long et al., 1995) and Bayesian algorithms (Stephens et al., 2001; Scheet and Stephens, 2006; Niu et al., 2002), which will give haplotype frequency estimates across all L loci. These haplotype estimates can then be used to estimate the two-locus haplotype frequencies.

3.2 Proposal distributions for tip sequences

Since we have introduced latent variables for the haplotypes of the tip sequences, we will require proposal distributions that sample new haplotype configurations. Like the topology updates, we have found that it is beneficial to make updates to the tips of the sequences using two different proposal distributions that make changes to the sequences of a different nature. By performing updates from both proposal distributions mixing of the sampler may be improved.

The first proposal distribution, the allele-swap, is based on the proposal distribution described in Kuhner and Felsenstein (2000) and it involves selecting a single heterozygous locus in a single individual and swapping the alleles on the two sequences corresponding to this individual. Since the sequence at a node depends on the sequence of it's parent, it is possible that even such a minor change will have low probability of acceptance because of mutation or recombination events being required to explain the difference between the tip and parental sequence. However, the change of haplotype at a tip sequence is a discrete change and it isn't possible to make a less drastic change than swapping the alleles at one locus in one individual.

The second proposal distribution changes the haplotypes by choosing a new haplotype configuration from those that are likely given the current haplotypes at the tip nodes. This produces a sequence that is much different than the parent's sequence and we therefore also do topology changes to find new siblings having sequences similar to the new configuration. As described in detail in Section 3.2.2, the topology change here is like completing two major topology changes of the phase-known sampler, then also changing recombination and sequence variables for all nodes ancestral to the tips that have moved in order to ensure compatibility of the values that have changed with those that have not changed. The resulting update involves modification to many variables and it is therefore quite a drastic change that is made.

The two proposal distributions make complementary changes to the sequence. The first proposal distribution makes the smallest change possible by swapping the allele at one of the heterozygous loci in the sample. However, two haplotype configurations that are compatible with a genotype may be different by more than one allele swap. To move from one configuration to another would require multiple allele swaps to be accepted. Since few allele swaps are accepted in general, it may take many iterations for this to occur and mixing will therefore be slow. The second proposal distribution chooses a completely different haplotype configuration, but it is restricted to strongly favour configurations that are combinations of the haplotypes currently seen in the tip sequences. The number of steps required to go from one configuration to another is therefore much smaller than if just allele swaps are attempted. However, the penalty for making such a large change to the sequence is that topology changes are needed.

During the development phase, small datasets were used to test the performance of the various different haplotype updates. In particular, the main test dataset consisted of 30 sequences of 18 loci in a 10,000 base-pair region. The data were simulated using MS (Hudson, 2002), assuming a coalescent model with constant population size and $\theta = \rho = 4$. Haplotypes were randomly paired and only genotype information was retained. Based on performance on the test dataset, many other versions of these two proposal distributions were implemented and subsequently discarded. In Section 3.2.4, we give a brief discussion of the variations that were attempted.

3.2.1 Allele-swap

This update is completed as follows:

1. Select an individual, say the i^{th}, from the set of n_u individuals with uncertain phase.

2. Select an alternate haplotype configuration for that individual by choosing one locus from the n_h heterozygous sites and swapping phase configuration only at that locus.

A similar update was proposed for the genotype-based sampler described in Kuhner and Felsenstein (2000). They also explored making a topology change after the allele swap by dissolving and reconnecting either one or two of the branches connecting the new tips to the tree. They found that even with Metropolis-coupled MCMC tempering (Geyer, 1991), neither produced high acceptance probabilities and both were inferior to the swap with no-topology change.

The proposal probability for this update is

$$Q(\tilde{\mathbf{A}}|\mathbf{A}) = \frac{1}{n_u}\frac{1}{n_h}$$

where n_u is the number of individuals with uncertain phase and n_h is the number of heterozygous sites in individual K_i's genotype. This is a symmetric update, that is $Q(\tilde{\mathbf{A}}|\mathbf{A}) = Q(\mathbf{A}|\tilde{\mathbf{A}})$, since the number of individuals with unknown phase is the same in $\mathbf{A}$ and $\tilde{\mathbf{A}}$, and the number of heterozygous sites in the chosen individuals genotype is the same in $\mathbf{A}$ and $\tilde{\mathbf{A}}$. The acceptance probability for this update will therefore consist only of those terms that are not the same in $f(\mathbf{A}|\mathbf{G},\gamma)$ and $f(\tilde{\mathbf{A}}|\mathbf{G},\gamma)$. This update changes only the sequence for tip nodes K_{i_1} and K_{i_2}. Therefore $\Pr(\mathbf{s}_{i_1}|\mathbf{s}_{a_{i_1}}, b_{i_1}, r_{i_1}, z_{i_1}, \theta, \gamma)$ and $\Pr(\mathbf{s}_{i_2}|\mathbf{s}_{a_{i_2}}, b_{i_2}, r_{i_2}, z_{i_2}, \theta, \gamma)$ will have different values in the numerator and denominator of the Metropolis-Hastings ratio. These are computed based on the probability of a sequence conditional on the parent's sequence for loci that were inherited from this parent and it is determined by the Poisson mutation model described in Section 2.3.3. For loci that were not inherited from this parent, the probability of the sequence is based on the haplotype frequency model parametrized by $\boldsymbol{\gamma}$.

In tests of this proposal distribution on the main test dataset of 15 individuals, the update had a low overall acceptance probability of approximately 1%. First, as mentioned, although only a minor change is made if the allele-swap induces a mutation on the branch between parent and child it may be unlikely to be accepted due to mutation events being relatively rare. Second, the genotype data provides more information about phase for some individuals than for others. For example, when examining the posterior haplotype configuration probabilities from PHASE output on the test dataset, some individuals are estimated to have over 99% probability of a particular configuration. Similarly, the information provided by the genotypes about the phase is also not equal for all heterozygous loci. Because this update samples each individual and heterozygous

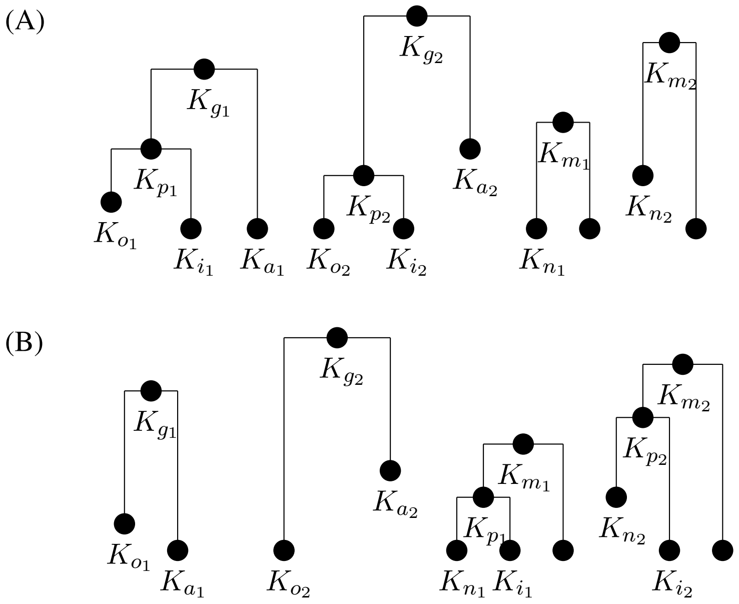


FIGURE 3.1: Illustration of the topology change resulting from the haplotype update. (A) Nodes K_{i_1} and K_{i_2} have their sequences updated and are moved to another location in the tree. (B) Resulting tree after the two topology changes. Node K_{n_1} is sampled from the set of candidate nodes that K_{i_1} can coalesce with, based on sequence similarity with K_{i_1}'s new sequence. After the topology change to K_{i_1} is made, K_{n_2} is then selected as K_{i_2}'s new sibling in a manner similar to that for K_{n_1} and K_{i_1}.

locus equally, we are expending effort trying to update loci and individuals to configurations that are extremely unlikely given the observed genotypes.

3.2.2 Dictionary Rephaser

The dictionary rephaser attempts to address some of these limitations with the allele swap. First, it makes a larger change to the tip sequences by sampling the new haplotype configuration for an individual from the set of likely haplotype configurations given the current set of tip sequences. It can therefore quickly move between configurations that differ by multiple allele swaps. Second, topology changes are also done to avoid forcing mutations between parent and offspring sequences. Finally it is structured to concentrate sequence updates to nodes where multiple configurations are likely and thereby avoids attempts to update sequences that are quite certain.

The topology change for this haplotype update resembles the major topology update described in Section 2.5.4, except two nodes must be moved rather than one. In order to move the two nodes without causing incompatibilities with the r values on the tree, it is necessary to not only update the r's corresponding to close relatives of the node that moved, but also the r and therefore s values to all the ancestors of the node that moved and it's relatives. This topology change therefore results in a fairly drastic change to both the haplotypes at the tips and the topology.

The proposal procedure is now summarized:

1. For each individual in the dataset, enumerate all possible haplotype configurations compatible with their genotype. We call the list of all possible configurations for all individuals the set of candidate configurations. The goal of the update is to select a particular rephasing (for a particular individual) from the set of candidate configurations. As the set of candidate configurations can be large, its enumeration can be computationally intensive, as described below.

2. The current tip haplotypes make up the haplotype dictionary. Provided the tip haplotypes are not changed, the dictionary does not change.

3. Compute a similarity score for each of the candidate configurations by comparing the two haplotypes to the dictionary. Details about the score is given below. The scores are re-weighted to sum to 1 and a new configuration is chosen based on the re-weighted scores. Because we are more likely to sample a new configuration if it is similar to the current tip haplotypes, the idea is to concentrate the haplotype updates on individuals with haplotype configurations that have higher probability given the current tip haplotypes. Computing and sampling the score can be computationally intensive if the dictionary changes frequently, as described below.

4. The new configuration should be similar to the current tips, but it is not necessarily similar to the current parents. Therefore, the nodes corresponding to the sequences $\mathbf{s}_{i_1}$ and $\mathbf{s}_{i_2}$ are moved. Referring to Figure 3.1, the branches connecting K_{i_1} and K_{i_2} to the tree are dissolved by removal of the parental nodes K_{p_1} and K_{p_2}. The siblings, K_{o_1} and K_{o_2}, now have parents K_{g_1} and K_{g_2} rather than K_{p_1} and K_{p_2} therefore their branch lengths are changed by this update.

5. For new node K_{i_1}

(a) Select a node for K_{i_1} to coalesce with, K_{n_1}, based on sequence similarity. This similarity score is also described below.

(b) Modify the topology so that K_{n_1} is now K_{i_1}'s sibling, and their parent is K_{p_1}.

(c) Sample a new node time for K_{p_1}, from a uniform distribution with minimum value $\max(t_{i_1}, t_{o_1})$ and maximum value t_{m_1}.

6. Given the new topology, repeat the above for node K_{i_2}

7. With both topology changes made, sample new values for $r_{i_1}, r_{i_2}, r_{o_1}, r_{o_2}, r_{n_1}, r_{n_2}$ since their branch lengths have been changed by the topology update and the previous values may be unlikely given the new lengths. Sample new values for the r and s variables for all nodes ancestral to $K_{i_1}, K_{i_2}, K_{o_1}, K_{o_2}, K_{n_1}, K_{n_2}$. This ensures that the proposed r values for the relatives of K_{i_1} and K_{i_2} are possible given the topology. The proposal distributions for the r and s values are based on the recombination and mutation models respectively. In particular, the proposal distribution for the r is based directly on the recombination model given in Section 2.3.2. No constraints on the support are required because this update changes r values for all nodes that are ancestral to the nodes that have moved. For sampling new s values, the proposal distribution is similar to that given in Section 2.4. However, because this update modifies parental sequence for all nodes ancestral to the nodes that have moved, we do not condition on parental sequence when sampling the new sequence for a node. Instead, the new sequence is sampled conditional on the children's and sibling's sequences.

Two similarity scores are required for this update. Recall that a similarity score was used for the major topology update in Section 2.5.4 that counted the number of identical alleles between two sequences. Although two sequences that are identical will have a higher score than two sequences that differ by one allele, the difference in the score is not so great that we don't periodically attempt to make a coalescence event between two nodes that differ by at least one allele in the major topology update. This is acceptable because the purpose of the major topology change is to create large changes in the topology. However, the primary goal of the dictionary rephaser update is to update the haplotypes, not perform a topology change. The topology changes are done only to improve acceptance of the haplotype update. Therefore, stricter similarity scores are used that strongly favour coalescence events between identical sequences. This is achieved by accounting for the probability of mutation events between two sequences when constructing the scores.

The first score is used to sample a new haplotype configuration from the set of candidate configurations. It quantifies the similarity between a particular haplotype configuration and the set of tip haplotypes. The score for the j^{th} configuration is computed as follows:

1. Compare the first haplotype of configuration j to all of the tip sequences and store the maximum of the number of loci with identical alleles in both haplotypes. Call it max_1; it has a maximum value of L, the total number of loci, if the two sequences are identical.

2. Compare the second haplotype of configuration j to all of the tip sequences and store the maximum of the number of identical alleles. Call it max_2.

3. max_1 and max_2 correspond to the similarity for the most similar haplotypes at the tips. There are $d = 2L - max_1 - max_2$ differences between the new sequences and this closest pair. Assuming the differences between the two sequences correspond to mutation events, the score accounts for the probability of the mutation events. In Section 2.3.3, we assumed that mutation occurs according to a Poisson process with rate $\theta/2$ per SNP per unit of coalescent time. For two tip nodes having first common ancestor t time units ago, the total branch time separating the two nodes will be $2t$ and, since mutation events on the two branches are independent, the number of mutations occurring on the branches separating the two nodes will therefore be Poisson distributed with mean $\theta/2 \times 2t$. However, since the j^{th} configuration doesn't yet correspond to an actual node in the tree, we let t=1, which is the average coalescence time for a sample of two sequences under the neutral coalescent model. The score is

$$p_j = \Pr(d \text{ mutation events}|\theta) = \frac{\theta^d * \exp(-\theta)}{(d)!},$$

   where the current value of θ is used to compute the score. If the j^{th} configuration is the same as $\mathbf{s}_{i_1}/\mathbf{s}_{i_2}$ then let $p_j = 0$.

4. Calculate the score for all j configurations and rescale all the scores so they sum to 1. This defines the proposal distribution for sampling a new haplotype configuration.

The second similarity score is used to choose the new sibling K_n for the node K_i. The new sibling is chosen based on its sequence's similarity to the new haplotype assigned to node K_i. We will require a similarity score that strongly favours coalescence events between identical or very similar sequences in order to increase the probability of accepting the update. As with choosing a new configuration, this score weights the number of differences between two sequences by the probability of mutation events. The derivation of this score is also explained in a similar manner. The allelic differences in the portion of sequence that both nodes inherited from their parent are due to mutation events occurring on the branches, b_i and b_n, from each node to its parent. For a given difference between the two sequences, since we don't yet know the sequence of the parent, we also don't know which of the two children received a mutated allele and which received the parent's allele. We know only that a mutation occurred somewhere over the total branch time separating the two nodes, $t = b_i + b_n$. For computing this score, since we haven't yet sampled time and recombination variables, we let the parent's time be the mean of its proposal distribution and we let both K_i and K_n inherit the maximum amount of sequence from their parent. Let the number of different alleles on the portion of the sequence that both could inherit from their parent be d_{ij} and assume that each difference corresponds to a distinct mutation event. The number of mutations is assumed to be Poisson distributed with mean $t\theta/2$ so the score for this node is proportional to

$$\Pr(d_{ij} \text{ mutations}|t\theta/2) = \frac{(t\theta/2)^{d_{ij}} \exp(-t\theta/2)}{d_{ij}!}.$$

Note that we do not allow the nodes to coalesce with their current parents in the tree, as this leads to a nonsensical topology.

The update to $\mathbf{s}_{i_1}$, $\mathbf{s}_{i_2}$, τ_x, the time of the parent node, and all the r and s values is either accepted or rejected as a block with probability given by the Metropolis-Hastings ratio.

3.2.3 Running the sampler with both updates

Although the dictionary rephaser can propose quite different haplotypes for an individual, it is limited by the haplotypes currently observed in the tips. Theoretically, a configuration involving two haplotypes that are not observed in the dictionary, but are similar to haplotypes in it, could be chosen. However, since the score is weighted by the mutation rate it is quite unlikely that this configuration would be chosen. Hence, in practice, the haplotypes can mix faster with the dictionary rephaser if other configurations are possible given the dictionary, but new haplotypes won't generally be introduced into the dictionary unless the allele swap is also used.

When running the sampler, both proposal distributions should therefore be used and the user can specify the proportion of time each is called. Both tend to have low acceptance rates, but the acceptance rate of the dictionary rephaser tends to be lower. Since a dictionary rephasing includes two topology changes, the lower acceptance rate is expected.

The main test dataset consisting of 15 individuals was used to compare results when the sampler performed (a) only the allele swap and (b) the allele swap and the dictionary rephaser. The two versions of the sampler were run on the test dataset ten times under the same initial conditions and the results were compared after one million iterations. For each individual with unknown phase, the proportion of time that individual was at their true haplotype configuration was stored. The standard deviation of this proportion across the ten replicates provides a measure of how close the proportions are to each other after one million iterations. If the standard deviation is low for a particular individual, all ten replicates estimate a similar posterior probability for the true haplotype configuration; if it is high, the estimated posterior probabilities for the true haplotype configuration differ between replicates.

Figure 3.2 shows boxplots of the standard deviations for the combined 13 individuals with unknown phase in this dataset. The standard deviations are lower when both the allele swap and dictionary rephaser are used, indicating that the ten estimates of the posterior probability for the true configuration are closer to each other after one million iterations than when the allele swap alone is used. Although this is a single dataset, it does provide evidence that the dictionary rephaser can improve mixing when used in conjunction with the allele swap.

Finally, the dictionary rephaser can be both time and memory intensive due to the need to create and store large score and haplotype configuration vectors. The length of these vectors are

$$\sum_{i=1}^{n} 2^{h_i - 1}$$

where h_i is the number of heterozygous loci of individual i. If a dataset has many large h_i values, then these vectors can get very large. Since both vectors are created once and stored throughout the run length, they can require large amounts of RAM. In example runs, more than 1GB was needed to run a particular dataset. The scores are only recomputed when the dictionary or mutation rate has changed, but if the score vectors are large this can also make run times so long that using the dictionary rephaser is infeasible. Preliminary runs to determine whether it is feasible to use this proposal type should first be done. If the run time seems too long, one could use only the allele swap proposal. Based on Figure 3.2, one might want to increase the number of MCMC samples taken when the allele swap is used alone since the allele swap on its own may have slower convergence to the target distribution.

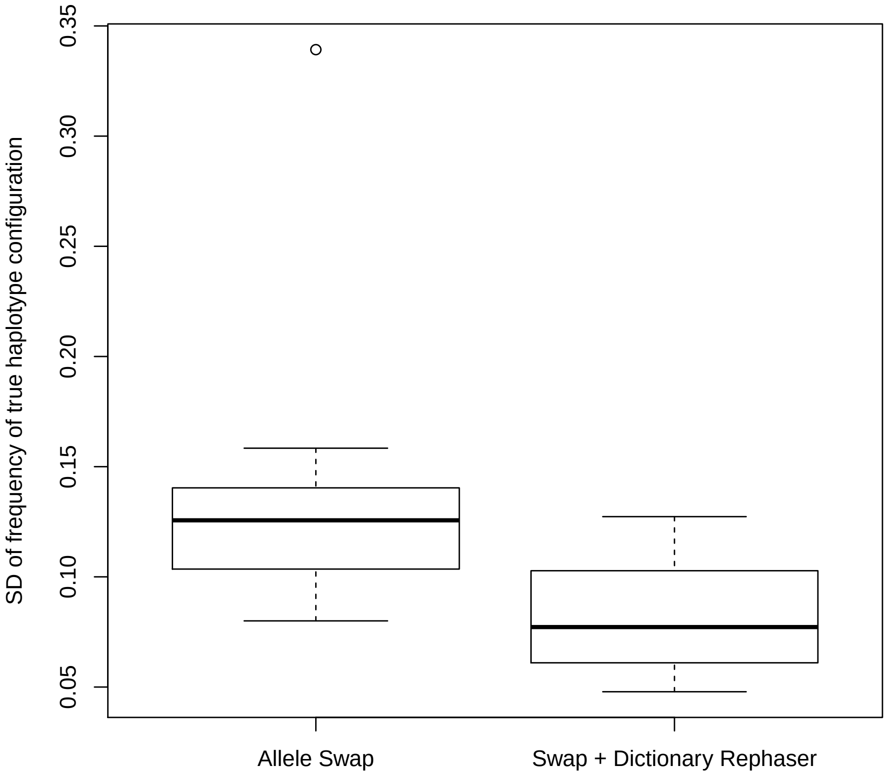


FIGURE 3.2: Comparison of mixing of the allele swap and dictionary rephaser updates. For ten replicate runs of the same dataset under the same conditions, the proportion of time the sampler spent in the true haplotype configuration for each individual was saved. Boxplots of the standard deviation across replicates of this proportion, for all individuals with uncertain phase, are shown when only the allele swap is used (left) and the dictionary rephaser and allele swap are both used (right).

3.2.4 Proposal types considered but not pursued

Although the allele-swap and dictionary rephaser are highlighted above, other variations of these updates were attempted but eventually abandoned. The variations centred on (1) how the new haplotype was sampled; (2) how the individual/locus was sampled; and (3) how the post-topology updates are made.

1. The two methods described above for sampling a new haplotype involve the single locus allele swap update or a multiple locus update using the dictionary of current tip sequences. The Markov haplotype frequency model was also explored as a means to update multiple loci by sampling a new haplotype configuration with and without the topology change. However, the haplotype frequency model is not a particularly good one for sampling haplotype configurations. The model does not account for allelic association beyond the adjacent loci and therefore the sampled haplotypes may be unlike other haplotypes currently seen at the nodes of the tree.

2. Not all individuals with unknown haplotypes have equal uncertainty about the true configuration. There is a great deal of information in the data about some individual's haplotypes and therefore any attempt to change their configuration in the allele swap update is rejected. We looked into non-uniform sampling of either individual or locus in order to focus the allele swap updates on individuals or loci where there was less phase information available in the sample. Both the number of heterozygous loci and entropy scores based on PHASE output were used to generate a sampling weight. However, when both weights were tested, the acceptance rate for this update was not improved over uniform sampling. This modification to the allele swap update was not pursued further and the dictionary rephaser, which implicitly samples haplotype configurations that are more uncertain, was introduced instead.

3. Finally, after the topology change, r and s variables are modified for all branches that are *ancestral* to the nodes in the neighbourhood of the topology change. We also looked at updating these variables for all branches that *descend* from these nodes. The benefit to this direction of updates is that because we are updating tip nodes, the siblings are more likely to be closer to the tips and therefore fewer values would need to be updated. Although this is the case, the recursive nature of the r values makes it easier to update ancestral branches, since no restrictions on r values are required, than it is to update descendent branches. This lead to

an update with many special cases and ultimately it was abandoned.

3.3 Initial Sequence Values

Initial values for the haplotypes at the tip nodes are required to begin the MCMC sampling. By default the program is coded to use the first-order Markov haplotype frequency model to sample alleles at a locus conditional on the previous locus as follows:

Genotype at current locus is homozygous -

Since this indicates that there are two copies of either the 0 or 1 allele, assign this allele to each of the two sequences.

Genotype at current locus is heterozygous -

Case 1 - Previous locus is homozygous. There is no information about the haplotype configuration of the two alleles from the previous locus and so randomly choose which of the two sequences will receive the '0' allele and which will receive the '1' allele.

Case 2 - Previous locus is a heterozygous There are two possible choices for the two-locus haplotype consisting of the previous and current loci: 00/11 or 01/10. Sample the two-locus haplotype configuration based on the haplotype frequencies.

The difficulty with using the haplotype frequency model to sample an initial set of sequences is that it doesn't capture the phase information from non-adjacent loci. Loci that are not adjacent to a particular locus, but are close to that locus, do provide phase information since few, if any, recombination events may have occurred between the two loci. This model also doesn't incorporate the fact that haplotypes tend to be similar to each other, due to the underlying but unknown genealogy, and so the initial haplotypes can be unrealistically diverse. This can lead to a set of initial sequences with low posterior probability and it can take the sampler a large number of iterations before more likely haplotype configurations are found.

It may be more generally desirable to choose initial values that are in a high probability region of the target distribution (Besag, 2001). The program therefore allows the user to provide an initial set of sequences to start the sampler at a more realistic configuration. These could be based on an estimate from a haplotype reconstruction program, such as PHASE (Stephens et al., 2001). This is the approach that was taken in the simulation study described in the next section.

3.4 Examining the performance of the genotype-based sampler with simulated data

Simulated data was used to evaluate the performance of the sampler. A total of 50 datasets were simulated using the MS program (Hudson, 2002) assuming a coalescent model with recombination. A total of 50 sequences were simulated for each dataset and they were randomly paired to make a sample of 25 individuals. Both the mutation and recombination rates were set to 10^{-8} per base pair per generation, the effective population size was 10,000 individuals and a 20,000 base pair region was simulated. These values give scaled mutation and recombination rates per base pair of $\theta = \rho = 0.0004$. In order to simulate SNP ascertainment and to get a SNP density of approximately 1 per 1000 bp, which is the typical estimate for the human genome, a minor allele frequency cutoff of 10% was used. A minimum cutoff of 15 SNPs was also set, so that datasets with fewer than 15 SNPs were discarded. The median number of SNPs across all 50 datasets was 18 and the maximum number of SNPs in a dataset was 22. As discussed in Section 2.2.3, the mutation model we consider does not model the biological mutation rate of a random base pair and instead models the mutation rate of a pre-ascertained SNP. Due to our ascertainment scheme, θ is the mutation rate of a SNP conditional on the frequency of the minor allele being greater than 10%. We therefore would not expect that the mutation rate that we estimate is close to the true value used to simulate the data. In Appendix B, we provide calculations for comparing the rate used to generate the data and the rate estimated by the sampler. Finally, the purpose of this simulation study is to evaluate the sampler when there is missing haplotype phase and therefore we did not simulate any trait values associated with a mutation.

3.4.1 Prior distributions and initial values

Each of the 50 replicates was run with the same set of initial conditions. The focal points were all chosen to be the midpoint of the simulated region at 10,000 bp. Since the location of mutations is random, the focal point does not necessarily have an equal number of markers to the left and right. The prior distribution for θ was set to U(0.0001,10) and an initial value of one was used. The prior distribution for ρ was gamma with shape parameter 1 and scale parameter 0.08, which was selected based on earlier test runs on datasets with fewer sequences. The initial ρ was set to 0.004, which is a factor of 10 larger than the generating value. At a given iteration of the sampler, the update

TABLE 3.1: Probabilities of sampling each of the different update types at a given iteration for the simulated datasets

Update type	Sampling probability
theta	0.1
rho	0.15
local updates	0.35
major topology	0.1
minor topology	0.1
allele swap	0.1
dictionary rephaser	0.1

type to perform was selected based on the probabilities given in Table 3.1. With the addition of the two new haplotype updates, the frequencies with which the other updates are selected must be adjusted. The local updates are performed most often and approximately every third update. The sampling probabilities for the three other types, a rate update, a topology update, and a haplotype configuration update, were each selected to be approximately equal. The recombination rate was set slightly higher to provide more opportunities for the rate to adapt to the changing tip sequences.

The haplotype frequency model's two-locus haplotype probabilities, γ, also need to be specified. These were chosen based on estimates of the population haplotype frequencies returned by PHASE. To estimate these frequencies, PHASE was run with non-default values of 1000 iterations, a thinning size of 10 and burn-in of 500; the default is (100, 1, 100) for iterations, thinning and burn-in respectively. The documentation for PHASE suggests running PHASE multiple times to examine the consistency of the estimated values. PHASE was run a second time and there were no substantial differences between the frequency estimates; they were the same in the two runs up to two decimal places.

3.4.2 Pilot runs

Pilot runs were first used to determine whether the initial conditions were adequate. The results from these runs suggested that setting the initial haplotypes based on the haplotype frequency model did not yield a set of initial haplotypes that were likely given the data (see discussion of this problem in Section 3.3) and the sampler was taking too many iterations before it was sampling reasonable haplotypes. For this reason, PHASE was also used to generate initial values for the sampler. For each individual in the dataset, PHASE provides estimates of the posterior probabilities of possible haplotype configurations. The PHASE-estimated posterior probability distribution for each individual were therefore used to sample an initial haplotype configuration.

The pilot runs also provided information about the expected run length for each of the 50 iterations. Even though the data were all simulated under the same parameter values the run time is quite variable, ranging from one hour to over 24 hours. The variability in run time can be explained by the number of haplotype configurations, as shown in Figure 3.3. Recall that the dictionary rephaser update requires enumerating all possible haplotype configurations compatible with the observed genotypes and computing a score for each configuration. The score vector, of length equal to the total number of configurations in the datasets, must be recomputed when a haplotype or θ update is accepted. Figure 3.3(A) plots the times, in hours, to complete one million iterations by the total number of possible haplotype configurations in the dataset for all 50 datasets. As expected, the run time increases with the number of configurations. Figure 3.3(B) focuses on run times where the total number of configurations was less than 500,000 in order to see whether the linearity of the relationship was due to the scale of the plot. Although after 100,000 configurations the variability in run time seems to increase, which may be due to differences in acceptance rates across datasets, the relationship between run time and number of configurations is roughly linear.

The four datasets that did not finish one million iterations in under 24 hours all had a relatively high number of haplotype configurations in their set of candidate configurations. Due to their long run times, it is impractical to use the dictionary rephaser for these datasets and the allele swap was exclusively used instead. Although the number of haplotype configurations was high for these datasets, the haplotype uncertainty was not higher than the haplotype uncertainty of the datasets with fewer haplotype configurations. For three of the four datasets, the PHASE output puts greater than 95% posterior probability on a single configuration on all but two individuals. These two individuals have greater than 90% posterior probability for a single configuration in all three datasets. The fourth dataset corresponds to the point in Figure 3.3(A) having over two million different possible configurations. The majority of individuals in this dataset have greater than 97% posterior probability for a single configuration, although three individuals in the dataset have 50% posterior probability for each of two configurations. However, having some individuals with 50% posterior probability on each of two haplotype configurations is also not unusual; some of the other datasets with faster run times also have individuals with this type of haplotype uncertainty.

Five duplicate runs were also completed for 10 different datasets and estimates of θ, ρ and the distributions of sampled haplotype configurations were compared after 10 and 20 million iterations.

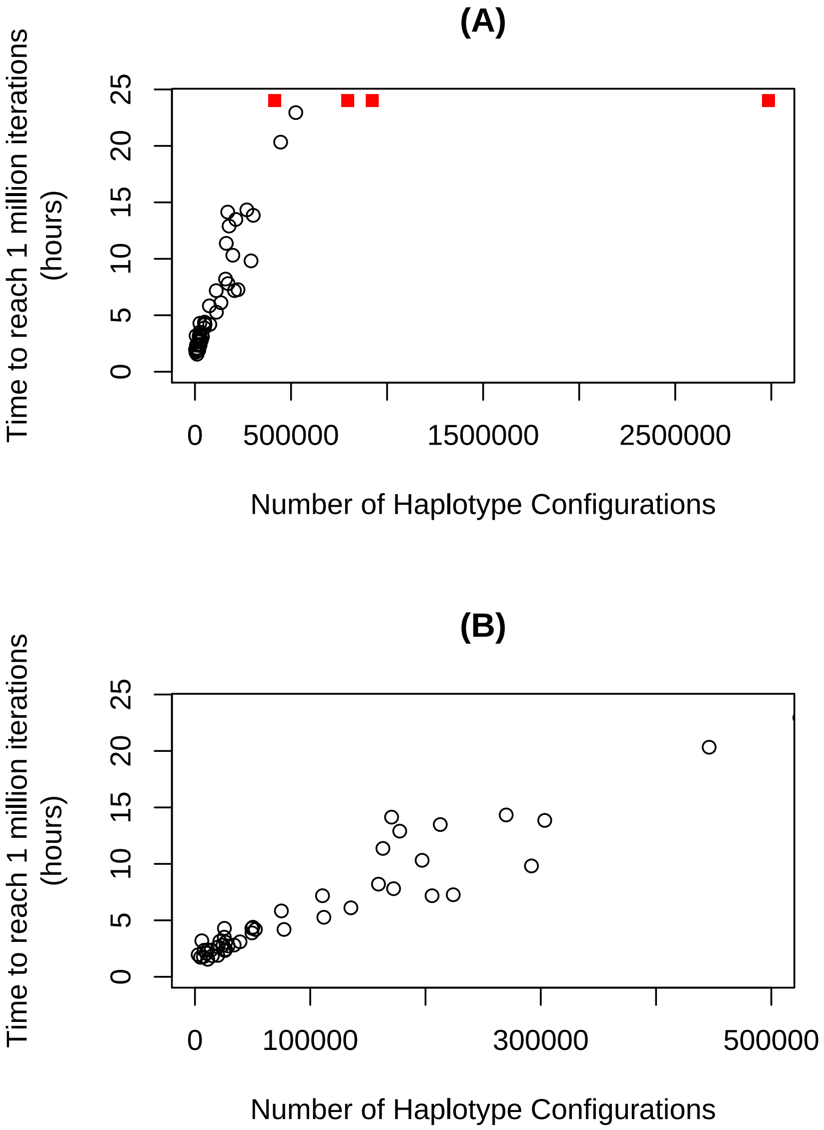


FIGURE 3.3: Plot of time to complete one million iterations by number of haplotype configurations (A) Across all 50 simulated datasets. The red squares indicate the number of haplotype configurations for the four runs that didn't finish in 24 hours. (B) Across only those datasets with fewer than 500,000 total possible configurations

For some datasets estimates of θ were fairly variable across the duplicates and the sampled haplotypes were quite different from run to run, both indicating that mixing was poor and that run length should be extended. Therefore the final results presented are for 40 million iterations, with only every $10,000^{th}$ tree saved. Even longer run lengths would be preferable; however, since some of the datasets require 24 hours to complete a million iterations, this is currently not feasible.

3.4.3 Results

To assess performance we use the sampler's ability to estimate ρ, θ and the tip haplotypes. Kuhner and Felsenstein (2000) also assess the performance of their genotype-based sampler using its ability to estimate θ and ρ; however, they do not present results on its ability to estimate the haplotypes themselves. Since there is likely to be a great degree of dependence between the topology and the tip haplotypes, it is important to examine the sampled haplotypes in order to ensure that both are mixing adequately. This is especially important for our purpose, since we are most interested in the genealogies sampled and not simply in estimating population genetic parameters. Since the data are simulated, we know the true haplotype configuration, but under most circumstances of haplotype uncertainty we would expect multiple haplotype configurations are likely given the genotype data. We are therefore interested in whether the distribution of sampled haplotypes over time is reasonable. We compare the distributions of sampled haplotype configurations for each individual to those obtained by PHASE (Stephens et al., 2001). Although there is no guarantee that the distribution estimated by PHASE is the true distribution, PHASE has been shown to be the most accurate of the haplotype inference programs (Marchini et al., 2006). Thus PHASE is reasonable to use as a “gold standard”, as far as haplotype estimation is concerned.

Estimating θ and ρ

All datasets were generated with the same true values of θ and ρ, the mutation and recombination rate respectively, and so each provides an independent estimate of these rates. The summaries of sampled θ and ρ values for each dataset are summarized by the interquartile range (IQR) and Median in Figures 3.4(A) and (B). Note that all datasets were run with the same initial values for θ and ρ of 1 and 0.004 respectively. As the IQR's for all datasets do not include these initial values, this suggests the true values were different.

The true value for ρ was 0.0004 and the estimated ρ values are all close to the truth. However, most of the mean and median estimates for ρ are above 0.0004. Across all datasets, the average of the median ρ is 0.00053. The ρ estimate for the 44^{th} dataset appears to be an outlier relative to the other datasets, in that the estimated ρ is much higher and the IQR is wider. It is unclear why this estimate is higher; the true number of recombinations, total number of SNPs, and number of SNPs with a minor allele frequency (MAF) greater than 0.1 are all similar to the other datasets. However, as will be seen, the haplotype distributions show more uncertainty than most datasets. Thus, the chance assignment of haplotypes to individuals may have lead to greater uncertainty in the recombination rate in this case.

The generating mutation rate per site was also 0.0004, but the θ estimated by the algorithm is not the standard population genetic mutation rate. The sampler will estimate the mutation rate of the pre-ascertained loci, conditional on the loci being polymorphic and having a MAF greater than 0.1 in the sampled haplotypes. The mutation rate of a pre-ascertained locus should be higher than that of a random base pair. Simplified calculations to provide an upper bound on θ are provided in Appendix B and show that the estimated θ should be less than 0.45. This calculation does not include the MAF criteria, which would be expected to reduce this upper bound further. As shown in Figure 3.4(B) the median θ values are all less than this upper bound. As mentioned, the 50 datasets are all estimating the same θ even if the truth is known only up to an upper bound. From Figure 3.4(B), we can see the distributions for sampled θ values are all fairly close to each other, except for the fourth dataset which appears to be an outlier. As with the 44^{th} dataset, the true number of recombinations, total number of SNPs, and number of SNPs with MAF>0.1 are also all similar to the other datasets. It is unclear why the estimate is higher but it may just be due to the underlying variation of the coalescent process used to simulate the data (i.e., a realized genealogical tree with long total branch length).

Traceplots of θ, ρ and tree summary statistics

Although it is not possible to show traceplots of sampled values for all 50 datasets, a representative plot for a single dataset across the 40 million iterations is shown in Figure 3.5. Plots (A) and (B) show the sampled θ and ρ values over time. Plots (C) and (D) provide traceplots of the total tree time and the Rand index, which are both tree summary statistics. The total tree time is the sum of the branch lengths of the tree. The Rand index, as described in Section 2.7.3, summarizes the similarity of clusterings between consecutive trees after thinning. It captures differences in the trees

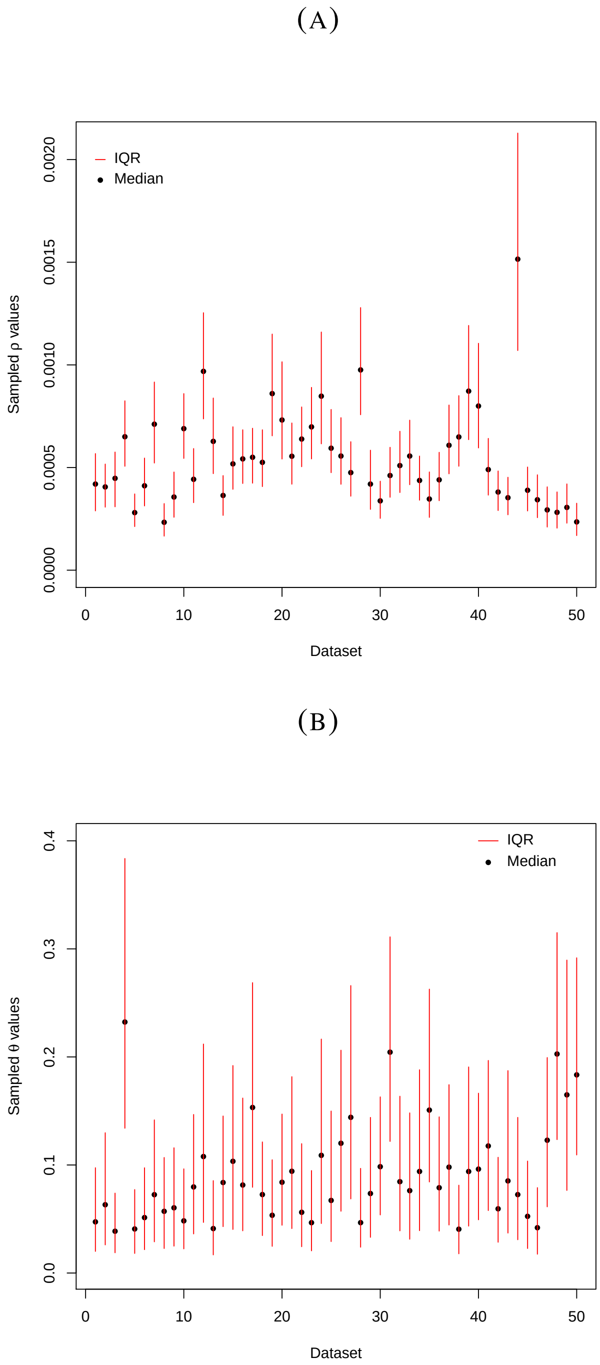


FIGURE 3.4: The marginal posterior distributions of (A) sampled ρ and (B) sampled θ values across 50 simulated datasets are represented by the interquartile range (IQR) and median.

that are due to both tip updates and major and minor topology changes. The majority of traceplots resembled the example shown in Figure 3.5, although for some datasets the sampled ρ or θ values showed a slight upward or downward trend even at 20 million iterations and therefore the long run lengths are justified.

Haplotype distributions

In order to assess how well the haplotypes are estimated, and therefore how well the sampled trees reflect haplotype uncertainty, we compare the posterior probabilities of the haplotype configurations for each individual estimated by our sampler to the corresponding estimates from PHASE (Stephens et al., 2001). The summary observations are of a qualitative nature; it is important to show that the sampler is mixing with respect to haplotypes (i.e., different haplotype configurations are sampled) and that the haplotypes that are sampled are reasonable ones. It isn't required or expected that the sampled configurations be identical to PHASE. However, if the results differed significantly it would lead to questions about our sampler's ability to sample over haplotypes.

The first note to be made is that the distributions of haplotypes sampled by our sampler and by PHASE are in general remarkably similar. For most individuals with uncertain phase, if the configuration is sampled more than approximately 5% of the time, the actual configurations that are sampled using both approaches are usually the same although the frequencies differ. For some individuals, PHASE samples the true haplotype configuration more than our sampler; the reverse is also frequently seen. With 50 datasets of 25 individuals we can't show all possible comparisons; however, a sample that summarizes the observations made on all 50 datasets is shown in Figure 3.6(A) and (B). Each consists of a barplot showing the proportion of time a particular haplotype configuration was sampled (the estimated posterior probabilities of each haplotype configuration) for each individual with uncertain phase. Haplotype configurations that were sampled less than 1% of the time are not shown. The bar labelled ‘P’ gives the PHASE-estimated posterior probabilities of the haplotype configurations; the bar labelled ‘S’ gives the posterior probabilities estimated by the sampler. For a single individual, a particular grey shade represents the same haplotype configuration in both the ‘P’ and ‘S’ bar, so that the results are similar if both bars are the same grey in roughly the same proportion. The segment corresponding to the true configuration is marked by an asterisk in the ‘S’ bar.

Figure 3.6(A) shows an example where there was a great deal of haplotype certainty and both

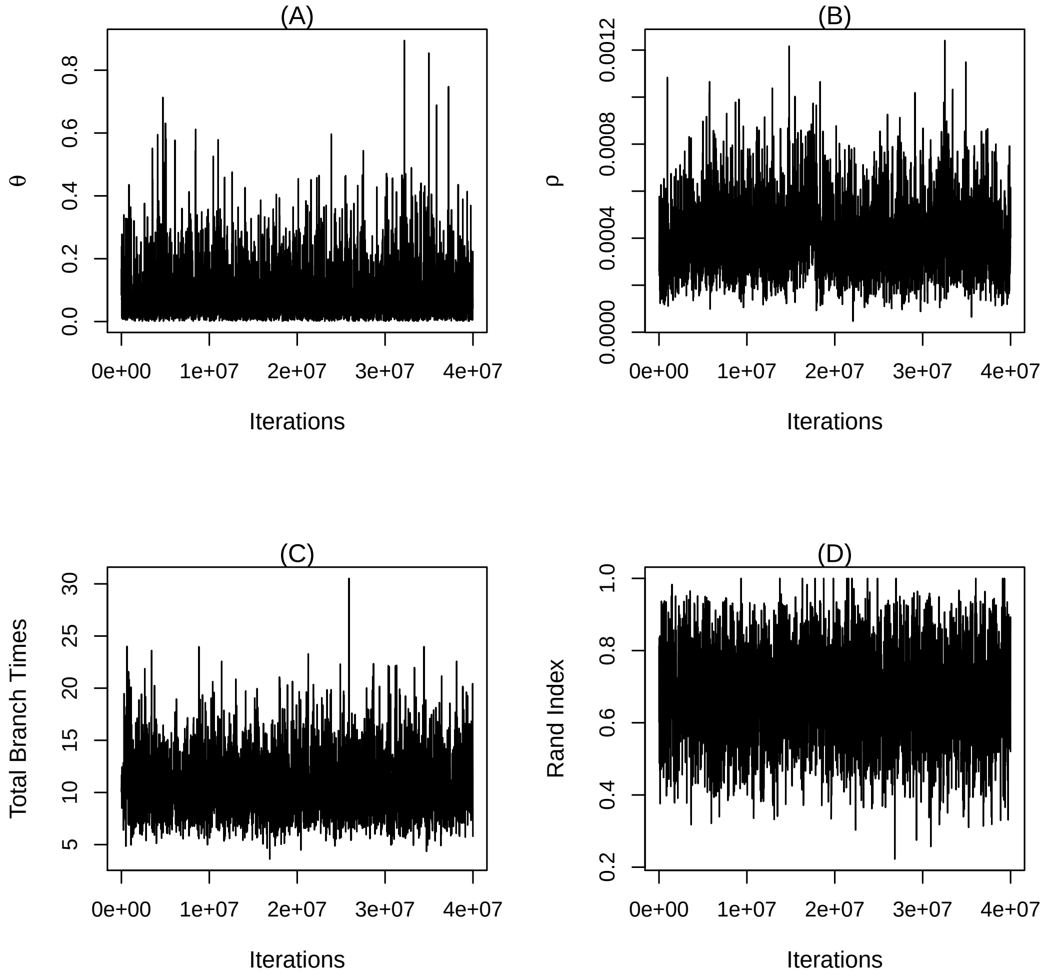


FIGURE 3.5: Example traceplots over 40 million iterations for a single simulated dataset. Sample points were recorded every $10,000^{th}$ iteration. (A) plots the samples of θ over time. (B) plots the samples of ρ over time. (C) plots the sum of the branch lengths of the tree over time.

sets of results are quite similar. Although the results differ by more than 10% for individual three, the traceplot of sampled haplotype configurations for this individual shows that the sampler moves frequently between the two configurations. Figure 3.6(B) shows a dataset with more haplotype uncertainty. As with (A), the haplotype configurations that are sampled by both approaches are the same for most individuals and each is sampled a similar proportion of the time. Individual 21 is an exception. The barplots do not include 'rare' haplotype configurations, that is, configurations estimated to have less than 1% posterior probability. Half of the bar for the PHASE-estimated posterior probability distribution appears to be missing because the remainder is composed of only rare haplotype configurations. Therefore, PHASE samples at least 50 rare haplotype configurations and so this individual is estimated to have very high haplotype uncertainty. On the other hand, our sampler does not sample rare configurations for this individual and they would therefore be considered to have more moderate haplotype uncertainty. Our sampler may be out-performing PHASE for this individual as it estimates a higher posterior probability for the true haplotype configuration than PHASE estimates. Conversely, our sampler may be less likely to move to rare haplotype configurations than PHASE.

Although the results are similar between PHASE and, when they do differ, we often spend more time at the true configuration than PHASE, recall that the initial sequences were sampled from the posterior probabilities estimated by PHASE. Therefore, if we started at a configuration with high probability and our sampler was not mixing well, we would artificially appear to have as high accuracy as PHASE. It is therefore important to examine the traceplots and ensure that for those individuals having multiple potential haplotype configurations the sampler makes many traversals between them.

Examples of sampled haplotype configurations and the corresponding traceplots are shown for two individuals in Figure 3.7(A) and (B). Panel (A) gives an example where our sampler sampled the true configuration more often than PHASE and the traceplot for this individual shows that the sampler travelled frequently between the two common configurations. Panel (B) gives an example of an individual with a lot of haplotype uncertainty. Both our sampler and PHASE sampled many different configurations for this individual. The traceplot also shows that our sampler travels frequently between the eight most common configurations. Note that this dataset had more haplotype uncertainty than most datasets and it corresponds to the outlying high estimated ρ value in Figure 3.5.

Two more problematic individuals are shown in 3.8(A) and (B). Dataset 48 had too many po-

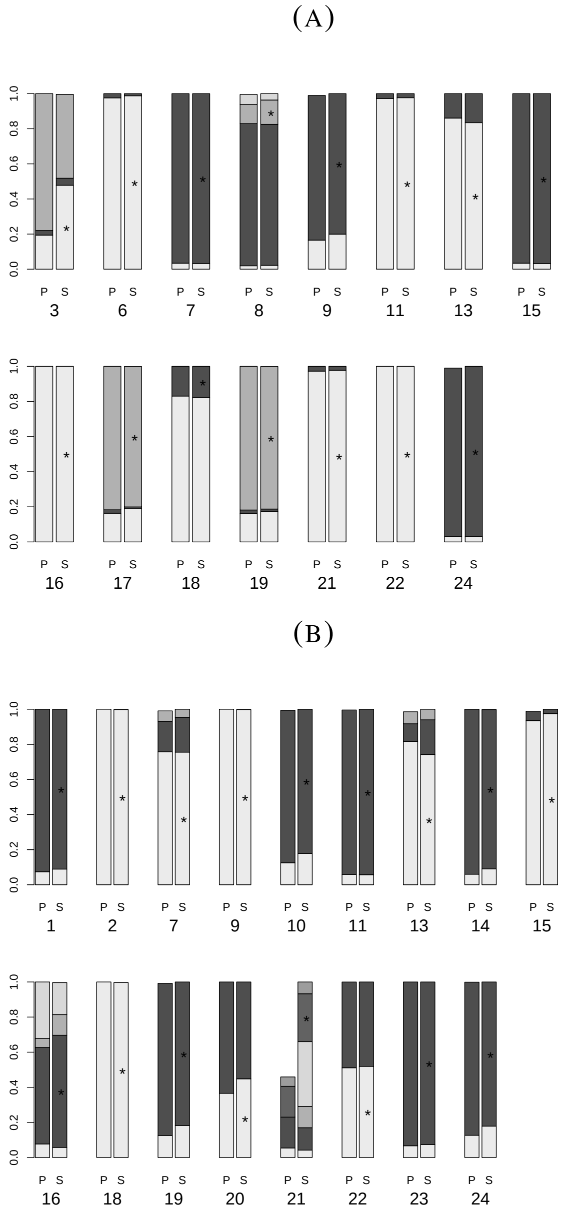
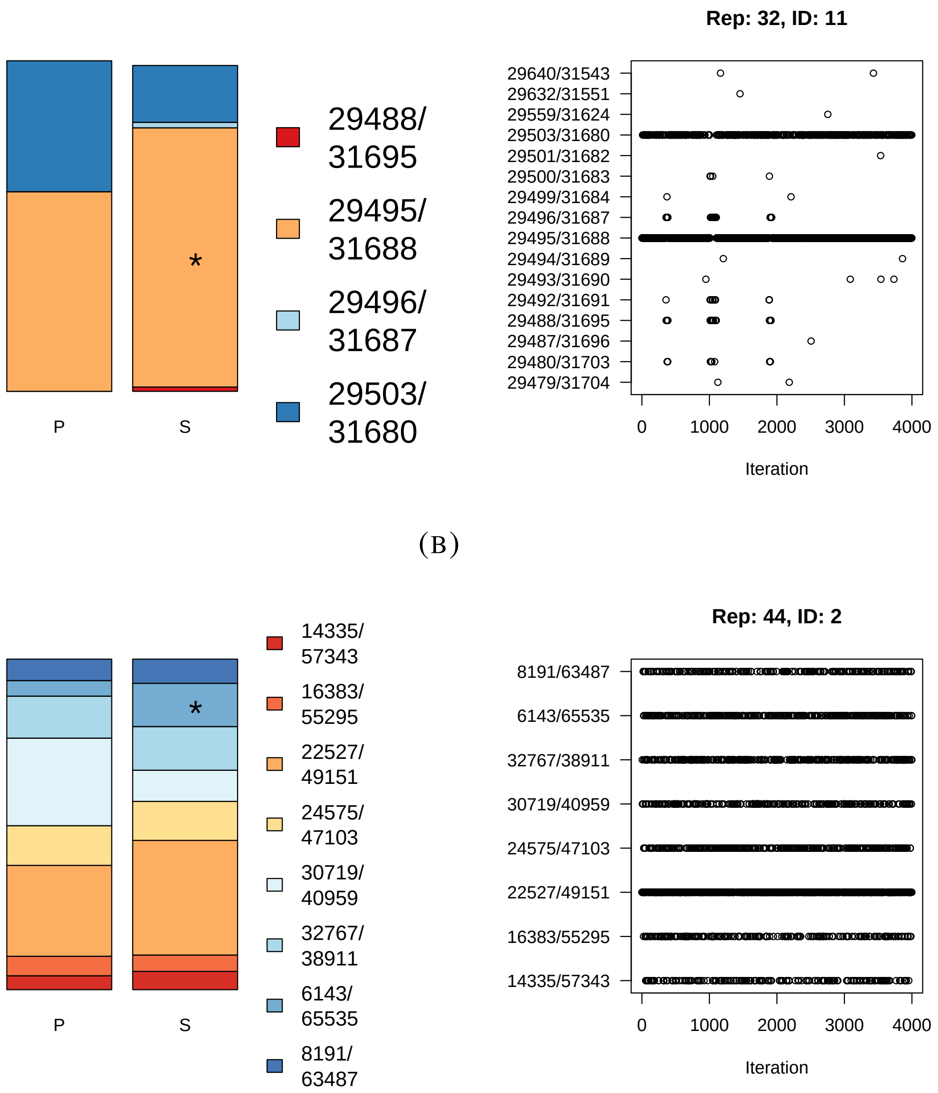


FIGURE 3.6: Barplots comparing haplotype configuration distributions for (A) dataset 3 and (B) dataset 39. Each set of bars consists of the PHASE-estimated posterior probabilities of haplotype configurations (column ‘P’) and the posterior probabilities estimated by the sampler (column ‘S’) for each individual with unknown phase in the dataset. Results are based on 40 million total iterations and samples were recorded every $10,000^{th}$ iteration. Only haplotype configurations estimated to have higher than 1% posterior probability are included in each bar. Therefore, the bars will not sum to 1. The true haplotype configuration is marked by an asterisk.


(A)

(B)

FIGURE 3.7: Sampled haplotype configurations and traceplots after 40 million iterations for (A) Individual 11 in dataset 32 and (B) Individual 2 in dataset 44. In the barplots, the columns labelled 'P' give the PHASE-estimated posterior probabilities of the haplotype configurations and the columns labelled 'S' give the posterior probabilities estimated by the sampler. The true configuration is marked by an asterisk. In the legend to the right of each barplot, the labels for the two haplotypes in each configuration have been converted from the binary sequence to the base-10 number system.

tential configurations to perform the dictionary rephaser update and so only the allele swap is used. Since the mixing may be slower when only the allele swap is used it would be expected that longer run lengths would be necessary. As can be seen from the traceplot in panel (A), although the configurations are sampled with similar frequency by both PHASE and our sampler, our sampler tends to stick at a single configuration for a long period of time. For this individual in particular, a much longer run length would be necessary to feel confident in the haplotype estimation. A similar result is seen for the individual from dataset 43 shown in panel (B), which would again call for longer run lengths. Finally, although the traceplots for these individuals show signs of poor mixing, this tends to be a property of the individual within the dataset rather than the dataset. That is, the sampling of different haplotype configurations may be adequate for most individuals but inadequate for a single individual in a given dataset.

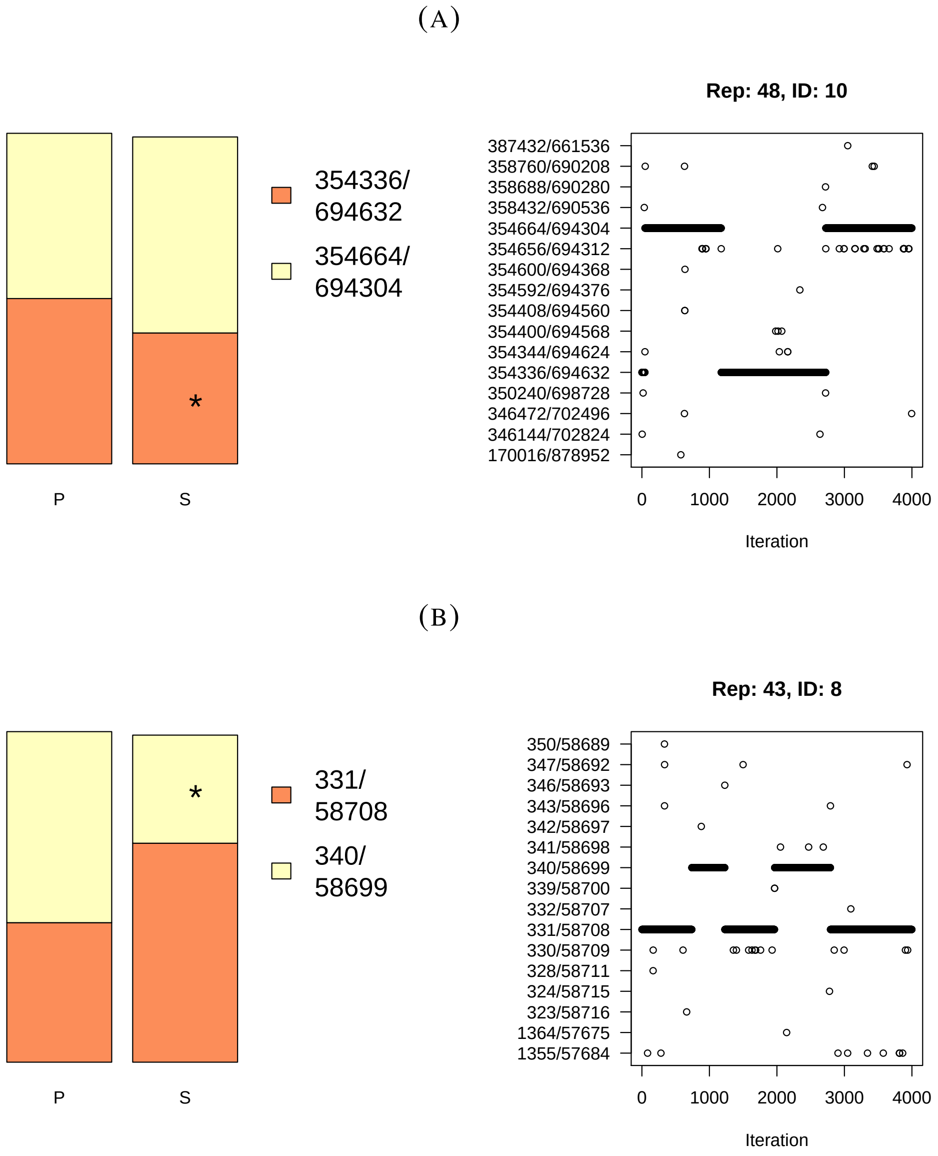
3.4.4 Sensitivity to prior distributions

Finally, our sampler requires specifying a prior distribution for θ, ρ and for the haplotypes that recombine in to the tree. To assess the sensitivity, two other prior distributions for each of θ, ρ and the haplotypes were used on ten of the fifty datasets.

Recall that the time to complete one million iterations ranged from 1.5 to 24 hours. To complete 40 million iterations could therefore take up to 40 days with the dictionary rephaser. Due to time constraints, to run the sampler with different prior distributions, we selected a subset of ten datasets based on having run times of one million iterations in two hours or less. By examining the barplots of the sampled haplotypes like those shown in Figure 3.6, these datasets do not appear to be unusual in their haplotype certainty, which provides some reassurance that the results would hold more generally.

Modifying the prior distribution for ρ

The prior distribution for ρ used for the main runs of the 50 datasets was an exponential distribution with mean 0.08 and the initial rho was set to 0.004. The true value of ρ is 0.0004. Two other prior distributions for ρ were run, with a higher and lower mean. Figure 3.9 shows estimated marginal posterior densities for ρ based on 40 million iterations using the original prior and two alternate prior distributions: exponential with mean 0.01 and exponential with mean 1. The results for the two lowest means of 0.08 and 0.01 are quite similar. The mode of the sampled ρ values when the


(A)

(B)

FIGURE 3.8: Sampled haplotype configurations and traceplots after 40 million iterations for (A) Individual 10 in dataset 48 and (B) Individual 12 in dataset 43. In the barplots, the columns labelled 'P' give the PHASE-estimated posterior probabilities of the haplotype configurations and the columns labelled 'S' give the posterior probabilities estimated by the sampler. The true configuration is marked by an asterisk. In the legend to the right of each barplot, the labels for the two haplotypes in each configuration have been converted from the binary sequence to the base-10 number system.

prior distribution for ρ is exp(1) is higher, indicating some sensitivity to the prior distribution for ρ. However, since the true value for ρ is 0.0004, an exponential distribution with a mean of 1 is a poor choice for the prior distribution. Although only results for four datasets are shown, similar results are seen for all ten of the datasets where alternate prior distributions were evaluated.

Modifying the prior distribution for θ

The prior distribution for θ used for the main runs of the 50 datasets was uniform on (0.001,10) and the initial θ value was 1. As outlined in Appendix B, the true value of θ should be less than 0.44. Two other prior distributions for θ were examined, one that was wider and one that was narrower. Figure 3.10 shows the estimated marginal posterior densities of θ over 40 million iterations using the original prior and two alternate prior distribution: $U(0.005, 1)$ and $U(0.0001, 100)$. The initial values were 0.1 and 1 respectively. The estimated densities for sampled θ values are very close to each other. Although only results for four datasets are shown, similar results are seen for all ten of the datasets where alternate prior distributions were evaluated.

Haplotype frequency model

As mentioned, the haplotype frequency parameter, $\boldsymbol{\gamma}$, was set based on the population haplotype frequencies estimated by PHASE. To examine the sensitivity of the sampler to $\boldsymbol{\gamma}$, the sampler was run with two different sets of frequencies. The first set consisted of the true two-locus frequencies based on the known data, which were available because the data are simulated. The second set consists of the values estimated by PHASE, but perturbed by adding noise from a Normal distribution with mean centred on the actual value and standard deviation 0.05. If the perturbed value was less than zero, a new value was sampled. The perturbed frequencies were then rescaled to sum to one.

To assess sensitivity, we compared the estimated posterior probabilities of each haplotype configuration for each individual with uncertain haplotype phase, for both frequency models, pooling across 10 of the 50 datasets. As shown in Table 3.2, for the majority of haplotype configurations, the difference between the estimated posterior probabilities under the two models is less than 15% after 40 million iterations. There were eight individuals (16 haplotype configurations) where the difference was greater than 15%. For six of these eight individuals, the large differences could be explained by poor mixing, as assessed by the traceplots, rather than by the underlying model for the population haplotype frequencies.

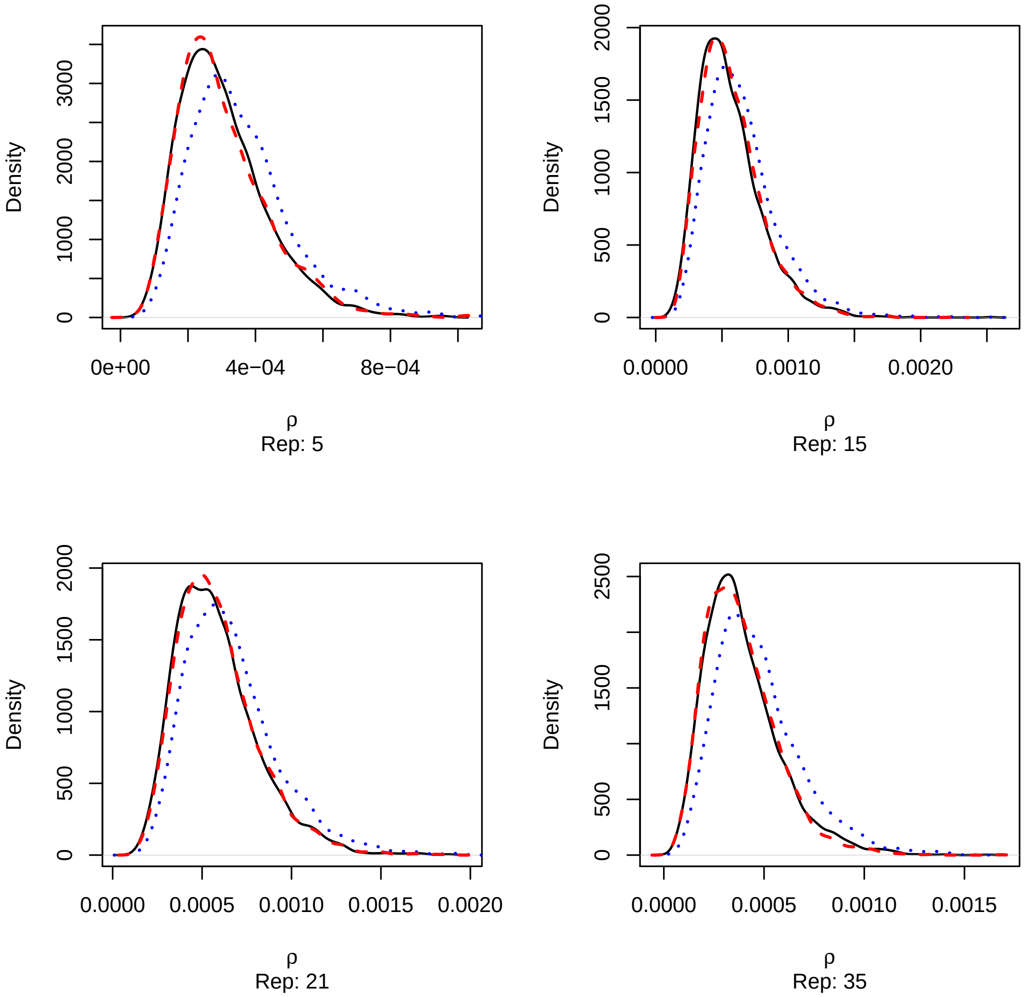


FIGURE 3.9: Estimated posterior density for ρ under alternate ρ prior distributions for four datasets. The black line corresponds to an exponential prior with mean of 0.08, the dashed red line to an exponential with mean 0.01 and the dotted blue line corresponds to an exponential with mean 1.

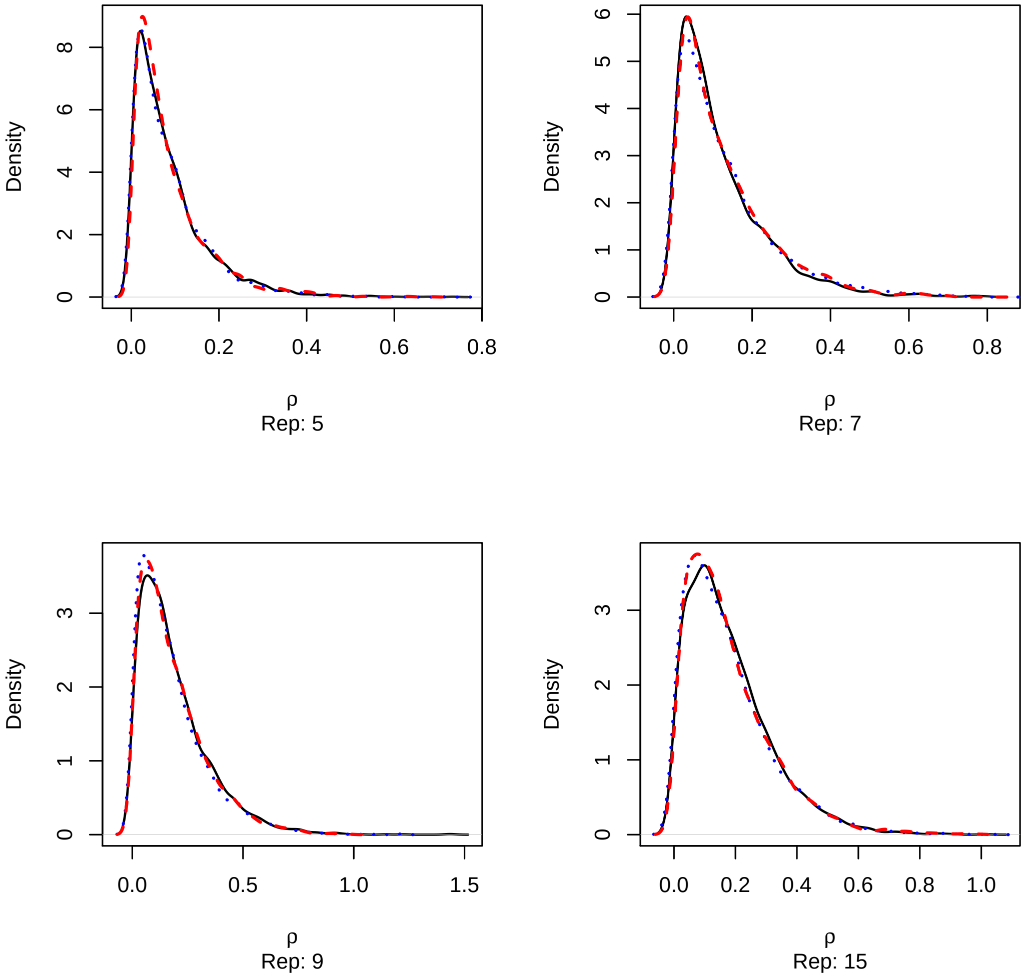


FIGURE 3.10: Estimated posterior density for θ under alternate θ prior distributions for four datasets. The black line corresponds to a $U(0.001, 10)$ prior, the dotted red line to a $U(0.005, 1)$ prior and the dashed red line corresponds to a $U(0.0001, 100)$ prior.

TABLE 3.2: Summary of differences in estimated posterior probabilities of haplotype configurations. The differences compare a run of the sampler using the true haplotype frequencies from the known data to a run using perturbed values of the PHASE estimated population frequencies. Results are pooled across all individuals with uncertain phase in 10 of the 50 datasets.

Frequency Difference	Count
0-0.05	1282
0.05-0.10	23
0.10-0.15	4
0.15-0.20	12
>0.2	4

Six of the eight individuals with differences greater than 15% were in a single dataset, the 29^{th}. For two of these individuals having the same genotypes in this dataset, the large difference could potentially be explained by the underlying haplotype model, as the traceplots showed that mixing was adequate. One of the two individuals is shown in Figure 3.11. From left to right, the bars in this figure correspond to posterior probabilities of haplotype configurations estimated by (i) PHASE and by the sampler using two-locus haplotype frequencies taken: (ii) directly from the true haplotype data (True), (iii) from the population haplotype frequencies estimated by PHASE (Estimated) and (iv) from the perturbed population haplotype frequencies estimated by PHASE (Perturbed). The resulting estimates of the posterior probability of the true configuration are quite similar for all but the sampler using two-locus haplotypes taken directly from the true haplotype data. The estimated proportion for the true configuration is larger when the actual haplotype frequencies from the known data are used. However, the differences in results could also be due to variation of the MCMC sampling, so in the next section we compare results when analyses on each of the ten datasets are replicated.

Multiple runs of the same dataset

Finally, to examine the consistency of results, five additional runs were performed on each of the ten datasets and the haplotype distributions were compared. The initial values, prior distributions and haplotype frequency model for these duplicate runs were all set to the same values that were used for the main runs, as described in Section 3.4.1. However, each of the five runs were started at a different set of initial sequences and tree topology.

For seven of the ten datasets, the sampled haplotype distributions were quite similar across the

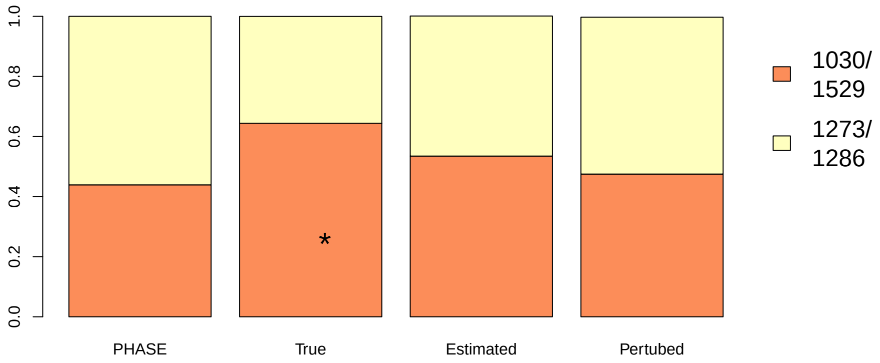


FIGURE 3.11: Barplot of estimated haplotype configurations for individual 10 of dataset 20 under different haplotype frequency models. The column labelled "PHASE" gives the posterior probabilities estimated by PHASE. The column labelled "True" gives the posterior probabilities estimated by the sampler when the two-locus haplotype frequencies from the true data were used. The column labelled "Estimated" gives the posterior probabilities estimated by the sampler when the two-locus haplotype frequencies were obtained from PHASE estimates of the population haplotype frequencies. The column labelled "Perturbed" gives the posterior probabilities estimated by the sampler when the two-locus haplotype frequencies were obtained from PHASE estimates of the population haplotype frequencies then perturbed by a small amount (see text for full detail). The labels of the haplotype configurations are given to the right of the barplot. The binary haplotype sequences are represented in the base-10 number system.

five duplicated analyses. Figure 3.12 shows barplots of the sampled haplotype configurations for the 25 individuals after 40 million iterations for one of these seven datasets. Although there is some variability in the proportions of the sampled haplotypes, the results are remarkably close across the five runs.

Across all ten datasets, six individuals could be found with differing results across the five runs. Four of these six individuals also had different results when the different haplotype frequency models were compared in the previous section. For most of the runs, the traceplots of the sampled configurations for these individuals show that the sampler is mixing poorly and that it tends to get stuck on a single configuration for many iterations. There is typically one or two of the five runs where only a single configuration is sampled with any frequency. This mixing problem would only be obvious when multiple runs were performed.

As with the results on the sensitivity to the haplotype frequency model, dataset 29 appears to have mixing issues with two individuals in particular across the five duplicated analyses. An example individual with results that differed across runs is shown in Figure 3.13. The first and third run primarily sample one main configuration, while the second, fourth and fifth runs spend more time on alternate configurations. From the traceplots we can see that, for all five runs, the sampler does not travel well between these two configurations. These results are based on only 40 million iterations; however, since the sampler can spend more than 10 million iterations at a single configuration, exceptionally long run lengths would be required to be confident that sampling over haplotype configurations was adequate for this dataset.

Finally, in the previous section examining the sensitivity to the haplotype frequency model, individual 10 in dataset 29 had different results when the true haplotypes were used to specify the haplotype frequency model than when other haplotype frequency models were considered, as seen in Figure 3.11. Figure 3.14 shows the sampled haplotype configurations for this individual across the duplicated runs. For the duplicate runs, the two-locus haplotype probabilities, γ, are obtained from the PHASE population haplotype frequency estimates, as in the third column of Figure 3.11. Therefore, the results can be compared. We now use the results across duplicated runs in Figure 3.14 to argue that the differences across haplotype frequency models in Figure 3.11 are due to Monte Carlo error rather than the haplotype model. First, the results are quite consistent across the duplicated analyses and, unlike the PHASE results in the first column of Figure 3.11, the posterior probability of the true configuration is around 0.6 for all five duplicated runs. This value is similar to the posterior probability estimated when the two-locus haplotype probabilities, γ, were

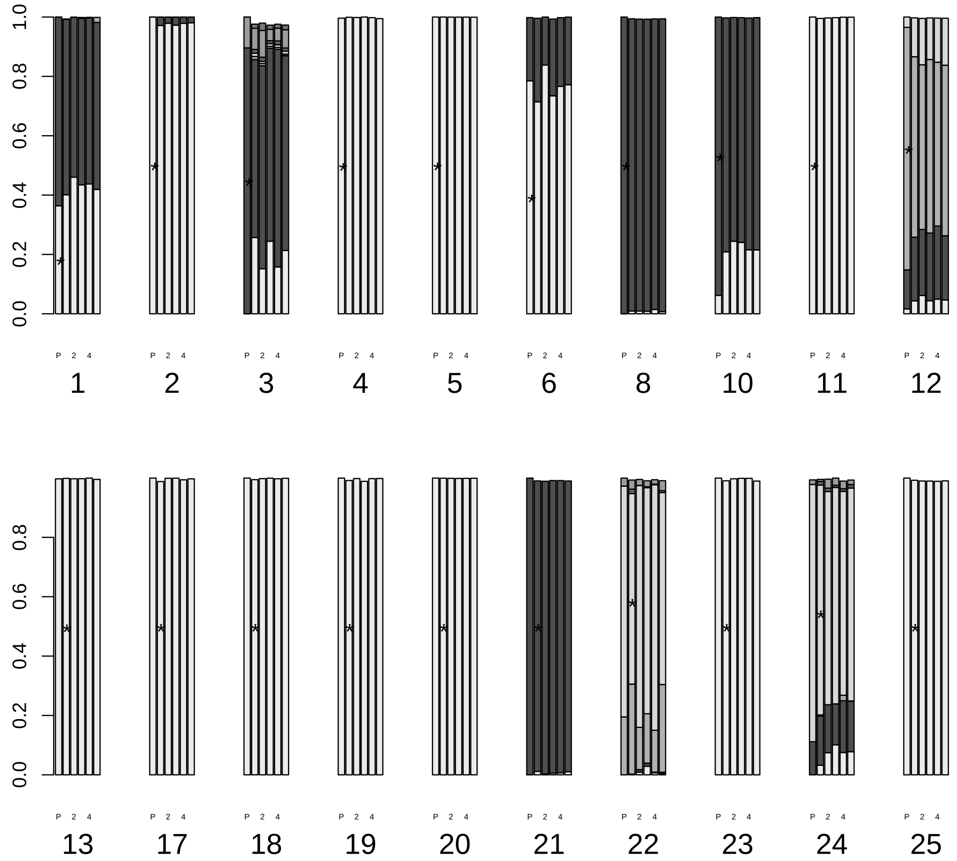


FIGURE 3.12: Barplots of estimated haplotype configurations over five runs on dataset 15. Each set of bars consists of, from left to right, the PHASE-estimated posterior probabilities of haplotype configurations (column ‘P’), and the posterior probabilities estimated by our sampler in the five runs (columns 1 through 5). Only configurations having higher than 1% probability are plotted so bars won’t sum to 1. The true haplotype configuration is marked by an asterisk.

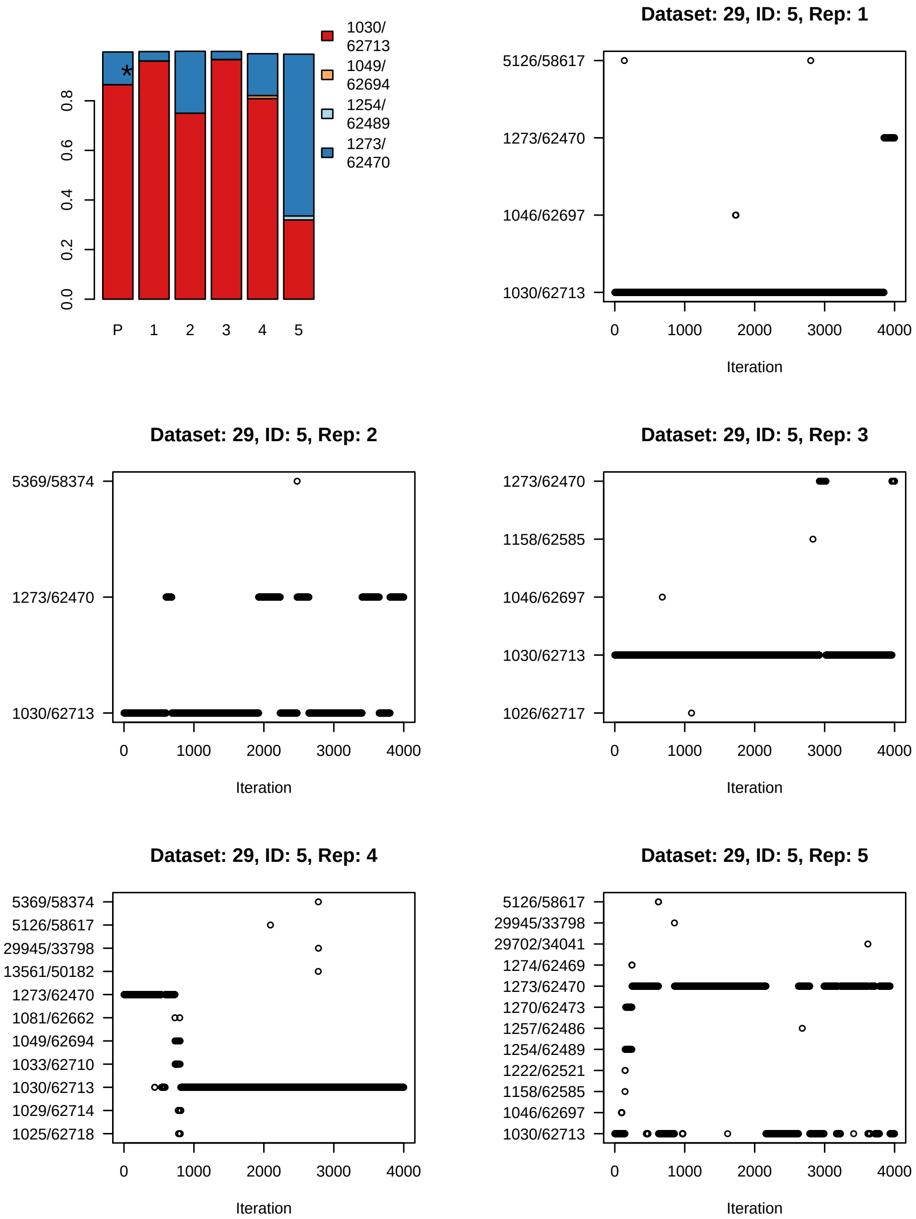


FIGURE 3.13: Barplot of sampled haplotype configurations and the corresponding traceplots for individual 5 over five runs of dataset 29. Each run consisted of 40 million iterations and sample points are recorded every 10,000 iterations. In the barplot, the column labelled "P" gives the posterior probabilities estimated by PHASE; the columns labelled 1 through 5 give the posterior probabilities estimated by the five runs of the sampler. The labels for the two haplotypes in each configuration have been converted from the binary sequence to the base-10 number system.

set to the true known values (Figure 3.11, column 2). In addition, the estimate from the fourth run (column 4 in Figure 3.14) is close to the estimate in the third column of Figure 3.11 and both were estimated under the same haplotype frequency model. Therefore, we attribute the apparent difference in posterior probability estimates between the second and third columns of Figure 3.11 to sampling variability rather than the haplotype frequency model.

3.5 Discussion

In this chapter, an extension to the sampler to deal with missing haplotype data was presented. We described two proposal distributions that update the sequence at the tips for a single individual. The allele-swap is the minimal change possible as it swaps alleles at a heterozygous locus from one sequence to the other. Even though it is a small change, it can have low probability of acceptance because it causes the parental sequence to be different from its offspring at the locus that was swapped. The dictionary-rephaser is a more drastic change to the tip sequences for an individual in that the proposed haplotypes could be completely different from the current haplotypes. However, because the new configuration is chosen from the set of likely configurations given the other tips, it should be similar to the current set of tip haplotypes. Because the sequence is quite different, two topology changes are done to find siblings with similar haplotypes to the proposed haplotypes. Since the major topology update in the haplotype-based sampler has around a 5% acceptance probability even without a sequence change, we can't expect very high acceptance probabilities for the dictionary rephaser, which performs essentially two major topology changes, in addition to the change in tip sequence and the new sequence and recombination variables for nodes ancestral to the sequences that have moved.

We illustrated the use of the sampler with simulated data consisting of 50 datasets of 25 individuals each. In order to assess how well the sampler was working, we ran it for 40 million iterations and compared the consistency of estimates for θ and ρ across datasets. The similarity of the sampled haplotypes to the true haplotypes and of the estimated posterior haplotype distributions to the PHASE-estimated posterior haplotype distributions was also examined to determine how well the sampler performs when there is missing phase. In general, the proportion of sampled configurations for our sampler and PHASE were quite similar. For some individuals in some datasets, PHASE sampled the true configuration more frequently than our sampler; in others, our sampler sampled the true configuration more frequently. Although the distributions of sampled hap-

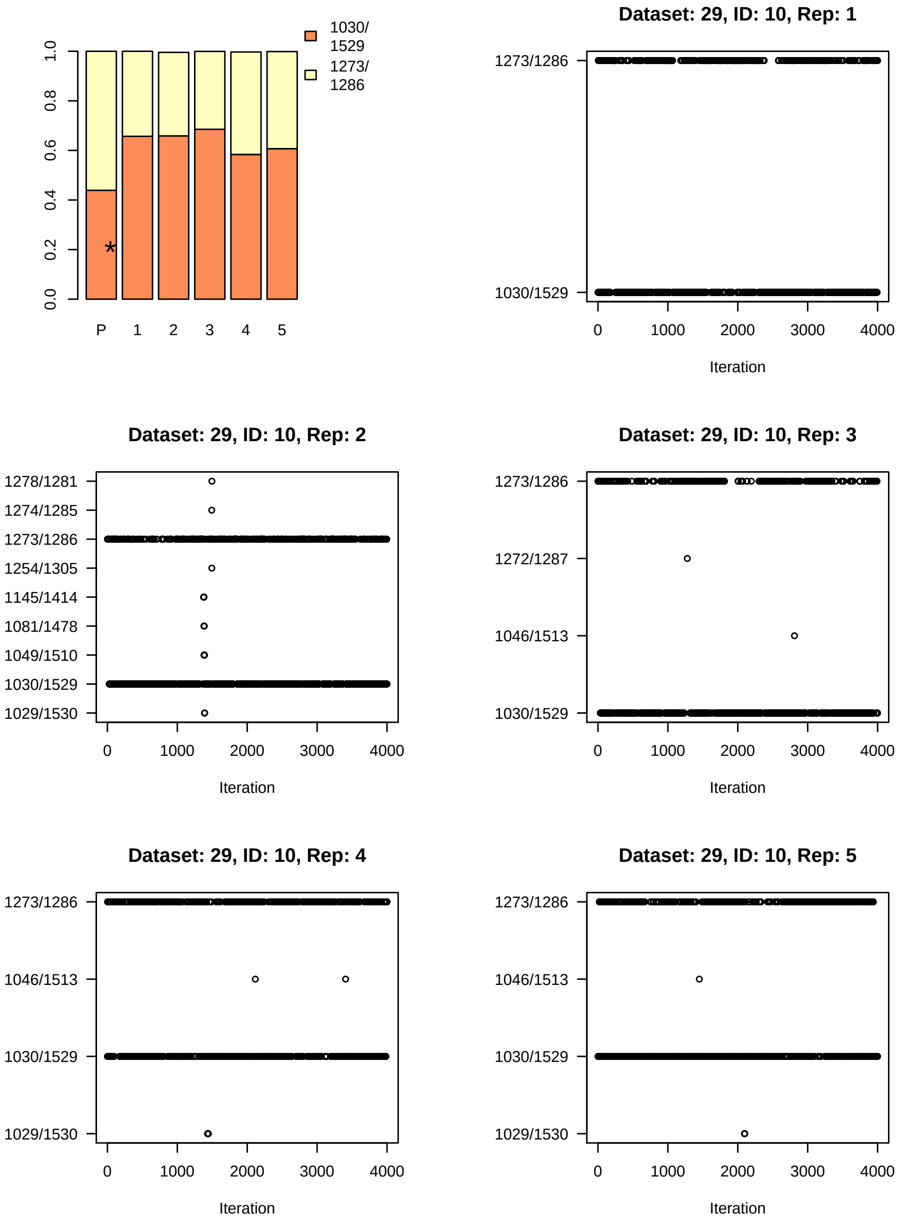


FIGURE 3.14: Barplot of sampled haplotype configurations and the corresponding traceplots for individual 10 over five runs of dataset 29. Each run consisted of 40 million iterations and sample points are recorded every 10,000 iterations. In the barplot, the column labelled "P" gives the posterior probabilities estimated by PHASE; the columns labelled 1 through 5 give the posterior probabilities estimated by the five runs of the sampler. The labels for the two haplotypes in each configuration have been converted from the binary sequence to the base-10 number system.

lotypes are similar, for many individuals the traceplots indicate that our sampler mixes poorly over haplotype configurations. Hence, longer run lengths would be advised for our sampler.

We also performed limited simulations to examine whether results differed significantly when the prior distributions for ρ and θ and the population haplotype frequencies underlying the haplotype frequency model were changed. T here seemed to be little difference in the distributions of sampled ρ and θ values when the prior distributions for each were changed. Some differences were seen in the sampled haplotype distributions, but these could be explained by poor mixing rather than sensitivity to the haplotype frequency model. The sensitivity simulations for the prior distributions of ρ and θ were only performed on 10 datasets, and these datasets were chosen for convenience, because they had short run times, rather than randomly selected. Although their PHASE-estimated posterior probabilities of haplotype configurations did not seem more uncertain than those of the remaining 40 datasets, it is possible that these results will not generalize to all 50 datasets. However, due to time constraints it was infeasible to run the sampler with the different models on all 50 datasets.

Multiple runs of the same dataset were compared to determine if the sampled haplotypes were consistent across runs. For seven of the ten datasets, the distributions of sampled haplotype configurations were similar across the five runs. For three of the ten datasets, the sampler did not adequately sample a second configuration for six individuals. In these cases, the traceplots of the sampled haplotype configurations for these individuals often showed that the haplotypes were mixing poorly. However, in some cases a single configuration was selected and therefore we would not be able to differentiate between poor mixing or an individual with very certain haplotypes. Comparing the frequency of the sampled configurations to those estimated by PHASE can help to differentiate the two cases. Even longer run lengths are justified for these datasets in order to overcome the poor mixing.

Long run lengths for a genealogy sampler are not unexpected. Kuhner (2009) states that genealogy samplers “vary from slow to excruciatingly slow” and recommends budgeting months for the analysis phase. Since the time to complete one million iterations could be as long as 24 hours, increasing the number of MCMC samples for all datasets would be infeasible given the current run times unless we are able to streamline the implementation to improve efficiency. Without such streamlining, it may also be infeasible if the size of the datasets are increased. However, the dictionary rephaser, as currently implemented, is very computationally intensive. First, it can require a large amount of RAM to store the vectors corresponding to all possible haplotype configurations. Second, this update requires recomputing the scores for each of these configurations when the dictionary at the tips has changed, which can be slow if the size of this vector is large.

In general, even if there are many haplotype configurations compatible with a given genotype, the majority of the haplotypes are not actually seen in the general population. Due to the evolution of haplotypes that occurs on the genealogical tree, haplotypes will tend to be similar to other haplotypes. Therefore, instead of enumerating all haplotype configurations, most of which will be extremely unlikely to be observed, an approximation could be programmed that is less computationally intensive. For example, say that for each individual with unknown phase a maximum of 100 configurations will be included in the set of configurations to sample over. In order to focus on the more likely haplotypes, a list of haplotypes estimated to exist in the general population can first be created. For each individual with unknown phase, the configurations in their set can be started with combinations of haplotypes that are likely to actually exist.

For example, say that an individual has genotype (0/1, 0/1, 0/1, 0/1, 0/1). Since there are five heterozygous loci, there are $2^5 = 32$ different configurations compatible with their genotype data. However, say that only the haplotype configurations

$$\begin{gathered} 11100/00011 \\ 11101/00010 \\ 11000/00111 \\ 11001/00110 \end{gathered}$$

are estimated to be present in the population. The set of configurations for this individual can first be populated with these four configurations. Any additional configurations can be added by swapping alleles with the allele-swap update. For example the fourth configuration could produce

$$1\underline{0}001/0\underline{1}110$$

where the loci that had their alleles swapped are underlined.

This approximation to the set of all haplotype configurations could be time consuming to set up at the beginning of the run, but it would presumably reduce the run time for subsequent iterations. Although it is beyond the scope of the thesis to evaluate this version of the dictionary rephaser, there are plans to implement it in the future.

For the most problematic individuals, longer run lengths may not be sufficient to ensure that the haplotype configurations are adequately sampled. It is therefore important to determine which diagnostics are most useful to highlight these datasets. In the examples shown here, the combination of comparing results to the PHASE output and examining traceplots of the sampled configurations would have been enough to determine which datasets were not mixing adequately. Duplicating the analysis was also quite helpful, and if computing resources are available, multiple runs should be tried. It is not unusual that such a strategy would be useful for MCMC samplers that sample trees - it was also recommended in the review by Stephens (2003) and in Zöllner and Pritchard (2005) to assess convergence.

As a final caution however, as with all MCMC samplers, not all convergence problems will be evident from diagnostic plots or statistics. In our application in particular, for some individuals the haplotype configurations do not mix easily and the sampler can spend a good proportion of the run at a particular configuration. For this reason, alternative MCMC schemes that were developed to improve mixing are potentially beneficial. In the next section, we describe our experiences with simulated tempering as a means to improve mixing of the sampler.

Chapter 4

Simulated tempering to improve mixing of the genealogy sampler

When the target distribution is multi-modal or high-dimensional, convergence to the target distribution may be exceptionally slow with MCMC methods. In these cases, it may be difficult to reach all regions of the state space having high posterior probability, as reaching such states may require first accepting moves to states having extremely low probability. Moves to states having low posterior probability will rarely be accepted. Therefore, the mixing of the sampler will be poor and convergence to the target distribution will be slow. Longer run lengths may overcome some of the mixing issues; however, for some MCMC algorithms, the run lengths may need to be so long that the approach becomes impractical.

Concerns about slow mixing are especially relevant to genealogy samplers, as there is strong dependence between the tree structure and the latent data at the nodes of the tree. We saw evidence of slow mixing with our genotype-based sampler. In Section 3.4, for some individuals in some datasets, a new haplotype configuration was accepted in approximately one of every 10 million MCMC iterations. The time to complete one million iterations was quite variable and could be as long as 24 hours. In order to ensure adequate mixing of the haplotype configurations on one of these datasets, the sampler might need to be run for many weeks or months.

Extensions to the standard MCMC approach, such as Metropolis-coupled MCMC (Geyer, 1991) and Simulated Tempering (Marinari and Parisi, 1992; Geyer and Thompson, 1995), were proposed in order to improve mixing. In this chapter, we focus on simulated tempering, but both simulated tempering and Metropolis-coupled MCMC (MCMCMC) involve the introduction of a temperature parameter to the MCMC scheme. In simulated tempering, the state space is augmented by a discrete temperature variable, which is also updated by the algorithm. The temperature variable corresponds to the index of a parameter of a probability distribution. At the "cold" temperature, sampling is from the distribution of interest. If the heated distributions are well chosen, when the temperature is "hot", sampling is from a distribution having state space that is more easily explored by the MCMC algorithm than the target distribution. Therefore, by cycling between hot and cold temperatures the sampler is able to reach all states of interest.

In this chapter, we describe our experiences using simulated tempering to improve mixing of the sampler. Since the recombination rate can be used to control the dependence between the tip haplotypes and the topology, our heated distributions involve modifying the distribution for the recombination rate so that very high recombination rates are sampled. In the next section, we provide background information on simulated tempering and contrast it to MCMCMC as MCMCMC is more commonly used to improve mixing of MCMC samplers. We then describe our approach for incorporating simulated tempering into the sampler. Our implementation of simulated tempering applies to both the haplotype and genotype-based samplers; however, since poor mixing of the haplotype configurations was observed for some of the genotype datasets, we focus the detail in this chapter on the genotype-based sampler. We apply the simulated tempering version of the genotype-based sampler to one of the problematic simulated genotype datasets examined in Section 3.4.4. Finally, we briefly describe our experiences and thoughts on other approaches to specifying the heated distributions.

4.1 Background on simulated tempering

General extensions to improve mixing and convergence of MCMC samplers have been proposed. In particular, two papers describe related approaches that use "heating" to improve mixing (Geyer, 1991; Geyer and Thompson, 1995). In both approaches, a series of m unnormalized densities, labelled $f_i(\mathbf{x})$ are chosen and defined on the same state space. The density $f_1(\mathbf{x})$, the cold distribution, is the original distribution that we wanted samples from and $f_m(\mathbf{x})$ is the hot distribution. The hot distribution is chosen so that all regions of the state space can be reached easily using MCMC. The data are augmented to account for these m distributions. With MCMCMC (Geyer, 1991), the data are augmented by $m-1$ copies of the variable; that is, $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m)$ on the state space

$\mathcal{X} \times \cdots \times \mathcal{X}$, and the joint distribution is proportional to

$$\prod_{i=1}^{m} f_i(\mathbf{x}_i).$$

Each component of $\mathbf{x}$ is updated and periodically a swap update that proposes swapping $\mathbf{x}_i$ with $\mathbf{x}_j$ is performed. With simulated tempering (Geyer and Thompson, 1995; Marinari and Parisi, 1992), the data are augmented by the temperature index, I, where $I = \{1, \ldots, m\}$. The state space of the augmented data, $(\mathbf{X}, I)$, is $\mathcal{X} \times \{1, \ldots, m\}$ and the unnormalized joint distribution is

$$f(\mathbf{x}, i) = \pi(i) f_i(\mathbf{x}), \tag{4.1}$$

where $\pi(i)$ is called the pseudoprior of I. The MCMC sampler with simulated tempering therefore has stationary distribution given in equation (4.1). The conditional distribution for $\mathbf{x}$ given $I = 1$ is the cold distribution, $f_1(\mathbf{x})$. Therefore, to obtain the sample from the distribution of interest, that is, the cold distribution, we run the chain and save only states of $\mathbf{X}$ sampled when $I = 1$.

In terms of which approach might be better suited to our sampler, each has advantages and disadvantages. MCMCMC only requires that the temperatures of the heated distributions be chosen. It does not require that the pseudoprior be specified. However, in MCMCMC, m copies of the data vector must be created, which requires more storage. With simulated tempering, extending our sampler to also sample I is straightforward. However, in addition to the choice of temperatures, simulated tempering requires that the pseudoprior be specified. Simulated tempering may perform better: Geyer and Thompson (1995) state that simulated tempering mixes better than MCMCMC. It has also been shown to be more efficient theoretically, provided the pseudoprior is well chosen (Zheng, 2003; Atchadé et al., 2010). We chose to use simulated tempering since it could more easily be adapted into our existing sampler and, theoretically, has improved performance over MCMCMC.

Details for implementing simulated tempering are now given. As mentioned, simulated tempering, as described in Geyer and Thompson (1995), involves sampling over a set of unnormalized distributions $f_i(\mathbf{x})$, $i = 1, \ldots, m$, where I is the temperature index and the cold distribution $f_1(x)$ is the original distribution that we wanted to sample from. Sampling is now over the pair $(\mathbf{X}, I)$; however, in practice, the components are updated separately. A Metropolis-Hastings or Gibbs update is used to update $\mathbf{X}$ for a fixed value of $I = i$. At the t^{th} iteration, the temperature index, I, is updated with proposal distribution

$$q_I(I^{(t)} = j | I^{(t-1)} = i) = \begin{cases} 1 & i = 1; j = 2 \\ 1 & i = m; j = m - 1 \\ 1/2 & i = 2, \ldots, m; j = i \pm 1 \end{cases} \tag{4.2}$$

The Metropolis-Hastings acceptance probabilities are

$$\min(1, \frac{f_j(\mathbf{x})\pi(j)q_I(i|j)}{f_i(\mathbf{x})\pi(i)q_I(j|i)}) \tag{4.3}$$

for updates from $I = i$ to $I = j$ and

$$\min(1, \frac{f_i(\tilde{\mathbf{x}})q_X(\mathbf{x}|\tilde{\mathbf{x}})}{f_i(\mathbf{x})q_X(\tilde{\mathbf{x}}|\mathbf{x})}), \tag{4.4}$$

for updates from $\mathbf{x}$ to $\tilde{\mathbf{x}}$, where $q_X(\tilde{\mathbf{x}}|\mathbf{x})$ is the proposal distribution that updates $\mathbf{X}$.

In order to implement simulated tempering, the heated distributions must be chosen. If a bad choice of heated distributions is made, simulated tempering can fail (Geyer and Thompson, 1995). “Powering up”, where the unnormalized density for the distribution of interest is raised to incrementally decreasing powers, is a common form of heating. Raising the density to a power that is less than one can have the effect of flattening the modes of the distribution, essentially making the heated distributions increasingly like a uniform distribution. However, this form of heating does not work for all densities: the “witch’s hat” distribution from Geyer and Thompson (1995) provides an example where powering up would fail.

If the hot distribution is chosen so that independent sampling is possible, then the chain is regenerating (see, for example, Ripley (1987)). Regeneration is not required for simulated tempering; however, Geyer and Thompson (1995) state that it is this aspect of simulated tempering that provides all the mixing and therefore it is safer to use regenerating samplers. Other authors do not make regeneration a goal of simulated tempering. For example, in their version of simulated tempering, Marinari and Parisi (1992) do not use regeneration and Neal (1996) questions whether there is any advantage to regeneration in simulated tempering. Liu and Sabatti (1999) note that, in general, simulated tempering can force the Markov chain to spend too much time in regions with low target probabilities and that, in doing so, the advantage of MCMC over independent Monte Carlo sampling will be lost. Therefore, a simulated tempering scheme that actually performs independent sampling at hot temperatures might be expected to spend too much time sampling unlikely states due to the algorithm's construction.

Simulated tempering also requires specification of the pseudoprior and temperatures. The temperatures should be chosen so that the algorithm accepts temperature moves often. Atchadé et al. (2010) show that acceptance rates of 0.234 for temperature moves are optimal for a specific form of target distribution and heating by powering up. A range of 20 to 40% is given by Geyer and Thompson (1995).

To ensure that the sampler is able to traverse the temperatures, the ideal pseudoprior would spend roughly an equal amount of time sampling each distribution. That is, the marginal distribution for I,

$$\Pr(I = i) \propto \int_{\mathcal{X}} \pi(i) f_i(x) dx = \pi(i) \int_{\mathcal{X}} f_i(x) dx \tag{4.5}$$

has value $1/m$ for all $i = 1 \ldots m$. This is achieved if

$$\frac{1}{m} \propto \pi(i) c(i),$$

where $c(i) = \int_{\mathcal{X}} f_i(x) dx$ is the normalizing constant. Thus, the ideal $\pi(i)$ is proportional to the inverse of the normalizing constant, which will not be known in most applications requiring MCMC.

Geyer and Thompson (1995) give *ad hoc* methods for selecting the pseudoprior and temperatures. To choose the pseudoprior and temperatures for the next run from the current run, these methods use two diagnostics: (a) the amount of time spent in each distribution – that is, the occupation numbers $o(i)$, and (b) the average acceptance rates, a_i, for updates between the i^{th} and $i + 1^{th}$ temperature. For example, if the acceptance rate for temperature transitions, a_i, is too low, the temperatures are too far apart. If the occupation numbers $o(i)$ are very unequal, the sampler is not mixing well and a better pseudoprior is needed. In particular, they start the tuning process in the hot end with a few distributions, presumably, because the improvements in mixing come from the hotter temperatures and effective tuning requires a simulated temperating sampler that mixes well. They first use the Robbins-Munro stochastic approximation method (Robbins and Monro, 1951), a stochastic optimization approach, to get an initial estimate of the pseudoprior. They re-run the algorithm with the new pseudoprior and, in subsequent runs, adjust the temperatures, as described below. After an adjustment to the temperatures, an estimate of the pseudoprior for the new temperatures is found using cubic spline interpolation. The adjustment of the temperatures/pseudoprior proceeds iteratively until the occupation numbers are approximately uniform and the acceptance rates for temperature transitions are in the correct range. Colder temperatures are added when the sampler is traversing the current temperatures well, and the tuning process begins again.

The pseudoprior can be adjusted by using equation (4.5) to estimate $1/c(i)$ up to a constant. Specifically, since $\Pr(I = i) \propto o(i)$, from equation (4.5) and the definition of $c(i)$, we have

$$\frac{1}{c_i} \propto \frac{\pi(i)}{o(i)}.$$

Hence, at the t^{th} iteration of the tuning process, let the pseudoprior for the $t + 1^{th}$ iteration be

$$\pi^{(t+1)}(i) \propto \pi^{(t)}(i)/o^{(t)}(i), \tag{4.6}$$

where $o^{(t)}(i)$ are the observed occupation numbers from the t^{th} run and $\pi^{(t)}$ is the pseudoprior used for the t^{th} run. Note that the choice of pseudoprior values will affect the transition probabilities for temperature moves since they are included in the Metropolis-Hastings ratio for temperature updates (equation 4.3)

The temperatures are adjusted by modeling the average acceptance rates a_i between temperatures i and $i + 1$ as

$$a_i = \exp(-\int_{\lambda_i}^{\lambda_{i+1}} b_i ds), \tag{4.7}$$

where b_i is an unknown step function assumed to be constant between the λ_i. For the current temperature values $\lambda_i^{(t)}$ and current average acceptance rates $a_i^{(t)}$, which are found by averaging the acceptance rate for transitions from λ_i to λ_{i+1} and the acceptance rate for transitions from λ_{i+1} to λ_i, $b_i^{(t)}$ is estimated by solving equation (4.7) for b_i:

$$b_i^{(t)} = \frac{1}{\lambda_{i+1}^{(t)} - \lambda_i^{(t)}} \log \frac{1}{a_i^{(t)}}.$$

The estimate $b_i^{(t)}$ is then plugged into equation (4.7) and the $\lambda_i^{(t+1)}$ are chosen so that the acceptance rate between temperatures is α. As an example, assume that all $\lambda_i^{(t)}$ with $i \leq j$ give acceptance rates in the correct range; therefore, $\lambda_i^{(t+1)} = \lambda_i^{(t)}$ for $i \leq j$. In addition, assume that $a_j^{(t)} < \alpha$. Then, $x = \lambda_{j+1}^{(t+1)}$ is estimated by solving

$$\alpha = \exp(-\int_{\lambda_j^{(t+1)}}^{x} b_j^{(t)} ds)$$

for x, which gives

$$x = (\lambda_{j+1}^{(t)} - \lambda_j^{(t)}) \frac{\log \alpha}{\log a_j^{(t)}} + \lambda_j^{(t)}.$$

Since we assumed that $a_j^{(t)} < \alpha$, the estimate for $\lambda_{j+1}^{(t+1)}$ is less than $\lambda_{j+1}^{(t)}$, which is as desired: if the acceptance rate between two temperatures is too low, the next set of temperatures should be closer together. Although no explanation is given for this model, Geyer and Thompson (1995) found it performed well but tended to over-correct the temperatures. For example, if a lower acceptance rate was desired between two temperatures, the acceptance rate after a temperature adjustment would be too low.

This particular method is not the only *ad hoc* approach for determining a good pseudoprior and temperatures. For example, in powering up versions of heating, using geometric spacing for the temperatures seems to be common. Neal (1996) used trial-and-error to select the number of temperatures and the pseudoprior, and geometric spacing of temperatures in an example comparing simulated tempering to his tempered transitions approach. In their description of an approach to estimate the normalizing constants for Boltzmann-type distributions (of the form $f(x) = \frac{1}{Z} \exp -H(x)/\tau$), Liang (2005) mention the tuning approach outlined in Geyer and Thompson (1995), but also state that trial-and-error is the most promising approach to tuning for hard problems, that it is "non-automatic" and "time consuming" and that the resulting estimates are often not accurate. Atchadé et al. (2010) suggest a temperature selection scheme for MCMCMC that would also apply to the selection of temperatures for simulated tempering. They propose starting at the cold temperature and iteratively adding temperatures such that the acceptance probability for adjacent temperature moves is approximately 0.23.

Finally, although many authors note that the pseudoprior must be accurately estimated for simulated tempering to perform well (for example, Jasra et al. (2007)), they did not provide examples of the poor performance of simulated tempering when the pseudoprior is poorly estimated. On the other hand, Geyer and Thompson (1995) note that the sampler will mix faster if the pseudoprior approximates the inverse of the normalizing constants, but that the pseudoprior need not be estimated with high precision and that the sampler will have the correct stationary distribution for any strictly positive pseudoprior.

In the next section, we describe an implementation of our sampler that incorporates simulated tempering. For this version, we assume that the distribution of interest has a fixed and known value for the recombination rate, ρ. Heating then involves specifying warmer values for ρ. We also describe the strategies that we used for choosing the temperatures and pseudoprior when we applied simulated tempering to an example dataset.

4.2 Using the recombination rate to specify heated distributions

Geyer and Thompson (1995) give a statistical genetic example that involves sampling genotypes on an extended pedigree in order to estimate the probabilities that certain members of the pedigree are "carriers" of the allele that causes cystic fibrosis. The latent variables they sample are the genotypes at the locus that causes cystic fibrosis, on pedigree members with unknown genotype. The data they condition on is cystic fibrosis disease status, which is known for some pedigree members. Cystic fibrosis is known to be recessive, which means that an individual requires two copies of the mutated allele in order to have the disease. Therefore, the *penetrance model*, that is, the probability of disease given the genotype, is known. Let the allele for cystic fibrosis be 'a', the normal allele be 'A', and disease status 'D' be 1 if the individual has cystic fibrosis and 0 otherwise. The "cold" penetrance model is

$$\begin{aligned} \Pr(D=1|G=A/A) &= 0 \\ \Pr(D=1|G=A/a) &= 0 \\ \Pr(D=1|G=a/a) &= 1. \end{aligned} \tag{4.8}$$

In addition, historically, cystic fibrosis was a lethal disease in childhood; therefore, even if disease status of a pedigree member is not known, if they had children, then they could not have had cystic fibrosis. Their genotype at the cystic fibrosis locus must either be homozygous for the normal allele or they are a *carrier* (i.e., they are heterozygous).

In this example, heating involves altering the penetrance model given in equation (4.8). They considered two forms of heating but we will discuss only one. In their "gene-drop" form of heating, the penetrance model for the hot distribution is uniform. That is, the probability of disease is the same regardless of genotype and therefore the observed data on disease status gives us no information about the latent genotype. The genotypes can be sampled for the founders then 'dropped' to descendants using Mendelian inheritance. The warm distributions at temperature λ are convex combinations of the hot and cold penetrance models.

By analogy, in our approach, we can also motivate our hot distribution as the distribution of the latent genealogy unconditional on the observed data on genotypes. The distribution for the topology and node times unconditional on the genotype data is the standard neutral coalescent (see Section 1.1.1). In particular, in the standard neutral coalescent model all topologies are equiprobable, which can be achieved by forcing the recombination rate, ρ, to be large. If ρ is made so large that we are certain of a recombination event between the focal point and its adjacent markers in all tip branches, then the observed data gives us no information about the topology. High values of ρ mean less sequence is passed to the present from the ancestors on the tree. Thus, for large ρ, the genotypes at the observed markers are all inherited from other ancestors than those on the tree of the focal point.

To encourage sampling of large ρ values, we might consider altering the prior distribution for ρ but this would not be in strict analogy to our motivating pedigree example. In this example, the penetrance values in equation 4.8 are fixed and known. In contrast, we assume that the true value of ρ is not known and our sampler provides an estimate of it. Therefore, to implement simulated tempering, we make the simplifying assumption that ρ is known and fix the value for the cold distribution. The heated distributions, f_i, are then the same as the target distribution except that they have a higher value, ρ_i, for the recombination rate. The drawback to this approach is that we must assume that ρ is known; in Section 4.3, we discuss our experiences when we instead heat on the parameter of the prior distribution for ρ.

Although we make the assumption of known ρ at the target distribution in order to test simulated tempering, this is not necessarily a bad approach in general. First, as mentioned in Section 2.3.1, estimates of the recombination rate across the genome are available and could be used to specify ρ for the target distribution. Second, assuming known values for parameters that are truly unknown has been a frequent strategy used in genetic applications to handle complex problems. For example, in their approach to estimate the recombination rate, McVean et al. (2002) fix the mutation rate at a value that is estimated based on the observed data. In parametric linkage analysis, which aims to localize trait-influencing mutations using pedigree data, aspects of the model are often also fixed in advance, in particular the penetrance model and the marker allele frequencies in the founder individuals, even though these are often not known with certainty. Finally, for our sampler, we assumed that the historical two-locus haplotype frequencies, γ, are known and we used estimates from the data to specify their values.

We now describe how simulated tempering was implemented. Let the i^{th} recombination rate be

ρ_i. The i^{th} heated distribution is

$$f(\mathbf{A}|\mathbf{G}, \rho_i) = \Pr(\mathbf{G}|\mathbf{S}) \Pr(\mathbf{S}|\mathbf{R}, \boldsymbol{\Omega}, \tau, \theta) \Pr(\mathbf{R}|\boldsymbol{\Omega}, \tau, \rho_i) h(\boldsymbol{\Omega}) \Pr(\tau) h(\theta). \tag{4.9}$$

We do not change the proposal distributions for the heated distributions; the rationale behind these distributions is not changed by the changes to ρ_i. However, the proposal distribution that updates ρ is no longer required. To update the temperature, we use the proposal distribution given in equation (4.2). The proposal from temperature $I = i$ to $I = j$ is accepted with probability

$$\begin{aligned}\alpha(i,j) &= \min(1, \frac{\Pr(\mathbf{G}|\mathbf{S}) \Pr(\mathbf{S}|\mathbf{R}, \boldsymbol{\Omega}, \tau, \theta) \Pr(\mathbf{R}|\boldsymbol{\Omega}, \tau, \rho_j) h(\boldsymbol{\Omega}) \Pr(\tau) h(\theta) \pi(j) q_I(I = i|I = j)}{\Pr(\mathbf{G}|\mathbf{S}) \Pr(\mathbf{S}|\mathbf{R}, \boldsymbol{\Omega}, \tau, \theta) \Pr(\mathbf{R}|\boldsymbol{\Omega}, \tau, \rho_i) h(\boldsymbol{\Omega}) \Pr(\tau) h(\theta) \pi(i) q_I(I = j|I = i)}) \\ &= \min(1, \frac{\Pr(\mathbf{R}|\boldsymbol{\Omega}, \tau, \rho_j) \pi(j) q(i|j)}{\Pr(\mathbf{R}|\boldsymbol{\Omega}, \tau, \rho_i) \pi(i) q(j|i)}).\end{aligned}$$

For very large ρ, the probability distribution for the r values will be concentrated on $r = 1$. When $r = 1$, the sequence is not passed from the parent to the offspring nodes, and therefore the sequence at internal nodes is '-' with probability 1. Since the sequence and recombination terms $\Pr(\mathbf{S}|\mathbf{R}, \boldsymbol{\Omega}, \tau, \theta)$ and $\Pr(\mathbf{R}|\boldsymbol{\Omega}, \tau, \rho_i)$ will all be 1 when the r values for internal nodes approach 1, equation (4.9) simplifies to

$$\Pr(\mathbf{A}|\mathbf{G}, \rho_m) \approx h(\boldsymbol{\Omega}) \Pr(\tau) h(\theta) \prod_{i=1}^{n} \Pr(\mathbf{s}_{i_1}, \mathbf{s}_{i_2}|\gamma).$$

Therefore, at very large ρ, the sequences and the topologies are effectively independent. Updates to the tip sequences will be more likely to be accepted since they will not be limited by the topology.

With the prospect of increased acceptance of sequence updates, we do not use the dictionary rephaser update with the simulated tempering version of the sampler. As discussed in Section 3.2.3, the dictionary rephaser requires that the probability distribution for sampling a new haplotype configuration be recomputed when the tip sequences are changed. As a result, if the tip sequences change frequently, many more computations are done and the sampler becomes too slow. Therefore, for the simulated tempering version of the sampler, the allele-swap provides all of the haplotype configuration mixing.

Typical descriptions of simulated tempering propose alternating between an update to $\mathbf{X}$ and an update to the temperature I. For our sampler with simulated tempering, we have five proposal distributions to update the augmented data $\mathbf{A}$. If we alternate between an update to $\mathbf{A}$ and an update to I, there will be little opportunity for $\mathbf{A}$ to be affected by ρ_i before it is changed to ρ_j. We have therefore chosen to treat the updates to I as an additional proposal distribution, the sixth, that is chosen with some probability p_I. By default, we let the probability of selecting the update to I be 30%.

As mentioned in the previous section, a general criticism of simulated tempering is that too much time might be spent exploring regions of low posterior probability, which correspond to sampling low probability haplotype configurations in our context. Requiring independent sampling at the hottest temperatures might exacerbate this effect. We will therefore not require that ρ is so large that independent sampling is achieved. Although we cannot provide any guarantees about the adequacy of mixing, satisfactory improvements to haplotype sampling might be seen with more moderate ρ values.

An additional benefit with increasing the ρ values for the heated distributions is that at hotter temperatures the computation time for updates to $\mathbf{A}$ will theoretically decrease as less sequence material is updated at the internal nodes. During the testing phase of implementation, shorter run times were seen with very large ρ values ($\rho \approx 4$). However, detailed testing of run times at moderate ρ values has not been done.

4.2.1 Simulated tempering to improve mixing of a simulated genotype dataset

To test this version of simulated tempering, we use the most problematic genotype dataset from the simulated data presented in Section 3.4. Barplots of the estimated posterior probabilities of the haplotype configurations are shown in Figure 4.1. For two individuals in this dataset, 5 and 20, the haplotype configurations were not mixing well. The estimated posterior probabilities of the haplotype configurations for each individual were quite different across five runs. Traceplots of sampled haplotype configurations for individual 5 were shown in Figure 3.13 and show that updates to the haplotype configuration are only accepted about once every ten million iterations on average. We therefore use the frequency with which these haplotype configurations are updated to assess the performance of simulated tempering. We continue to add heated distributions until the acceptance rates of the haplotype configuration updates are reliably non-zero over the tuning run of up to 2.5 million iterations.

For the distribution that we actually want samples from, the cold distribution, we assume a fixed and known $\rho = 0.0004$. This is the true value used to simulate the data. To find heated distributions for this dataset, we started with five distributions: the cold distribution and four distributions with gradually increasing ρ values. We iteratively updated the pseudoprior and temperatures, as described below. When the results on the tuning output were satisfactory, we ran the sampler again and examined the sampled haplotype configurations for the problematic individuals. If no improvement was seen, we increased the number of distributions by increasing the ρ values and completed the process again.

Temperatures were tuned so that the average acceptance rates for temperature transitions between adjacent temperatures was around 0.4, although acceptance rates as low as 0.3 and as high as 0.5 were acceptable. This is outside the range of 0.2-0.4 that was suggested by Geyer and Thompson (1995) and the optimal rate of 0.234 found by Atchadé et al. (2010). However, in initial tuning attempts it was difficult to ensure that all temperatures were visited. Therefore, higher acceptance rates seemed preferable to acceptance rates that were too low. After an adjustment to the temperatures, cubic spline interpolation was used to select their pseudoprior values. If the interpolation yielded a negative value for $\pi(i)$, we set $\pi(i) = \pi(i+1)/2$.

If the transition from i to $i+1$ has higher probability than the transition from $i+1$ to i, or vice versa, the average of the two acceptance rates, a_i, is not as useful as a tuning diagnostic. For example, if the acceptance rate for transitions up is 0.1 and transitions down is 0.98, $a_i = 0.54$, which is above the tolerance set. The tuning procedure, as described on page 130, would make the two temperature values further apart in order to decrease this acceptance rate. However, with the transition rate for moves up being so low, this tuning decision will only make the probability of moving up even smaller. We therefore also required that the difference between the transition rates for moves up versus moves down not be greater than 0.1. Recall from equation (4.3), the acceptance probability for temperature updates depends on the ratio of the pseudoprior values so that an adjustment to the pseudoprior will influence these transition probabilities. In addition, note that the occupation numbers at each temperature will also not be equal if the transition probabilities for moves up and down are not equal. Therefore, an adjustment to the pseudoprior using equation (4.6) should also equalize the transition rates for moves up in temperature and moves down in temperature.

The *ad hoc* tuning procedure of Geyer and Thompson (1995) described in Section 4.1 was attempted to select the temperatures and the pseudoprior; however, in our context it did not perform well. First, even after a pseudoprior and/or temperature adjustment using equations (4.6) and (4.7), the occupation numbers in the next iteration were often very non-uniform, indicating that the sam-

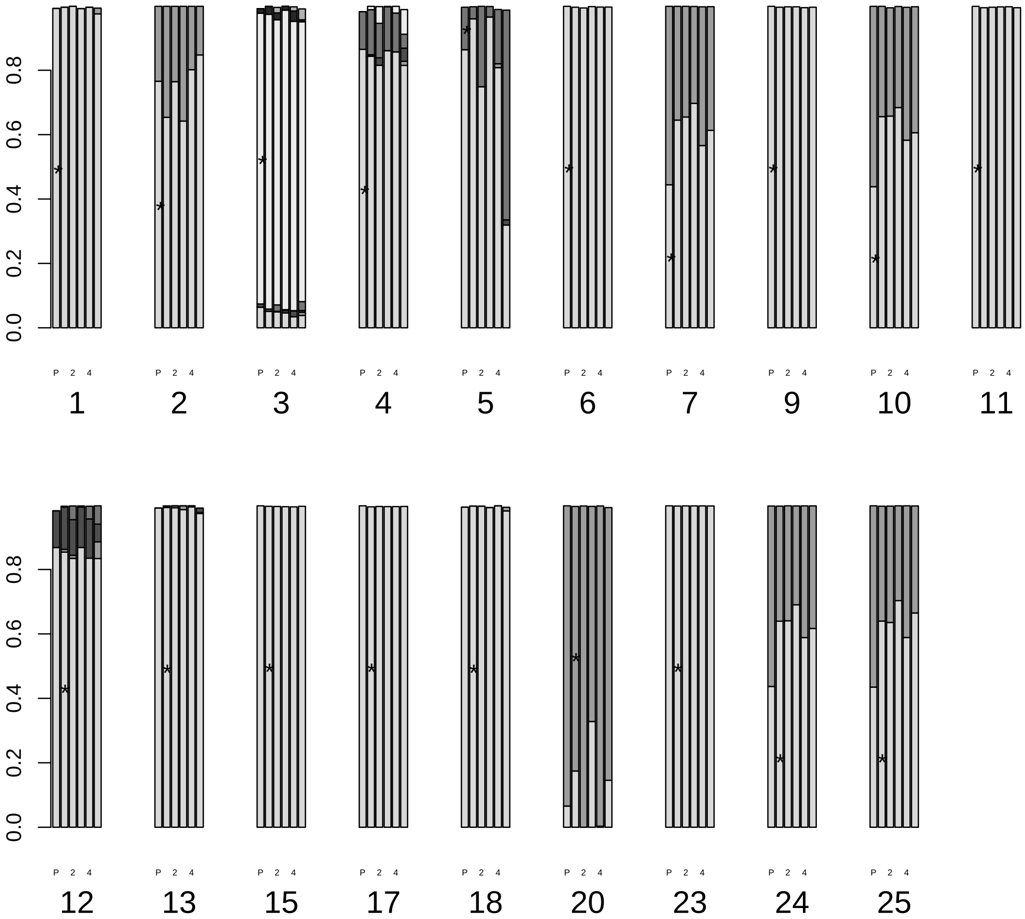


FIGURE 4.1: Barplots of estimated posterior probabilities of haplotype configurations for dataset 29. Each set of bars consists of the PHASE-estimated posterior probabilities, marked 'P', and the same probabilities estimated in each of five duplicate runs of the sampler. Only configurations sampled more than 1% of the time are plotted so bars will not sum to 1. The true haplotype configuration is marked by an asterisk.

pler was not mixing well. Before adjusting the temperatures, we would attempt to improve mixing by adjusting the pseudoprior using equation (4.6). However, selecting the next pseudoprior using equation (4.6) tended to result in extreme over-corrections. For example, if in the current run the sampler spent too much time at the hottest temperatures, in the next run it would spend too much time at the coldest temperatures. The over-corrections became more frequent as warmer temperature distributions were added. The challenges with tuning the pseudoprior using equation (4.6) are at least partly due to the run length of the tuning iterations. During the tuning stage, it is not feasible to run the chain long enough for the t^{th} tuning run to get the accuracy required for good estimation of the pseudoprior for the $(t+1)^{th}$ run. Therefore, the resulting occupation numbers observed after the $(t+1)^{th}$ run are not necessarily an improvement over those observed at the t^{th} run. Finally, using equation (4.7) to tune the temperatures also tended to result in temperature over-corrections, particularly once hotter temperatures were added. Geyer and Thompson (1995) also observed over-corrections using equation (4.7) and therefore this phenomenon may be explained by the procedure itself. When the sampler appeared to be cycling between over-corrections, which happened more often when many hotter temperature distributions had been added, trial-and-error was instead used to select better values.

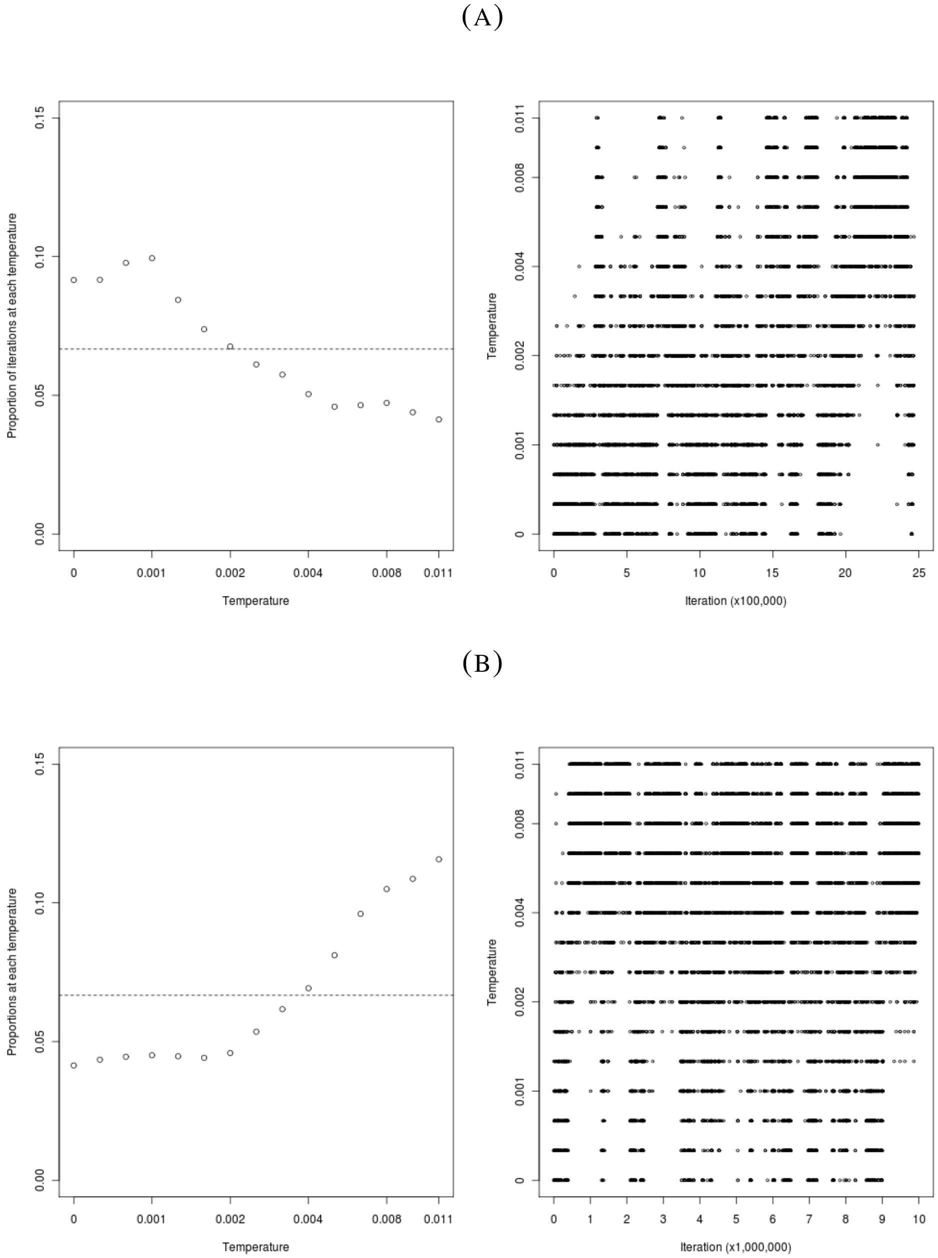
During our *ad hoc* tuning procedure, as new heated distributions were added, we increased the number of MCMC iterations in order to improve the estimates of the occupation numbers and acceptance probabilities for temperature transitions. The maximum number of MCMC iterations was set to 2.5 million, which generally took over six hours to complete. Therefore, tuning the last set of temperatures took the longest amount of time as only two or three iterations of the tuning process could be completed in a day. In total, approximately four weeks were required to find a pseudoprior and temperatures that performed adequately for this dataset; however, for the majority of this time, the sampler was running and only periodic monitoring was required to make tuning decisions.

When the tuning process was terminated, a total of 15 temperature values and the corresponding pseudoprior had been selected. For these values, the occupation numbers and acceptance rates were within the appropriate range. The hottest ρ value was 0.011, which was nearly 100 times the ρ of the cold distribution. After tuning was complete, two additional runs were completed with the same temperature/pseudoprior values. The first run of the sampler had a total of 2.5 million MCMC samples in total. The second run had 10 million iterations and was completed in 34 hours. Although the occupation numbers were not as uniform as seen during the final tuning run, adequate time was spent at each of the temperatures. In Figure 4.2, the plots on the left show the proportion of time spent at each temperature for the two post-tuning runs. In the first run consisting of 2.5 million iterations, shown in panel (A), too much time is spent at the cold temperature; however, in the second run consisting of 10 million iterations and shown in panel (B), the reverse is true. The traceplots of the temperature over time are given in the plots on the right of Figure 4.2. They confirm that the sampler cycles well between all temperatures.

As the temperature, ρ, is increased, there is a corresponding reduction in the $\mathbf{R}$ values for the tip nodes. This effect is illustrated in Figure 4.3, which gives boxplots of the sample means of the tip r values for different values of ρ. Theoretically the heating comes from less genetic material being passed to the tip nodes from the internal nodes. Therefore, the reduction in the mean r values for the tip nodes should correspond to shorter internal node sequences.

When ρ is at the coldest temperature, samples are taken from the distribution of interest. We can therefore compare the estimates from samples at the coldest temperature found in the second run to the results on this dataset without simulated tempering, which were summarized in Section 3.4.4. However, there are important differences between the two runs. The results without simulated tempering were based on 40 million MCMC samples, with every $10,000^{th}$ sample saved, and both the dictionary rephaser and allele swap updates were used to sample new haplotype configurations. With simulated tempering, due to time and memory constraints, the second run consisted of a total of 10 million MCMC samples and only approximately 400,000 of those samples were from the distribution of interest. The results with simulated tempering should therefore be interpreted with caution as more samples are required. In addition, the dictionary rephaser is not used with simulated tempering, and therefore many swap updates might be required to move between two configurations with high posterior probability.

The estimated posterior probabilities of the haplotype configurations for four selected individuals in the dataset are given in Table 4.1. Even with the much smaller sample size at the target distribution, the results with simulated tempering (the column labelled ST) are similar to the five runs without simulated tempering (the columns labelled Run 1 through Run 5) for the two individuals, 12 and 24, that had adequate mixing of the haplotype configurations in the results without simulated tempering. The third configuration for individual 12 is sampled less frequently in the results with simulated tempering than the results without; however, the number of MCMC iterations is too small to have much confidence in this difference. Traceplots of the sampled haplotype configurations for these two individuals are given in Figure 4.4(A) and (B). Both show that the sam-



(A)

(B)

FIGURE 4.2: Plots of occupation numbers at each temperature, given as a proportion of the total number of iterations, by temperature (left) and traceplots of sampled temperature values (right) for two runs of the sampler on the genotype dataset. Both sets of results are based on the same temperatures and pseudoprior. The dotted line corresponds to uniform occupation numbers. Panel (A): Run with 2.5 million iterations. Panel (B): Run with 10 million iterations.

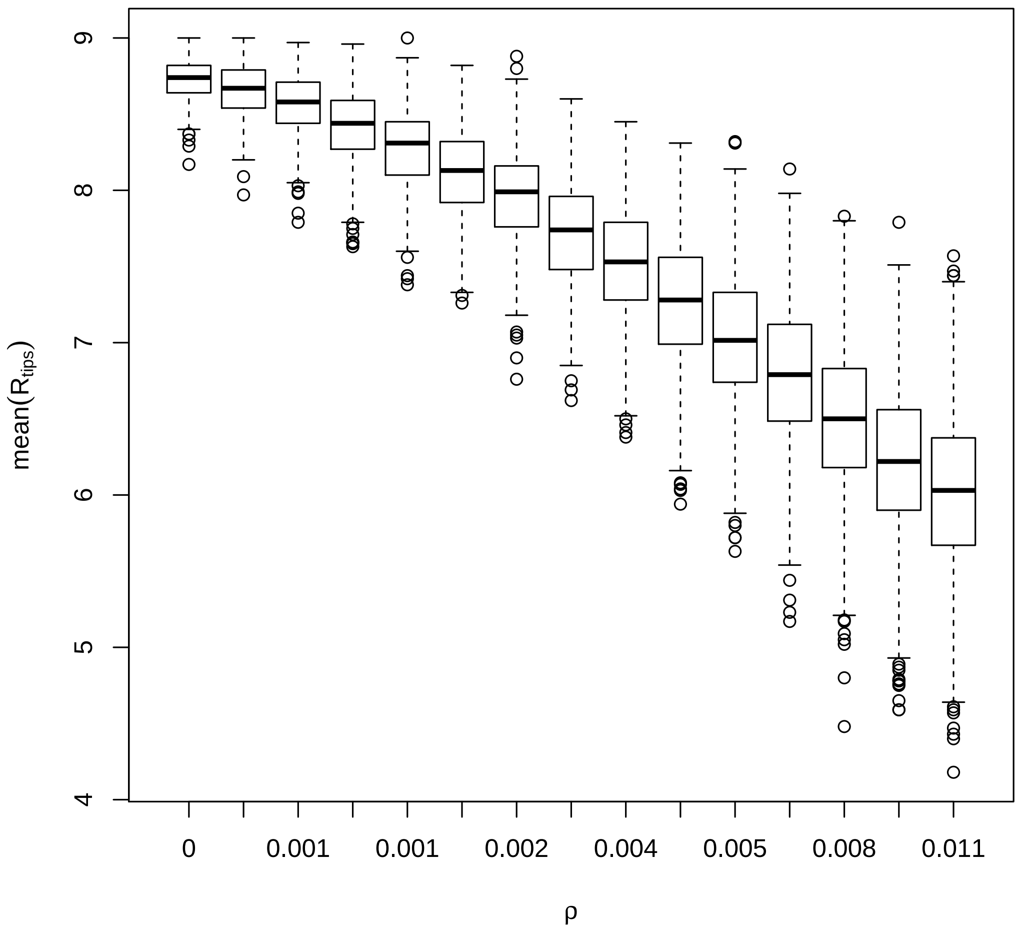


FIGURE 4.3: Boxplot of sample averages of the r values for tip nodes by ρ value. The sampler was run for 10 million iterations and $\mathbf{R}$ and ρ were periodically saved. For each saved sample, the average of the r values of the tip nodes was computed. Each boxplot summarizes the distribution of the averages for the given value of ρ.

TABLE 4.1: Table of estimated posterior probabilities of the haplotype configurations for four individuals in dataset 29. Posterior probabilities of haplotype configurations were estimated across five runs of 40 million iterations each of the sampler without simulated tempering (columns labelled Run 1 through Run 5) and in one run of 10 million iterations of the sampler with simulated tempering (column labelled ST). The bold haplotype configuration gives the true configuration for each individual.

id	haplotype configuration	Run 1	Run 2	Run 3	Run 4	Run 5	ST
5	1030/62713	0.961	0.75	0.967	0.809	0.32	0.406
5	**1273/62470**	0.038	0.25	0.033	0.169	0.652	0.593
12	**1030/62969**	0.854	0.834	0.868	0.835	0.834	0.841
12	1273/62726	0.009	0.01	0	0	0.052	0.007
12	1286/62713	0.131	0.111	0.125	0.122	0.055	0.025
12	1529/62470	0.003	0.043	0.003	0.04	0.057	0.124
20	1030/2809	0.175	0	0.329	0.003	0.146	0.0045
20	**1273/2566**	0.823	0.999	0.669	0.996	0.848	0.995
24	**1030/1529**	0.641	0.642	0.692	0.59	0.618	0.611
24	1273/1286	0.358	0.358	0.307	0.41	0.381	0.387

pler with simulated tempering continues to sample alternate haplotype configurations frequently. To summarize, these results show that for the individuals where mixing of the haplotype configurations previously seemed adequate, based on traceplots and comparison with PHASE results, simulated tempering does not change the resulting configurations that are sampled.

In order to determine whether simulated tempering has improved mixing, we examined the results of the two individuals, 5 and 20, for whom the haplotype configuration appeared to stick when the sampler was run without simulated tempering. Estimated posterior probabilities of the haplotype configurations for these individuals are also provided in Table 4.1; however, we would not expect these estimates to be similar to previous runs of the sampler as the mixing over haplotype configurations was previously inadequate. The traceplot for individual 5, given in Figure 4.4(C), shows that a number of transitions between the two most probable configurations are made. This traceplot can be contrasted with the traceplots for this individual given in Figure 3.13. In particular, in the five traceplots there is no segment of length 10 million iterations (1000 samples after thinning) where the haplotype configuration jumps as frequently between the two main haplotype configurations as it does in Figure 4.4(C). Therefore, for this individual, we are starting to see improved mixing at 15 heated distributions. The results on individual 20 are less promising; only three transitions are

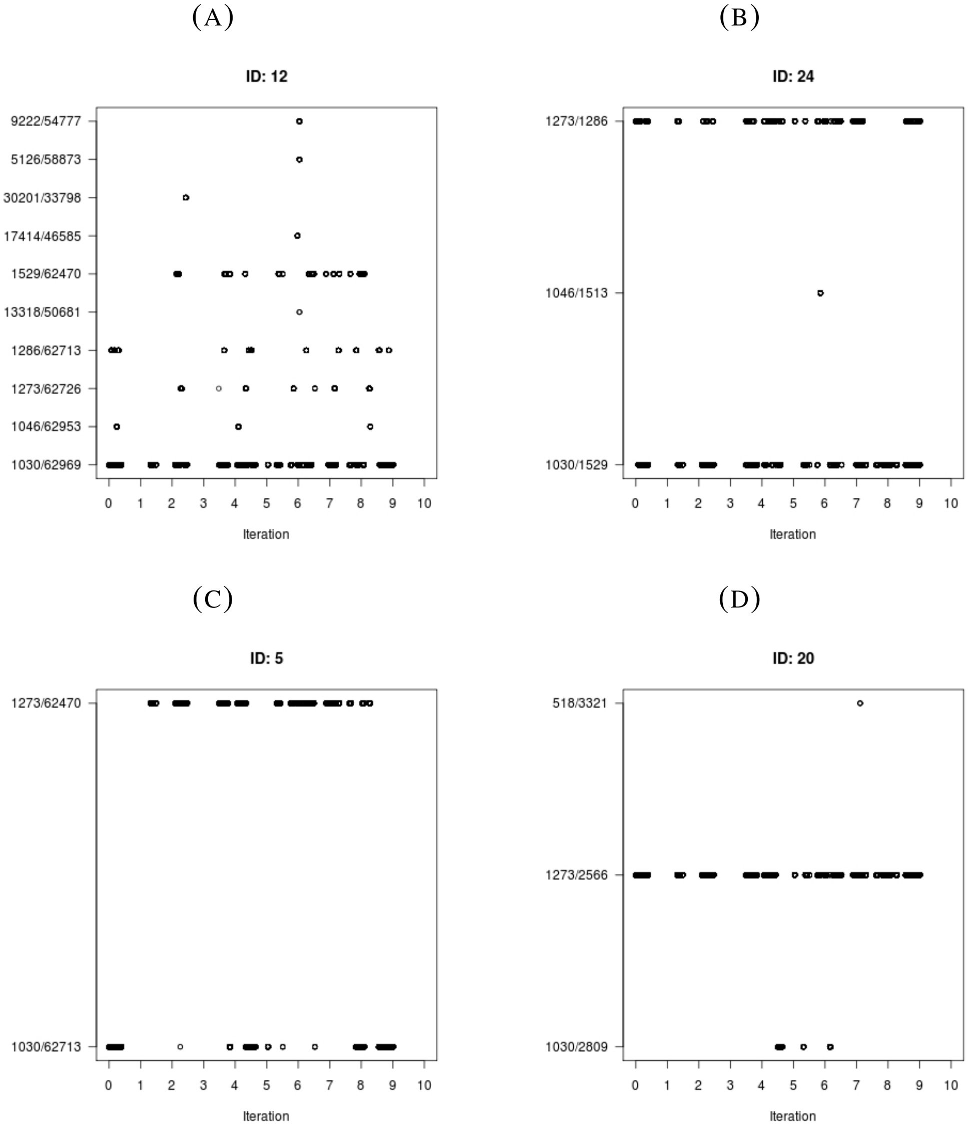


FIGURE 4.4: Traceplots of sampled haplotype configurations for four individuals in dataset 29. The sampler with simulated tempering was run for 10 million iterations. Output was saved only at the cold temperature, when sampling was from the target distribution, so these results are based on 400,000 MCMC samples from the target distribution.

made between the two main configurations. However, given the improved results on individual 5, it is possible that adding more heated distributions will help for this individual as well.

After significant effort to adjust the temperatures and pseudoprior, promising improvements in mixing of the haplotype configurations were seen. It would be worthwhile to examine longer run lengths; however, the current implementation of simulated tempering will not allow longer run lengths due to the massive file sizes produced when simulated tempering is used. For example, the total sizes of the output files produced by the final run of 10 million are close to 1 GB because all samples taken at the cold temperature are stored. Improvements to the implementation of the sampler with simulated tempering to store output more efficiently are now required.

4.3 Simulated tempering using other heating strategies

The results presented in the previous section used heated distributions where the recombination rate ρ was assumed known and set to gradually increasing values. In this section, we briefly describe our experiences and thoughts on other approaches to specify the heated distributions.

4.3.1 Altering the prior distribution for ρ

We also examined simulated tempering when the prior distribution for ρ was assumed to be a gamma distribution. Heating was achieved by letting the temperature be the shape parameter of the gamma distribution. As there were large differences in the occupation numbers and acceptance probabilities for temperature transitions between multiple runs with the same temperatures and pseudoprior, we concluded that infeasibly long runs would be required for adequate tuning.

Tuning output for two runs of 1.5 million MCMC samples with simulated tempering are shown in Figure 4.5. Note that these results do not correspond to the same genotype dataset used in Section 4.2.1; these results correspond to a dataset used exclusively for testing and where we know that the non-simulated tempering version of the sampler performs adequately. The two top plots, (A) and (B), show the occupation numbers and a traceplot of the sampled temperatures for one of the runs; the bottom two plots, (C) and (D), show the same for a second run. In the first run, for most of the 1.5 million iterations, the sampler spends a roughly uniform amount of time at each of the seven temperatures, as seen in (A), and the sampler is traversing well between temperatures, as seen in the traceplot of the temperatures in (B). On the other hand, in the second run, the sampler spends too much time at the coldest temperatures, as shown in the spike at 1 in (C). From the traceplot of the temperatures for this run (D) we can see why it spends so much at this temperature: after roughly 300,000 iterations the sampler is unable to escape the three coldest temperatures.

With results that are unpredictable from run to run, tuning becomes challenging. Recall that we use the occupation numbers and acceptance rates for temperature transitions from a run to estimate the temperatures/pseudoprior for the next run. For example, based on the first run, no further tuning would be required. Based on the second run drastic changes to the pseudoprior and temperatures would likely be required. The instability of occupation numbers observed in Figure 4.5 between runs having identical pseudoprior and temperatures makes tuning the pseudoprior/temperatures based on sampler output from runs of this length very difficult.

With much longer run lengths for tuning, the instability of the occupation numbers and acceptance probabilities for transitions might be reduced since eventually all runs will converge to the same target distribution. With these long run lengths for tuning it may be possible to find a particular combination of pseudoprior and temperature that allows the sampler to move across the temperatures. However, requiring excessively long run lengths for the tuning phase would not be practical for most users of the sampler. As a result, this form of heating was not pursued further.

4.3.2 Heating using “Powering up”

A more common form of heating is sometimes called "powering up". With powering up, the unnormalized density $f(\mathbf{x})$ is heated by raising it to a power that is less than 1, $f(\mathbf{x})^{1/t_i}$. As the temperature t_i is increased, the modes of the distribution are flattened and at high enough values the heated distribution is essentially uniform. Although this form of heating is quite common, powering up will not work as a heating mechanism for all distributions. For example, in the “witch’s hat" example of Geyer and Thompson (1995), the target distribution has two levels and is therefore not continuous over the support. Raising it to a power will not eliminate the two levels. For this reason, and since the genetic example of Geyer and Thompson (1995) also did not use powering up, we decided to heat by increasing ρ based on a direct appeal to population-genetic principles. In essence, increasing ρ allows the haplotype configurations to be freed from the constraints of the genealogical tree.

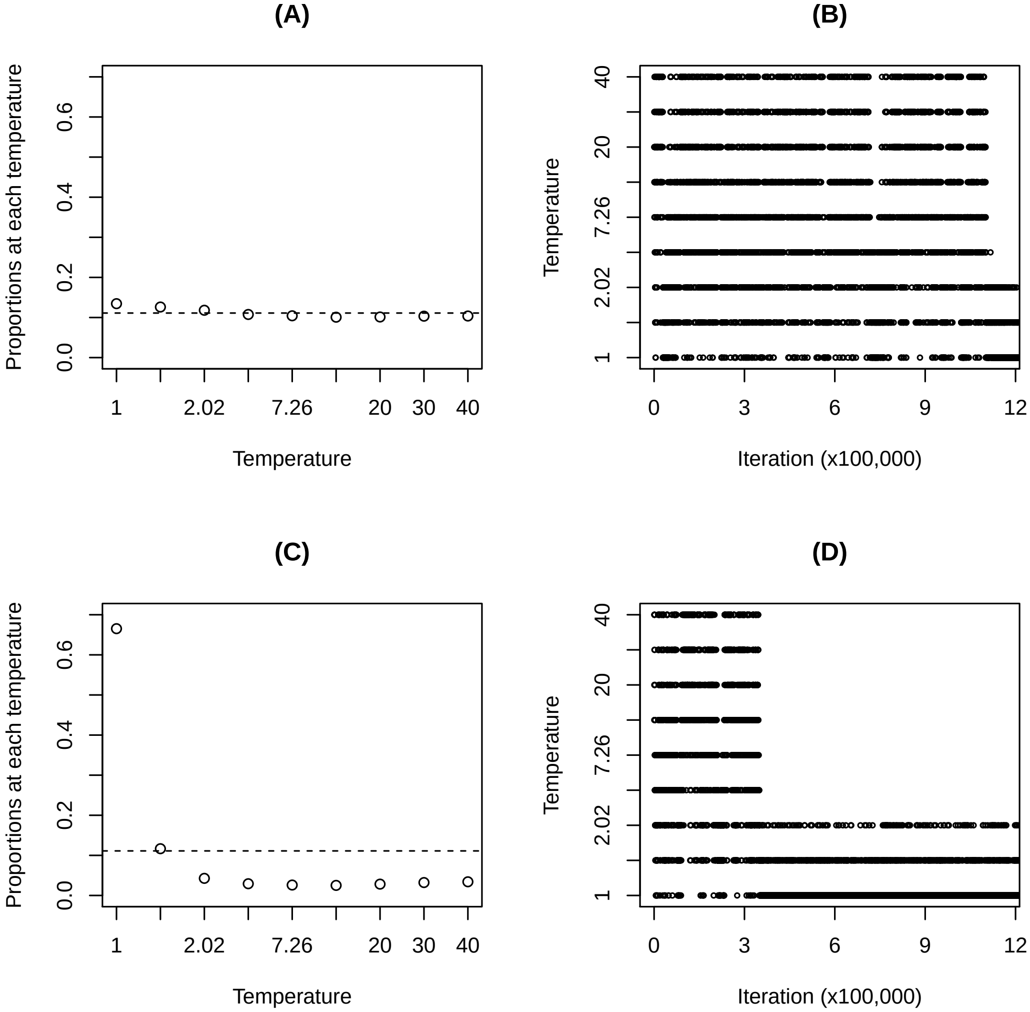


FIGURE 4.5: Plots of occupation numbers at each temperature, given as a proportion of the total number of iterations, for each temperature (left) and traceplots of sampled temperature values (right) for two runs of the sampler on a genotype dataset. Each row corresponds to a single run. The dotted line in (A) and (C) correspond to uniform occupation numbers. The temperatures and pseudoprior were the same for both runs.

4.4 Concluding remarks

Outside of the physics literature, there seem to be few examples of simulated tempering with real data. In this chapter, we have given an example of simulated tempering to improve mixing when the latent variable of interest is a tree structure. We have described how heating the recombination rate, which controls the information passed down to nodes on the tree, can result in improved mixing.

Our investigation of simulated tempering with fixed ρ values should be considered as a proof-of-principle. Now that we have some evidence that this form of simulated tempering does indeed improve mixing, further improvements can be pursued. First, as mentioned, in order to reduce file sizes, programming can be done that outputs the data more efficiently. We can also examine whether it is necessary to fix ρ at a known value for the cold distribution. In particular, it may be possible to assume that for the target distribution ρ has an exponential prior distribution, but that for the heated distributions ρ is assumed known and set to increasingly larger values.

Simultaneously tuning the pseudoprior and temperatures proved to be an extremely time consuming process. Difficulties associated with simultaneously finding a pseudoprior and temperatures have previously been noted (for example, (Jasra et al., 2007)). Although Geyer and Thompson (1995) provided strategies for tuning, these did not perform well in our context. For estimating the pseudoprior/normalizing constants, as a minimum, a desirable quality of a tuning strategy is that, after every iteration, our estimates are closer to the true values (i.e., the occupation numbers are closer to uniformly distributed). However, in our experience, the tuning process on one iteration could produce results having more unequal occupation numbers in the next iteration, possibly because we are not able to run the sampler long enough during the tuning phase.

The challenges associated with tuning limit the general applicability of this approach in this context. It is probably impractical for most users to commit as much time to selecting the pseudoprior and temperatures for a single dataset, especially since the process might be required for multiple datasets or multiple focal points. Therefore, investigation into how well the tuned temperatures/pseudoprior transfer to other datasets must be done. Ideally, for datasets of similar size and similar joint genotypic distributions, the temperatures selected based on one dataset or focal point would have reasonable acceptance probabilities for temperature transitions on a second dataset. Then, for a given set of temperatures, only the pseudoprior would need to be tuned. Restricting the tuning to one set of values would hopefully be less time consuming for the user. On the other hand, it is possible that tuning on very small subsets of a larger dataset might produce a reasonable set of temperatures/pseudoprior to apply to the larger dataset. This would improve the computation time for each iteration of the tuning process.

Other heating-based approaches could also be explored if the temperatures/pseudoprior values prove to be useful only for a single dataset. In particular, it might be less time consuming to select the temperatures for an MCMCMC algorithm than the temperatures/pseudoprior for a simulated tempering algorithm. There have been some successes using MCMCMC in a genetic setting. For example, Kuhner et al. (2000) used a powering up version for their genotype-based genealogy sampler. Although it initially did not perform well with a model that included recombination, MCMCMC is still an option in their estimation program, LAMARC (Kuhner, 2006). It may therefore be worth exploring this approach as an alternative to simulated tempering if the temperatures and pseudoprior selected for a single dataset are not applicable to another dataset.

Chapter 5

Summary and Future Work

The gene genealogy is a tree describing the ancestral relationships among a sample of genes from unrelated individuals. Although the true genealogy cannot be known, the shared ancestry is reflected in the genes themselves and genetic data can therefore be used to gain insight about this latent genealogy. In this thesis, we have described an approach to sample genealogical trees conditional on genetic data. In this chapter, we summarize our results and discuss future work.

In Chapter 2, we described our MCMC-based approach to sample genealogical trees conditional on haplotype data. Our implementation was based on the genealogy sampler outlined in Zöllner and Pritchard (2005); however, we made some modifications to the sampler that generally make sampling more efficient. In particular, we structured our proposal distributions so that the proposed values will not automatically be rejected. We also simplified the sampler by eliminating the need for one of the six update types, the time swap. With our implementation, we did not find it necessary to complete the update types in the fixed order proposed by Zöllner and Pritchard (2005); in test datasets, the strategy of ordering updates did not improve mixing.

We illustrated the use of the sampler on a publicly-available dataset to assess whether a region on chromosome five was associated with Crohn's disease. We sampled genealogies at 100 focal points and computed a cluster-based association statistic on each genealogy. We used the fuzzy p-value approach to evaluate whether focal points in this region were associated with case/control status. The peak areas of association were similar when the cluster-based results were compared to the single-locus results; however, the results based on clustering appear much smoother and the signal is more distinct. Since the true disease-predisposing mutations have not been established with certainty in this region, it is not clear whether our approach offers additional information to localize the mutation over the single-locus results.

In addition to the genealogy being unknown, commonly used genotyping technology measures genotypes rather than haplotypes. Therefore, haplotype phase is also unknown. In Chapter 3, we presented an extension to our sampler to handle missing haplotype phase. We tested various proposal distributions to update the tip haplotypes. The two selected proposal distributions make complementary changes to the haplotypes. The allele-swap (Kuhner et al., 2000) shuffles the allele at a heterozygous locus on a single individual's two sequences. Our novel dictionary rephaser samples a new haplotype configuration from a distribution that puts more weight on configurations that are combinations of the current tip haplotypes.

We examined the performance of the sampler on simulated data consisting of 50 datasets of 25 individuals each. The estimates for the mutation rate, θ, and recombination rate, ρ, were similar across datasets. The similarity of the sampled haplotype configurations to the PHASE-estimated haplotype configurations was also examined to assess the sampler's ability to handle missing phase. In general, the estimates of the posterior probabilities of the different haplotype configurations were quite similar for PHASE and for our sampler. For some individuals in some datasets, PHASE sampled the true haplotype configuration more frequently than our sampler; in others, our sampler sampled the true haplotype configuration more frequently. Although the traceplots generally indicated good mixing, for some individuals in some datasets, the traceplots of the haplotype configuration suggested that our sampler moves to new haplotype configurations too infrequently. Therefore, longer run lengths or further improvements to the sampler are necessary.

To assess the consistency of results, ten of the 50 datasets were run an additional five times, with each run started at a different set of initial sequences and tree. Three of the ten datasets showed signs of poor mixing with respect to the sampled haplotypes. However, the problem was severe for only six individuals within the three datasets.

To assess the sensitivity of the genotype-based sampler to our selected prior distributions for the mutation rate, θ and the recombination rate, ρ, we tested two alternate prior distributions for each parameter on ten of the datasets. We saw that the estimates of the marginal posterior density for θ were very similar for the different prior distributions. There was more sensitivity to the prior distribution for ρ: the results with an exponential prior distribution with mean 1 were different than the results with lower means. However, genomic estimates of the recombination rate are around 0.0004 per pair of base pairs per coalescent time unit. Therefore, a prior distribution with such a large mean would be a poor choice. We also assessed sensitivity to the haplotype frequency parameter, γ, on the same ten datasets by examining results with two alternate values of γ. There were some differences in the haplotype configurations sampled for the different choices of γ. However, these differences are most likely explained by poor mixing rather than sensitivity to the haplotype frequency model.

As mentioned, for some of the datasets, longer run times are necessary. However the dictionary rephaser is computationally intensive as currently programmed, and for some datasets many weeks or months would be required to complete a sufficient number of iterations to be confident in the mixing. As discussed in Section 3.5, it is possible to improve the speed of the dictionary rephaser by reducing the number of haplotype configurations contained in the set of candidate configurations. We are planning to implement these changes soon in order to make run times more similar between datasets of the same size and allow the dictionary rephaser to be used on all datasets. This modification will also be required in order to use the sampler on larger datasets.

Finally, MCMC is prone to mixing and convergence issues when the state space has high dimension or is multi-modal. When duplicate runs of the genotype-based sampler were performed on ten simulated datasets, three of the ten datasets contained individuals with haplotype configurations that tended to get stuck at a single configuration. We therefore evaluated the use of simulated tempering in order to improve the mixing of the Markov chain. We focused on improving mixing of the genotype-based sampler for one of these three datasets where mixing was particularly poor. For two individuals in this dataset, the sampler run without simulated tempering rarely sampled alternative haplotype configurations.

We focused on the recombination rate, ρ, for our heating efforts since it controls how much sequence is passed down the tree along with the focal point. At higher values of ρ, recombination events are located closer to the focal point and therefore less sequence is co-inherited with the focal point. Our heated distributions therefore consist of the original distribution at fixed, but increasing, values of ρ. Although we were able to improve mixing for the challenging dataset, the tuning process was extremely time consuming and the process is not easily automated.

Our experiences with selecting the pseudoprior and temperatures for simulated tempering lead to questions about the general feasibility of this approach in this context. Although one might have the resources to devote to finding a good pseudoprior and temperatures for a single dataset, we would anticipate that the sampler would be used on multiple focal points or multiple datasets and a subset of these might show signs of poor mixing. Therefore, finding a tuning strategy that requires less user intervention will be necessary. One approach to examine is whether the temperature/pseudoprior values selected for one dataset can be applied to another dataset. As mentioned in Section 4.4, there has also been some success using MCMCMC in genetic applications. It may therefore be worthwhile to explore the performance of MCMCMC on our sampler for the challenging datasets. Sequential Monte Carlo is another sampling approach that has recently been applied to sample phylogenetic trees (Bouchard-Côté et al., 2011). Therefore, it may also be applicable to our problem of sampling genealogical trees.

Although there may be multiple applications for a genealogy sampler, the initial motivation was for its use in finding regions in the genome that harbour disease-predisposing mutations. With the completion of a sampler that works reasonably well, it is now possible to begin using the sampler for this purpose. There are many different tree-based association statistics whose properties and performance in localizing disease-predisposing mutations can be evaluated. In Section 1.1.3, we provided a summary of some proposed tree-based association statistics and we used a cluster-based tree statistic to illustrate the use of the haplotype-based sampler. The evaluation of these tree-based statistics could include data simulated with disease models where there are multiple disease mutations or rare mutations.

We can also develop and evaluate novel tree-based association statistics. For example, under the assumption that individuals affected with disease are more closely related than unaffected individuals due to the shared ancestry at the disease-predisposing mutation, distance on the tree between pairs of individuals might discriminate affected individuals from unaffected individuals. This is similar in spirit to haplotype sharing methods. Haplotype sharing (see for example Allen and Satten (2007)) is based on the observation that in the region of a disease-predisposing mutation case haplotypes should be more similar to each other than to randomly selected haplotypes. These approaches assess whether the similarity of pairs of haplotypes is correlated with the similarity of disease status. However, the haplotypes of cases will be similar because of the shared ancestry at the disease-predisposing locus. Therefore, haplotype similarity is really a proxy for the relatedness of individuals on the ancestral tree. This motivates using proximity on the ancestral tree rather than between haplotypes to define a similarity measure between individuals.

Finally, there are also many different comparisons to examine between typical approaches and a genealogy-based approach that incorporates tree and haplotype uncertainty. First, we could examine the impact of sample size on tree-based association statistics to determine whether this approach requires the large sample sizes of the single-locus association approaches. Second, many of the clustering based approaches use single imputation of a tree in computing their tree-based association statistic. With a sampler that can actually capture the uncertainty of tree reconstruction, the impact of single imputation in the context of genealogy-based association statistics can now be assessed. Third, Zöllner and Pritchard (2005) suggested using a phase reconstruction program and treating the reconstruction as known data in order to handle haplotype uncertainty. With the genotype-based sampler, we can actually evaluate this strategy and quantify the effects of treating a haplotype reconstruction as if it were known.

In summary, we have presented an algorithm for sampling genealogical trees conditional on haplotype or genotype data. It can be used in conjunction with tree-based association statistics to find disease-predisposing mutations. Further work on evaluating the performance of different tree-based statistics can now be done, as well as comparing the standard approaches to an approach that uses the sampler to account for tree/haplotype uncertainty.

Bibliography

Albatineh, A. N., Niewiadomska-Bugaj, M., and Mihalko, D. (2006). On similarity indices and correction for chance agreement. *Journal of Classification*, 23(2):301–313.

Allen, A. S. and Satten, G. A. (2007). Statistical models for haplotype sharing in case-parent trio data. *Human Heredity*, 64(1):35–44.

Andrés, A. M., Clark, A. G., Shimmin, L., Boerwinkle, E., Sing, C. F., and Hixson, J. E. (2007). Understanding the accuracy of statistical haplotype inference with sequence data of known phase. *Genetic Epidemiology*, 31(7):659–671.

Atchadé, Y., Roberts, G., and Rosenthal, J. (2010). Towards optimal scaling of Metropolis-coupled Markov chain Monte Carlo. *Statistics and Computing*, 21:555–568.

Awadalla, P., Gauthier, J., Myers, R. A., Casals, F., Hamdan, F. F., Griffing, A. R., Côté, M., Henrion, E., Spiegelman, D., Tarabeux, J., Piton, A., Yang, Y., Boyko, A., Bustamante, C., Xiong, L., Rapoport, J. L., Addington, A. M., DeLisi, J. L. E., Krebs, M., Joober, R., Millet, B., Fombonne, E., Mottron, L., Zilversmit, M., Keebler, J., Daoud, H., Marineau, C., Roy-Gagnon, M., Dubé, M., Eyre-Walker, A., Drapeau, P., Stone, E. A., Lafrenière, R. G., and Rouleau, G. A. (2010). Direct measure of the de novo mutation rate in autism and schizophrenia cohorts. *American Journal of Human Genetics*, 87(3):316–324.

Bardel, C., Danjean, V., Hugot, J., Darlu, P., and Génin, E. (2005). On the use of haplotype phylogeny to detect disease susceptibility loci. *BMC genetics*, 6(1):24.

Beerli, P. and Felsenstein, J. (1999). Maximum-likelihood estimation of migration rates and effective population numbers in two populations using a coalescent approach. *Genetics*, 152:763–773.

Besag, J. (2001). Markov chain Monte Carlo for statistical inference. Working paper No. 9, Center for Statistics and the Social Sciences, University of Washington. Available from http://www.csss.washington.edu/Papers/.

Besag, J., Green, P., Higdon, D., and Mengersen, K. (1995). Bayesian computation and stochastic systems. *Statistical Science*, 10:3–41.

Bouchard-Côté, A., Sankararaman, S., and Jordan, M. I. (2011). Phylogenetic inference via sequential Monte Carlo. *Systematic Biology (In Press)*.

Brooks, S. P. (1998). Markov chain Monte Carlo method and its application. *The Statistician*, 47(1):69–100.

Browning, B. L. and Browning, S. R. (2009). A unified approach to genotype imputation and haplotype-phase inference for large data sets of trios and unrelated individuals. *American Journal of Human Genetics*, 84(2):210–223.

Clark, A. G. (1990). Inference of haplotypes from PCR-amplified samples of diploid populations. *Molecular Biology and Evolution*, 7:111–122.

Cooney, R. and Jewell, D. (2009). The genetic basis of inflammatory bowel disease. *Digestive Diseases*, 27(4):428–442.

Daly, M. J., Rioux, J. D., Schaffner, S. F., Hudson, T. J., and Lander, E. S. (2001). High-resolution haplotype structure in the human genome. *Nature Genetics*, 29(2):229–232.

Duret, L. (2009). Mutation patterns in the human genome: More variable than expected. *PLoS Biology*, 7(2):e1000028.

Durrant, C., Zondervan, K. T., Cardon, L. R., Hunt, S., Deloukas, P., and Morris, A. P. (2004). Linkage disequilibrium mapping via cladistic analysis of single-nucleotide polymorphism haplotypes. *American Journal of Human Genetics*, 75(1):35–43.

Epstein, M. P. and Kwee, L. C. (2009). Haplotype association analysis. In Lin, S. and Zhao, H., editors, *Handbook on Analyzing Human Genetic Data*, pages 241–276. Springer Berlin Heidelberg, Berlin, Heidelberg.

Excoffier, L. and Slatkin, M. (1995). Maximum likelihood estimation of molecular haplotype frequencies in a diploid population. *Molecular Biology and Evolution*, 12:921–927.

Felsenstein, J. (2004). *Inferring phylogenies*. Sinauer Associates, Sunderland, Mass.

Felsenstein, J. (2006). Accuracy of coalescent likelihood estimates: Do we need more sites, more sequences, or more loci? *Molecular Biology and Evolution*, 23(3):691–700.

Felsenstein, J., Kuhner, M. K., Yamato, J., and Beerli, P. (1999). Likelihoods on coalescents: A Monte Carlo sampling approach to inferring parameters from populatoin samples of molecular data. In Seillier-Moiseiwitsch, F., editor, *Statistics in Molecular Biology*, volume 33, pages 163–185. American Mathematical Society, first edition.

Fisher, R. A. (1930). *The Genetical Theory of Natural Selection*. Clarendon Press, first edition.

Geyer, C. (1991). Markov chain Monte Carlo maximum likelihood. In *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, pages 156–163.

Geyer, C. J. and Thompson, E. A. (1995). Annealing Markov chain Monte Carlo with applications to ancestral inference. *Journal of the American Statistical Association*, 90(431):909–920.

Gilks, W. and Roberts, G. (1996). Strategies for improving MCMC. In Gilks, W. R., Richardson, S., and Spiegelhalter, D. J., editors, *Markov Chain Monte Carlo in Practice*, pages 89–110. Chapman and Hall/CRC, first edition.

Gilks, W. R., Richardson, S., and Spiegelhalter, D. J. (1998). *Markov Chain Monte Carlo in Practice*. Chapman and Hall/CRC, Boca Raton, Florida, first edition.

Hawley, M. E. and Kidd, K. K. (1995). HAPLO: a program using the EM algorithm to estimate the frequencies of multi-site haplotypes. *The Journal of Heredity*, 86(5):409–411.

Hein, J., Schierup, M., and Wiuf, C. (2005). *Gene Genealogies, Variation and Evolution: A primer in coalescent theory*. Oxford University Press.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2(1):193–218.

Hudson, R. (1990). Gene genealogies and the coalescent process. In Futuyma, D. and Antonovics, J., editors, *Oxford Surveys in Evolutionary Biology*, pages 1–44. Oxford University Press: Oxford.

Hudson, R. R. (1983). Properties of a neutral allele model with intragenic recombination. *Theor. Pop. Biol.*, 23:183–201.

Hudson, R. R. (2002). Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics*, 18(2):337–8.

International HapMap Consortium (2005). A haplotype map of the human genome. *Nature*, 437(7063):1299–1320.

Jasra, A., Stephens, D. A., and Holmes, C. C. (2007). On population-based simulation for static inference. *Statistics and Computing*, 17(3):263–279.

Kimmel, G., Karp, R. M., Jordan, M. I., and Halperin, E. (2008). Association mapping and significance estimation via the coalescent. *The American Journal of Human Genetics*, 83(6):675–683.

Kingman, J. F. C. (1982). The coalescent. *Stochastic Processes and their Applications*, 13(3):235 – 248.

Kuhner, M. K. (2006). LAMARC 2.0: maximum likelihood and Bayesian estimation of population parameters. *Bioinformatics*, 22(6):768 –770.

Kuhner, M. K. (2009). Coalescent genealogy samplers: windows into population history. *Trends in Ecology & Evolution*, 24(2):86–93.

Kuhner, M. K. and Felsenstein, J. (2000). Sampling among haplotype resolutions in a coalescent-based genealogy sampler. *Genetic epidemiology*, 19 Suppl 1:S15–21.

Kuhner, M. K., Yamato, J., and Felsenstein, J. (1995). Estimating effective population size and mutation rate from sequence data using Metropolis-Hastings sampling. *Genetics*, 140:1421–1430.

Kuhner, M. K., Yamato, J., and Felsenstein, J. (1998). Maximum likelihood estimation of population growth rates based on the coalescent. *Genetics*, 149(1):429–34.

Kuhner, M. K., Yamato, J., and Felsenstein, J. (2000). Maximum likelihood estimation of recombination rates from population data. *Genetics*, 156:1393–1401.

Liang, F. (2005). Determination of normalizing constants for simulated tempering. *Physica A: Statistical Mechanics and its Applications*, 356:468–480.

Lin, D. Y. and Huang, B. E. (2007). The use of inferred haplotypes in downstream analyses. *American Journal of Human Genetics*, 80(3):577–9.

Liu, J. S. and Sabatti, C. (1999). Simulated sintering: Markov chain Monte Carlo with spaces of varying dimensions. In M., B. J., O., B. J., P., D. A., and M., S. A. F., editors, *Bayesian Statistics 6*. Oxford University Press.

Long, J. C., Williams, R. C., and Urbanek, M. (1995). An E-M algorithm and testing strategy for multiple locus haplotypes. *Am J Hum Genet*, 56:799–810.

Mailund, T., Besenbacher, S., and Schierup, M. (2006). Whole genome association mapping by incompatibilities and local perfect phylogenies. *BMC Bioinformatics*, 7(1):454.

Marchini, J., Cutler, D., Patterson, N., Stephens, M., Eskin, E., Halperin, E., Lin, S., Qin, Z. S., Munro, H. M., Abecasis, G. R., and Donnelly, P. (2006). A comparison of phasing algorithms for trios and unrelated individuals. *American Journal of Human Genetics*, 78(3):437–50.

Marinari, E. and Parisi, G. (1992). Simulated tempering: a new Monte Carlo scheme. *EPL (Europhysics Letters)*, 19:451–458.

McPeek, M. S. and Strahs, A. (1999). Assessment of linkage disequilibrium by the decay of haplotype sharing, with application to fine-scale genetic mapping. *American Journal of Human Genetics*, 65(3):858–75.

McVean, G., Awadalla, P., and Fearnhead, P. (2002). A coalescent-based method for detecting and estimating recombination from gene sequences. *Genetics*, 160(3):1231–1241.

McVean, G. A. T., Myers, S. R., Hunt, S., Deloukas, P., Bentley, D. R., and Donnelly, P. (2004). The fine-scale structure of recombination rate variation in the human genome. *Science*, 304:581–584.

Mensah, F. K., Gilthorpe, M. S., Davies, C. F., Keen, L. J., Adamson, P. J., Roman, E., Morgan, G. J., Bidwell, J. L., and Law, G. R. (2007). Haplotype uncertainty in association studies. *Genetic Epidemiology*, 31(4):348–57.

Minichiello, M. J. and Durbin, R. (2006). Mapping trait loci by use of inferred ancestral recombination graphs. *The American Journal of Human Genetics*, 79(5):910–22.

Molitor, J., Marjoram, P., and Thomas, D. (2003). Fine-scale mapping of disease genes with multiple mutations via spatial clustering techniques. *The American Journal of Human Genetics*, 73(6):1368–1384.

Morris, A. P., Whittaker, J. C., and Balding, D. J. (2002). Fine-scale mapping of disease loci via shattered coalescent modeling of genealogies. *American Journal of Human Genetics*, 70(3):686–707.

Neal, R. (1996). Sampling from multimodal distributions using tempered transitions. *Statistics and Computing*, 6:353—-366.

Niu, T., Qin, Z. S., Xu, X., and Liu, J. S. (2002). Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms. *American Journal of Human Genetics*, 70:157–169.

Nordborg, M. (2003). Coalescent theory. In Balding, D. J., Bishop, M., and Cannings, C., editors, *Handbook of Statistical Genetics*, pages 179–212. John Wiley and Sons, second edition.

Paradis, E., Claude, J., and Strimmer, K. (2004). APE: analyses of phylogenetics and evolution in R language. *Bioinformatics*, 20(2):289–290.

Peltekova, V. D., Wintle, R. F., Rubin, L. A., Amos, C. I., Huang, Q., Gu, X., Newman, B., Oene, M. V., Cescon, D., Greenberg, G., Griffiths, A. M., St George-Hyslop, P. H., and Siminovitch, K. A. (2004). Functional variants of OCTN cation transporter genes are associated with Crohn disease. *Nature Genetics*, 36(5):471–475.

Rioux, J. D., Daly, M. J., Silverberg, M. S., Lindblad, K., Steinhart, H., Cohen, Z., Delmonte, T., Kocher, K., Miller, K., Guschwan, S., Kulbokas, E. J., O'Leary, S., Winchester, E., Dewar, K., Green, T., Stone, V., Chow, C., Cohen, A., Langelier, D., Lapointe, G., Gaudet, D., Faith, J., Branco, N., Bull, S. B., McLeod, R. S., Griffiths, A., Bitton, A., Greenberg, G. R., Lander, E. S., Siminovitch, K. A., and Hudson, T. J. (2001). Genetic variation in 5q31 cytokine gene cluster confers susceptibility to Crohn disease. *Nature Genetics*, 29:223–228.

Rioux, J. D., Silverberg, M. S., Daly, M. J., Steinhart, A. H., McLeod, R. S., Griffiths, A. M., Green, T., Brettin, T. S., Stone, V., Bull, S. B., Bitton, A., Williams, C. N., Greenberg, G. R., Cohen, Z., Lander, E. S., Hudson, T. J., and Siminovitch, K. A. (2000). Genomewide search in Canadian families with inflammatory bowel disease reveals two novel susceptibility loci. *Americal Journal of Human Genetics*, 66:1863–1870.

Ripley, B. (1987). *Stochastic simulation*. Wiley Series in Probability and Statistics. J. Wiley.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407.

Robinson, D. and Foulds, L. (1981). Comparison of phylogenetic trees. *Mathematical Biosciences*, 53:131–147.

Scheet, P. and Stephens, M. (2006). A fast and flexible statistical model for Large-Scale population genotype data: Applications to inferring missing genotypes and haplotypic phase. *American Journal of Human Genetics*, 78(4):629–644.

Seltman, H., Roeder, K., and Devlin, B. (2001). Transmission/Disequilibrium test meets measured haplotype analysis: Family-based association analysis guided by evolution of haplotypes. *American Journal of Human Genetics*, 68(5):1250–1263.

Seltman, H., Roeder, K., and Devlin, B. (2003). Evolutionary-based association analysis using haplotype data. *Genetic Epidemiology*, 25(1):48–58.

Stephens, M. (2003). Inference under the coalescent. In Balding, D. J., Bishop, M., and Cannings, C., editors, *Handbook of Statistical Genetics*, pages 636–661. John Wiley and Sons, second edition.

Stephens, M., Smith, N. J., and Donnelly, P. (2001). A new statistical method for haplotype reconstruction from population data. *American Journal of Human Genetics*, 68:978–989.

Tachmazidou, I., Verzilli, C. J., and Iorio, M. D. (2007). Genetic association mapping via evolution-based clustering of haplotypes. *PLoS Genetics*, 3(7):e111.

Takahata, N. (1993). Allelic genealogy and human evolution. *Molecular Biology and Evolution*, 10:2 –22.

Templeton, A. R., Boerwinkle, E., and Sing, C. F. (1987). A cladistic analysis of phenotypic associations with haplotypes inferred from restriction endonuclease mapping. I. Basic theory and an analysis of alcohol dehydrogenase activity in Drosophila. *Genetics*, 117(2):343 –351.

The 1000 Genomes Project Consortium (2010). A map of human genome variation from population-scale sequencing. *Nature*, 467(7319):1061–1073.

Thompson, E. A. and Geyer, C. J. (2007). Fuzzy p-values in latent variable problems. *Biometrika*, 94(1):49–60.

Tierney, L. (1994). Markov chains for exploring posterior distributions. *The Annals of Statistics*, 22:1701–1728.

Waller, S., Tremelling, M., Bredin, F., Godfrey, L., Howson, J., and Parkes, M. (2006). Evidence for association of OCTN genes and IBD5 with ulcerative colitis. *Gut*, 55(6):809–814.

Weir, B. S. (1996). *Genetic Data Analysis II.* Sunderland, Mass. : Sinauer Associates, second edition.

Wilson, I. J. and Balding, D. J. (1998). Genealogical inference from microsatellite data. *Genetics*, 150:499–510.

Wright, S. (1931). Evolution in Mendelian populations. *Genetics*, 16:97–159.

Zheng, Z. (2003). On swapping and simulated tempering algorithms. *Stochastic Processes and their Applications*, 104:131–154.

Zöllner, S. (2005). *Documentation for TreeLD, version 1.0.* `http://pritch.bsd.uchicago.edu/treeld/TreeLD_Doc/Documentation.html`.

Zöllner, S. and Pritchard, J. K. (2005). Coalescent-based association mapping and fine mapping of complex trait loci. *Genetics*, 169:1071–1092.

Appendix A

Background on Genetic Concepts

A.1 Genetic Terminology and Background

DNA is the molecule storing the hereditary material in most organisms, including humans. It is composed of *base pairs* which can be one of four types. The human genome contains 3 billion base pairs, most of which are identical from individual to individual. A *locus* has been historically used to denote any small DNA segment in the genome, but for the purposes of this document it will be used to denote a single base pair of DNA somewhere in the genome. A locus that varies from individual to individual is known as a *polymorphism*; the different types of a polymorphism are called the *alleles*.

In humans, not all polymorphic loci have been discovered and in particular there is much interest in finding loci that predispose an individual to a disease or trait. A polymorphic locus with a known location and alleles that can be easily observed can be used as a *genetic marker*. The observable genetic markers are used to map the locations of unknown polymorphic loci in the genome. Single nucleotide polymorphisms (*SNPs*), which differ at a single base pair, are commonly used genetic markers and our approach assumes SNP data.

Strictly speaking, a *gene* is a DNA segment that codes for a protein. However the term “gene" is often used with slightly different meanings. For example, if a person has “the gene for cystic fibrosis” what is meant is that they have a mutation in a gene which causes cystic fibrosis. Much of the genome is made up of DNA sequence that is not part of a gene, including important regulatory regions that could be implicated in diseases, and many genetic markers are not located in genes. For this reason, in this document, the use of the word “gene” will be avoided as much as possible in favour of locus or marker.

In humans, the genetic material is divided into 23 *chromosomes*. Each individual inherits two copies of every chromosome, one from each parent, so each individual has 46 chromosomes. Therefore, at any given marker locus, an individual receives one copy of the locus from their mother and one from their father. The combination of maternal and paternally inherited alleles observed at a locus is called the *genotype*. If the genotype of a locus is composed of two identical alleles, the genotype is *homozygous*; conversely, it is *heterozygous* if the two alleles are different. The set of alleles at multiple loci that are inherited together on the same chromosome from a single parent is called a *haplotype*.

SNP loci have two alleles that can be labelled as 0 and 1. At any SNP locus, the three possible genotypes are $g = \{0/0, 0/1, 1/1\}$, with the slash separating the two alleles from each other. The genotypes at three loci might be 0/0, 0/0, 0/1 for example. If these loci are on the same chromosome, this multi-locus genotype can also be written as 000/001 to indicate that the haplotype sequences 000 and 001 were inherited. The slash is now used to separate the two haplotypes from each other. For three SNP loci, the set of 8 possible haplotypes that can be observed is $\{000, 001, 010, 100, 011, 101, 110, 111\}$.

Meiosis is the biological process that creates gametes (i.e., sperm or egg cells) by dividing an individual's genomic material in half. During meiosis, each chromosomal pair, consisting of a maternally and paternally derived chromosome, is separated and a random copy of each chromosome is passed to the gamete cell. Thus, each gamete consists of a random assortment of maternal and paternal chromosomes.

During meoisis, the process of *recombination* in these chromosomal pairs swaps the parental origin of genetic material on a chromosome so that one chromosome might be made of a mixture of paternal and maternal genetic material. Figure A.1 gives a cartoon depicting recombination. A recombination event or a crossover is said to occur if the DNA at one locus is of a different parental origin than the DNA at the second locus. In the example shown in Figure A.1, locus A has a different parent than locus B in the top chromosome in panel (c) as indicated by the two different colours of the chromosomal segments.

When considering the joint inheritance from parent to offspring of two loci, if the two loci are on different chromosomes, the allelic type of the first locus is independent of the allelic type of the second due to the random assortment of chromosomes during meiosis. On the other hand, if the two loci are on the same chromosome, the two alleles are more likely to be inherited together since the

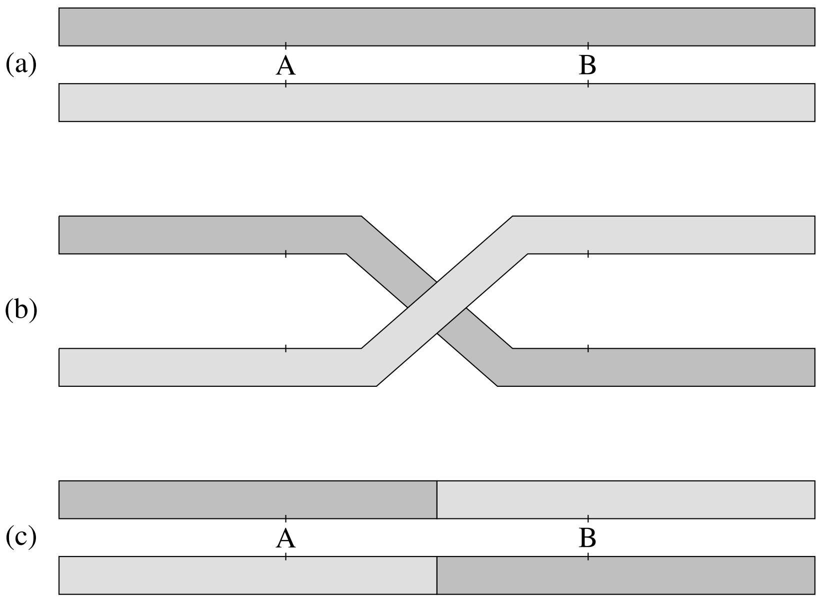


FIGURE A.1: Illustration of the process of recombination. (a) An individual has inherited two copies of a chromosome, one from each parent. The light gray chromosome was inherited from the mother; the dark gray chromosome was inherited from the father. Two loci, A and B, are shown on each chromosome. (b) During meiosis, a recombination event occurs at a random location between loci A and B. A crossover of genetic material between the two chromosomes takes place. (c) The resulting two chromosomes after the recombination are mixtures of the two original parental chromosomes. Locus A in the top chromosome is of paternal origin while locus B is of maternal origin. The reverse holds for the bottom chromosome. An offspring of this individual would randomly inherit one of these two chromosomes from this parent.

chromosome is passed as a block to the offspring cells. This leads to an association in the population of alleles at loci on the same chromosome. The degree of association in the population between the alleles at two loci on the same chromosome depends on the probability of recombination events occurring between the two loci. This probability depends on the distance between the two loci. The term for population dependence between alleles on the same chromosome is *allelic association* or *linkage disequilibrium*.

A.2 Finding disease-predisposing loci with association studies

A mutation that predisposes an individual towards a particular *trait*, for example a disease, occurred on a background haplotype at some point in the past (see Figure 1.4). At first, the mutated allele is completely associated with the other alleles on the same haplotype. Thus, there is complete allelic association between the other loci and the mutation locus. Over time, recombination will break down the association between these alleles. For loci that are really close to the trait locus, the allelic association degrades more slowly since few recombination events will occur between the two loci. Those who have the trait are therefore more likely to share alleles in common at loci in the vicinity of the trait-predisposing mutation due to this association.

For example, suppose that a mutation m occurs on a haplotype with alleles a_1 and b_1 at two loci A and B. The initial mutation-bearing haplotype is a_1mb_1 and only individuals with a_1 and b_1 at the adjacent loci will have the mutation allele. If A is assumed to be the closer of the two loci to the trait locus, after many generations, mutation-bearing haplotypes could be, for example, a_1mb_2 and a_1mb_1 in proportion to the frequencies of b_1 and b_2 in the population. The a_1 allele is still associated with the mutation allele but the b_1 is no longer associated with the mutation allele.

Gene mapping by association methods exploits the allelic association between a trait locus and a marker that is very close to the trait locus. The methods typically used for population-based samples examine marker alleles in affected individuals and unaffected individuals to determine if there are differences in allele frequencies between the two groups. A significant difference between frequencies in the two groups is taken as evidence that the allele either predisposes an individual towards the trait or that it is in linkage disequilibrium with the trait-predisposing allele. Therefore, the marker information is used as a proxy for the hidden trait locus since the genotype at the trait locus is most likely not among the observed genotypes. A chi-squared test or logistic regression is most often used to test whether the allele frequencies differ between the two groups. If the trait is continuous, linear regression based approaches are used to detect association. Association studies examine association of the trait with SNP markers in a candidate gene, multiple genes, or a genomic region that might have been linked to the trait using linkage analysis methods. Genomewide association studies (GWAS), which genotype hundreds of thousands of SNPs throughout the genome, are now routinely being done. In GWAS, no *a priori* assumptions are made about which genes or regions are likely to be associated and instead all markers are scanned for marginal association with the trait of interest.

A problem with candidate gene/region studies, and to a lesser extent GWAS, is that the results at SNPs in the same region are likely to be correlated due to the underlying allelic association; therefore, it might seem more efficient to consider their joint effect. Haplotypic tests, which often use a small set of markers and a "sliding window" approach, use the information of multiple markers at a time; however, choosing the number of markers to use for a given focal point can be a challenge. If too few are chosen the results may be no better than single-locus tests; if too many are chosen, there can be a loss of power due to increased degrees of freedom of the association statistic since the corresponding chi-squared test or regression model treats each different haplotype as a separate category. The haplotypic approaches also suffer from low counts of certain haplotypes, which can be due to linkage disequilibrium between alleles on a haplotype and even to genotyping error. The common practice is to bin low-count haplotype categories together in testing for association, which creates essentially a meaningless category that is unlikely to be associated with disease. It is probably a better strategy to bin similar haplotypes together since they are more likely to also be similar at the trait locus. Ideally, all haplotypes in a bin will have the same allele at the trait locus and will therefore have the same risk of developing the trait. However without knowing the location of the mutation and the ancestral history, it isn't possible to know which haplotypes share the same risk and should be binned together. A discussion of attempts to address these problems is given in Section 1.1.3.

Appendix B

Mutation rate of a pre-ascertained SNP

Let $X = 1$ if a mutation event occurs at a locus in a generation and 0 otherwise. Assume that mutation events in a single generation have a Bernoulli distribution with probability p. We would like to compute the probability of a mutation event at the locus in a single generation given that we know that at least one mutation event occurred at that locus in the history of a sample of n sequences. Let Y be the number of mutations occurring at the locus in the history. Therefore, we would like to find

$$p_s = \Pr(X = 1|Y \geq 1) = \frac{\Pr(X = 1)}{\Pr(Y \geq 1)} = \frac{p}{\Pr(Y \geq 1)}. \tag{B.1}$$

If we knew the history, τ and Ω, then we could compute $\Pr(Y \geq 1)$. Letting $T_{total} = \sum_{i=2}^{n} i\omega_i$ be the total tree time measured in generations, the number of mutation events at the locus over T_{total} generations has a binomial distribution with probability μ and T_{total} trials. Therefore,

$$\Pr(Y \geq 1) = 1 - \Pr(Y = 0) = 1 - (1 - p)^{T_{total}}. \tag{B.2}$$

However, since we do not know T_{total}, we integrate over the intercoalescence times. From the coalescent model, the i^{th} intercoalescence time has an exponential distribution with rate $\frac{\binom{i}{2}}{2N}$, where N is the effective population size. Therefore,

$$\Pr(Y \geq 1) = \int_{\boldsymbol{\omega}} \Big(1 - (1 - p)^{\sum_{i=2}^{n} i\omega_i}\Big)\Big(\prod_{i=2}^{n} \frac{\binom{i}{2}}{2N} \exp(-\frac{\binom{i}{2}}{2N}\omega_i)\Big) d\omega. \tag{B.3}$$

Since p is on the order of 10^{-8}, we can use the approximation that $1 - p \approx \exp(-p)$ when p is close to 0. Equation (B.3) can then be written as

$$\begin{aligned}
\Pr(Y \geq 1) &= \left(\prod_{i=2}^{n} \frac{\binom{i}{2}}{2N}\right) \int_{\boldsymbol{\omega}} \left(1 - \exp(-p\sum_{i=2}^{n} i\omega_i)\right) \exp(-\sum_{i=2}^{n} \frac{\binom{i}{2}}{2N}\omega_i) d\omega. \\
&= 1 - \left(\prod_{i=2}^{n} \frac{\binom{i}{2}}{2N}\right) \int_{\boldsymbol{\omega}} \exp\left(- \sum_{i=2}^{n} (ip + \frac{\binom{i}{2}}{2N})\omega_i\right) d\omega. \\
&= 1 - \left(\prod_{i=2}^{n} \frac{\frac{\binom{i}{2}}{2N}}{ip + \frac{\binom{i}{2}}{2N}}\right) \\
&= 1 - \prod_{i=2}^{n} \frac{i-1}{i-1+4Np}. \qquad \text{(B.4)}
\end{aligned}$$

Combining equations (B.1) and (B.4) gives

$$p_s = \frac{p}{1 - \prod_{i=2}^{n} \frac{i-1}{i-1+4Np}}. \qquad \text{(B.5)}$$

In our model, we assumed that mutation events at the polymorphic loci were Poisson distributed with rate $\theta_s/2$ per unit of coalescent time or μ_s per generation. Assuming the Poisson model, the probability of at least one mutation event at a site known to be polymorphic is

$$1 - \exp(-\mu_s) \approx \mu_s$$

using the approximation for μ_s when μ_s is small. Therefore, the probability of a mutation event in a single generation, p_s found in equation (B.5) is also approximately the rate of the Poisson distribution, μ_s, provided that the rate is small.

The simulated data in Section 3.4 were generated assuming a per site per generation mutation rate of $\mu = 10^{-8}$ and $N = 10000$, which leads to $\theta = 4N\mu = 0.0004$. Using equation (B.5) gives $\mu_s = 5.59 \times 10^{-06}$ and $\theta_s = 0.22$. However, in our model, half of all mutation events lead to a different allele. Since in generating the data all mutation events lead to a new allele, our estimated value for θ_s should be two times the value used to generate the data. In our simulated datasets, the estimated θ_s values are typically around 0.1, which is much lower than 0.44. Recall, however, that our datasets only included loci where the minor allele frequency was greater than or equal to 0.1. We would therefore expect θ_s to be lower than 0.44.