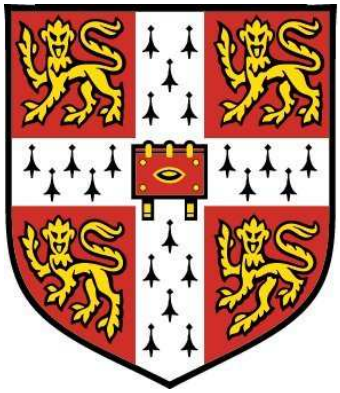
A Bayesian approach to Nested Clade Analysis

Ioanna Manolopoulou

Trinity College and Statistical Laboratory

University of Cambridge



A thesis submitted for the degree of
Doctor of Philosophy
September 2008

Abstract

The purpose of this study is to identify genetically distinct clusters of individuals based on related characteristic traits (namely phenotypic data) or geographical locations (namely phylogeographic data). There are 2 main steps to this process: inferring the genetic history of the sequences under study, and subsequently identifying significant clusters according to the phenotypic/phylogeographic measurements. Based on an evolutionary model and an appropriate model for the distribution of the phenotype, such inference is possible in a number of different ways. However, due to the multiple level uncertainty and the complexity of the models, it is essential that the methods avoid stepwise optimization in order to give statistically reliable conclusions.

The main methods currently used for analysis of this type are called Nested Clade Analysis (NCA) and Nested Clade Phylogeographic Analysis (NCPA) for phenotypic and phylogeographic data respectively. In short, they rely on finding the optimal genetic history based on a simplified evolutionary model, and identifying significantly different clusters for the phenotype/geography (assuming the inferred genetic history as fixed) by using Nested Analysis of Variance and permutation tests. Such methods do not allow for the uncertainty of each step to fully propagate through the model and have been shown by simulations often to lead to false conclusions.

Here we describe a coherent statistical framework for NCA/NCPA by taking a (Reversible Jump) Markov chain Monte Carlo approach to the genetic clustering problem. By considering a general evolutionary model and clustering constructions using haplotype trees for the phenotypic and phylogeographic analysis respectively, we construct a holistic method in order to obtain the global optimum of the parameters of interest.

Several challenges arise in this process. The presence of homoplasy (representing convergent evolution, usually through back mutations) can obscure the analysis, increasing the number of possible histories that underly the data. This leads to intractable likelihoods and normalisation constants. Here we use Approximate Bayesian Computation to address these issues. In addition, the parameter space of clusterings is vast, so we employ adaptive methods and efficient proposals to ensure mixing and convergence. Lastly, we address inherent issues of similar clustering and phylogenetic inference problems such as label-switching (for the cluster parameters) and representation of trees (essential for convergence assessment). We implement our method for 3 datasets and discuss the results in relation to NCA and NCPA.

Acknowledgements

First of all I would like to thank my supervisor, Professor Simon Tavaré, for offering to supervise me and always providing his encouragement and inspiration. I would also like to thank my former supervisor, Professor Steve Brooks, for his help during the first two years of my PhD, leading to our paper Brooks et al. (2007). I am very lucky to have worked under their guidance.

Much of the biological aspect of my work was motivated and carried out by my collaborators Dr Brent Emerson and Dr Lorenza Legarreta at the University of East Anglia and Dr Neil Gemmell at the University of Otago. Specifically, Brent and Lorenza collected the phylogeographic datasets presented in Chapter 4, and helped me develop the phylogeographic methods of Chapter 2. Some of our collaborative work is included in our joint papers; see Manolopoulou et al. (2008), Legarreta et al. (2008). Neil provided me with the phenotypic dataset analyzed in Chapter 4 and offered me the biological interpretation of the results. I am very grateful to all of them for inspiring me to work on this project and for being am endless source of biological insight.

This work was made possible by a studentship from Trinity College. Special thanks go to my Tutor, Professor David McKitterick, who was always generous with his time and resources. I would like to thank everyone in the Cambridge University Statistical Laboratory for making it such an enjoyable place to work. I am also eternally grateful to all my friends, and especially to Olli, Sophie and David, without whose support and help I would never have been able to complete this thesis. Finally, I would like to thank my family who always encouraged and supported me throughout my education.

This dissertation is the result of my own work and contains nothing which is the outcome of work done in collaboration with others, except where specifically indicated in the text. This dissertation has not been submitted for any other degree or qualification at any other university.

Contents

Chapter 1

Introduction

The key motivation for this study is the popularity of Nested Clade Phylogeographic Analysis (NCPA) amongst evolutionary biologists, despite its frequent criticism (see Petit and Grivet, 2002; Knowles, 2004; Panchal, 2007; Petit, 2008; Panchal and Beaumont, 2007), and the need to provide a solid statistical framework for the existing methodology so as to draw statistically reliable inferences from phylogeographic data. NCPA is a statistical method for reconstructing the demographic history of spatially distributed populations from genetic data (see Templeton, 1998). Following an introduction to the methods hitherto employed (see Chapter 1), we take a coherent model-based Bayesian approach to NCPA in order to draw inferences about the geographical clustering (see Chapter 2) and the phylogeny simultaneously (see Chapter 3), using Markov chain Monte Carlo (MCMC) and Approximate Bayesian Computation (ABC). This approach is applicable to both the phenotypic and phylogeographic clustering problems described below, by plugging in different models for the distribution of the data. We present our method and implement it by applying it to two phylogeographic datasets from beetles and weevils respectively, and a phenotypic dataset from salmon (see Chapter 4). The thesis concludes with a recapitulation of the findings presented, together with recommendations for future work in the field (see Chapter 5).

In **phenotypic** cluster analysis, the objective is to identify Single Nucletotide Polymorphisms (SNPs) in DNA sequences which are associated with changes in characteristic traits. Drawing inferences about population structure and subdivision in combination with a phenotype can yield valuable information for SNP analysis (see Abecasis et al., 2007). Although SNPs are rarely solely responsible for the expression of a characteristic (such as disease), partly due to the presence of linkage disequilibrium (see Nelson, 2001), it is often the case that associations can be made. For example, there is increasing evidence that there is an association between mitochondrial SNPs and fertility (see Montiel-Sosa et al., 2002). One of the existing methods of phenotypic cluster analysis is Nested Clade Analysis, developed by

Templeton et al. (1987).

In **phylogeographic** clustering problems, the objective is to draw conclusions about the geographical history of a population (of the same species) by identifying geographically and genetically distinct population clusters. The main method used for analysis of this type is NCPA which was developed by Templeton (1998). Other methods aim at identifying subpopulations using migration models in combination with the coalescent (see De Iorio and Griffiths, 2004a,b). Although reliable inference for human populations usually requires very complex models (see Jow et al., 2007; Liu et al., 2006), for many other species often simpler models can be used, allowing for computationally less intensive methods.

Both clustering problems described above may be addressed simultaneously within the same statistical framework. Given a dataset comprising a DNA sequence and a measurement (phenotypic or geographical) for each individual, we want to infer the genetic history of the individuals and identify clusters of phenotypes/geographical locations that are consistent with the phylogeny. This requires a model for the evolutionary process underlying the sampled DNA sequences and a model for the structure and distribution of the clusters, which can subsequently be used to draw inferences using Bayesian computational methods.

In this Chapter we introduce the necessary tools for the Bayesian approach described in subsequent chapters. To this end, we present some relevant molecular genetics (see Section 1.1), including the mutation process, the coalescent model and various graphical representations of intraspecific genetic relationships between individuals. For a detailed explanation of the biology involved in population genetics, see Balding (2003). These are necessary for the phylogenetic side of the analysis, i.e., inferring the genetic history. We then present several methods that have been proposed for inferring phylogenies (see Section 1.2), and describe the approach taken in NCA in detail.

In Section 1.3 we describe a number of approaches to identifying associations between DNA changes and phenotypes based on the inferred tree (e.g. Templeton et al., 1987, 2005; Posada et al., 2005). Subsequently, we present a few different methods of analyzing phylogeographic data (Section 1.4), some of which are based on tree inference (Handley et al., 2007; Templeton, 1998; De Iorio and Griffiths, 2004a,b) and some which are not (e.g. Falush et al., 2003). Since NCPA is widely used by biologists, we focus on the methods developed by Templeton et al..

Finally, in Section 1.5 we introduce the basic principles of Markov chain Monte Carlo and Approximate Bayesian Computation methods and describe some standard techniques of overcoming common problems associated with the convergence of MCMC samplers.

1.1 Overview of genetics

All living organisms have a DNA sequence of nucleotides A, G, C, T (representing the four nucleotide subunits adenine, guanine, cytosine, and thymine bases) which contains the genetic instructions used for their development and function. Each nucleotide position is called a nucleotide site. In a sample of DNA sequences, a nucleotide site which is not identical across all sequences is called a Single Nucleotide Polymorphism (SNP). Although for most species more than 99% of their DNA sequences is the same across the population, variations in DNA sequences can have a major impact on the expression of a phenotype.

Within cells, most DNA is organized into structures called chromosomes, which in eukaryotic organisms (animals, plants and fungi) are stored in the cell nucleus. Genes are hereditary units found at specific loci on each chromosome. Chromosomes are duplicated before cells divide in a process called DNA replication, and genes are passed onto offspring. During reproduction, diploid eukaryotes generate offspring that contain a mixture of genetic material inherited from two different parents.

In the process of reproduction, two events may occur which may increase the genetic variation. Firstly, **mutations** may occur, usually through an error during copying of the DNA strand, resulting in changes in the DNA sequence. Secondly, chromosomal **recombination** may occur, referring to crossover between the paired chromosomes during meiosis. This leads to offspring having different combinations of genes from their parents.

Although most DNA present in eukaryotic organisms is contained in the cell nucleus, mitochondrial DNA (mtDNA) is located in organelles called mitochondria which are found in the cytoplasm. Unlike nuclear DNA, which is inherited from both parents and in which genes are rearranged in the process of recombination, mitochondrial DNA is maternally inherited. As a result, and owing to the faster mutation rate compared to nuclear DNA, mtDNA is a powerful tool for tracking ancestry through females. Throughout this thesis, we use mitochondrial data. Their haploid nature and maternal inheritance allow us to use simpler evolutionary models and to relax some model assumptions, so that the complexity of the phylogenetic problems is greatly reduced.

In practice, usually only a small region of the DNA sequences is studied. Distinct sequences are called haplotypes; note that the number of sequences in a sample may be larger than the number of haplotypes. Although no two individuals ever have identical DNA sequences throughout their length, it is frequently the case that two individuals will share quite long stretches of DNA.

The evolutionary process underlying the region under study can be viewed in the following way; see Wright (1951). As organisms reproduce, three possible events occur independently at

different rates under no selection: new sequences arise through mutation and recombination, sequences are replicated when no mutations or recombination occur, and others disappear through extinction. The study of the evolutionary history of individuals, whether of the same species (intraspecific) or between different species (interspecific) is called phylogenetics (see Nei and Kumar, 2000; Semple and Steel, 2003). In this thesis we are concerned with datasets of the same species, and hence we mainly discuss intraspecific inferences. Several approaches have been presented, attempting to draw reliable inferences and addressing many of the challenges of population genetics (see Wakeley and Hey, 1997; Wilson et al., 2003; Nei and Kumar, 2000).

The evolutionary history of a sample of sequences may be represented in a number of ways through a graph, depending on the objective of the analyses (see Hein et al., 2005; Rosenberg and Nordborg, 2002). We describe two main ones, namely haplotype trees and rooted ancestral trees.

Haplotype trees The haplotype tree[1] depicts the mutational steps that relate the observed haplotypes to one another. Each node in the tree represents a haplotype, and two haplotypes are connected by an edge if they are one mutation apart. Although a haplotype tree may also be rooted, meaning that the oldest haplotype is represented in the graph, a haplotype tree generally provides little or no information about the relative time-scale of mutation and replication events.

A haplotype tree may also include information about the number of times each haplotype appears in the sample, so that it contains information about sequences and not just haplotypes. In fact, in that case the haplotype tree provides all information available in the sequence data; the haplotype tree determines the number of times each haplotype appears in the sample, and specifies all missing intermediate (extinct or unsampled) sequences. Although in the absence of additional data (such as a phenotype, geographical location, pedigree information) it is not possible to distinguish between copies of the same haplotype (see Posada and Crandall, 2001), the number of copies itself is a useful source of information about history of each haplotype in relation to the rest of the sample.

An alternative form of trees which are very similar to haplotype trees does not rely on an evolutionary model, but is simply based on the phenetic distances between sequences. Such networks are Neighbour-Joining graphs, which connect sequences that are closer together, with appropriate weightings (see Gascuel and Steel, 2006; Atteson, 1997, 1999). In practice, they have the same structure as a haplotype tree, but a different method of obtaining the

[1] In earlier work, Templeton et. al. refer to haplotype trees as cladograms, but in their recent literature the term haplotype tree is used. They are also occasionally referred to as haplotype networks or mutational trees (see Sankoff, 1975).

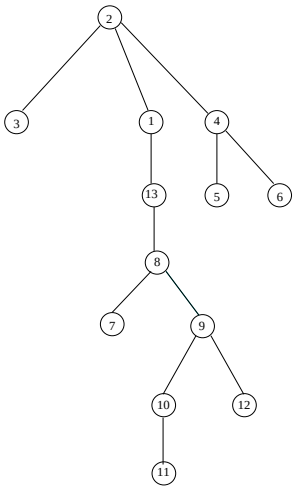


Figure 1.1: *Example of a haplotype tree. In this figure, initially there is one haplotype present, named 2. It is inherited by its descendants, until at some point random mutations occur, and haplotypes 1,3,4 appear (in any order). This process continues until eventually we have 13 haplotypes in the sample.*

optimum one for each case.

Rooted genealogical trees In contrast to haplotype trees, a rooted genealogical tree (also sometimes called a gene tree) is a binary tree showing the timewise evolutionary relationships among a set of individual sequences (rather than haplotypes) of the same species. Each internal node represents the most recent common ancestor (MRCA) of its descendants, and the edge lengths correspond to time estimates. This implies that the root of the tree represents the MRCA of the sample. Looking at evolution backwards in time, the time taken for two individuals to reach their MRCA is called the time to coalescence, and equivalently the event is called a coalescence event. Considering the process forwards in time, the equivalent event of a node splitting into two lineages is called a divergence event.

In the case of *interspecific* datasets (meaning datasets involving different species), rooted genealogical trees are called phylogenetic trees. In *intraspecific* population genetics datasets such as the ones addressed in this thesis, rooted genealogical trees usually assume some version of the coalescent model described in Subsection 1.2.1, and are thus named coalescent trees. A typical coalescent tree has the following binary form shown in Figure 1.2.

The information obtained from the tree is the order in which coalescence events occured, as well as the relative times at which they happened. However, precise temporal locations of mutations cannot typically be extracted, and the exact states of unobserved ancestral sequences is not usually inferred (see Wilson et al., 2003). A variation of the usual coalescent is one where the states of unobserved ancestral sequences are determined (see Li et al., 2000),

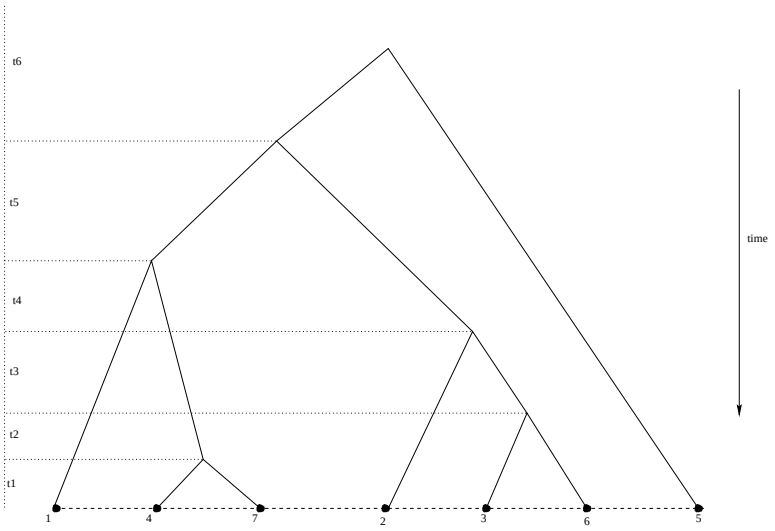


Figure 1.2: *Example of a coalescent tree. In this tree, nodes 4 and 7 are the most closely related, having their Most Recent Common Ancestor (MRCA) time t_1 ago. The second most recent coalescence event was between sequences 3 and 6, which coalesced a time $t_1 + t_2$ ago. The process continues until the top of the tree, where the MRCA of all sequences appears, a time $\sum_i t_i$ ago.*

or the precise times when specific mutations occurred are specified (see Markovtsova et al., 2000).

1.1.1 Challenges in tree estimation

Haplotype and coalescent trees are both used to summarize the evolutionary history of a sample of sequences. In practice, the true tree is unknown, as is the underlying evolutionary process generating the sequences. Drawing inferences about the tree is difficult for many reasons:

- The exact model of the evolutionary model is not known. Although model-based inference is possible, assessing model-fitness is very difficult.
- The parameters of the evolutionary model are not known and have to be estimated. Owing to the complexity of the model and sparsity of the data, such estimates are rarely accurate, and there is often great variation between different samples.
- Often sequences which existed in the past have gone extinct, or are not sampled, and there are multiple possibilities of what they may have been; see Figure 1.3.
- **Homoplasy** is the result of convergent evolution and is present when sequences are similar, but are not derived from a common ancestor.

The presence of homoplasy (e.g. due to back-mutations) hugely increases the number of possible histories consistent with the data. An example of homoplasy is given in Figure 1.4. Homoplasy leads to the presence of loops (called reticulations) in a haplotype tree, as displayed in the example. There have been several approaches to address reticulate evolution; see Xu (2000).

- The **parsimony** assumption implies that if two sequences are one DNA change apart, they are assumed to be one mutation apart. Although it can greatly reduce the computational complexity of ancestral tree inference, it may be untrue when homoplasy is present. Even if all sequences are sampled and none has gone extinct, it is not possible to determine the history uniquely, since the parsimony assumption can never be verified. For example, we cannot verify that 3 sequences arose as in the left-hand panel of Figure 1.5 rather than the right-hand one.

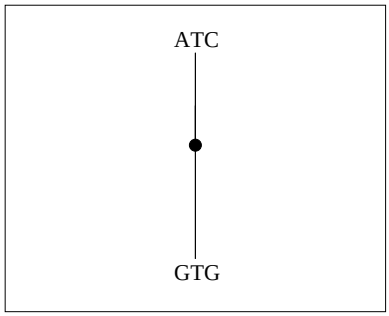


Figure 1.3: *Example of unobserved ancestral sequence: even if we know that precisely two mutations occurred between the two sequences shown, it is not possible to know in which order they occurred, and thus it is not possible to determine whether the unknown intermediate sequence is GTC or ATG.*

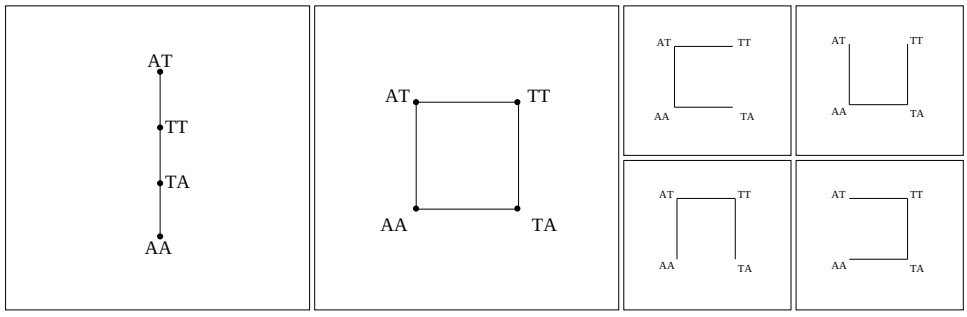


Figure 1.4: *Example of homoplasy: the tree on the left represents the true mutational history. However, given the data, it is not possible to distinguish between the 4 possibilities given by the four sub-trees shown on the right of the network in the middle.*

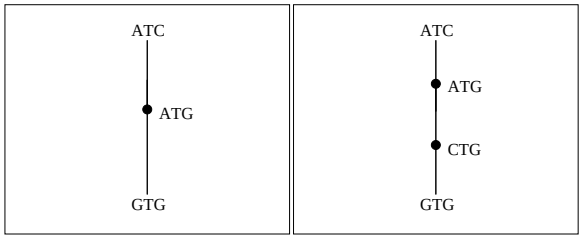


Figure 1.5: *Example of unknown 2 possible mutational histories. In the absence of homoplasy, the only possible mutational history which could yield the observed three sequences is the one on the left. However, homoplasy cannot be checked, and it is not possible to know whether there was an unobserved mutation as in the figure on the right.*

1.1.2 Coalescent trees versus haplotype trees

Both coalescent and haplotype trees are used to represent the evolutionary history of sequences, but they focus on different aspects of the process.

Haplotype trees specify every ancestral sequence that has been present in the evolutionary history of interest, including those that may be extinct, but provide no information about the event times; see Posada and Crandall (2001). They usually rely on the parsimony principle, which is often regarded as an inherent disadvantage (see Felsenstein, 1978). This means that sequences are generally assumed to evolve according to the "minimum-evolution" principle (see Rzhetsky and Nei, 1993; Desper and Gascuel, 2002). When multiple sequences corresponding to the same haplotype are present in the sample, they are simply collapsed onto their haplotypes. If a mutation is undetected (i.e., a mutation which occurred is not depicted on the haplotype tree), this will usually not affect the shape of the haplotype tree (see Templeton et al., 1987). However, in the presence of homoplasy, or when datasets contain very distantly related sequences (this phenomenon is called deep divergence), the parsimony assumption may lead to false conclusions; see Felsenstein 1983).

In addition, haplotype trees do not explicitly draw inferences about the ancestral haplotype. Although there have been some studies on root inference (see Castelloe and Templeton, 1994), rooting a haplotype tree is still a challenging problem. In some cases it is possible to infer the oldest haplotype by using a haplotype of a different species (called an outgroup) which is known to be ancestral to the taxon under study. The haplotype of the sample which is genetically closest to the outgroup is assumed to be the ancestral haplotype; see Maddison et al. (1984). However, it is often not possible to determine the genetically closest haplotype and hence outgroups cannot always be used.

On the other hand, coalescent trees give a precise order and time frame of all divergence events, but do not usually specify the mutations which occurred in history. Sequences are not collapsed onto haplotypes, which allows homoplasy of whole sequences (i.e., two sequences

are identical by state, but not identical by descent) to be detected. When event times are of interest, coalescent trees are more reliable and can yield more accurate conclusions (see Felsenstein, 1978) based on reasonably simple models. However, the computational complexity of coalescent trees is frequently prohibitive. A number of very sophisticated statistical tools have been developed to reduce the computational complexity of coalescent tree inference (see Huelsenbeck and Ronquist, 2001), but they usually do not allow for specific mutational steps to be determined, and they often exhibit extremely long run times.

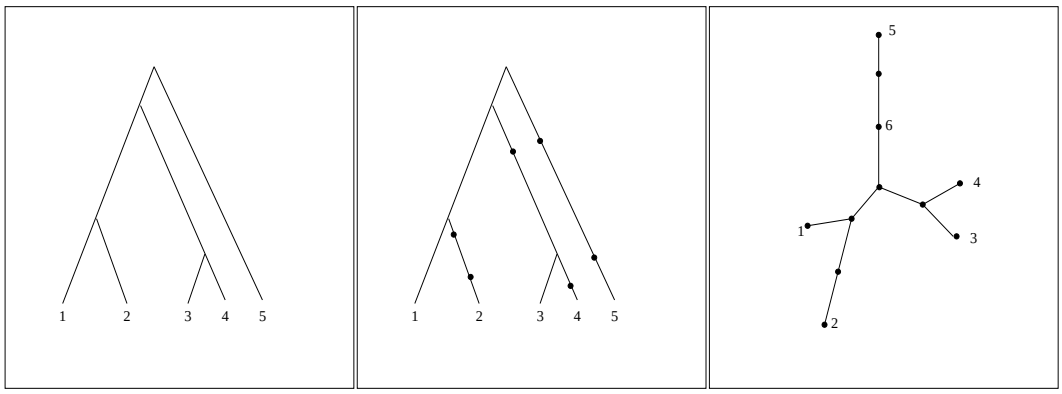


Figure 1.6: *Examples of a typical coalescent tree, a coalescent tree specifying mutations, and a haplotype tree. In the figure on the left, the history is summarized by a coalescent tree. In this example, individuals 3 and 4 are the ones which are the most closely related with their MRCA being the youngest one in the sample. However, it is not possible to identify the sequence state of their MRCA, since any number of mutations may have occurred along any of the branches of the tree. The middle figure shows a coalescent tree which also specifies mutations. In this example mutations are shown on the tree by small black dots on the branches. In this case the MRCA of 3 and 4 has the same type (i.e., sequence) as sequence 3, whereas sequence 4 is one mutation apart. On the right, we show an example of a haplotype tree, where unnumbered nodes represent sequences which are not sampled. We cannot distinguish the relative times in which events occurred. However, we can determine the precise number of mutations which occurred, as well as the relative genetic similarity of individuals.*

In other words, although coalescent trees theoretically allow for more accurate inference, they can be computationally prohibitive, with haplotype trees being the only feasible alternative.

1.2 Inference about the tree

Before describing particular approaches, we give a brief overview of coalescent theory (Subsection 1.2.1) and discuss several properties of mutation models (see Subsection 1.2.2), many of which are used in the methods described subsequently.

We then (Subsections 1.2.3 - 1.2.6) present a number of different methods which have been

developed (see Griffiths and Tavaré, 1995; Felsenstein, 2001, 2003, 1983; Templeton et al., 1992; Meligkotsidou and Fearnhead, 2005; Tavaré, 1986; Desper and Gascuel, 2002; Gascuel and Steel, 2006; Atteson, 1997, 1999) in order to draw inferences about the haplotype or coalescent tree relating to the sequence data in question. We divide the methods into four categories: coalescent-based, distance-based, maximum-likelihood and parsimonious methods. Many of these are summarized and compared by Kuhner and Felsenstein (1994); Makarenkov et al. (2006); Holder and Lewis (2003); Stamatakis (2004).

1.2.1 The coalescent

Wakeley (2008) noted that "Coalescent theory provides the foundation for molecular population genetics and genomics". Coalescent theory, first developed by Kingman (1982), is a retrospective model relating a set of sequences back to their MRCA through a series of coalescence events. It is regarded as an fundamental model of population genetics.

Kingman (1982) described evolution by viewing it backwards in time, based on the assumptions of constant population size and random mating. In a sample of N sequences, the time to the next coalescence event which reduces it to $N-1$ sequences is proportional to $\binom{N}{2}$.

The coalescent does not model the mutation process. However, given a mutation model, the two can be combined to express the probability of a coalescent with mutations. Assuming that mutations occur independently as a Poisson process at rate $\theta/2$, they can be thought of as being poured down the coalescent tree (see Tavaré, 2003).

Using the coalescent model with mutations, it is possible to simulate a sample from a coalescent tree with mutations of size N through the following algorithm which is due to Ethier and Griffiths (1987):

Algorithm 1.2.1.

1. To start with, choose an initial DNA sequence from the stationary nucleotide distribution π, and immediately split that node. This is because the first event necessarily has to be a split rather than a mutation. If that were not the case, then the MRCA of the sample would be the mutated sequence, which breaches the assumption that the root of the tree is the MRCA of the sample.

2. Thereafter, if there are k lines in the ancestry, select one at random. Wait an exponential amount of time with parameter $k(k-1+\theta)/2$, and then decide to split at that point with probability $(k-1)/(k-1+\theta)$ or mutate otherwise according to P (for example this may be given by (1.1)), the mutation process matrix.

3. Continue until there are $N+1$ lines, and throw away the last sequence.

Several extensions of the coalescent model have been developed, to account for a variable population size (see Slatkin, 2001), selection (see Neuhauser and Krone, 1997) and recombination (see Hudson, 1983).

1.2.2 Mutation models

A number of models have been developed to represent the mutation process (see Hein et al., 2005). There are several desirable properties for the representation of the process, which simplify inference and computation of likelihoods. We highlight the main ones.

A1 No recurring mutations, which implies that no more than one mutation occurs at the same nucleotide site. This is the key assumption in the infinitely-many-sites model, but it may be invalid since recurring mutations do occur.

A2 Equal mutation rates across all sites. Mutation rates across all sites are frequently not equal, since certain sites mutate are much more common than on others.

A3 Equal mutation rates between all nucleotides. Nucleotides A-G and C-T have similar chemical properties, and mutations within the two pairs are much more common than the remaining ones, so equal mutation rates between nucleotides is usually not a valid assumption.

A4 Models that assume independence of sites are usually constructed for statistical convenience, notwithstanding the fact that biological feautures (e.g. codon usage) render this assumption questionable.

A5 Stationarity of nucleotide frequencies in the ancestral sequence. This assumption implies that the relative frequency of the four nucleotides remains the same throughout time. There have been several studies on the applicability of stationary models (see Gu and Li, 1998)

A6 Time-reversibility of the chain is a stronger assumption than stationarity. It implies that not only are the nucleotide frequencies at equilibrium, but also that the mutation process is identical forwards and backwards in time.

A7 Models assuming parent-independent mutations imply that the probability of obtaining type j after a mutation is independent of the parent type i. This assumption is usually biologically unrealistic, but is used for statistical convenience.

A8 Models assuming that no selection is possible, which means that mutations cannot result in an individual having a higher probability of survival and hence a higher probability of its descendants prevailing in the population.

Here we present the most general time-reversible mutation model under assumptions A4, A5, A6 and A8, namely the Generalised Time-Homogeneous Time-Reversible model (REV) (see Tavaré, 1986). We consider L (the length of the sequences) parallel independent mutation processes and represent the state of each nucleotide sire l of sequence i as X_l^i. Mutations occur as a Markov Process with generator Q-matrix

$$Q = \phi_j \begin{pmatrix} \cdot & v_1\pi_G & v_2\pi_C & v_3\pi_T \\ v_1\pi_A & \cdot & v_4\pi_C & v_5\pi_T \\ v_2\pi_A & v_4\pi_G & \cdot & v_6\pi_T \\ v_3\pi_A & v_5\pi_G & v_6\pi_C & \cdot \end{pmatrix}$$

where the π_is ($i = A,\ G,\ C,\ T$) represent the equilibrium probabilities of the nucleotides, and the mutation coefficients $v_1, \ldots, v_6$ the relative mutation probabilities. The extra parameter ϕ_j denotes the site-specific mutation rate for each site j. A Markov process at time t with generator matrix Q and initial distribution equal to the distribution δ_i (here δ is the Kronecker δ and represents a known initial state i at time 0) can be viewed as a Markov chain with transition matrix

$$P_{ij}^{(t)} \quad = \quad \{\exp(Qt)\}_{ij}\,. \tag{1.1}$$

This implies that mutations happen as a Poisson Process with rate $\phi_j q_i$ when at each state i in site j, where q_i is the sum along row i of the rates of jumping to other possible states.

In this model the v_is are important because certain mutations are more likely than others (for example as with transition/transversion bias), whereas the ϕ_is allow for different mutation rates between each nucleotide site, and represents the fact that certain sites mutate more frequently than others, but the relative probabilities of mutating to each possible nucleotide remain the same. From now on for notational simplicity we refer to ($\pi_A,\ \pi_G,\ \pi_C,\ \pi_T$) as ($\pi_1,\ \pi_2,\ \pi_3,\ \pi_4$) respectively. This process is time-reversible since it satisfies the detailed-balance equations $\pi_i q_{ij} = \pi_j q_{ji},\ \ i, j = 1, \ldots, 4$ (see Norris, 1997), which is a sufficient (and necessary) condition for time-reversibility.

1.2.3 Coalescent-based Bayesian methods

Inference about the coalescent tree is a challenging problem. Here we present a number of Bayesian model-based methods which have been developed. In order to draw likelihood-based inferences on coalescent trees, calculating the probabilities of trees is essential. After describing the peeling algorithm which enables the calculation of the probability of a coalescent tree,

the key difficulty becomes to devise an efficient method of exploring the space of possible trees within the search algorithm used, whether this is Importance Sampling (IS), Markov chain Monte Carlo or Approximate Bayesian Computation (ABC). In Subsection 1.2.3 we present some importance sampling techniques (including some valuable approximations) have been suggested. We continue with some MCMC techniques, concentrating on tree representation and moves, since these are two the main challenges that will become relevant later. More recently, Approximate Bayesian Computation (ABC) methods have been implemented to address intractable likelihood issues (see Beaumont et al., 2002), presented in Subsection 1.2.6.

The peeling algorithm

Although simulating from a coalescent tree is relatively straightforward, directly calculating the probability of a given tree of size N with sequences of length L is not immediately possible. Assuming that nucleotide sites evolve independently, the likelihood is the product of L terms, so the likelihood calculation is of complexity $O(L)$. Since most of the sites are not variable, this can be reduced to $O(m)$ where m is the number of SNPs, with $m \leq N$. Using the Markovian nature of the coalescent process, the evaluation of the likelihood can be split into L steps using a peeling (also called pruning) algorithm (see Felsenstein, 1983). Specifically, let u_i denote an unknown nucleotide in the ancestral sequence which is represented by node i of the tree. We label the two descendant sequences of u_i as $A(i)$ and $B(i)$. By independence, on a given tree topology T and given u_i we have:

$$\mathbb{P}(A(i), B(i) \,|\, u_i, T) = \mathbb{P}(A(i) \,|\, u_i, T) \times \mathbb{P}(B(i) \,|\, u_i, T),$$

where the probabilities may be readily calculated using $P_{ij}^{(t)} = \{\exp Qt\}_{ij}$. Then the total probability of the tree can be written as

$$\sum_{\text{nodes}} \prod_{\text{nodes}} \mathbb{P}(A(i), B(i) \,|\, u_i, T),$$

summing over all the possible states of each internal node, and multiplying over all the probabilities of coalescence events (which are independent using the Strong Markov property). The peeling algorithm reduces the complexity of the above calculation from $O(N \times 4^L)$ to $O(N \times 4 \times L)$ by summing starting from the leaf rather than the root nodes and moving to the top of the tree, in other words starting with the innermost sum.

Calculating the exponential of a matrix may be carried out in many different ways (see Van Loan, 1978; Moler and Van Loan, 2003), many of which fail at singularities (e.g. repeated eigenvalues) or do not always form a converging series. In this case owing to the special

property of Q having zero row sums, we use Poisson embedding in order to calculate $\exp\{Qt\}$ by transforming Q into a stochastic matrix which has a convergent series.

We express the probability as

$$\exp(Qt) \quad = \quad \exp\left\{q_{\max}t\left(\frac{Q}{q_{\max}}+I\right)\right\}\exp\left(-q_{\max}tI\right)$$

where $q_{\max} = \max_i |\, q_{ii} \,|$.

Since the Q-matrix has the property that its rows sum to 0, the matrix $\frac{Q}{q_{\max}} + I$ will be stochastic, as will $\left(\frac{Q}{q_{\max}}+I\right)^n$ for any integer n. We can then write down the Taylor expansion of

$$\exp\left\{q_{\max}t\left(\frac{Q}{q_{\max}}+I\right)\right\}$$

as

$$\exp\left\{q_{\max}t\left(\frac{Q}{q_{\max}}+I\right)\right\} \quad = \quad \sum_i \left(q_{\max}t\right)^i \frac{\left(\frac{Q}{q_{\max}}+I\right)^i}{i!}.$$

Since $(\frac{Q}{q_{\max}} + I)^i$ is stochastic, it will clearly be bounded for any i. Moreover, $\frac{(qt)^i}{i!} = \prod_{1\le j\le i}\frac{qt}{j} \to 0$ as $i \to \infty$, because for large values of i the most terms of the product will be less than 1 and tending to 0. Hence we see that

$$\left(q_{\max}t\right)^i \frac{\left(\frac{Q}{q_{\max}}+I\right)^i}{i!} \to 0 \qquad \text{as } \; i \to \infty.$$

Thus, we can use the first few terms of the power series to find an approximation of the probability.

Using the time-reversibility of the evolutionary process, it can be shown that the likelihood of the tree is independent of the location of the root. For example, considering the two descendant braches of the root, subtracting t from the time to the next split of one and adding it to the time to the next split of the other one preserves the likelihood. In effect, we can view this as "picking up" an unrooted tree from any point on its branches without affecting the likelihood. Felsenstein (1983) called this the Pulley Principle.

Importance sampling approaches

One of the first methods of inferring phylogenies using importance sampling (see Section 1.5) is described in a number of papers by Griffiths and Tavaré and is implemented in the software genetree available at `http://www.stats.ox.ac.uk/~griff/software.html`; see Grif-

fiths and Tavaré (1994a,b, 1995). Their method was subsequently improved with a more efficient proposal (see Stephens and Donnelly, 2000) by introducing an approximation for the stationary distribution of the mutation process. Later on De Iorio and Griffiths (2004a) generalized the approximation by showing that it can be viewed as a diffusion-process generator.

The algorithm by Stephens and Donnelly (2000) is defined for a sample of N chromosomes of two distinct types α and β, but can easily be generalised for SNPs by introducing four types and replacing chromosomes with nucleotide positions. The key approximation considered by Stephens and Donnelly (2000) and De Iorio and Griffiths (2004a) is based on the fact that the probability $\pi(\cdot \,|\, A_N)$ of selecting a random individual from a set of genetic types A_N and mutating it according to a mutation probability matrix P a geometric number of times with parameter $\frac{\theta}{N+\theta}$ can be approximated by

$$\hat{\pi}(\beta \,|\, A_N) = \sum_{\alpha \in E} \sum_{m=0}^{\infty} \frac{N_\alpha}{N} \left(\frac{\theta}{N+\theta} \right)^m \frac{n}{N+\theta} \, (P^m)_{\alpha\beta} \,,$$

where N_α is the number of chromosomes of type α in A_N. This approximation satisfies a number of important properties of the true distribution π.

Tree moves described by Newton et al. (1999)

One of the earliest Markov chain Monte Carlo algorithms (see Section 1.5) for inferring *phylogenetic* trees from a set of N DNA sequences was presented by Yang and Rannala (1997), and was later improved by Newton et al. (1999). Although here we are interested in coalescent trees, the tree representation and moves described are applicable to both phylogenetic and coalescent trees.

Concentrating on inference about the tree topology determined by a permutation parameter σ (determining the order in which sequences coalesced) and the divergence times t, we present the tree representation used by Newton et al. (1999) and describe the tree moves they developed, updated through a Metropolis-Hastings (MH) step.

The following representation of tree topologies is used, using nested parentheses, such as

$$top(\tau) \quad = \quad (((1,(4,7)),(2,(3,4))),5) \tag{1.2}$$

to represent the tree in Figure 1.7.

To account for the 2^{N-1} equivalent tree topologies, the convention when joining up two branches is to place the branch which contains the smallest number on the left, defining a *canonical ordering*.

In order to draw conclusions about the joint distribution of σ and t, a MCMC sampler

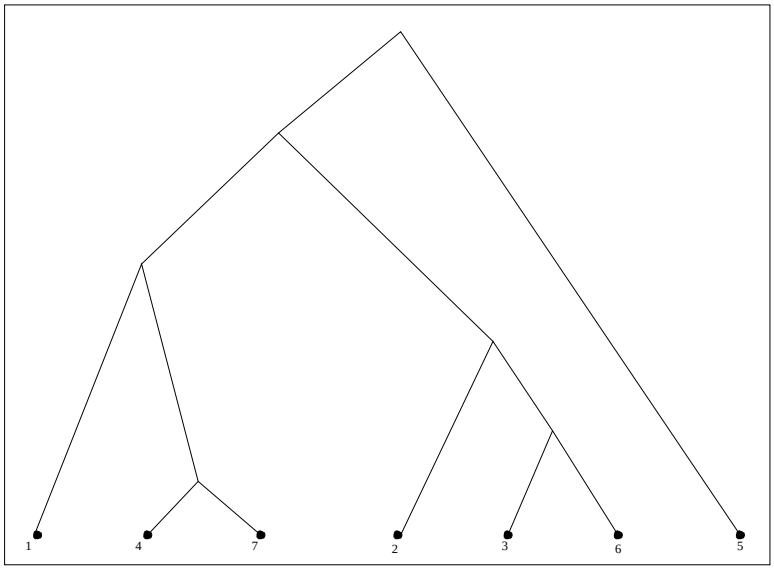


Figure 1.7: *The tree topology represented by (1.2)*

with target distribution

$$\pi(t, \sigma \mid D)$$

for sequence data D is constructed. The chain is initialized by a tree topology (t, σ), where σ is a permutation of $\{1, 2, \ldots, N\}$. Then they carry out the following update. Choose at random one of the 2^{N-1} equivalent trees, and perturb the inter-mutation times slightly according to a uniform random variable. In particular, starting from a vector of times t, generate a new vector t' of times by

$$t'_i = t_i \oplus \epsilon_i, \qquad \text{for } i = 1, 2, 3, ..., N-1$$

where ϵ_i are independent identically distributed Uniform$(-\delta, \delta)$ random variables for some tuning parameter δ, and $\oplus$ indicates addition reflected into the interval $(0, t_{\max})$. Although this changes the times only by a small amount, the change may alter the branching structure, yielding a very different tree topology. The proposed move is then accepted or rejected according to the corresponding MH ratio.

The above proposal method ensures that the tree proposed is quite 'near' the current tree. However such a proposal also allows for enough mixing as the candidate tree can have quite a different branching structure even though the likelihood will be similar.

A similar approach is presented by Li et al. (2000), where the ancestral nucleotide sequences are added as an auxiliary parameter, updated (rather than summed over) at each MCMC iteration.

Tree moves described by Mau et al. (1999)

As in the previous method, Mau et al. (1999) propose a similar algorithm for inferring phylogenetic trees from N DNA sequences from a slightly different perspective. Again, the tree representation and moves decribed can also be used for coalescent trees.

Evolution has two components that may be modelled as a stochastic process: the branching created by speciation and extinction to form a phylogeny, and the propagation of characters along the branches of that phylogeny. In the method by Mau et al. (1999) the phylogeny is treated as a parameter in a model for the propagation of data along each lineage.

A phylogenetic (or coalescent) tree may be viewed as a weighted tree Ψ, in which each edge has an associated positive weight. The branch lengths (edge weights) are the vertical distances between connected nodes. The ordering in which the mergings occur define coalescent levels, whereas the times at which these mergings occur denote coalescent times. Such a tree can be uniquely defined either by its labelled history and coalescence times as described by Newton et al. (1999) in the previous section, or by its topology and branch lengths, as described by Mau et al. (1999). The number of topologies and labelled histories grows rapidly with N, equal to $(2N-3) \times (2N-5) \times \ldots \times 1$ (inductively) and $N! \times (N-1)!/2^{N-1}$ respectively.

We form the matrix whose entries are determined by the within-tree distances between leaf nodes. Each permutation of the leaves generates a different matrix, and a rooted tree where all leaf nodes are equidistant from the root is called *cophenetic*. Clearly, such matrices are composed of at most N distinct entries. A cophenetic matrix with a canonical ordering has the important property that its super-diagonal (the diagonal of the sub matrix formed when deleting the first column and nth row) contains each distinct non-zero cophenetic distance. Below is an example of a canonical cophenetic matrix, describing the distances for the tree in Fig. 1.8, where coalescent times T are set at (0.8, 0.3, 0.7, 0.5, 0.9, 1.5):

	5	7	4	1	2	6	3
5	0	9.4	9.4	9.4	9.4	9.4	9.4
7		0	1.6	4.6	6.4	6.4	6.4
4			0	4.6	6.4	6.4	6.4
1				0	6.4	6.4	6.4
2					0	3.6	3.6
6						0	2.2
3							0

Here notice that $2.2 = 2 \times (t_1 + t_2)$ so that the distance between 3 and 6 is the vertical distance

that has to be travelled to get from 3 to 6 on the haplotype tree (so up and down again). 5 is the last one to coalesce and so the distance from all other nodes is equal to 9.4, which is twice the total height of the tree.

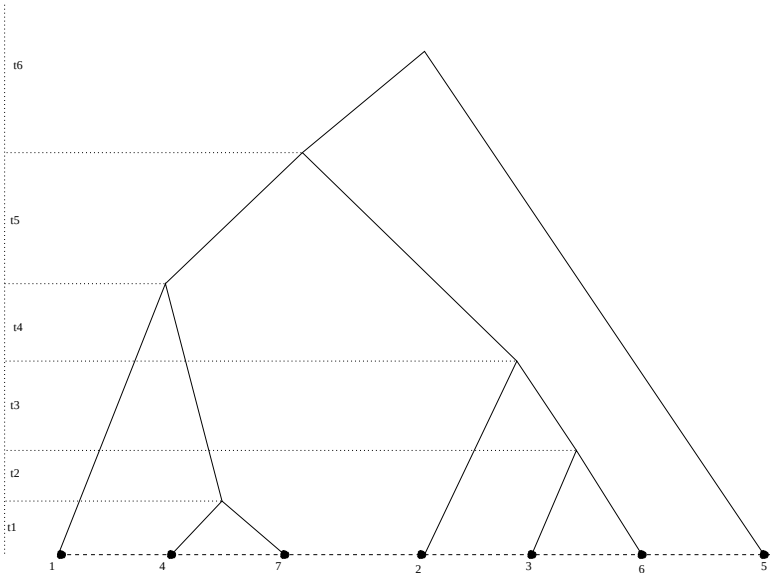


Figure 1.8: *Sample phylogeny on seven taxa*

The stochastic model used by Mau et al. (1999) describes the joint distribution of $y = \{y_v,\ v \in V = \mathcal{I} \cup \mathcal{L}\}$, the historical record at $\mathcal{I}$ and current status at $\mathcal{L}$ for a given site. This method describes a different way to represent phylogenies by considering the stochastic process above, and thus a different way of proposing and updating trees. More detail may be found in Newton et al. (1999).

Newton et al. (1999) propose a two-stage proposal distribution. The first stage uses the current tree Ψ to propose a canonical representation (σ, t), where t is the set of times-to-coalescence, and the second stage perturbs t. Specifically, by using a random binary variable at each of the $N-1$ internal nodes, a particular super-diagonal $\{d_{i,i+1} : i = 1, \ldots, n-1\}$ of a canonical cophenetic matrix is selected. Then, the elements of t are independently perturbed by a uniform ϵ, which can be selected to have small or large interval to moderate the acceptance rate.

Tree moves described by Larget and Simon (1999)

Larget and Simon (1999) summarized some of the existing MCMC methods suggested, and proposed a new approach for updating the trees, which is local rather than global. This is achieved by picking one of the internal edges (i.e., not connected to any of the leaves) at random, and rearranging the nodes to which it is connected according to a distribution which

based on the lengths of the edges replaced. More detail may be found in Larget and Simon (1999).

Altekar et al. (2004) suggest a similar Metropolis-Coupled Markov Chain Monte Carlo ($(MC)^3$) method which addresses slow mixing and getting stuck in local optima. The method is similar to simulated tempering. Two chains run in parallel, a "hot" and a "cold" one, and the overall chain jumps between the two. This approach is implemented by the software MrBayes (see `http://mrbayes.csit.fsu.edu`), one of the most sophisticated phylogenetic MCMC inference programs available; see also Huelsenbeck et al. (2002).

Tree moves described by Markovtsova et al. (2000)

A similar MCMC method is proposed in this article, implemented for a variety of different models. The basics steps are the same as in the subsections above, but an alternative sampler is proposed. We describe it in detail since it will be used later.

Algorithm 1.2.2.

1. Pick a level, l say ($l = N, N-1, \ldots, 3$), according to some proposal kernel.

2. For the chosen l observe the structure of coalescence at levels $l-1$ and $l-2$. There are two possible structure types, depending on whether the coalescence at level $l-2$ involves the line which results from the coalescence at level $l-1$. These are shown in Figure 1.9. When the structure has type A, the kernel randomly chooses one of the three possiblities shown in Figure 1.10. When level l has structure B, then the kernel always swaps to the symmetric structure as shown in Figure 1.11.

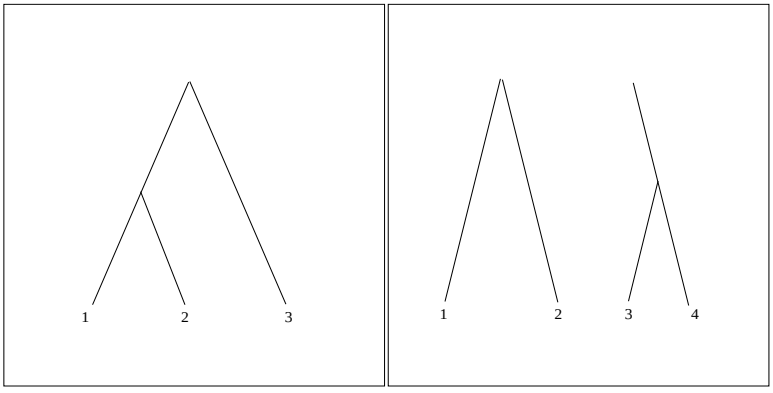


Figure 1.9: *The two structure types A and B for a coalescence level.*

3. Generate new times T_l' and T_{l-1}' according to an arbitrary distribution, and leave the other times unchanged. Thus only alter the times corresponding to the levels at which

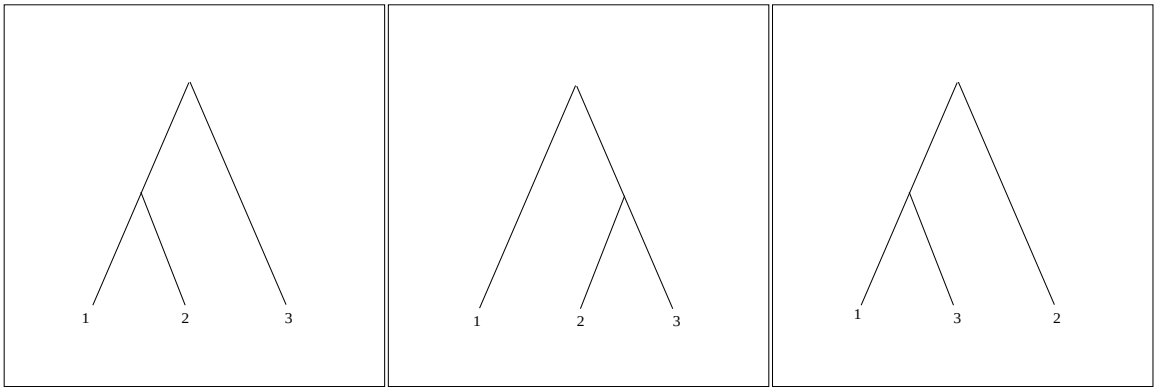


Figure 1.10: *The three possibilities of moving to, when the structure of level l is of type A. In this case the kernel randomly chooses one of the three.*

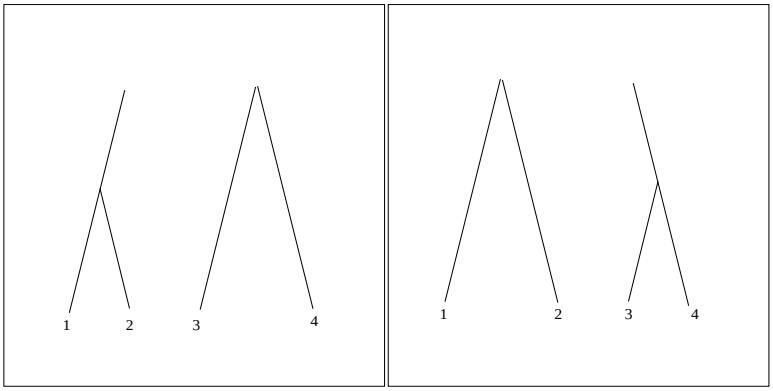


Figure 1.11: *The two possibilities of arrangements when the structure of level l is of type B. In this case the kernel always chooses to swap.*

the topology has been changed. This ensures that $(\Lambda', \boldsymbol{T}')$ is similar to $(\Lambda, \boldsymbol{T})$ and therefore has a reasonable probability of being accepted.

By using appropriate proposal distributions, the Hastings ratio can be simplified. Specifically, since pairs of lines are chosen uniformly to coalesce, all topologies are equiprobable a priori. Furthermore, if the updated times are generated from an exponential distribution with parameter $l(l-1)/2$ (where l represents the level), and mutation rates are proposed independently of the current values, the Hastings ratio becomes

$$\min\left\{1, \frac{\mathbb{P}(\mathcal{D} \,|\, G')}{\mathbb{P}(\mathcal{D} \,|\, G)}\right\}$$

This approach proves efficient enough to allow mixing, at the same time being straightforward to implement.

1.2.4 Distance-based methods

Distance-based methods (including Neigbour-Joining and Median-Joining methods) construct a branch-weighted haplotype network based on pairwise phenetic distances of sequences which are computed a priori. If the sequence distances are sufficiently close to the number of evolutionary events between them, these methods can provide sufficiently accurate results (see Desper and Gascuel, 2002; Rzhetsky and Nei, 1993; Atteson, 1997, 1999) based on the phenetic distance matrix. However, as mentioned before (see Subsection 1.1.2), this is frequently not the case, and also cannot reliably be verified. The main advantage of distance-based methods is their small time complexity that makes them applicable to the analysis of large datasets. When homoplasy or deep divergence have occurred, they have little chance of success at inferring the true evolutionary history of the individuals.

Median-joining networks and Reduced Median networks (see Bandelt et al., 1999, 1995) are implemented in the software Network found in `http://www.fluxus-engineering.com/sharenet.htm`.

1.2.5 Maximum-Likelihood methods

The maximum likelihood approach for inferring phylogenies from sequence data was introduced by Felsenstein (1983). Felsenstein's method does not assume a constant evolutionary rate, and it compares possible histories by assigning probabilities to them based on an evolutionary model.

Maximum likelihood methods are powerful and flexible tools in model-based inference and can give statistically reliable conclusions. Likelihood functions are known to be a consistent and powerful basis for statistical inference. Their main drawback is the prohibitive computational complexity, as well as the usual problem of assessing model-fitness. Currently they are implemented through software packages such as PHYLIP (see `http://evolution.genetics.washington.edu/phylip.html`).

1.2.6 Parsimonious methods

Recall that the **parsimony** assumption implies that if two sequences are one DNA change apart, they are assumed to be one mutation apart. Even though parsimonious methods are similar to distance-based methods, they are different in that parsimony infers the trees by evaluating the possible mutations between the sequences. The overall objective of maximum parsimony is to infer the tree with minimum total length, i.e., with the smallest total number of evolutionary changes which explain the observed data. For example, see Figure 1.12 below.

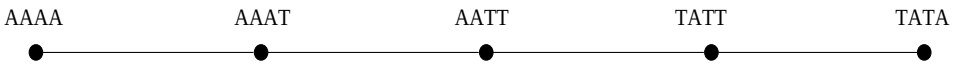


Figure 1.12: *Given a dataset of sequences AAAA, AAAT, AATT, TATT, TATA, the unique minimum possible tree is shown above.*

In contrast to the parsimonious minimum-tree inferred in the figure, distance based methods would not be able to distinguish between pairs of sequences TATA-AATT, which are both two mutations and two SNPs apart, and TATA-AAAA, which are four mutations apart but still 2 SNPs apart.

Parsimonious methods can, under certain conditions, provide estimates of the true tree which are as accurate as Maximum-Likelihood estimates; see Tuffley and Steel (1997). As with distance methods, parsimonious inferences may lead to false conclusions if extensive homoplasy is present, or deep divergence is observed, resulting in long unbranched lineages.

Nested Clade Analysis: forming the haplotype tree. Here we describe the parsimonious method used by Templeton et al. (1992) in order to reconstruct the haplotype tree. Their method is implemented in the software TCS (see Clement et al., 2000, 2002), available at `http://darwin.uvigo.es/software/tcs.html`.

Templeton et al. (1992) define the parsimony assumption as the probability that any two haplotypes which differ at j sites are actually j mutations apart, denoted by P_j. The aim is to estimate P_j for all j and investigate the limits of parsimony. In order to test whether the parsimony assumption is valid, the *maximum* parsimony probability of the data has to be estimated. Templeton et al. (1992) suggest setting the acceptance level by convention at 95%. This means that the assumption that the number of mutations leading from one haplotype to another one is no more than the number of observed mutations (i.e., the number of different nucleotides) will be rejected if the probability of the data based on that assumption is less than 5%. If the assumption is rejected, then it is not possible to obtain accurate results.

An estimator for evaluating the limits of parsimony (meaning the smallest probability of maximum parsimony being true) is constructed based on a simplified evolutionary model of a fixed probability of mutations. Ideally, all sites will be parsimonious, although this is rarely true in reality. In order to estimate the probability that the maximum parsimony assumption is not true, consider the oldest polymorphic site, the *index site* (without specifying it). The total probability that two haplotypes A and B differ at the index site, differ at $j-1$ other polymorphic sites, and share in common the presence of m cut sites (meaning that they have

m letters in common in the DNA sequence under consideration) is approximated by:

$$\begin{aligned} L(j,m,q_1) &= (1-q_1)\left[1-q_1/b\right](1-q_1)^{2m} \times \left\{2q_1\left[2-q_1(b+1)/b\right]\right\}^{j-1} \\ &\quad \times \left\{1-2q_1\left[1-q_1/b\right]\right\} = \\ &= (2q_1)^{j-1}(1-q_1)^{2m+1}\left[1-q_1/b\right] \times \left[2-q_1(b+1)/b\right]^{j-1} \\ &\quad \times \left\{1-2q_1\left[1-q_1/b\right]\right\}. \end{aligned} \tag{1.3}$$

Here q_1 is the probability of a nucleotide change in a single site of the two haplotypes A, B since their respective lineages diverged at the index site, m is a constant based on the similarity between the two haplotypes and $b \in [1,3]$ represents the transition bias (compared to transversion), so that $b = 3$ if there is no bias and $b = 1$ if there is an extreme bias. The value of b is taken to be 3 unless there is evidence to suggest otherwise from previous experiments. A detailed explanation for the derivation of the above expression may be found in Templeton et al. (1992).

Combining (1.3) with a uniform prior on q_1, a standard Bayesian estimator of q_1 is thus

$$\hat{q}_1 = \frac{\int_0^1 q_1 L(j,m,q_1)dq_1}{\int_0^1 L(j,m,q_1)dq_1} \tag{1.4}$$

Now consider mutations that arose after the second oldest mutation associated with a different site. The probability of these mutations in a block of r nucleotides is designated by q_2. Similarly q_i represents the probability of mutations which arose after the jth oldest mutation associated with a different site. An estimator for P_j, the probability that two haplotypes differing at j sites but sharing m have a parsimonious relationship, is:

$$\hat{P}_j = \prod_{i=1}^{j}(1-\hat{q}_i). \tag{1.5}$$

Having established a formula for estimating the parsimony limits between haplotypes, Templeton et al. (1992) iteratively calculate it for pairs of haplotypes starting from 1-step parsimony and moving on to 2-step and so on, until a complete haplotype tree is obtained. The steps followed are described below.

Algorithm 1.2.3.

Step 1: Take $j = 1$ and thus estimate P_1 using (1.4), (1.5), i.e., the probability of parsimony of haplotype pairs that only differ in one site. If any of them is less than 95%, terminate the algorithm. If not, link up all haplotypes that differ by one site. In

addition, it is often the case that other mutational changes are obvious, and so they can be integrated into our 1-step network (see Lloyd and Calder, 1991).

In this step, homoplasies may be observed. However, even at homoplasy events, no loops should be formed.

Step 2: Increase j by 1, and calculate P_j for all possible pairs. If parsimony is accepted, unite the two $(j-1)$-step haplotype networks through the two haplotypes that differ by j steps to form a j-step network.

Repeat step 2 until all haplotypes are in a single connected graph, or in connected subgraphs which between them do not necessarily have a parsimonious relationship. In the case of a high probability of parsimony, a spanning tree is obtained which includes all the observed haplotypes as nodes, and the process termninates. So far no loops should be present, since only sites which are parsimony informative are considered. The graph is not connected, move to step 4.

Step 3: Unite the separate networks identified in the previous step into a single haplotype tree, considering both parsimonious and non-parsimonious linkages. Let x be the number of mutational steps involving sites that connect two networks under maximum parsimony. Then, the probability that y or fewer of the x polymorphic site mutations are non-parsimonious is:

$$\sum_{i=0}^{y} \sum_{I} q_{j(k)} \prod_{k=1}^{x} \left(1 - q_{j(k)}\right) \tag{1.6}$$

where I refers to the set of all permutations of the x age ranks of the mutations. Here only the total number of mutations that occurred beyond those required by parsimony is of interest. As a result, consider all permutations of the age ranks (with which these additional mutations are associated) that yield the same number of total additional mutations. This is achieved by placing age ranks into two classes of size i and $x-i$, and then summing over all permutations of the age ranks that result in these class sizes. These alternative permutations are indicated by $j(k)$, which refers to the kth permutation in the set I. The first product in (1.6) is defined to be 1 when $i = 0$.

Find the minimum value of y such that (1.5) is greater than or equal to 0.95. The set of plausible haplotype trees contains all connections between disjoint networks that include the maximum parsimony solutions as well as any connections involving up to y additional mutational steps.

This results in a set of both parsimonious and non-parsimonious networks. In this step,

some loops may appear, implying that the tree is not unique. Although homoplasy is frequently present, it is highly unlikely that evolution actually formed a full loop (Templeton et al., 1992). In the case of ambiguity of the haplotype network, when there is no unique most likely tree, various criteria are used Crandall and Templeton (1993); Templeton et al. (1992). For example, within a haplotype tree, rare haplotypes are more likely to be leaf haplotypes, and common ones more likely to be interior. Also, in the case of phylogeographic data, singleton haplotypes are more likely to be connected to haplotypes from the same population as opposed to haplotypes from different populations (see Templeton, 1998).

Approximate Bayesian Computation in population genetics

MCMC algorithms rely upon evaluation of the likelihood of the data given the model parameters. We showed how the peeling algorithm can be used to calculate the probability of a coalescent tree based on a simple evolutionary model. However, when more complex models are required, the calculation becomes intractable. To overcome intractable likelihoods, ABC methods have been developed (see Section 1.5). Beaumont et al. (2002) describe how ABC can be employed in population genetics using summary statistics based on the number of segregating sites. An appropriate metric ρ is proposed, as well as criteria for choosing the tolerance level ϵ. Beaumont et al. (2002) describe how a series of simulated statistics S' from different values of a model parameter ϕ can be used within a linear regression in order to adjust the weighting of each ϕ based on the deviation of the simulated statistics S' from the true S, and to weaken the effect of the discrepancy between S and S'.

More recently, Beaumont (2003) developed an improved ABC algorithm where the simulated statistics S' are treated as an auxiliary parameter in the model and are updated within MCMC in the usual way, preserving time-reversibility of the sampler.

1.3 Phenotypic clustering analysis

The objective of phenotypic clustering analysis is to identify nucleotide mutations which are associated with a significant change in the phenotypic effect. We describe two methods of phenotypic clustering analysis (see Subsections 1.3.1 and 1.3.2), both developed by Templeton et al. based on a haplotype tree (inferred through the method described in the previous Subsection 1.2.6).

1.3.1 Nested clade analysis for phenotypic data

The first step of NCA is defining a nesting on the haplotype tree. Once the nested tree is obtained, nested levels are tested for significant associations with phenotypic effects. The main assumption in NCA is that if an undetected mutation causing a phenotypic effect occurred at some point in the evolutionary history of the population, it would be embedded in the same historical structure represented by the haplotype tree. In other words, even if some mutation is not detected, the shape of the predicted haplotype tree would still be correct, and hence the hidden mutation would only be present in the correct branch.

The nesting algorithm is as follows (see Templeton et al., 1987). The 0-step clades are just the haplotypes represented as leaves in our haplotype tree. Given the n-step clades, the $(n+1)$-step clades are formed by taking the union of all n-step clades which can be joined up by moving one mutational step back from the terminal node of each n-step clade. Any internal n-step clades which are not included in one of the nested clades are nested by considering nodes which are adjacent to a nested n-step clade as terminal. The process continues recursively until all the nodes in the haplotype tree have been nested. The nesting process is easier understood through an example; see Figure 1.13.

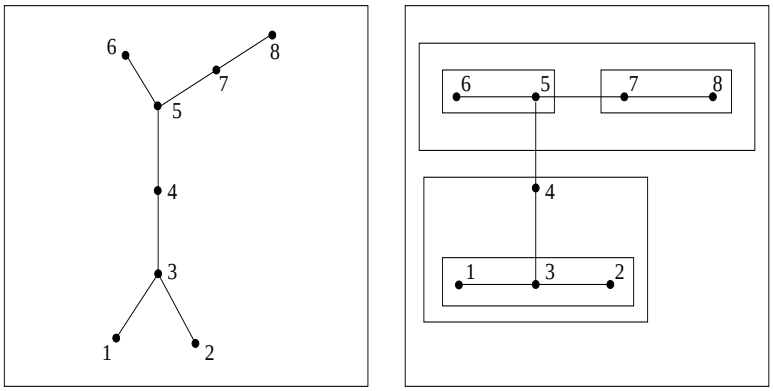


Figure 1.13: *An example of a haplotype tree, unnested (left) and nested (right). The procedure of nesting the tree on the left follows a number of steps. First we joined up leaves to their neighbours. In the case where two leaves are joined to the same neighbour, the two leaves are nested together (i.e., clade 1-3-2). So we obtain 5-6, 7-8 and 1-3-2 as our 1-step clades. In effect, these 3 groups are collapsed onto only 1 node each, behaving like a leaf: we now have 5, 7 and 3 representing the 3 groups (7, 8, 1, 2 were 'chopped'). In the next step, we join up 7 (and the nodes associated with it from the previous step) and 5 (likewise), and we also join up 3 and 4 (as shown in diagram). Finally, our last step only involves joining the whole thing together.*

The nesting algorithm is not well defined; for example, see Figure 1.14. In these cases Templeton and Sing (1993) provide extra criteria on how to proceed. If an unnested node

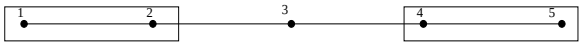


Figure 1.14: *An example of incomplete nesting. The procedure described above is not always well-defined, as there are special cases which are ambiguous, or where nesting is not complete. For example, in this figure, on the first step we nest 1-2 and 4-5, leaving 3 stranded (as the next nesting step is just nesting the whole thing which has no practical use in terms of the analysis*

(or clade) represents an unobserved haplotype, then it is simply left ungrouped with no complications. However, if it belongs to the sample, it needs to be grouped with another clade. The following guidelines are suggested: first, the stranded clade should be grouped with the nesting category that has the smallest sample size because such a grouping tends to maximise statistical power. Secondly, if the smallest sample size is observed in more than one alternative, then the stranded clade should be nested with the alternative to which it is connected through a non-polymorphic site mutation.

Once the nested haplotype tree is formed, it is used to associate it with significant (or insignificant) geographical or phenotypic data through the following algorithm. Starting from level 1, at each level, an ANOVA is performed, and Residual Sum of Squares (RSS) contributions are examined to identify clades which contribute most. To avoid the possibility of "overspill", where the effect of significant mutations carries through to different clades and masks the true effect, significant clades are examined and compared using Bonferroni comparisons.

1.3.2 Tree scanning

An improvement to Nested Clade Analysis is achieved with the tree-scanning method described by Templeton et al. (2005), available at `http://darwin.uvigo.es`. In this approach, all possible individual mutations are tested for significance, by separating the haplotype tree into two parts which are treated as ANOVA groups. The mutations which exhibit the most significant effect are then assumed to be associated with a change in the phenotype.

1.4 Phylogeographic analysis

There have been several studies on different aspects of phylogeographic analysis. Avise (2000) present a detailed introduction to phylogeographic inference. A review of a few existing methods is given by Pearse and Crandall (2004), Crandall and Templeton (1993), Knowles and Maddison (2002) and Emerson et al. (2001), where various approaches to identification of population structure, quantification of gene flow and inference of demographic history are

described.

We present four main approaches. Firstly, methods which are not based on phylogenetic inference are described in Subsection 1.4.1. We then describe how the change of various genetic characteristics may be used in order to draw phylogeographic conclusions in Subsection 1.4.2. In Subsection 1.4.3 we describe how population subdivision can be investigated using migration models. Finally, we describe the clustering approach of NCPA in Subsection 1.4.4.

1.4.1 Non phylogeny-based approaches

One of the main population clustering algorithms based on genotype data is presented by Falush et al. (2003) and implemented in the software STRUCTURE available at `http://pritch.bsd.uchicago.edu/structure.html`. The method is based on treating the the genotypes as categorical data, and inferring population clusters which show a better fit with the data based on a simple model of population structure.

In their paper, Falush et al. (2003) define two possible population models, with and without admixing respectively. Suppose genotypes of N diploid individuals for a total of L loci are given. In the first model of no admixture, each individual is assumed to originate from one of the K populations. Here X denotes the observed genes, Z the populations of origin of individuals (which will be inferred) and P the unknown allele frequencies in the populations. Specifically:

$$\begin{aligned}
(x_l^{(i,1)}, x_l^{(i,2)}) &= \text{genotype of the } i\text{th individual at the } l\text{th locus,} \\
&\quad \text{where } i = 1, 2, \ldots, N \text{ and } l = 1, 2, \ldots, L; \\
z^{(i)} &= \text{population from which individual } i \text{ originated;} \\
p_{klj} &= \text{frequency of allele } j \text{ at locus } l \text{ in population } k, \\
&\quad \text{where } k = 1, 2, \ldots, K \text{ and } j = 1, 2, \ldots, J_l.
\end{aligned}$$

For a model which allows admixing, Falush et al. (2003) introduce a parameter Q which represents the admixture proportions for each individual, so that

$$q_k^{(i)} = \text{proportion of the genome of individuals } i \text{ which originated from population } k,$$

and the vector Z becomes

$$z_l^{(i,a)} = \text{ population of origin of allele copy } x_l^{(i,a)}.$$

These define a multinomial-type likelihood which can be used to update the parameters of

interest using an MCMC algorithm. The number of clusters K is allowed to vary, and an approximation is proposed to overcome computational complexity caused by the K updates.

1.4.2 Using the change in characteristics along clines

In contrast to clustering approaches, there has been substantial evidence that global genetic variation in humans is mainly *clinal* (see Handley et al., 2007). Clines represent the linear change along a character with increasing geographic distance. This implies that genetic characteristics vary in a continuous way, rather than being governed by the extistence of break-points in the geography or throughout history. As a result, clustering approaches are not always appropriate, rather models which investigate the change in genetic characteristic quantities along a cline can yield significant conclusions. For example, measuring the change in genetic differentiation through F_{ST}, which represents the correlation between two randomly selected genes from the population (see Wright, 1951) shows an approximately linear relationship with respect to pairwise geographical distance between populations, as shown in Linz et al. (2007).

Liu et al. (2006) use a stepping-stone model to analyze the worldwide demography of human populations and investigate past colonization events. Their approach is based upon the estimated coalescence times and their variation across the globe. Five parameters are considered in the study: the time since the spread of modern humans, the growth rate in a new (i.e., colonized) population, the migration rate, the maximum capacity of the initial and all subsequent populations. The model provides an excellent fit to the data.

Handley et al. (2007) discuss the two different representations of human demography and genetic variation, namely clusters and clines. However, although the genetic variation is mostly clinal (with clines explaining more than 75% of the total F_{ST}), there is evidence that breakpoints exist, and a synthetic model is probably most appropriate. Introducing clustering information to the model adds an extra 2% to the amount of genetic variance explained by the model.

1.4.3 Migration models

A number of approaches have been developed to investigate migration patterns in subdivided populations. They are mostly based on introducing a migration probability in the coalescent process, so that the possible events are coalescence, mutation and migration. If an individual migrates, a new tree is built afresh in the new location. This may be viewed as a single tree, where each lineage has a specific colour representing the population it belongs to, with coalescence events only allowed between individuals which belong to the same population

(i.e., bear the same colour). Several migration models have been developed. Two of the main ones are island models which assume equal rates of migration between any two subpopulations (see Latter, 1973), and stepping-stone models where population structure is represented as a set of subpopulations which can send and receive migrants to their left and right-hand neighbours only (see Kimura and Weiss, 1964). Some more sophisticated models asusme that migrations occur to the four nearest neighbouring populations as points on a lattice (lattice migration models, see Matsuda et al., 1992), or take into account environmental factors which encourage or discourage migration (see Ray et al., 2005).

Bahlo and Griffiths (2000) proposed an algorithm which De Iorio and Griffiths (2004b) subsequently improved by taking an approach which relies on Importance Sampling (IS) proposal distributions based on the diffusion-generator approximation of gene frequencies for a single population, as presented by De Iorio and Griffiths (2004a).

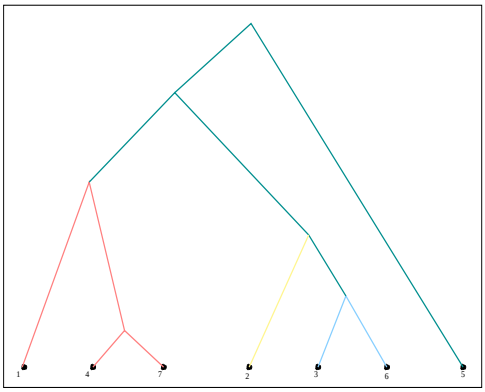


Figure 1.15: *A coalescent tree with population subdivision. Here, three migration events can be identified, all originating from the green population to found the pink, yellow and light blue populations.*

A similar approach is used by Nielsen and Wakeley (2001) in order to construct a MCMC algorithm for assessing whether an isolation or migration model shows a better fit with the data.

1.4.4 Nested Clade Phylogeographic Analysis

Following the haplotype tree inference described in Subsection 1.2.6, NCPA was developed by Templeton (1998) in order to draw phylogeographic conclusions. Three major geographical dispersal patterns that can cause a significant spatial/temporal association of haplotype variation are considered (see Templeton, 1998; Avise et al., 1987; Ibrahim et al., 1996):

- **Restricted gene flow** occurs when haplotypes take time to spread geographically. This implies that the spread of clades increases with time, and that descendant haplotypes

have a geographical range smaller than their ancestors and remain within the area for a number of generations. As a result, the geographical centres of all the clades nested together are close under restricted gene flow, and nested clade distances exhibit similar patterns.

- **Past fragmentation** occurs when environmental changes force a population to be fragmented, i.e., split into subpopulations. It restricts the geographical distribution, so that a clade distance cannot increase beyond the geographical range of the fragmented subpopulations. In the case of old fragmentation events, clade distances which remain constant will on average be longer than the average length of the branch in the tree.

- Finally, **range expansion** results in haplotypes with a geographically broad range, and descendant haplotypes which arose post-expansion become increasingly distant from the haplotypes in the original area.

NCPA follows similar steps to NCA, adapting the ANOVA with appropriate testing of geographical clustering patterns. Given the haplotype tree, the nested clades are formed as in Subsection 1.3.1. The geographical data are quantified in two ways: the clade distance D_c, which represents the geographical range of a clade, and D_n, which measures how that particular clade is geographically distributed relative to its closest evolutionary clades (i.e., clades in the same higher-level category). Specifically, D_c is the average distance of haplotypes from that clade from the geographical centre of the clade. D_n is the average distance of a haplotype form that clade to the geographical centre of all higher-level clades which contain the clade in question. Both these distances are a measure of the spatial spread of a clade, where only physically feasible paths are taken into account (i.e., routes which are possible for the species to have taken).

Templeton (1998) suggest that associations between the clade distance and the nested clade distance should be tested. The hypothesis of no association is tested by permutation tests. The precise calculations of the algorithm (implemented in the software Geodis available at `http://darwin.uvigo.es/software/geodis.html`, see Posada et al. 2000) were recently published by Posada et al. (2006) and assessed in Templeton (2004). If the null hypothesis of no association is rejected, precise phylogeographic events of restricted gene flow, fragmentation and range expansion are predicted following a descriptive inference key (see Appendix in Templeton, 1998).

Both NCA and NCPA suffer from a number of drawbacks, and have been frequently criticized; see Petit (2008), Petit and Grivet (2002), Knowles (2004), Panchal (2007), Panchal and Beaumont (2007). They rely upon choosing a unique tree at the first stage of the analysis, and do not allow for the uncertainty of the tree to be taken into account at the subsequent

steps. In addition, NCA involves ANOVA and as a result suffers from problems associated with multiple testing and poor resolution in terms of identifying the level of clade at which significant mutations occur (see Brooks et al., 2007). In many cases the criteria provided can be interpreted in several ways, hence the results can be subjective; see Panchal (2007), Panchal and Beaumont (2007). The phylogeographic hypotheses are based upon a descriptive inference key, which does not explicitly take into account a rigorous probability model for patterns of population dispersal. Finally, the permutation testing of NCPA has been shown to lead to false conclusions based on simulations (see Petit, 2008).

1.5 Overview of Markov chain Monte Carlo

In this section we present a brief overview of MCMC, describing various MCMC samplers. We introduce some convergence diagnostics which are used to assess whether the MCMC samplers have reached equilibrium. We then discuss the limitations caused by intractable likelihoods and how they can be addressed. Finally, we briefly present some alternatives to MCMC.

MCMC methods are some of the main methods implemented in Bayesian inference in order to estimate the posterior distribution of a set of parameters given the data. The key idea behind MCMC is to simulate a sequence of datapoints by constructing a Markov chain whose stationary distribution is the posterior distribution of our parameters given our data; see Geyer (1991), Geyer (1992), Gilks et al. (1995), Brooks (1998), Green (2000), Robert and Casella (2004), Green et al. (2003). It is based upon the ergodic theorem, which states that if $(X_n)_{n\geq 0}$ is an **irreducible** Markov chain on state space I with initial distribution λ and transition matrix P, then for any bounded function $f : I \to \mathbb{R}$ we have

$$\mathbb{P}\left(\frac{1}{n}\sum_{k=0}^{n-1} f(X_k) \to \overline{f}\ as\ n \to \infty\right) = 1,$$

where

$$\overline{f} = \sum_{i\in I} \pi_i f_i,$$

and $\boldsymbol{\pi}$ is the stationary distribution of the chain. Specifically, if the function f is the identity function, so that $f(X_k) = X_k$, then

$$\frac{1}{n}\sum_{k=0}^{n} X_k \to \mathbb{E}_\pi\{X\}$$

almost surely; see Norris (1997). If, in addition, the Markov chain is **aperiodic**, then the

chain converges to equilibrium almost surely. This means that

$$\mathbb{P}\left(X_n = j\right) \to \pi_j \text{ as } n \to \infty \ \ \forall\, j.$$

Finally, if $(X_n)_{n\geq 0}$ satisfies the detailed-balance equations

$$\pi_i p_{ij} = \pi_j p_{ji} \ \ \forall i, j,$$

the chain is also **time-reversible**. This means that the chain $(Y_n)_{0\leq n\leq N} = (X_{N-n})_{0\leq n\leq N}$ for some N is also a Markov chain, with transition matrix P and stationary distribution π. In other words, the chain is the same whether we view it backwards or forwards in time.

Markov chain Monte Carlo is very useful in situations where the posterior distribution is not exact or is intractable, and it allows inference about the joint posterior distribution of a large number of parameters.

One of the main algorithms for constructing a chain with equilibrium distribution equal to the posterior distribution of the parameters is the **Metropolis-Hastings** (MH) algorithm. Suppose we wish to draw inferences about a parameter θ given data $\mathcal{D}$. Then we define a Markov chain which moves from $\theta^{(t)}$ to $\theta^{(t+1)}$ according to the following transition kernel.

We propose a θ' using a proposal distribution $q(\theta^{(t)} \to \theta')$. The proposed state is accepted (implying that $\theta^{(t+1)} = \theta'$) with probability $\alpha = \min(1, A)$ where

$$A = \frac{q(\theta' \to \theta^{(t)})}{q(\theta^{(t)} \to \theta')} \frac{\pi(\theta' \,|\, \mathcal{D})}{\pi(\theta^{(t)} \,|\, \mathcal{D})}$$

If the proposed value is rejected, then we set $\theta^{(t+1)} = \theta^{(t)}$. The distribution q is virtually arbitary, provided it ensures irreducibility and aperiodicity of the chain, but it affects the speed at which the chain reaches equilibrium. It is easy to check that the MH chain satisfies the detailed-balance equations and thus is time-reversible.

Using Bayes' theorem,

$$\pi(\theta \,|\, \mathcal{D}) = \frac{f(\mathcal{D} \,|\, \theta) p(\theta)}{\mathbb{P}(\mathcal{D})},$$

where $p(\theta)$ is the prior distribution for θ, and A becomes

$$A = \frac{q(\theta' \to \theta^{(t)})}{q(\theta^{(t)} \to \theta')} \frac{f(\mathcal{D} \,|\, \theta')}{f(\mathcal{D} \,|\, \theta^{(t)})} \frac{p(\theta')}{p(\theta^{(t)})}$$

There are a few special cases of the MH algorithm. For example, if $q(\theta \to \theta')$ is chosen to be precisely the posterior distribution $q(\theta \to \theta') = \pi(\theta'|\mathcal{D})$, then the acceptance probability becomes 1 (this is called a **Gibbs' sampler**). Also, if the transition kernel $q(\theta \to \theta')$ is

symmetric, i.e., $q(\theta \to \theta') = q(\theta' \to \theta)$, then the acceptance probability becomes

$$A = \frac{f(\mathcal{D} \mid \theta')}{f(\mathcal{D} \mid \theta^{(t)})} \frac{p(\theta')}{p(\theta^{(t)})},$$

called a **random walk Metropolis** algorithm.

It is sometimes the case that the dimension of the parameter θ is not known. This implies that, from iteration to iteration, the size of the parameter space Θ changes. In those cases, **Reversible Jump** (RJ) MCMC is implemented; see Green (1995). Suppose we propose to move from a point $\theta^{(t)} \in \Theta^{(t)}$ to a point $\theta' \in \Theta'$. Then let g be the dimension-matching map $g(\theta^{(t)}) = \theta'$. The acceptance probability becomes $\alpha = \min(1, A)$ where

$$A = \frac{q(\theta' \to \theta^{(t)})}{q(\theta^{(t)} \to \theta')} \frac{\pi(\theta' \mid \mathcal{D})}{\pi(\theta^{(t)} \mid \mathcal{D})} \left| \frac{\partial g(\theta^{(t)})}{\partial \theta^{(t)}} \right|,$$

where the matrix $J_{ij} = \frac{\partial g(\theta)_i}{\partial \theta_j}$ is the Jacobian. RJMCMC preserves reversibility of the chain, and ensures that the stationary distribution is indeed the posterior distribution of the parameters.

An MCMC sampler can never simulate an infinite number of observations. However, in order to obtain an accurate estimate of the posterior distribution of the parameter θ, it is essential to investigate whether the Markov chain has reached equilibrium. Although it is impossible to find a test which can provide a definitive result about whether the chain has converged to equilibrium, a number of diagnostics have been developed which enable us to assess the convergence (see Brooks and Roberts, 1998; Cowles and Carlin, 1996). In simple cases, observation of the trace plots of the chain for each parameter may suggest convergence, if the samples seem uncorrelated and move "adequately" around the parameter space.

One of the methods recommended by Brooks and Roberts (1998) was developed by Gelman and Rubin (1992) and later extended by Brooks and Gelman (1998), and is implemented within the coda package in R. It is based on running several independent chains with starting points which are overdispersed in terms of the posterior distribution of each parameter. It provides estimates of how much the convergence can potentially be improved by running the chain for longer, called **Potential Scale Reduction Factors** (PSRF). When the PSRF is close to one, the chain is assumed to have converged. Brooks and Gelman (1998) extended this method to take into account more information regarding PSRFs, suggesting that a graphical approach is more informative. The plots show whether the PSRF has really converged, or whether it is still fluctuating.

In practice, and especially in biological problems, it is frequently the case that the likelihood function $f(\mathcal{D} \mid \theta)$ is intractable or computationally expensive. Recently, **Approximate**

Bayesian Computation (ABC) methods have been developed (see Marjoram et al., 2003), which may be used within MCMC. The main idea is that instead of calculating $f(\mathcal{D} \mid \theta)$, we simulate $\mathcal{D}' \mid \theta$, so that the acceptance probability becomes

$$A = \frac{q(\theta' \to \theta^{(t)})}{q(\theta^{(t)} \to \theta')} \frac{p(\theta')}{p(\theta^{(t)})} \mathbb{1}_{\{\mathcal{D}=\mathcal{D}'\}},$$

and it is easy to check that this chain also has stationary distribution equal to the posterior distribution of the parameters. When a sufficient statistic S is available, the MH ratio can be reduced to

$$A = \frac{q(\theta' \to \theta^{(t)})}{q(\theta^{(t)} \to \theta')} \frac{p(\theta')}{p(\theta^{(t)})} \mathbb{1}_{\{\mathcal{S}=\mathcal{S}'\}}.$$

Notably, the indicator function $\mathbb{1}$ can only be used for discrete data. For continuous data, it can be substituted by $\mathbb{1}_{\{\rho(\mathcal{D},\mathcal{D}')<\epsilon\}}$, where ρ is a distance metric and ϵ is a small value (which is not necessarily fixed). The disadvantage of such methods is that they may naturally lead to very high rejection rates. Moreover, for continuous data, the use of the distance metric ρ and a threshold ϵ introduces a bias to the acceptance probability and hence to the estimates of the analysis. A number of variants of ABC have been suggested to improve the efficiency and accuracy of the algorithm in population genetics (see Beaumont et al., 2002; Beaumont, 2003; Beerli and Felsenstein, 2001; O'Neill et al., 2000).

An alternative to MCMC are **Sequential Monte Carlo** samplers (Doucet et al., 2006), which are based on the idea of **Importance Sampling** (IS) (see Ripley, 1987). IS is used when sampling from the posterior distribution $\pi(\theta \mid \mathcal{D})$ is not possible, but we can draw n values of θ from a distribution $q(\theta^{(t)})$ which is called the importance distribution. We then calculate

$$\hat{\theta} = \sum_{t=1}^{n} w^{(t)} \theta^{(t)},$$

where $w^{(t)} = \frac{\pi(\theta^{(t)} \mid \mathcal{D})}{q(\theta^{(t)})}$ are the importance weights. It is easy to check that θ' is an unbiased estimate of $\theta \mid \mathcal{D}$. The challenge here is to select an importance distribution which minimizes the variance of the estimator, for example by using a distribution which is "similar" to the target posterior distribution. SMC methods are based on a sequential implementation of the IS algorithm to achieve efficient and accurate estimators.

Chapter 2

A Bayesian approach to nesting

We now describe a Markov chain Monte Carlo method of analysing sequence data in which both steps of NCA (inferring the haplotype tree and identifying significant clusters) are carried out simultaneously. Our holistic approach ensures that the uncertainty of the tree is propagated through to the phenotypic/phylogeographic analysis. Furthermore, it allows the joint posterior distribution of the parameters to be quantitatively assessed, and offers itself to modifications and additions to the mutation and clustering models assumed.

In this Chapter we consider the simple case where the exact history of the region of the DNA sequences under study is known, and it is assumed that no homoplasy is involved. This new method is an alternative to the last step of NCA, namely the one where nested clades are inferred and tested for significance.

In Section 2.1 we develop an algorithm for analysing one-dimensional phenotypic data, aiming to identify SNPs which are correlated with a significant change in the phenotypic effect. To this end, we fix the number of significant mutations, and propose an appropriate model for the data so as to apply a clustering algorithm. We describe a MCMC sampler in order to draw inferences about the distribution parameters based on the clustering model and the known haplotype tree. This method is extended in Section 2.2 to phenotypic data of two or more dimensions, where a Reversible-Jump MCMC algorithm is constructed that identifies the specific dimensions (representing phenotypic characteristics) which show the highest significant change with mutations. The phenotypic clustering model is modified in Section 2.3 to fit phylogeographic data, so that geographical clusters can be identified, and we propose an adaptive MCMC technique to ensure convergence of the chains. In Section 2.4 we generalise our results for any number of clusters, by proposing an adaptive Reversible-Jump MCMC algorithm which draws inferences about the clustering of the data, allowing the number of clusters to vary.

2.1 Phenotypic clustering for one-dimensional traits

We are given a set of DNA sequences along with a measurement of some characteristic trait (i.e., phenotype) for each individual. It is assumed that the precise mutational history of the sampled sequences is known and that the sequences can be collapsed onto haplotypes. This implies that each haplotype may be observed more than once, with a different phenotypic measurement each time. The objective of the analysis is to identify whether one (or more) of the mutations in the sample of DNA sequences is associated with a significant change in the phenotypic trait. This is equilavent to a simple clustering problem, where clusterings are constrained so that they are consistent with the known haplotype tree. Since we have assumed that the exact mutational history is known, this reduces to identifying which mutations from a fixed set define the best partition of the phenotypic data.

If a mutation is associated with a change in the characteristic trait, then any individuals carrying haplotypes with that mutation will have significantly different phenotype values to individuals that do not. Accordingly, K significant mutations will split the data into $K + 1$ significantly different *hard clusters defined on haplotypes*. A coloured haplotype tree, where each node is a haplotype and each edge a mutation, is well suited to illustrate a clustering structure within this setting. The colour of each node denotes the phenotypic cluster of each haplotype, and the size of each node shows the number of times the respective haplotype was observed in the sample. For two significant mutations the corresponding haplotype tree may look like Figure 2.1 below.

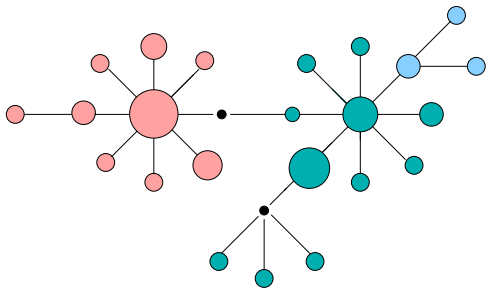


Figure 2.1: *Example of a haplotype tree as defined in the text under a clustering structure of two significant mutations. The black dots represent unsampled (but known) haplotypes. Here the pink star-like haplotype is the most common one as indicated by the size of the circle. Three population clusters are identified: pink, green and light blue. There are two mutations causing a significant effect in the phenotype: the one between the pink and green nodes, and the one between green and light blue nodes.*

Provided the true haplotype tree is known, identifying significant mutations is equivalent to partitioning the data into clusters which are defined by K edges on the tree. Since sequences are collapsed onto haplotypes, we note that observations from the same haplotype are always

clustered together.

Formally, we have a sample of size N of some phenotypic effect, which corresponds to a sample of $N_h \leq N$ haplotypes. We wish to infer mutations that are associated with a significant phenotypic change; in this section we assume that there exist precisely K of them, where K is known. These mutations are represented by a set of K edges on the haplotype tree, so that the resulting $K+1$ clusters are significantly different in terms of the distribution of the characteristic trait.

We introduce the following notation. We are given phenotypic data $\mathcal{Y} = \{Y_{ij}\}$ totalling N datapoints, so that Y_{ij} represents the jth datapoint of haplotype i. We denote the set of distinct significant edges of the known haplotype tree by $\boldsymbol{e}$, representing the mutations associated with a significant change in the phenotype. Here $\overline{y}_k$ denotes the sample mean of cluster k and n_k the sample size of cluster k, so that $\sum n_k = N$, the total sample size. Finally, we define an allocation variable $\boldsymbol{c}$, so that the jth datapoint of haplotype i belongs to cluster c_{ij}. Here c_{ij} for each i is the same for all j, since all observations from the same haplotype are forced to belong to the same cluster. This implies that $n_k = \sum_{j,l} \mathbb{1}_{\{c_{jl}=k\}}$.

Throughout this thesis we assume that our data are normally distributed because all datasets analyzed here are approximately normally distributed. In principle the method can be applied to any distribution by virtue of our computational, sample-based approach. In this section we consider the case where the phenotypic effect is one-dimensional and normally distributed, implying that the distribution-defining parameters of the $K+1$ clusters are simply $\mu_1,\ \mu_2, \ldots, \mu_{K+1}$ and $\sigma_1^2,\ \sigma_2^2 \ldots, \sigma_{K+1}^2$ The data are transformed so that the sample mean is 0 and sample variance is 1, without loss of information.

Specifically, we assume the following distributions (here k denotes the cluster):

$$\begin{aligned} \sigma_k^{-2} &\sim \mathcal{G}(a,b) \\ \mu_k &\sim \mathcal{N}(0,\sigma_\mu^2) \\ e_k &\sim \mathcal{U}\{1,\ldots,N_h-1\} \text{ without replacement} \\ Y_{ij} \,|\, \boldsymbol{e},\boldsymbol{\mu},\boldsymbol{\sigma^2} &\sim \mathcal{N}(\mu_{c_{ij}}, \sigma^2 c_{ij}), \end{aligned} \tag{2.1}$$

where $\mathcal{G}, \mathcal{N}, \mathcal{U}$ denote the Gamma, Normal and Uniform distribution respectively. In the absence of specific prior information, a and b are taken small so that the prior distribution of σ_k^{-2} has a large variance, and σ_μ^2 is set large. The hierarchical structure of the parameters in Model (2.1) is summarized in Figure 2.2.

Note that the conjugate prior for μ_k is actually $\mathcal{N}(0, \alpha\sigma_k^2)$, where α is a constant. This prior implies that a small variance of the phenotype within each cluster indicates a small variance between clusters, which imposes an unrealistic constraint on the prior belief for μ_k

(see Garthwaite and Al-Awadhi, 2001). As a result, we use an independent prior $\mathcal{N}(0, \sigma_\mu^2)$.

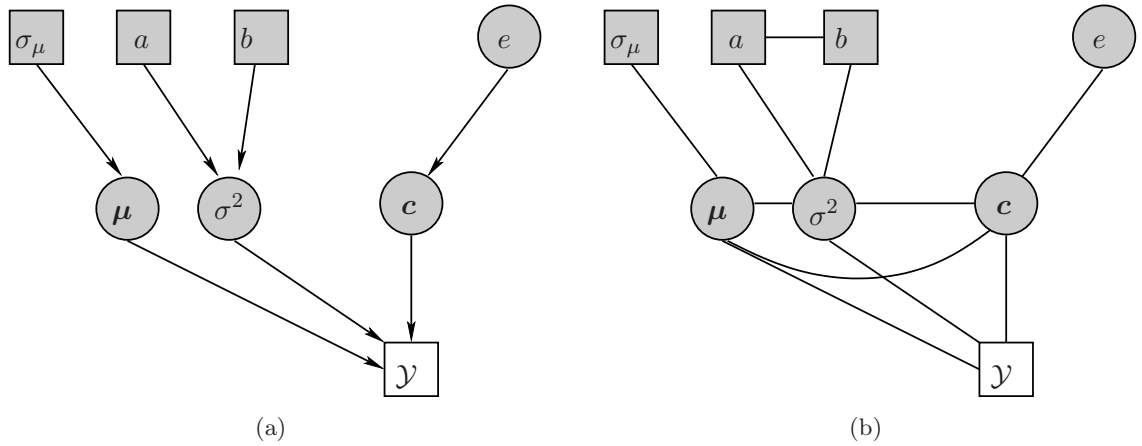


Figure 2.2: *Here we represent the parameters of our model as a Directed Acyclic Graph (DAG) on the left, and a conditional independence graph on the right. A DAG is interpreted as follows (see Gilks et al., 1995): for any node v, conditioning on the value of its parent nodes (i.e., the nodes which have an arrow directed towards v), means that no other nodes would be informative about v except its descendants. We adhere to the convention of representing fixed or observed quantities by squares, and circles for parameters which are estimated. The conditional independence graph is obtained by "moralising" (i.e., connecting) each node's parents. This graph will be augmented in later sections to demonstrate how parameters are added to the hierarchical parameter structure of our algorithms.*

From the given distributions we obtain the following conditionals:

$$\sigma_k^2 \,|\, \mathcal{Y}, \boldsymbol{e} \;\sim\; \mathcal{IG}\left(a + \frac{n_k}{2}, b + \frac{\sum_{j,l} \mathbb{1}_{\{c_{jl}=k\}} \left(y_{jl} - \overline{y}_k\right)^2}{2}\right), \tag{2.2}$$

$$\mu_k \,|\, \mathcal{Y}, \boldsymbol{e}, \tau \;\sim\; \mathcal{N}\left(\frac{\overline{y_k}\,\tau n_k}{\tau n_k + \frac{1}{\sigma_\mu^2}}, \left(\tau n_k + \frac{1}{\sigma_\mu^2}\right)^{-1}\right), \tag{2.3}$$

$$\sigma_k^2 \,|\, \mathcal{Y}, \boldsymbol{e}, \boldsymbol{\mu} \;\sim\; \mathcal{IG}\left(a + \frac{n_k}{2}, \frac{\sum_{j,l} \mathbb{1}_{\{c_{jl}=k\}} \left(y_{jl} - \mu_k\right)^2}{2} + b\right). \tag{2.4}$$

We construct an MCMC sampler which explores the space of possible clusterings, drawing inferences about the means of each cluster as well as the underlying common variance. Notably the cluster means and variances are only relevant to a specific clustering: for two different clusterings, the set of cluster means of one is nonsensical in relation to the other. This implies that the clustering cannot be updated without updating the means and variances simultaneously. We denote by $\mu_i^{(t)}$, $i = 1, \ldots, K+1$, the value of the mean of the ith cluster at iteration t, and similarly for the rest of the parameters.

We want to construct a MCMC sampler with target distribution

$$\pi(\boldsymbol{e}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2 \,|\, \mathcal{Y}) \quad \propto \quad f(\mathcal{Y} \,|\, e, \boldsymbol{\mu}, \boldsymbol{\sigma}^2) p(\boldsymbol{\mu}) p(\boldsymbol{\sigma}^2) p(\boldsymbol{e}). \tag{2.5}$$

Inference about (2.5) allows us to identify branches which are significantly different to each other in terms of the phenotype of interest, yielding the correspondence with NCA suggested by Templeton et al. (1987). Templeton uses nested clades in order to partition the data, and assesses their significance by using analysis of variance (ANOVA). Here we assess the significance of each clustering according to its posterior distribution.

We describe the MCMC algorithm in detail. The chain is initialized by generating a set of significant edges $\boldsymbol{e}^{(0)}$, $K+1$ different means $\mu_k^{(0)}$ and variances $\sigma_k^{2(0)}$, all from the prior distribution. We then iterate through the following steps[1].

A1a: First we update $\boldsymbol{e}$. We randomly pick one of the K edges $e_k^{(t)}$ uniformly from our haplotype tree (not allowing edges which are already in $\boldsymbol{e}^{(t)}$). We form the clusters defined by the new set of edges, and calculate their means $\overline{y}_k$ and their sample sizes n_k.

A1b: We propose $K+1$ variances $\sigma_k^{2'}$ from the conditional distribution $\sigma_k^2 \,|\, \mathcal{Y}, \boldsymbol{e}'$ given in Equation (2.2).

A1c: We propose $K+1$ means μ_k' from $\mu_k \,|\, \mathcal{Y}, \boldsymbol{e}', \sigma_k^{2'}$ given in Equation (2.3).

A1d: Compute

$$\begin{aligned} A_A &= \frac{\pi(\boldsymbol{e}', \boldsymbol{\mu}', \boldsymbol{\sigma}^{2'} \,|\, \mathcal{Y})}{\pi(\boldsymbol{e}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2 \,|\, \mathcal{Y})} \frac{q(\boldsymbol{\mu}' \to \boldsymbol{\mu})}{q(\boldsymbol{\mu} \to \boldsymbol{\mu}')} \frac{q(\boldsymbol{e}' \to \boldsymbol{e})}{q(\boldsymbol{e} \to \boldsymbol{e}')} \frac{q(\boldsymbol{\sigma}^{2'} \to \boldsymbol{\sigma}^2)}{q(\boldsymbol{\sigma}^2 \to \boldsymbol{\sigma}^{2'})} \\ &= \frac{f(\mathcal{Y} \,|\, \boldsymbol{e}', \boldsymbol{\mu}', \boldsymbol{\sigma}^{2'})}{f(\mathcal{Y} \,|\, \boldsymbol{e}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2)} \frac{p(\boldsymbol{\mu}')}{p(\boldsymbol{\mu})} \frac{p(\boldsymbol{\sigma}^{2'})}{p(\boldsymbol{\sigma}^2)} \frac{\pi(\boldsymbol{\mu} \,|\, \boldsymbol{e}, \boldsymbol{\sigma}^2)}{\pi(\boldsymbol{\mu}' \,|\, \boldsymbol{e}', \boldsymbol{\sigma}^{2'})} \frac{\pi(\boldsymbol{\sigma}^2 \,|\, \mathcal{Y}, \boldsymbol{e})}{\pi(\boldsymbol{\sigma}^{2'} \,|\, \mathcal{Y}, \boldsymbol{e}')}, \end{aligned}$$

where

$$\begin{aligned} p(e_k') &= \frac{1}{N_h - 1}, \\ p(\boldsymbol{\sigma}^{2'}{}_k) &= \frac{\sigma_k^{2'a-1} b^a \exp\left(-b\sigma^{2'}{}_k\right)}{\Gamma(a)}, \\ p(\mu_k) &= \frac{1}{\sqrt{2\pi\sigma_b^2}} \exp\left(-\frac{\mu_k^2}{2\sigma_b^2}\right), \end{aligned}$$

[1]Throughout this thesis, MCMC steps are marked by an uppercase letter representing the type of the analysis (for example phenotypic or phylogeographic), a number corresponding to a distinct Metropolis-Hastings update, and a lowercase letter indicating the proposal step within each MH update.

and

$$
\begin{aligned}
f(\mathcal{Y} \,|\, \mathbf{e}', \boldsymbol{\mu}', \boldsymbol{\sigma}') &= \prod_{i,l} \sqrt{\frac{\sigma^{2\prime}{}_k}{2\pi}} \exp\left(-\frac{\sigma^{2\prime}{}_k (y_{il} - \mu'_{c_{il}})^2}{2} \right), \\
\pi(\boldsymbol{\mu}' \,|\, \boldsymbol{e}', \boldsymbol{\sigma}^{2\prime}) &= \prod_{k=1}^{K+1} \sqrt{\frac{n_k \sigma^{2\prime}{}_k + \frac{1}{\sigma_\mu^2}}{2\pi}} \exp\left(-\frac{1}{2} \left(\mu'_k - \frac{\sigma^{2\prime}{}_k n_k \overline{y}'_k}{\sigma^{2\prime}{}_k n'_k + 1/\sigma_\mu^2} \right)^2 \left(\sigma^{2\prime}{}_k n'_k + \frac{1}{\sigma_\mu^2} \right) \right), \\
\pi(\boldsymbol{\sigma}^{2\prime} \,|\, \mathcal{Y}, \boldsymbol{e}') &= \prod_{k=1}^{K+1} \frac{\sigma^{2\,a+\frac{n_k}{2}-1}_k \left(\frac{\sum (y_{jl} - \overline{y}_k)^2}{2} + b \right)^{a+\frac{n_k}{2}}}{\Gamma(a + \frac{n_k}{2})} \exp\left\{ - \left(\frac{\sum (y_{jl} - \overline{y}_k)^2}{2} + b \right) \sigma_k^2 \right\}.
\end{aligned}
$$

With probability $\min(1, A_A)$ we accept

$$(\boldsymbol{e}^{(t+1)}, \mu'', \boldsymbol{\sigma}'') = (\boldsymbol{e}', \mu', \boldsymbol{\sigma}'),$$

and otherwise

$$(\boldsymbol{e}^{(t+1)}, \mu'', \boldsymbol{\sigma}'') = (\boldsymbol{e}^{(t)}, \mu^{(t)}, \boldsymbol{\sigma}^{(t)}).$$

A2: We then generate $\sigma_k^{(t+1)}$ for each k from the posterior conditional

$$\sigma_k^2 \,|\, \mathcal{Y}, \boldsymbol{e}^{(t+1)}, \mu''_k$$

given in Equation (2.4), and accept with probability 1 as in standard Gibbs sampler; see Brooks 1998.

A3: Finally, similarly to step A2, we generate $\boldsymbol{\mu}^{(t+1)}$ from

$$\mu_k \,|\, \mathcal{Y}, \boldsymbol{e}^{(t+1)}, \boldsymbol{\sigma}^{(t+1)},$$

and return to step A1.

The chain described above is aperiodic since there is a non-zero probability of staying at the same state. In addition, it is irreducible, since it is always possible to move from any clustering $\boldsymbol{e_1}$ to any other $\boldsymbol{e_2}$ in a maximum of K iterations by changing each of the edges in $\boldsymbol{e_1}$ to the corresponding ones in $\boldsymbol{e}_2$ at each iteration, and using the fact that the proposal distributions of $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$ have infinite support. Hence, the stationary distribution of the chain is indeed the target distribution (2.5) of the parameters of interest, and the chain converges to equilibrium as $t \to \infty$ almost surely (see Norris, 1997).

We continue with some technical and statistical issues.

1. As with all MCMC clustering methods, an inherent label-switching issue arises here because of the permutational symmetry of the cluster parameters. Clusters have to be labelled consistently from iteration to iteration in order to draw conclusions about the parameters of each one. For example, if there are three clusters with means $(-1, 0, 1)$ at one iteration and $(0, 0.5, 1)$ (subject to permutation) at the next, it is not clear how to relate one to the next. To overcome label-switching we employ two approaches, proposed by Stephens (2000) and Scott and Wang (2006) respectively, presented in Appendix A. For computational efficiency the preferred method is by Scott and Wang (2006), which relies upon inferring the optimal labelling at each iteration after burn-in by relating it to the maximum likelihood estimate of clusterings inferred during burn-in.

2. In problems involving very large trees, an efficient proposal for $\boldsymbol{e}$ is essential, since most clusterings are highly improbable. An alternative proposal for $\boldsymbol{e}$ at Step A1a is to propose an edge uniformly from the edges adjacent (i.e., ones that share one of the endpoints) to the current ones.

   The disadvantage of such a proposal kernal is that the chain moves slowly to and from distant regions in the graph and may get stuck in local optima. Although it is often beneficial, it should not be applied without strictly monitoring convergence to investigate multimodality of the clusterings.

   It is also possible to update all K edges simultaneously, but this leads to a very low acceptance rate, although it may improve mixing and perform better with multimodal clustering distributions.

3. If a cluster is empty, then we treat it as not containing any information, as we would prior to the data. Although such clusters are physically nonsensical, we will see in later sections (see Sections 2.4 and 3.9) that it is important to allow them.

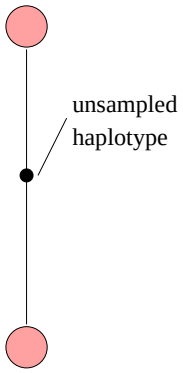


Figure 2.3: *An unsampled haplotype of degree two.*

4. Statistically more important, in a situation like Figure 2.3 where an unsampled haplotype is connected to only two other haplotypes (i.e., it has degree two), proposing either of the two edges to be significant will have exactly the same effect in terms of the clustering. It is not possible to distinguish between the two mutations involved, and the results of our posterior probability should be interpreted as the sum of the posterior masses of the two significant edges.

5. If a mutation which is subsequently reversed by back-mutation is suspected to be causal, then a move proposing that it divides the data into two different clusters naturally implies that, in fact, the data should be split into three clusters. This means that the dimension of the parameter space may vary from iteration to iteration. Especially when the number of causal mutations is assumed to be greater than 1, the complexity of the analysis may thus increase greatly. However, when merely associations are investigated, this is not necessary: even in the case where a mutation appears at more than one edge in the mutational tree, the two edges are analyzed independently.

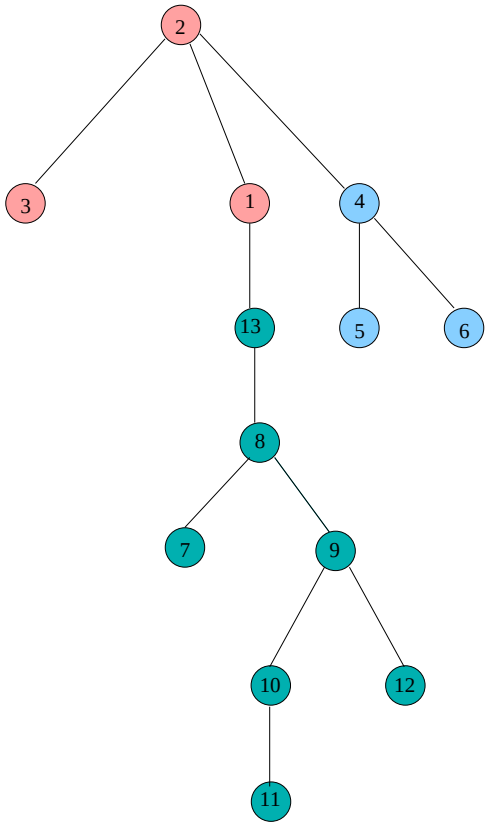


Figure 2.4: *The tree topology and clustering used in this example*

Example: simulated dataset We use the following simulated dataset S1 to demonstrate our algorithm. We consider a dataset of 13 haplotypes (shown in Figure 2.4), each of which appears 5 times in the sample. We pick two mutations to be associated with a significant change in a one-dimensional phenotype, and generate phenotypic data for each haplotype. Specifically, we ensure that the mutations that occurred between haplotypes 1-13 and 2-4 are significant, so that there are three clusters (depicted in green, light blue and pink in Figure 2.4). Observations from the light blue cluster are normally distributed with $\mathcal{N}(0,1)$, the pink cluster $\mathcal{N}(-1,1)$ and the green cluster $\mathcal{N}(1,1)$.

We specified the priors in Equation (2.1), with $a = b = 0.0001$ and $\sigma_b^2 = 1000$. We ran our analysis using a uniform significant edge proposal starting from two different starting points for 5000 steps, and discarded the first 500 as burn-in. The maximum a posteriori (MAP) estimate of the clustering coincides with the true one for both chains, with posterior mass > 0.998. The estimated cluster means and variances are given in Table 2.1.

(μ_1, σ_1^2)	(μ_2, σ_2^2)	(μ_3, σ_3^2)
(0.22,0.54)	(0.77,0.25)	(-1.25,0.56)

Table 2.1: *Estimated means and variances for the clusters with distributions $\mathcal{N}(0,1)$, $\mathcal{N}(1,1)$, $\mathcal{N}(-1,1)$ of dataset S1. The estimates coincide with the sample means and variances of the correct clusters.*

We investigate the rate of convergence of the sampled Markov chains. Considering the trace and density plots of the 6 parameters shown in Figure 2.5, the mixing for both chains seems very good and the estimates for the posterior densities for the two seeds match very well. The acceptance rate with a uniform proposal distribution was 0.06. In this case it is clear that convergence is almost guaranteed since there are only $\binom{13}{2} = 78$ different clusterings to be assessed within the MCMC.

Figure 2.6 shows the Gelman and Rubin Potential Scale Reduction Factor plots for S1, suggesting that our chain has indeed converged; see Gelman and Rubin (1992); Brooks and Gelman (1998).

Using an adjacent proposal for the significant edges $\boldsymbol{e}$ we repeated the same algorithm for the same dataset S1. The results are illustrated in Figure 2.7. In this case the acceptance rate rose to 0.10, and the speed of convergence was improved. This is also suggested by the Brooks-Gelman plot in Figure 2.8.

To investigate the accuracy of our analysis when the data are less clearly structured, we repeated the experiment by generating dataset S2 from $\mathcal{N}(0,16)$, $\mathcal{N}(-1,16)$ and $\mathcal{N}(1,16)$. We present the corresponding trace plots in Figure 2.9. Note that, due to the increased variance, a larger number of iterations was required, here 50,000. Although our method identified the

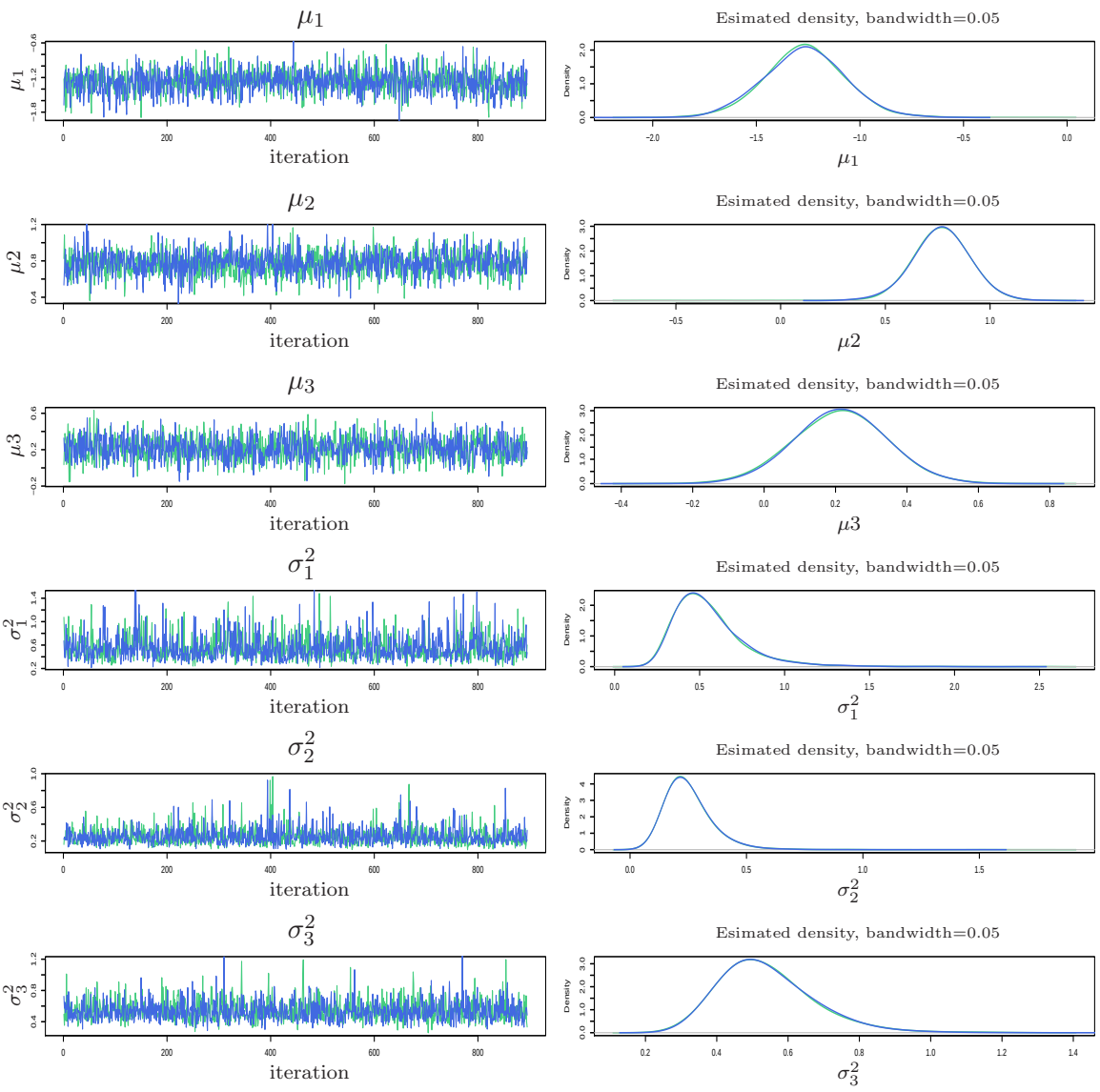


Figure 2.5: *Trace and density plots for the 6 parameters of the simulated dataset S1, using a uniform proposal. The burn-in period is not shown: this is because during burn-in, clusters are unidentifiable, because the labelling is initiated post burn-in. The trace plots are thinned by a factor of 5.*

correct mutations, observation of the density plots shows an unusual shape for the posterior distribution of μ_1 and σ_3^2, by a slight "bump" on the curves. This is because in this case, the clustering with second highest posterior mass (probability 0.05 vs 0.94 of the MAP estimate) causes multimodality in the distribution of the means.

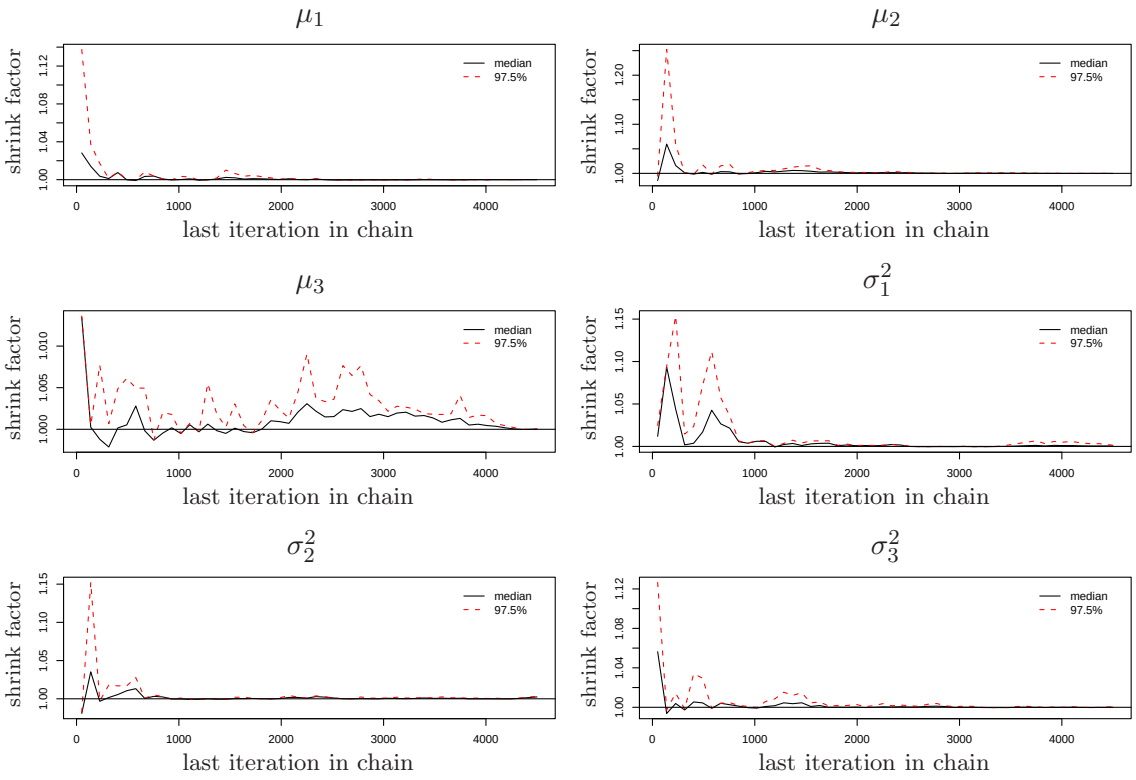


Figure 2.6: *Gelman and Rubin potential scale reduction factors using a uniform proposal for dataset S1. All shrink factors are below 1.1, suggesting that our chains have reached convergence.*

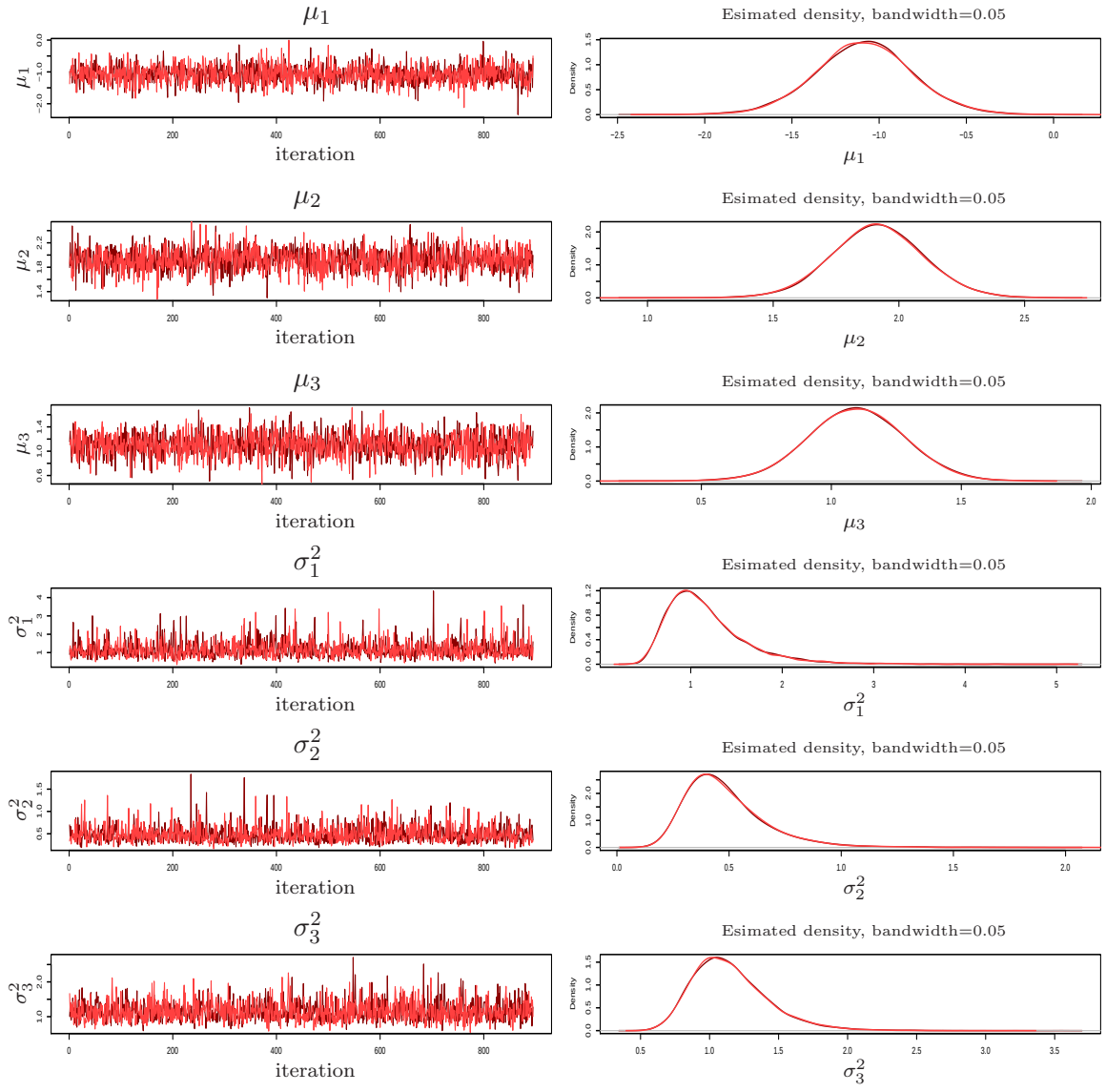


Figure 2.7: *Trace and density plots for the 6 parameters, using an adjacent proposal for dataset S1. The trace plots are thinned by a factor of 10.*

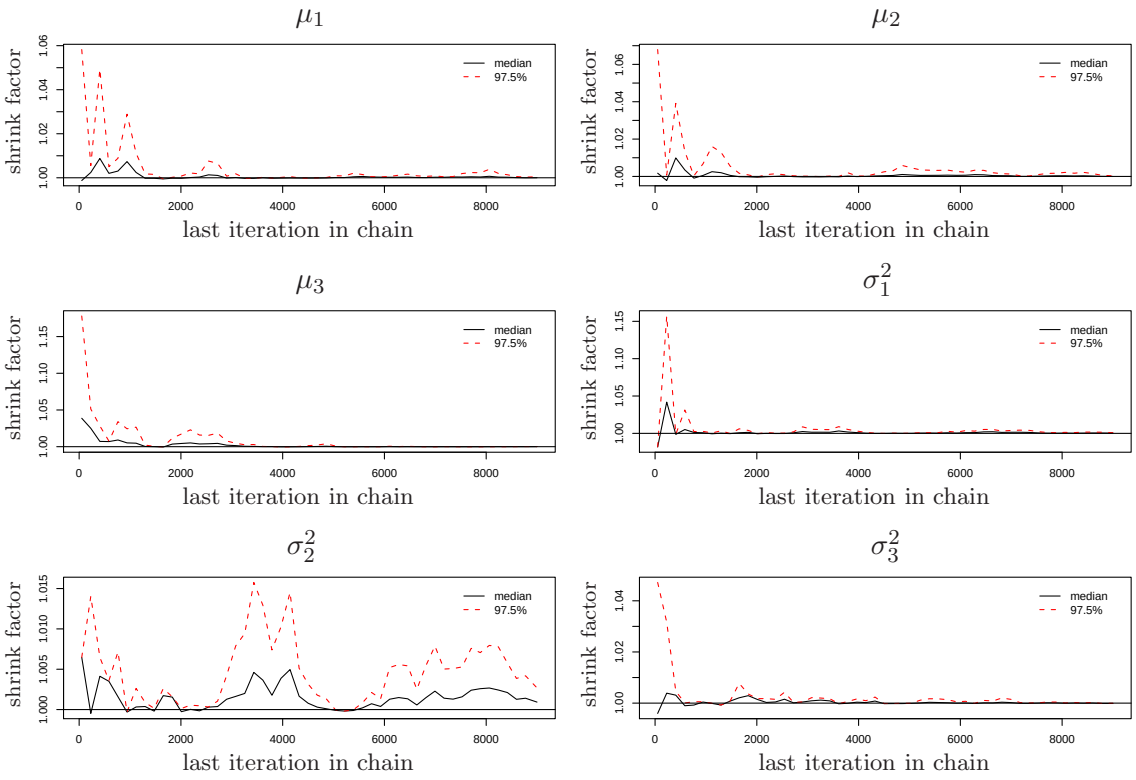


Figure 2.8: *Gelman and Rubin potential scale reduction factors for the adjacent proposal for dataset S1. By comparing the values of the shrink factors to the ones obtained using a uniform proposal, we see that the adjacent proposal results in a faster convergence.*

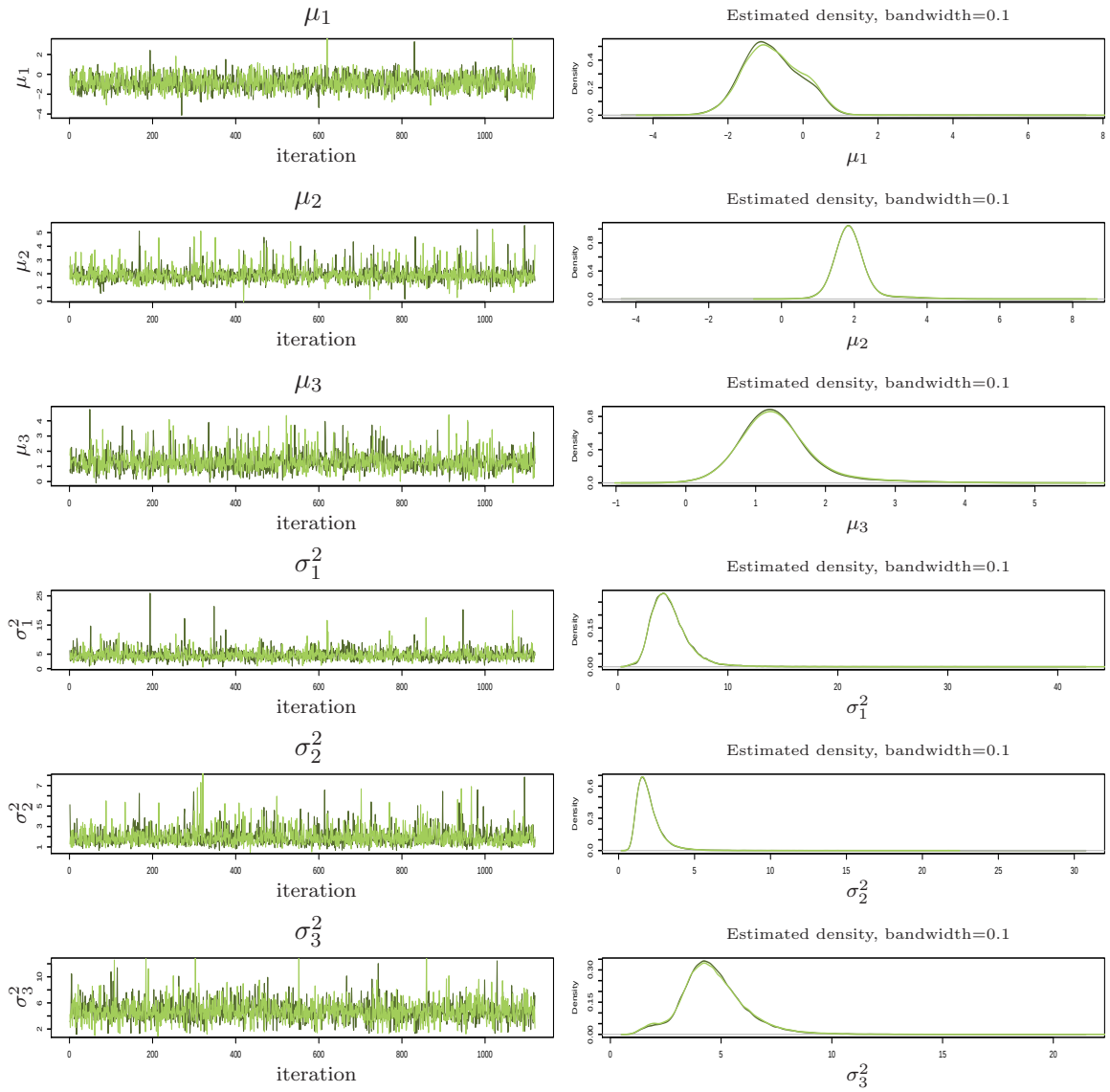


Figure 2.9: *Trace and density plots for normally distributed data with variance 4 using a uniform proposal for dataset S2. The trace plots are thinned by a factor of 40. Notice the "bump" of the first and last estimated density plots, indicating multimodality of the parameters.*

2.2 Phenotypic clustering for multi-dimensional traits

We now extend the analysis described in the previous section so that it is applicable to d-dimensional phenotypic data. As before, our objective is to identify mutations which are associated with a significant change in the phenotype, and we assume that the haplotype tree and the number of significant mutations K are known.

Phenotypic datasets often include a large number of measurements, many of which are independent of the mutations we are studying. Thus, we want to modify our method so that it is also possible to identify the specific measurements which show the most significant correlations with the mutations considered.

In order to allow the number of dimensions to vary, we introduce a binary variable z_i, $i = 1, \ldots, d$ which is 1 if that dimension is taken into account and 0 otherwise. Here we denote $|z|$ the magnitude of z, equal to the number of non-zero elements. This approach is presented by Tadesse et al. (2005).

Intuitively, the aim is to find which phenotypic traits maximise the ratio of the likelihood of the data before and after the clustering. The equation

$$\frac{\mathbb{P}(\mathcal{Y} \,|\, \mathbf{z}, \boldsymbol{c}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\mathbb{P}(\mathcal{Y} \,|\, \mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma})} = \frac{\mathbb{P}(\boldsymbol{c} \,|\, \mathcal{Y}, \mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\mathbb{P}(\boldsymbol{c} \,|\, \mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma})} \propto \mathbb{P}(\boldsymbol{c} \,|\, \mathcal{Y}, \mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

implies that we are interested in the posterior distribution of the clustering given the data. Note for the above identity that we are assuming that $\boldsymbol{c}$ has a uniform prior distribution.

As before, the data are transformed so that the sample mean of each component is 0 and the variance of each component is I_d, unless there is prior information suggesting otherwise.

For each cluster k the Model (2.1) thus extends to

$$\begin{aligned}
z_i &\sim \mathcal{B}(1, 0.5) \quad i = 1, \ldots, d, \\
\mathcal{Y} \,|\, \boldsymbol{e}, \boldsymbol{\mu}, \boldsymbol{\Sigma} &\sim \mathcal{N}_{|z|}(\boldsymbol{\mu}_i, \Sigma_i), \\
e_k &\sim \mathcal{U}\{1, \ldots, N_h\} \text{ without replacement} \\
\mu_k \,|\, \boldsymbol{z}, \Sigma_k &\sim \mathcal{N}_{|z|}\left(\mathbf{0}, V\right), \\
\Sigma_k &\sim \mathcal{IW}(\gamma, \Psi), \\
\gamma \,|\, \boldsymbol{z} &\sim \mathcal{U}\{1, \ldots, g\} \quad \gamma > |\boldsymbol{z}| + 1,
\end{aligned} \tag{2.6}$$

where $\mathcal{IW}$ denotes the Inverse Wishart distribution, V is set large and γ, g small to allow for vague priors. Note here that $\mu_k \,|\, z$ and $\Sigma_k \,|\, z$ have dimension $|z|$ and $|z| \times |z|$ respectively. In other words, the vectors and matrices in Model (2.6) change size depending on z, requiring a RJMCMC algorithm. The parameter structure of the hierarchical Model 2.6 is summarized

graphically in Figure 2.10.

Generalizing to $d > 1$ dimensions requires adaptation of the conjugate prior for the covariance matrices. A natural conjugate choice is the Inverse Wishart distribution, which is a generalization of the Gamma distribution in Model (2.1). However, it incurs some serious drawbacks. The prior variance of Σ decreases as the degrees of freedom γ increase, but is required to be a positive integer and cannot be arbitrarily small. This implies that we cannot have a prior as vague as $\mathcal{G}(0.0001, 0.0001)$ as we did in the one-dimensional case; in fact it is required that $\gamma > |z| + 1$, otherwise the prior becomes improper (but may still be usable). This means that for a moderately large d, the prior distribution for Σ_k has a very small variance, which is unrealistic since this is merely based on a statistical constraint rather than any prior knowledge of the data. As a result, it is often beneficial to assume a Generalized Inverse Wishart ($\mathcal{GIW}$); see Garthwaite and Al-Awadhi (2001). Alternatively, if the different phenotypic measurements can be assumed to be independent, independent Gamma distributions can be assumed for each component (in other words, the covariance matrices are diagonal).

The hyperparameter g is introduced to reduce the sensitivity of our results on the prior of Σ_k by letting the data choose an appropriate prior (see Richardson and Green, 1997). Different values of γ will dictate larger or smaller values for K. A prior which favours clusters with smaller variance will lead to a larger K being chosen, so that the data can be split into many small clusters. We will see in Section 2.4 that the posterior estimate of K is highly sensitive to the choice of γ. We wish to reduce this dependence, and hence introduce the hyperparameter.

From the distributions in Model 2.6 we obtain

$$\Sigma_k \,|\, \boldsymbol{z}, \mathcal{Y}, \boldsymbol{e}, \gamma \;\sim\; \mathcal{IW}^{\boldsymbol{z}}\left(n_k + \gamma, \Psi + \sum_{j,l} \mathbb{I}_{\{c_{jl}=k\}} \boldsymbol{y}_{jl}\boldsymbol{y}_{jl}^T - n_k \overline{\boldsymbol{y}}_i \overline{\boldsymbol{y}}_k^T\right), \tag{2.7}$$

$$\boldsymbol{\mu}_k \,|\, \boldsymbol{z}, \mathcal{Y}, \boldsymbol{e}, \Sigma_i \;\sim\; \mathcal{N}_d^{\boldsymbol{z}}\left(\frac{\Sigma_k^{-1} n \overline{\boldsymbol{y}}}{V^{-1} + n\Sigma_k^{-1}}, \frac{1}{V^{-1} + n\Sigma_k^{-1}}\right), \tag{2.8}$$

$$\Sigma_k \,|\, \boldsymbol{z}, \mathcal{Y}, \boldsymbol{e}, \gamma, \boldsymbol{\mu} \;\sim\; \mathcal{IW}^{\boldsymbol{z}}\left(N + \gamma, \Psi + \sum_{j,l} \mathbb{I}_{\{c_{jl}=k\}} (\boldsymbol{y}_{jl} - \boldsymbol{\mu}_k)^T (\boldsymbol{y}_{jl} - \boldsymbol{\mu}_k)\right). \tag{2.9}$$

We want to construct a MCMC sampler with target distribution

$$\pi(\boldsymbol{z}, \boldsymbol{e}, \gamma, \Sigma, \boldsymbol{\mu} \,|\, \mathcal{Y}) \propto f(\mathcal{Y} \,|\, \boldsymbol{z}, \boldsymbol{e}, \boldsymbol{\Sigma}, \boldsymbol{\mu}) p(K) p(\boldsymbol{\mu} \,|\, \boldsymbol{z}) p(\gamma \,|\, \boldsymbol{z}) p(\boldsymbol{\Sigma} \,|\, \boldsymbol{z}, \gamma) p(\boldsymbol{e}).$$

The MCMC algorithm takes the following form. The chain is initialized by generating $\boldsymbol{z}^{(0)}$, $\boldsymbol{e}^{(0)}$, $\gamma^{(0)}$, $\boldsymbol{\mu}^{(0)}$, $\boldsymbol{\Sigma}^{(0)}$ from their prior distributions. Subsequently iterate the steps described

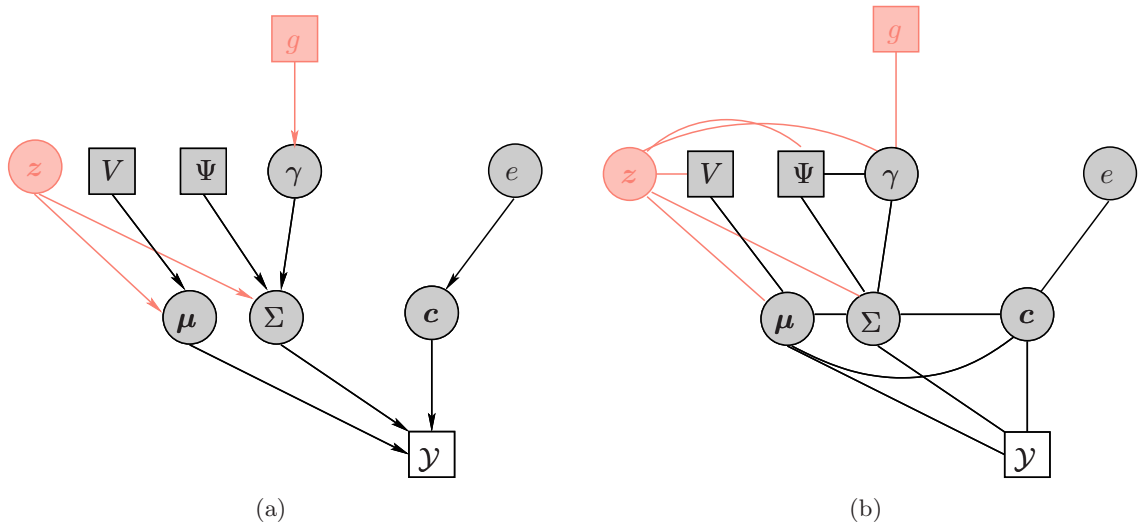


Figure 2.10: *The DAG for the phenotypic clustering of multi-dimensional traits, adapted for more dimensions. Here the black parameters are the same as in the one-dimensional case; new parameters are shown in pink.*

below.

B1: Propose $\boldsymbol{z}'$, increasing or decreasing $|\boldsymbol{z}|$ by 1. Note that $|\boldsymbol{z}|$ must be positive and cannot exceed d.

B2a: Propose a value for γ from its prior $\mathcal{U}\{|\boldsymbol{z}|+2,\ldots,g\}$ and conditional on $\gamma > |\boldsymbol{z}|+1$.

B2b: Update $\boldsymbol{e}$ in the same way as in the one-dimensional case in Step A1a. Pick one of the K edges e_i at random and change it to another one randomly from the tree with a uniform proposal. Identify the $K+1$ clusters and calculate their sizes n_k and sample means $\overline{\boldsymbol{y}}_k$.

B2c: Propose a set of new covariance matrices $\boldsymbol{\Sigma}'_k$ from the distribution $\Sigma_k \,|\, \boldsymbol{z}', \mathcal{Y}, \boldsymbol{e}', \gamma'$ given in Equation (2.7).

B2d: Propose $\boldsymbol{\mu}'_k$ from $\boldsymbol{\mu}_k \,|\, \boldsymbol{z}', \mathcal{Y}, \boldsymbol{e}', \Sigma'_k$ above (Equation 2.8).

B2e: The acceptance probability is given by $\min(1, A_B)$, where

$$\begin{aligned} A_B &= \frac{\pi(\boldsymbol{z}', \boldsymbol{e}', \gamma', \boldsymbol{\mu}', \boldsymbol{\Sigma}' \,|\, \mathcal{Y})}{\pi(\boldsymbol{z}, \boldsymbol{e}, \gamma, \boldsymbol{\mu}, \boldsymbol{\Sigma} \,|\, \mathcal{Y})} \frac{q(\boldsymbol{z}' \to \boldsymbol{z})}{q(\boldsymbol{z} \to \boldsymbol{z}')} \frac{q(\gamma' \to \gamma)}{q(\gamma \to \gamma')} \frac{q(\boldsymbol{\mu}' \to \boldsymbol{\mu})}{q(\boldsymbol{\mu} \to \boldsymbol{\mu}')} \frac{q(\boldsymbol{\Sigma}' \to \boldsymbol{\Sigma})}{q(\boldsymbol{\Sigma} \to \boldsymbol{\Sigma}')} \frac{q(\boldsymbol{e}' \to \boldsymbol{e})}{q(\boldsymbol{e} \to \boldsymbol{e}')} \\ &= \frac{f(\mathcal{Y} \,|\, \boldsymbol{z}', \boldsymbol{e}', \gamma', \boldsymbol{\mu}', \boldsymbol{\Sigma}')}{f(\mathcal{Y} \,|\, \boldsymbol{z}, \boldsymbol{e}, \gamma, \boldsymbol{\mu}, \boldsymbol{\Sigma})} \frac{p(\boldsymbol{\mu}' \,|\, \boldsymbol{z}')}{p(\boldsymbol{\mu} \,|\, \boldsymbol{z})} \frac{p(\boldsymbol{\Sigma}' \,|\, \boldsymbol{z}', \gamma')}{p(\boldsymbol{\Sigma} \,|\, \boldsymbol{z}, \gamma)} \frac{q(\boldsymbol{z}' \to \boldsymbol{z})}{q(\boldsymbol{z} \to \boldsymbol{z}')} \\ &\quad \times \frac{\pi(\boldsymbol{\mu} \,|\, \boldsymbol{z}, \boldsymbol{e}, \boldsymbol{\Sigma})}{\pi(\boldsymbol{\mu}' \,|\, \boldsymbol{z}', \boldsymbol{e}', \boldsymbol{\Sigma}')} \frac{\pi(\boldsymbol{\Sigma} \,|\, \boldsymbol{z}, \mathcal{Y}, \boldsymbol{e})}{\pi(\boldsymbol{\Sigma}' \,|\, \boldsymbol{z}', \mathcal{Y}, \boldsymbol{e}')} |J| \end{aligned}$$

Here

$$
\begin{aligned}
p(\boldsymbol{z}) &= \frac{1}{2^d} \\
p(\boldsymbol{e}) &= \prod_{i=1}^{K} \frac{1}{N-i+1}, \\
p(\gamma) &= \frac{1}{g} \\
q(\boldsymbol{z} \to \boldsymbol{z}') &= 0.5 + 0.5 \times \mathbb{1}_{\{|z|=0\}} + 0.5 \times \mathbb{1}_{\{|z|=d\}} \\
p(\boldsymbol{\mu} \,|\, \boldsymbol{z}) &= \prod_{k=1}^{K+1} (2\pi)^{-d/2} \times |V|^{-1/2} \exp\left(-\frac{1}{2}\boldsymbol{\mu}_k^T \, |V|^{-1} \, \boldsymbol{\mu}_k\right), \\
p(\Sigma \,|\, \boldsymbol{z}, \gamma) &= \prod_{k=1}^{K+1} \left(2^{\gamma d/2} \times \pi^{d(d-1)/4} \times \prod_{k=1}^{d} \Gamma\left(\frac{\gamma+1-k}{2}\right)\right)^{-1} \times |\Psi|^{\gamma/2} \\
&\quad \times |\Sigma_k|^{(d+\gamma+1)/2} \times \exp\left(-\frac{1}{2} tr\left(\Psi \Sigma_k^{-1}\right)\right),
\end{aligned}
\tag{2.10}
$$

and it can be calculated that

$$
\begin{aligned}
f(\mathcal{Y} \,|\, \boldsymbol{z}, \mathbf{e}, \boldsymbol{\mu}, \Sigma) &= \prod_{i,l} (2\pi)^{-d/2} \, |\Sigma_{c_{il}}|^{-1/2} \exp\left(-\frac{1}{2}\left(\boldsymbol{y}_{il} - \boldsymbol{\mu}_{c_{il}}\right)^T \Sigma_{c_{il}}^{-1} \left(\boldsymbol{y}_{il} - \boldsymbol{\mu}_{c_{il}}\right)\right), \\
\pi(\mu_k \,|\, \boldsymbol{z}, \boldsymbol{e}, \boldsymbol{\Sigma}) &= \prod_{i,j} \mathbb{1}_{\{c_{ij}=k\}} (2\pi)^{-d/2} \times \left(\frac{1}{V^{-1} + n_k \Sigma_k^{-1}}\right)^{-1/2} \times \\
&\quad \exp\left(-\frac{1}{2}\left(\boldsymbol{\mu}_k - \frac{n_k \boldsymbol{y}_{ij} \Sigma_k^{-1}}{V^{-1} + n_k \Sigma_k^{-1}}\right)^T \left(\frac{1}{V^{-1} + n_k \Sigma_k^{-1}}\right)^{-1} \left(\boldsymbol{\mu}_k - \frac{n_k \boldsymbol{y}_{ij} \Sigma_k^{-1}}{V^{-1} + n_k \Sigma_k^{-1}}\right)\right), \\
\pi(\Sigma_k \,|\, \boldsymbol{z}, \mathcal{Y}, \boldsymbol{e}, \gamma) &= \left(2^{(n_k+\gamma)d/2} \times \pi^{d(d-1)/4} \prod_{i=1}^{d} \Gamma\left(\frac{n_i + \gamma + 1 - i}{2}\right)\right)^{-1} \\
&\quad \times |S|^{(n_k+\gamma)/2} \times |\Sigma_k|^{-(n_k+\gamma+d+1)/2} \times \exp\left(-\frac{1}{2} tr\left(S \Sigma_k^{-1}\right)\right), \text{ where} \\
S &= \Psi + \prod_{i,j} \mathbb{1}_{\{c_{ij}=k\}} \boldsymbol{y}_{ij} \boldsymbol{y}_{ij}^T - n_i \overline{\boldsymbol{y}}_k \overline{\boldsymbol{y}}_k^T.
\end{aligned}
$$

Finally, the determinant of the Jacobian in this case is equal to one, since almost all parameters are generated afresh for $\boldsymbol{z}'$, independent of previous values. The only exception is $\boldsymbol{e}$, of which only one entry is changed. In other words, the move can be expressed as a set of independent moves (and leaving some parameters unchanged), yielding a Jacobian which is equal to the identity matrix with determinant $|J| = 1$; see Green (1995).

We calculate the ratio A_B and accept the step with probability $\min(1, A_B)$.

If we accept, we set

$$(\boldsymbol{z}^{(t+1)}, \boldsymbol{e}^{(t+1)}, \gamma^{(t+1)}, \boldsymbol{\mu}'', \boldsymbol{\Sigma}'') = (\boldsymbol{z}', \boldsymbol{e}', \gamma', \boldsymbol{\mu}', \boldsymbol{\Sigma}'),$$

otherwise we set

$$(\boldsymbol{z}^{(t+1)}, \boldsymbol{e}^{(t+1)}, \gamma^{(t+1)}, \boldsymbol{\mu}'', \boldsymbol{\Sigma}'') = (\boldsymbol{z}^{(t)}, \boldsymbol{e}^{(t)}, \gamma^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\Sigma}^{(\boldsymbol{t})}),$$

B3: We then generate $\boldsymbol{\Sigma}^{(t+1)}$ from the posterior conditional given in (2.9)

$$\boldsymbol{\Sigma} \,|\, \boldsymbol{z}^{(t+1)}, \mathcal{Y}, \boldsymbol{e}^{(t+1)}, \gamma^{(t+1)}, \boldsymbol{\mu}''$$

(and accept with probability one as in standard Gibbs sampler).

B4: Lastly, same as step 2, we generate $\boldsymbol{\mu}^{(t+1)}$ from

$$\boldsymbol{\mu} \,|\, \boldsymbol{z}^{(t+1)}, \mathcal{Y}, \boldsymbol{e}^{(t+1)}, \boldsymbol{\Sigma}^{(t+1)}.$$

B5 Repeat steps B2-B4, keeping the dimension indicator z constant, then go back to step B1.

As in the previous section, this chain is irreducible and aperiodic, since the clustering moves are the same and all proposal distributions ensure support over the whole parameter space.

Example: simulated dataset Using the same haplotype structure and significant mutations as the previous example (Section 2.1), we generated a three-dimensional dataset S3. In this case observations from the blue cluster are normally distributed with $\mathcal{N}_3(\mu_1, \Sigma_1)$, the red cluster $\mathcal{N}_3(\mu_2, \Sigma_2)$ and the green cluster $\mathcal{N}_3(\mu_3, \Sigma_3)$, where the means and variances are shown below. We assumed priors with $V = 1000 I_3,\ \Psi = I_3,\ g = 20$.

We repeated the phenotypic clustering algorithm for 5,000 iterations, this time aiming to both identify mutations associated with changes in the phenotype, but also to infer which phenotype shows the strongest significant change. In this case the distinction is clear in the data, and the first and third components are the ones which show a significant association with the clustering with postrior mass 1. We plot the most problematic trace and density plots for the first and third components in Figure 2.11.

$$\mu_1 = \begin{pmatrix} -2 \\ 0 \\ 0 \end{pmatrix} \quad , \quad \Sigma_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$
$$\mu_2 = \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} \quad , \quad \Sigma_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$
$$\mu_3 = \begin{pmatrix} 0 \\ 0 \\ -5 \end{pmatrix} \quad , \quad \Sigma_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Although for most of the chains the mixing appears very good, we notice that the trace plot for μ_{31}, for example, shows multiple jumps in the chain for both seeds, most likely caused by a slight multimodality of the clustering, causing the means to change significantly.

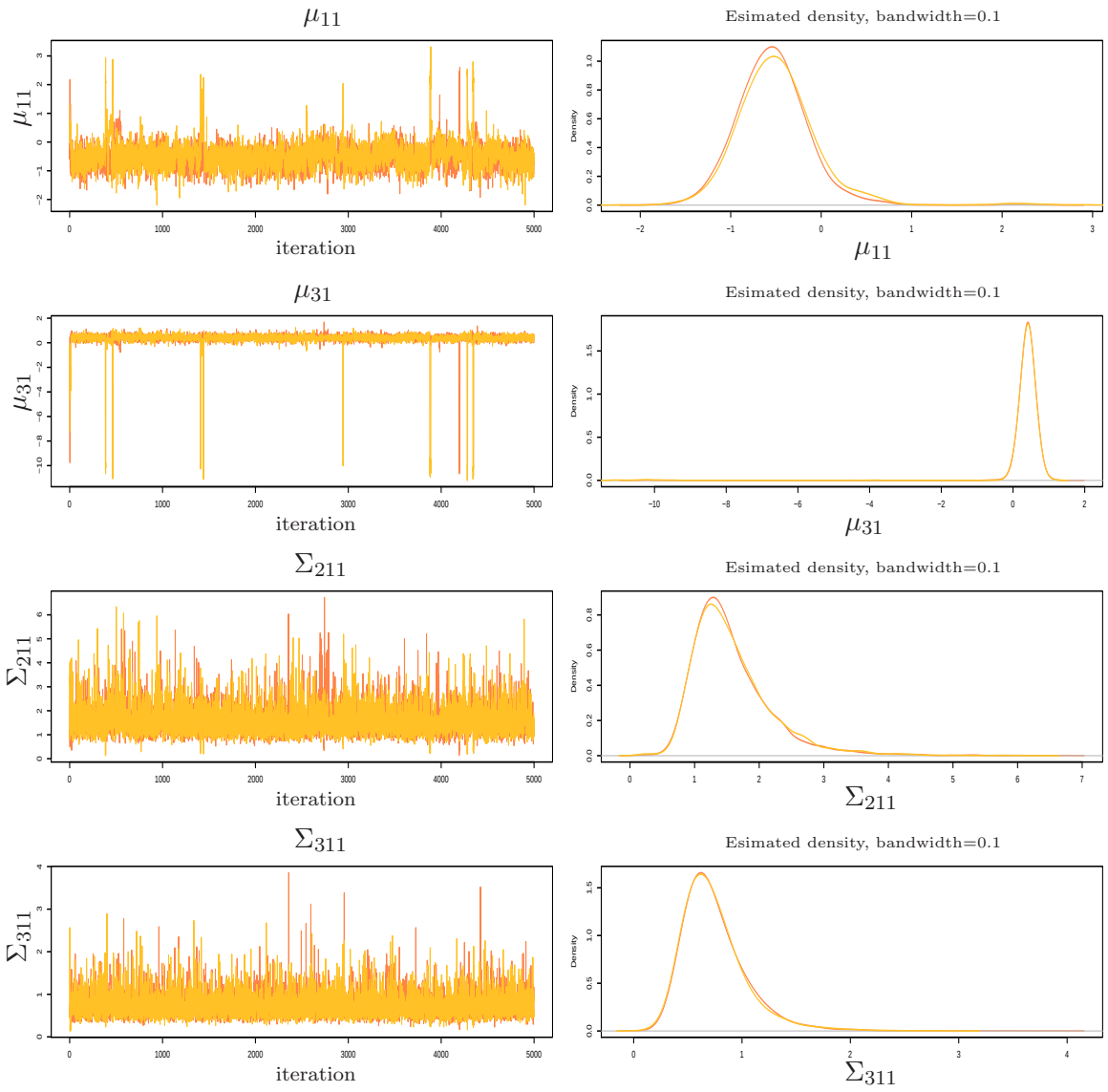


Figure 2.11: *Trace and density plots for dataset S3 of a few representative parameters in the three-dimensional case, showing excellent mixing.*

2.3 Phylogeographic clustering

In this section we are given a sample of N DNA sequences along with the corresponding geographical location (instead of the phenotype) where each sequence was sampled. The sequences correspond to N_h haplotypes, and we assume that the haplotype tree is fixed. We wish to identify population clusters which are consistent with the geographic and genetic information available.

Our first task is to devise clusterings defined on haplotype trees which are consistent with the geographical distribution of haplotypes, following phylogeographic phenomena such as range expansion (see Templeton, 1998; Avise, 2000). Because of the complexity of such events, we do not model them explicitly. The main assumption we make is that *sequences* (as opposed to haplotypes in the previous sections) can be divided into *hard geographical clusters*, implying that each sequence is assumed to belong to a specific geographical population cluster (see De Iorio and Griffiths, 2004b). We thus aim to cluster individuals such that, genetically, haplotypes may be shared across clusters due to moving individuals. Here we restrict the phylogeoraphic setting by considering a simple island migration model (see De Iorio and Griffiths, 2004b), but the approach can easily be extended.

2.3.1 Construction of phylogeographic clusters

In this subsection we develop a construction of phylogeographic clusters on haplotype trees which are consistent with simple migration island models, yielding shared haplotypes. We begin by presenting examples of population movement in order to provide an intuitive understanding of the effect of phylogeographic events on a haplotype tree rather than a coalescent. We then describe the properties of the phylogeographic clustering in detail, and finally we describe Algorithm 2.3.1 which defines all clusterings that are consistent with migrating haplotype.

First recall the simple migration model used by De Iorio and Griffiths (2004b), and consider a scenario where an ancestral population (depicted as green in Figure 2.12 below) is the source for the colonization of another three, shown in yellow, pink and light blue. We need to translate such migration scenarios in terms of their effect on haplotype trees.

Consider the following example. Assume that haplotype i belongs to population A. If one of the individuals carrying haplotype i migrates from population cluster A to start a new population B, then haplotype i will be found in both clusters A and B. Assuming that no more migration (or other phylogeographic) events occurred, all of the descendants of i will either belong to cluster A (if their ancestral sequence belongs to cluster A) or cluster B (if their ancestral sequence belongs to cluster B). An example of the resulting haplotype tree

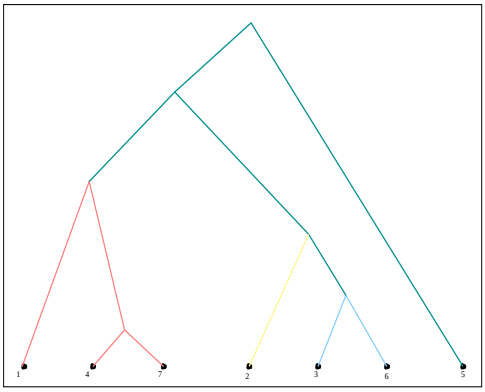


Figure 2.12: *A coalescent with subdivided populations: the green is the ancestral population, from which sequences subsequently migrated to found the pink, yellow and light blue populations.*

is illustrated in Figure 2.13, where as usual the colour indicates the cluster to which each sequence belongs. To illustrate that more than one migration event may occur, and also that some haplotypes may migrate to more than two clusters, we extend the above example in Figure 2.14, which represents the collapsed coalescent of Figure 1.15 onto a haplotype tree (here simplified for graphical simplicity).

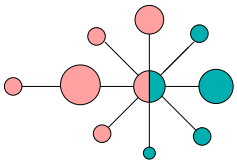


Figure 2.13: *Example of a migrating haplotype shared between two populations, here green and pink. As before, nodes represent haplotypes, with the size of the circle representing the number of times that haplotype appears in the sample. The colour of each node shows the population cluster in which it belongs, with one haplotype being shared between the two clusters. Two haplotypes are connected by an edge if they are one mutation apart.*

We now introduce a general setting in which phylogeographic clusterings can be described. The clusters are seeded by K migrating haplotypes denoted $s_1, \ldots, s_K$ (not necessarily distinct), where K is fixed in this section. Each of the migration events between two populations results in the migrating haplotype being present in both populations. We denote the clusters that haplotype s_k is shared between as the set $\mathcal{C}(s_k)$, where $|\mathcal{C}(s_k)|$ is directly reflected in the vector $\boldsymbol{s}$ by the number of times the value of s_k appears in $\boldsymbol{s}$, plus one.

Based on the vector $s_1, \ldots, s_K$, we describe how all the sequences are allocated to clusters given the set of migrating (and as a result shared) haplotypes $\boldsymbol{s}$. Remember that the main assumption of the clustering is that each sequence is allocated to precisely one cluster. All

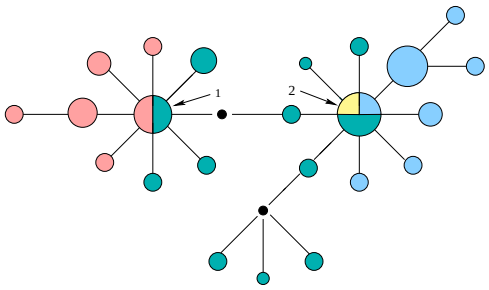


Figure 2.14: *Example of a migration haplotype tree. In this case the pink-green haplotype named 1 is shared between the pink and green population clusters with half of its copies found in each, whereas the green-blue-yellow named 2 is shared between the green, blue and yellow clusters with half of its copies found in the green cluster, and a quarter in each of the remaining two. In this case the yellow cluster only contains copies of haplotype 2. As in previous sections, black dots represent unsampled (but known) haplotypes.*

sequences corresponding to a migrating haplotype s_k belong to one of the $|\mathcal{C}(s_k)|$ clusters. Each migrating haplotype has a number of branches starting from it, which end either at a leaf node or at another migrating haplotype. For each of those branches, all sequences contained in it are clustered together in one of the $|\mathcal{C}(s_k)|$ clusters. Note that this implies that sequences (i.e., datapoints) corresponding to a haplotype which did not migrate are forced to belong to the same cluster, whereas sequences from a migrating haplotype may belong to different clusters.

It is perhaps easier to think of clusterings seeded by the vector $\boldsymbol{s}$ of migrating haplotypes in terms of migrations of the corresponding individuals. Taking the example in Figure 2.14 and assuming the green cluster is ancestral, the migrating haplotypes are $\boldsymbol{s} = 1, 2, 2$ and they are shared between two and three clusters respectively, i.e., $|\mathcal{C}(s_1)| = 2$, $|\mathcal{C}(s_2)| = 3$. This corresponds to the three migration events of an individual migrating from the green cluster to a new population (pink), and individuals with haplotype 2 migrating to two new populations (yellow and light blue). It is thus clear that the number of times a specific haplotype occurs in $\boldsymbol{s}$ is equal to the number of clusters it is shared between minus one.

Introducing K migrating haplotypes leads to the existence of $K + 1$ clusters. This is easy to see by considering the migration example described above. Each migrating haplotype represents a migration which introduces a new population cluster, thus K shared haplotypes result in $K + 1$ population clusters.

All such phylogeographic clusterings can be achieved by the following Algorithm 2.3.1 which describes a step-by-step method of constructing clusterings which are consistent with K migrating haplotypes based on a fixed haplotype tree of N_h haplotypes.

Algorithm 2.3.1.

1. Pick K of the N_h haplotypes with replacement, and denote them by $s_1, \ldots, s_K$.

2. Pick one of the K migrating haplotypes s_k. The number of clusters that s_k is shared between is equal to the number of times it appears in the vector $\boldsymbol{s}$, plus 1. If the selected haplotype is shared between $|\mathcal{C}(s_k)|$ clusters, introduce clusters $1, \ldots, |\mathcal{C}(s_k)|$ associated with that haplotype. Then iterate the following steps.

3a. Select one of the K haplotypes s_k which has at least one cluster associated with it. If the clusters associated with it are fewer in number than the clusters it is shared by, introduce new clusters associated with this haplotype to complete the set.

3b. Allocate each of the datapoints of the chosen haplotype s_k to one of the associated clusters $\mathcal{C}(s_k)$.

3c. Allocate each of its adjacent nodes along with their branches (until either a leaf or another migrating haplotype is reached) to one of the associated clusters. If a migrating haplotype is reached, associate it with that cluster. Go back to Step 3 until all haplotypes have been fully assigned to clusters.

The Algorithm 2.3.1 above is formed by following the properties of a phylogeographic clustering described in the current subsection, and as a result, any consistent clustering may be obtained.

Example

Using Algorithm 2.3.1 we demonstrate how the clustering of Figure 2.14 may be obtained from the haplotype tree.

- Start with Step 1. The three migrating haplotypes are picked to be $1, 2, 2$.
- Continue with Step 2. Pick haplotype 1, which is shared between two clusters, and assume that the two clusters are 1 and 2 (in this case pink and green).
- Move on to Step 3. Haplotype 1 is the only one which has any clusters associated with it, so pick haplotype 1.
- In Step 4, allocate each of the datapoints of 1 to the pink or the green cluster one by one. In this case half of them are allocated to the pink and half to the green cluster, as indicated by the proportions of pink and green on the haplotype node.
- In Step 5, allocate each of its adjacent branches to a cluster. All apart from one branch reach a leaf before reaching the other migrating haplotype. Those leaf branches are

allocated to the pink or green clusters (in the Figure the tree has been re-arranged so that all the pink ones lie on the left and all the green on the right; this need not be the case).

We allocate the branch connecting haplotype 1 and haplotype 2 to the green cluster, and thus assign one of the three clusters in which haplotype 2 is found to be the green one.

- Return to Step 3 and select haplotype 2 which now has the green cluster assigned to it. We assign yellow and light blue for the remaining two.
- Continue with Step 4 and assign each of the sequences of haplotype 2 into one of the three available clusters. In this case half of the sequences are allocated to the green cluster, a quarter to the light blue and a quarter to the yellow.
- Continue with Step 5 and assign each of the adjacent branches which have not yet been allocated to a cluster into green, light blue or yellow. Note here that none of the adjacent branches is allocated to the yellow cluster.

Note that the same clustering may be obtained for a number of different choices for the steps of Algorithm 2.3.1 (e.g. if we select haplotype 2 in step 2), up to re-arrangement of colours.

Once the complete clustering is determined, it is possible to separate all the datapoints into hard clusters. However, it is not possible to directly extract the historical information of the geographical movements. For example, in Figure 2.12, we cannot distinguish whether the yellow cluster was formed before or after the light blue, and whether it was e.g. a migration from the pink or green cluster. It is only possible to make a (subjective) interpretation of the output (for example using the fact that smaller populations are more likely to be younger; see Emerson and Hewitt 2005), also using external sources of information (for example about past glaciation of the area; see Hewitt 2000). Devising a method which would directly infer historical events requires modelling complex phylogeographic phenomena and would greatly increase the complexity of the algorithms.

The construction of clusters described in this subsection is analogous to an island model (Latter, 1973) for population subdivision, since any two populations are likely to share a haplotype, irrespective of their geographical distance. In addition, we remark that the phylogeographic clustering does not explicitly account for past fragmentation events. Notice that if a population undergoes fragmentation, a number of haplotypes which originally belonged to the same population will subsequently belong to the two fragment populations. As a result, all their descendants will belong to only one of the two. The resulting haplotype tree may look like Figure 2.15. The haplotype sharing construction described here does not allow for

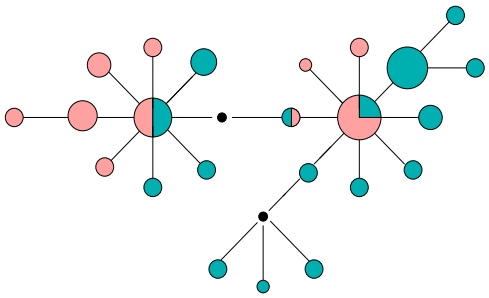


Figure 2.15: *A subdivided population which is a result of past fragmentation. Initially one population was present, to which the three pink/green haplotypes belonged. The population was subsequently fragemented, so that the three haplotypes are found in both fragments. All their descendants after the split belong exclusively to one of the two populations.*

such a clustering, but would instead only identify the three migrating haplotypes as being shared between four clusters. The construction could be extended to allow explicitly for such clusterings.

2.3.2 The clustering model

We now develop the phylogeographic clustering model. Because geographical data may be interpreted as two-dimensional distributed data (in this case assumed normal), the resulting model is very similar to the models described in previous sections. In fact, by virtue of the hierarchical approach, it is sufficient to replace only the parameters relating to the construction of clusterings.

We use Algorithm 2.3.1 to motivate a prior distribution for clustering constructions. Here we are assuming that a priori, any sequence is equally likely to correspond to a migrating haplotype. Referring back to the simplified migration setting described on page 60, this is equivalent to any individual being equally likely to migrate. This means that the probability of a haplotype being shared is proportional to the number of times it appears in the sample, yielding

$$p(\boldsymbol{s}') \quad = \quad \prod_{k=1}^{K} \frac{\min(|s_k|, 1)}{n},$$

where $|s_k|$ is the sample size of haplotype s_k.

As in previous sections, we use the notation c_{ij} to represent the cluster of the jth datapoint corresponding to haplotype i. In this case the allocation parameter c_{ij} is forced to be the same for all j for haplotypes which are not shared, but is allowed to take different values for shared ones. Assuming that the clusters chosen for each of the datapoints and branches of

haplotype s_k in Steps 3b and 3c of Algorithm 2.3.1 are selected randomly from the $|\mathcal{C}(s_k)|$ clusters, the priors for the clustering $\boldsymbol{c}$ can be written as:

$$\begin{aligned} p(\boldsymbol{c}) = p(\boldsymbol{c}, \boldsymbol{s}) &= p(\boldsymbol{s})p(\boldsymbol{c}|\boldsymbol{s}) \\ &= \prod_{k=1}^{K} \frac{\min(|s_k|, 1)}{N} |\mathcal{C}(s_k)|^{|s_k|+deg(s_k)}, \end{aligned} \tag{2.11}$$

where $deg(s_i)$ is the degree of haplotype s_k, i.e., the number of adjacent haplotypes. Note that here we correct $|s_k|$ by $\min(|s_k|, 1)$ to account for the fact that some haplotypes are extinct or unsampled, but may still have a non-zero probability of having migrated.

As before, we transform the data so that the sample mean is 0, but we do not transform the variance of each direction separately. In geographical terms, the North-South direction is spatially equivalent to the East-West, and hence should be treated identically.

Modifying the models (2.1), (2.6), the phylogeographic clustering model in full amounts to

$$\begin{aligned} \mathcal{Y} \,|\, \boldsymbol{e}, \boldsymbol{\mu}, \boldsymbol{\Sigma} &\sim \mathcal{N}_2(\boldsymbol{\mu}_i, \Sigma_i), \\ s_k &\sim \mathcal{U}\{1, \ldots, n\} \text{ with replacement}, \\ \boldsymbol{c}, \boldsymbol{s} &\sim \mathcal{U}\left\{ \prod_{k=1}^{K} \frac{|s_k|}{N} |s_k|^{|\mathcal{C}(s_k)|} deg(s_k)^{|\mathcal{C}(s_k)|} \right\}, \\ \boldsymbol{\mu}_k \,|\, \Sigma_k &\sim \mathcal{N}_2\left(\mathbf{0}, V\right), \\ \Sigma_k &\sim \mathcal{IW}(\gamma, \Psi), \\ \gamma &\sim \mathcal{U}\{4, \ldots, g\}. \end{aligned} \tag{2.12}$$

The hierarchical structure of the parameters is summarized by a DAG in Figure 2.16. The distributions in Model (2.12) give that

$$\Sigma_k \,|\, \mathcal{Y}, \boldsymbol{e}, \gamma \sim \mathcal{IW}\left(n_k + \gamma, \Psi + \sum_{j,l} \mathbb{I}_{\{c_{jl}=k\}} \boldsymbol{y}_{jl} \boldsymbol{y}_{jl}^T - n_k \overline{\boldsymbol{y}}_i \overline{\boldsymbol{y}}_k^T \right). \tag{2.13}$$

$$\boldsymbol{\mu}_k \,|\, \mathcal{Y}, \boldsymbol{e}, \Sigma_k \sim \mathcal{N}_2\left(\frac{\Sigma_k^{-1} n \overline{\boldsymbol{y}}}{V^{-1} + n\Sigma_k^{-1}}, \frac{1}{V^{-1} + n\Sigma_k^{-1}} \right) \tag{2.14}$$

$$\Sigma_k \,|\, \mathcal{Y}, \boldsymbol{e}, \gamma, \boldsymbol{\mu} \sim \mathcal{IW}\left(N + \gamma, \Psi + \sum_{j,l} \mathbb{I}_{\{c_{jl}=k\}} (\boldsymbol{y}_{jl} - \boldsymbol{\mu}_k)^T (\boldsymbol{y}_{jl} - \boldsymbol{\mu}_k) \right) \tag{2.15}$$

The objective of the analysis is to draw inferences about the target distribution

$$\pi(\boldsymbol{s}, \boldsymbol{c}, \gamma, \Sigma, \boldsymbol{\mu} \,|\, \mathcal{Y}) \propto f(\mathcal{Y} \,|\, \boldsymbol{s}, \boldsymbol{c}, \boldsymbol{\Sigma}, \boldsymbol{\mu}) p(\boldsymbol{\mu}) p(\gamma) p(\boldsymbol{\Sigma} \,|\, \gamma) p(\boldsymbol{s}, \boldsymbol{c}).$$

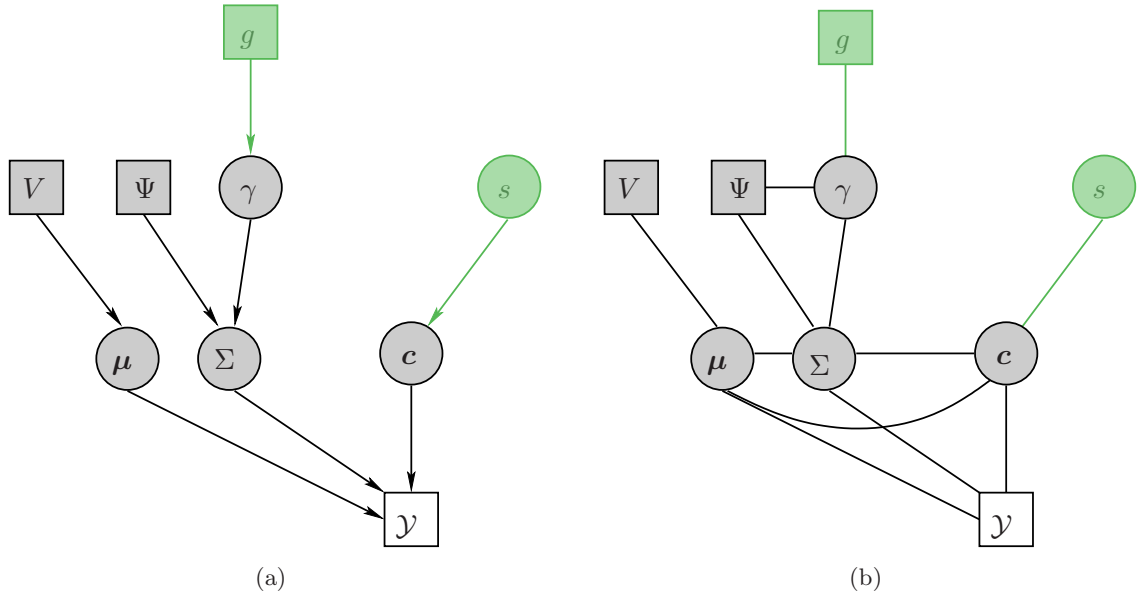


Figure 2.16: *As before, the DAG of the hierarchical parameter structure of the phylogeographic model. Here the black parameters correspond to the basic parameters of one-dimensional phenotypic clustering, and the parameters which were added to the model in this section are shown in green.*

The correspondence between our approach and the approach taken by Templeton (1998) is easy to see. Templeton uses two ways of quantifying geographical clusters: D_c which is the average distance of haplotypes within a clade from its centre, and D_n, which is the average distance of a haplotype from the geographical centre of all higher-level clades containing the clade of interest. In our approach, D_c represents Σ_k for cluster k, and we are looking for shared (migrating) nodes which will best explain the significance of the geographical location of haplotypes belonging the two adjacent clusters. This is the equivalent of inferring migrating nodes such that D_n is most significant in terms of D_c.

Following the phenotypic clustering analysis from the previous section, most parameters can be updated in the same way. However, we need to construct an MCMC update in order to move around the space of possible clusterings.

2.3.3 MCMC clustering moves

The nature of the phylogeographic clustering setting we are assuming implies that the size of the allocation parameter space is vast: there are N_h possibilities for each of the K migrating haplotypes, for which each of their datapoints as well as their adjacent haplotypes can be allocated to one of the available clusters. However, most of these combinations are highly unlikely given the data. Returning to Figure 2.14, if the pink and green cluster are geographically significantly different, it is improbable that one of the green adjacent nodes actually belongs to the pink cluster. In this subsection we develop a proposal kernel exploring the space of possible clusterings efficiently.

In Algorithm 2.3.1 we presented a method by which phylogeographic clusterings are chosen at random. In an MCMC setting, it can be modified so that the choices are made efficiently and allow mixing of the chains. To this end, we discuss some technical properties of the clustering algorithm.

First notice that is not easily possible to construct a local version of Algorithm 2.3.1; unless the algorithm is completed, the clustering cannot be updated, because the resulting clustering may be physically non sensical and contradict the migrating haplotype structure. Hence, for each MCMC iteration, all clusters are initially empty, datapoints are gradually added using a variant of Algorithm 2.3.1 until complete, and only then can the proposed move be accepted or rejected.

One possibility for improving the efficiency of Algorithm 2.3.1 is to iteratively calculate the sample mean and sample covariances of each cluster as datapoints are added to the clusters at each iteration. The calculated means and covariances may subsequently be used to assess which of the available clusters shows a better fit with the next datapoint (or haplotype branch) to be allocated.

It can be checked that, in the case of traditional unconstrained clustering, the allocation parameter c_{ij} corresponding to observation $\boldsymbol{y}_{ij}$ follows the distribution:

$$\mathbb{P}(c_{ij} = l \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{w}) \propto w_j \Sigma_j^{-1} \exp\left(-\frac{1}{2}(\boldsymbol{y}_{ij} - \boldsymbol{\mu}_l)^T \Sigma^{-1} (\boldsymbol{y}_{ij} - \boldsymbol{\mu}_l)\right),$$

(see Richardson and Green, 1997). The unconstrained clustering requires that each datapoint may be allocated to any of the clusters individually, which is not the case with the clustering structure described here.

Here clusters are constrained by the phylogeographic clustering structure on the haplotype tree, which dictates that allocating an adjacent node to one of the clusters implies allocating a whole branch of the tree to that cluster. A similar approach to unconstrained clustering can be taken in this case. For the assignment of each of the adjacent nodes of s_k into one

of the clusters $m \in \mathcal{C}(s_k)$, we recalculate the sample means and covariance matrices of the clusters, and allocate the next branch to cluster m according to

$$\mathbb{P}(c_{ij} = m \,|\, \text{nodes already assigned}) \propto \prod \overline{\Sigma}_m^{-1} \exp\left(-\frac{1}{2}(\boldsymbol{y}_{ij} - \overline{\mu}_m)^T \overline{\Sigma}_m^{-1}(\boldsymbol{y}_{ij} - \overline{\mu}_m)\right),$$

where the product is taken over the whole branch, and we take $\overline{\mu}_m$ to be the sample mean up to the latest addition of a datapoint, and similarly $\overline{\Sigma}_l$ to be the sample covariance. In this way, we allow the proposal to extract information about the clusters using the allocation values which have been proposed so far within the same MCMC iteration.

Investigating convergence of preliminary test runs, we choose the following proposal distribution Algorithm 2.3.2 for constructing a clustering at iteration t, given the clustering and cluster parameters of the previous iteration $t-1$. The algorithm is quite technical, but is essential for allowing satisfactory mixing of phylogeographic MCMC samplers. Intuitively, it is based upon attempting to preserve the clusters of as many datapoints as possible. This is achieved by preserving (where possible) the clusters of migrating haplotypes from the previous iteration.

Remember here that Algorithm 2.3.1 provides a kernel which proposes clusterings by cumulatively allocating datapoints to clusters until all datapoints have been clustered. The algorithm proceeds by iterating Steps 1-5, and is performed once during each MCMC iteration, at the end of which the proposed clustering is accepted or rejected. The Algorithm 2.3.2 described here is a variant of 2.3.1, using specific proposal distributions for each step which take into account the clustering of the previous iteration.

Algorithm 2.3.2.

During burn-in, for each iteration initially we set all clusters to be empty, with sample mean and covariances $\overline{\boldsymbol{\mu}}, \overline{\Sigma}$ equal to their prior estimates. After burn-in, initially set all clusters involved with migrating haplotypes which have not been changed since the previous iteration to have mean, variance and sample size as in the previous iteration $(\overline{\boldsymbol{\mu}}_i, \overline{\Sigma}_i, \overline{n}_i) = (\boldsymbol{\mu}_i^{(t-1)}, \Sigma_i^{(t-1)}, n_i^{(t-1)})$ respectively.

Then carry out the following steps:

1. Select one of the migrating haplotypes of $\boldsymbol{s}^{(t-1)}$ from the previous iteration and change it to s_k', proposing the new haplotype randomly.

2. For each of the migrating haplotypes s_k' shared by $|\mathcal{C}(s_k)'|$ clusters, if it was shared by $|\mathcal{C}(s_k)|^{(t-1)} \geq |\mathcal{C}(s_k)'|$ clusters at the previous iteration too, assign this migrating haplotype to be shared between the first $|\mathcal{C}(s_k)|^{(t-1)} - 1$ clusters of the set $\mathcal{C}(s_k)^{(t-1)}$, leaving the last cluster of $\mathcal{C}(s_k)'$ null.

3. Select at random one of the migrating haplotypes s'_k which has not been allocated to clusters; if none such exist, the algorithm has completed. If it was previously a migrating haplotype with at least $|\mathcal{C}(s_k)'|$ available clusters, then the last cluster of $\mathcal{C}(s_k)'$ is set to same one as the previous iteration. Otherwise the next available cluster from the list of all clusters is chosen.

4. Select at random one of the datapoints j of the migrating haplotype s_k which has not been assigned to a cluster, and assign it to one of the available clusters $m \in \mathcal{C}(s_k)$ with probability

$$\propto p(c_{s_kj} = m \,|\, \mathcal{Y}, \overline{\boldsymbol{\Sigma}}, \overline{\boldsymbol{\mu}}, \boldsymbol{s}) \propto \left|\overline{\Sigma}_m\right|^{-1/2} \exp\left(-\frac{1}{2}\left(\boldsymbol{y}_{s_kj} - \overline{\boldsymbol{\mu}}_m\right)^T \overline{\Sigma}_m^{-1}\left(\boldsymbol{y}_{s_kj} - \overline{\boldsymbol{\mu}}_m\right)\right).$$

   Update the sample means and covariances $\overline{\boldsymbol{\mu}}_{c_{s_kj}}, \overline{\Sigma}_{c_{s_kj}}$. If all datapoints of s_k have been assigned to a cluster, move on to the next step, else repeat this step.

5. Select one of the adjacent nodes l of s_k which has not been assigned to a cluster yet. Each adjacent node defines a branch, which starts at the adjacent node and ends either at a leaf node, or at another migrating haplotype. Assign all datapoints j of all the haplotypes i along the branch to one of the clusters $m \in \mathcal{C}f(s_k)$, with probability

$$\propto p(\cup_{i,j} c_{ij} = m \,|\, \mathcal{Y}, \overline{\boldsymbol{\Sigma}}, \overline{\boldsymbol{\mu}}, \boldsymbol{s}) \propto \prod_{i,j} \left|\overline{\Sigma}_m\right|^{-1/2} \exp\left(-\frac{1}{2}\left(\boldsymbol{y}_{ij} - \overline{\boldsymbol{\mu}}_m\right)^T \overline{\Sigma}_m^{-1}\left(\boldsymbol{y}_{ij} - \overline{\boldsymbol{\mu}}_m\right)\right),$$

   where the product is taken over all datapoints of all haplotypes along the branch. If the branch ends at a migrating haplotype, then assign one of its associated clusters to be m. If all adjacent branches have been allocated to clusters, go back to Step 3. Else repeat this step.

Using Algorithm 2.3.2, we adapt the MCMC algorithm described in previous sections for the phylogeographic data.

The chain is initialized by generating $\boldsymbol{s}^{(0)}$, $\boldsymbol{c}^{(0)}$, $\gamma^{(0)}$,$\boldsymbol{\mu}^{(0)}$, $\boldsymbol{\Sigma}^{(0)}$ from the prior distributions. Subsequently iterate the following steps.

C1a First we update $\boldsymbol{s}$. We randomly pick one of the K nodes s_i and change it to another one uniformly from the tree.

C1b Split sequences into clusters using Algorithm 2.3.2.

C1c Propose a new value γ' from its prior $\mathcal{U}\{4,\ g\}$.

C1d Propose new covariance matrices Σ_k' from the distribution $\Sigma_k \,|\, \mathcal{Y}, \boldsymbol{s}', \boldsymbol{c}', \gamma'$ given in Equation (2.13).

C1e Propose $\boldsymbol{\mu}_k'$ from $\boldsymbol{\mu}_k \,|\, \mathcal{Y}, \Sigma_k', \boldsymbol{s}', \boldsymbol{c}'$ given in Equation (2.14).

C1f Calculate

$$\begin{aligned} A_C &= \frac{\pi(\boldsymbol{s}', \boldsymbol{c}', \gamma', \boldsymbol{\mu}', \boldsymbol{\Sigma}' \,|\, \mathcal{Y})}{\pi(\boldsymbol{s}, \boldsymbol{c}, \gamma, \boldsymbol{\mu}, \boldsymbol{\Sigma} \,|\, \mathcal{Y})} \frac{q(\boldsymbol{s}', \boldsymbol{c}' \to \boldsymbol{s}, \boldsymbol{c})}{q(\boldsymbol{s}, \boldsymbol{c} \to \boldsymbol{s}', \boldsymbol{c}')} \frac{q(\gamma' \to \gamma)}{q(\gamma \to \gamma')} \frac{q(\boldsymbol{\mu}' \to \boldsymbol{\mu})}{q(\boldsymbol{\mu} \to \boldsymbol{\mu}')} \frac{q(\boldsymbol{\Sigma}' \to \boldsymbol{\Sigma})}{q(\boldsymbol{\Sigma} \to \boldsymbol{\Sigma}')} \\ &= \frac{f(\mathcal{Y} \,|\, \boldsymbol{s}', \boldsymbol{c}', \boldsymbol{\mu}', \boldsymbol{\Sigma}')}{f(\mathcal{Y} \,|\, \boldsymbol{s}, \boldsymbol{c}', \boldsymbol{\mu}, \boldsymbol{\Sigma})} \frac{p(\boldsymbol{s}', \boldsymbol{c}')}{p(\boldsymbol{s}, \boldsymbol{c})} \frac{p(\boldsymbol{\Sigma}')}{p(\boldsymbol{\Sigma})} \frac{p(\boldsymbol{\mu}')}{p(\boldsymbol{\mu})} \frac{q(\boldsymbol{s}', \boldsymbol{c}' \to \boldsymbol{s}, \boldsymbol{c})}{q(\boldsymbol{s}, \boldsymbol{c} \to \boldsymbol{s}', \boldsymbol{c}')} \frac{\pi(\boldsymbol{\mu} \,|\, \boldsymbol{s}, \boldsymbol{c}, \boldsymbol{\Sigma})}{\pi(\boldsymbol{\mu}' \,|\, \boldsymbol{s}', \boldsymbol{c}', \boldsymbol{\Sigma}')} \frac{\pi(\boldsymbol{\Sigma} \,|\, \mathcal{Y}, \boldsymbol{s}, \boldsymbol{c})}{\pi(\boldsymbol{\Sigma}' \,|\, \mathcal{Y}, \boldsymbol{s}', \boldsymbol{c}')}. \end{aligned}$$

Accept the move with probability $\min(1, A_C)$ and set

$$(\boldsymbol{s}^{(t+1)}, \boldsymbol{c}^{(t+1)}, \gamma^{(t+1)}, \boldsymbol{\mu}'', \boldsymbol{\Sigma}'') = (\boldsymbol{s}', \boldsymbol{c}', \gamma', \boldsymbol{\mu}', \boldsymbol{\Sigma}'),$$

otherwise set

$$(\boldsymbol{s}^{(t+1)}, \boldsymbol{c}^{(t+1)}, \gamma^{(t+1)}, \boldsymbol{\mu}'', \boldsymbol{\Sigma}'') = (\boldsymbol{s}^{(t)}, \boldsymbol{c}^{(t)}, \gamma^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\Sigma}^{(t)})$$

C2 Generate $\boldsymbol{\Sigma}^{(t+1)}$ directly from the posterior conditional $\Sigma \,|\, \mathcal{Y}, \boldsymbol{s}^{(t+1)}, \boldsymbol{c}^{(t+1)}, \gamma^{(t+1)}, \boldsymbol{\mu}''$ given in Equation (2.15).

C3 Generate $\boldsymbol{\mu}^{(t+1)}$ directly from the posterior conditional $\boldsymbol{\mu} \,|\, \mathcal{Y}, \boldsymbol{s}^{(t+1)}, \boldsymbol{c}^{(t+1)}, \boldsymbol{\Sigma}^{(t+1)}$ of Equation (2.14). Go back to step C1.

Lemma 2.3.3. *The algorithm 2.3.2 described above preserves irreducibility and aperiodicity of the chain.*

Proof. Clearly, it is always possible to change one of the migrating haplotypes to be haplotype 1, without loss of generality. Similarly, we may repeat the same, until haplotype 1 is the only migrating haplotype. Hence, we can get to this clustering from any other clustering, and the chain is irreducible.

Aperiodicity is guaranteed because there is always a positive probability of staying in the same state. □

Lemma 2.3.4. *Randomizing the order in which datapoints and branches are clustered in Steps 3 and 4 of algorithm 2.3.2 described above preserves time-reversibility of the chain.*

Proof. Notice first that the move $\boldsymbol{c}^{(t-1)} \to \boldsymbol{c}^{(t)}$ may be achieved in a number of different combinations of steps in the algorithm, depending on the order in which we choose to propose the migrating haplotypes and their datapoints; remember that the move can only be accepted

or rejected once they have all been proposed. Randomizing the order in which the migrating haplotypes are proposed is equivalent to having a pool of proposals q_i and randomly selecting one (see Geyer, 1992, 1991).

In the standard MCMC setting, the ratio of the proposal distributions would then be equal to:

$$\frac{q(\boldsymbol{c}^{(t)} \to \boldsymbol{c}^{(t-1)})}{q(\boldsymbol{c}^{(t-1)} \to \boldsymbol{c}^{(t)})} = \frac{\sum_i q_i(\boldsymbol{c}^{(t)} \to \boldsymbol{c}^{(t-1)})}{\sum_i q_i(\boldsymbol{c}^{(t-1)} \to \boldsymbol{c}^{(t)})},$$

where the sum is taken over all the possible step combinations which may lead to the same clustering.

However, in this setting we use the order of the update as an extra parameter, say $\boldsymbol{z}_c$, and assume that all step combinations have equal probability a priori. At each iteration we propose a step combination and then update the clustering using the proposal distribution

$$q(\boldsymbol{c}^{(t-1)} \to \boldsymbol{c}^{(t)}) = \sum_i \mathbb{I}_{i=\boldsymbol{z}_c} q_i(\boldsymbol{c}^{(t-1)} \to \boldsymbol{c}^{(t)}).$$

Clearly q is a distribution, since all but one term will be zero, and q_i is a distribution. This means that the overall proposal ratio becomes simply

$$\frac{q(\boldsymbol{z}_c^{(t)} \to \boldsymbol{z}_c^{(t-1)})}{q(\boldsymbol{z}_c^{(t-1)} \to \boldsymbol{z}_c^{(t)})} \frac{q(\boldsymbol{c}^{(t)} \to \boldsymbol{c}^{(t-1)})}{q(\boldsymbol{c}^{(t-1)} \to \boldsymbol{c}^{(t)})} = \frac{q(\boldsymbol{z}_c^{(t)} \to \boldsymbol{z}_c^{(t-1)})}{q(\boldsymbol{z}_c^{(t-1)} \to \boldsymbol{z}_c^{(t)})} \frac{q_{\boldsymbol{z}^{(t-1)}}(\boldsymbol{c}^{(t)} \to \boldsymbol{c}^{(t-1)})}{q_{\boldsymbol{z}^{(t)}}(\boldsymbol{c}^{(t-1)} \to \boldsymbol{c}^{(t)})},$$

and this can be treated as a standard time-reversible MCMC sampler. □

Returning to the phenotypic analysis, we can see that the phylogeographic clustering can also be thought of as a phenotypic clustering, when the significant mutation occurs at a nucleotide site which is not sampled. Much like a migration event, one of the sequences corresponding to the same haplotype would show a significant change in the phenotype. The phenotypes of the descendants of that haplotype will then depend on whether or not the sequences involve the unsampled mutation. The analysis of the previous sections can easily be extended to compare between the two possible clustering constructions in order to infer whether it is more likely that the significant mutation is included in the sample or not.

Example: simulated dataset We generated dataset S4 using the same haplotype tree as before, assuming that two haplotypes are shared between two and three populations respectively, namely haplotypes 1 and 9, shown in Figure 2.17.

$$\mu_1 = \begin{pmatrix} -3 \\ -3 \end{pmatrix} \quad , \quad \Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$
$$\mu_2 = \begin{pmatrix} 5 \\ 5 \end{pmatrix} \quad , \quad \Sigma_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$
$$\mu_3 = \begin{pmatrix} 0.1 \\ 0.2 \end{pmatrix} \quad , \quad \Sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$
$$\mu_4 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad , \quad \Sigma_4 = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix}.$$

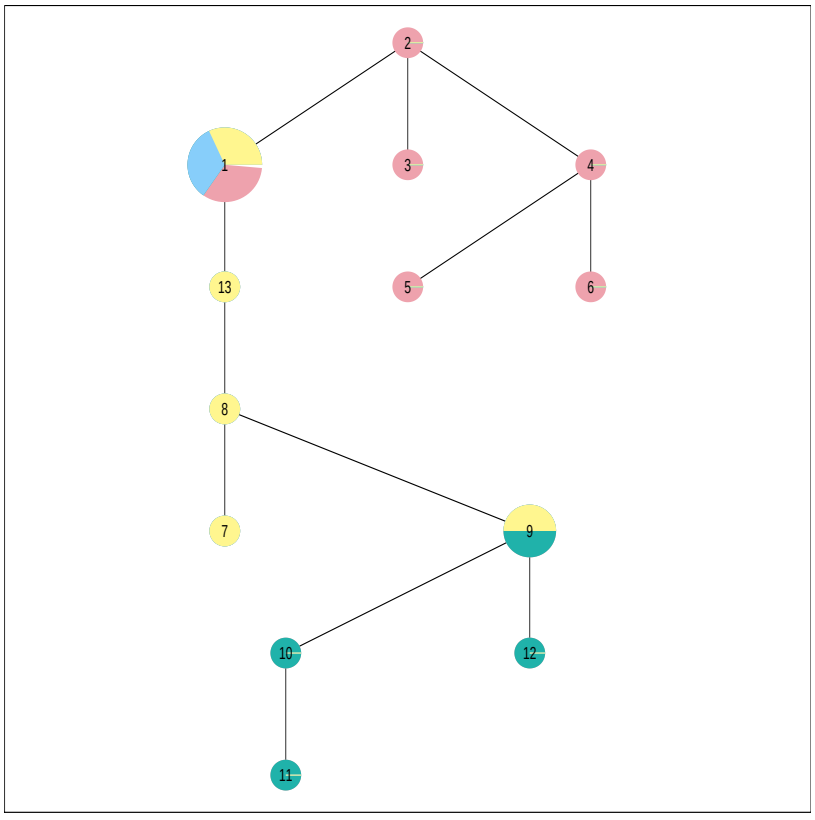


Figure 2.17:

The MCMC algorithm correctly identified the clustering structure, and showed excellent mixing as shown in Figure 2.18.

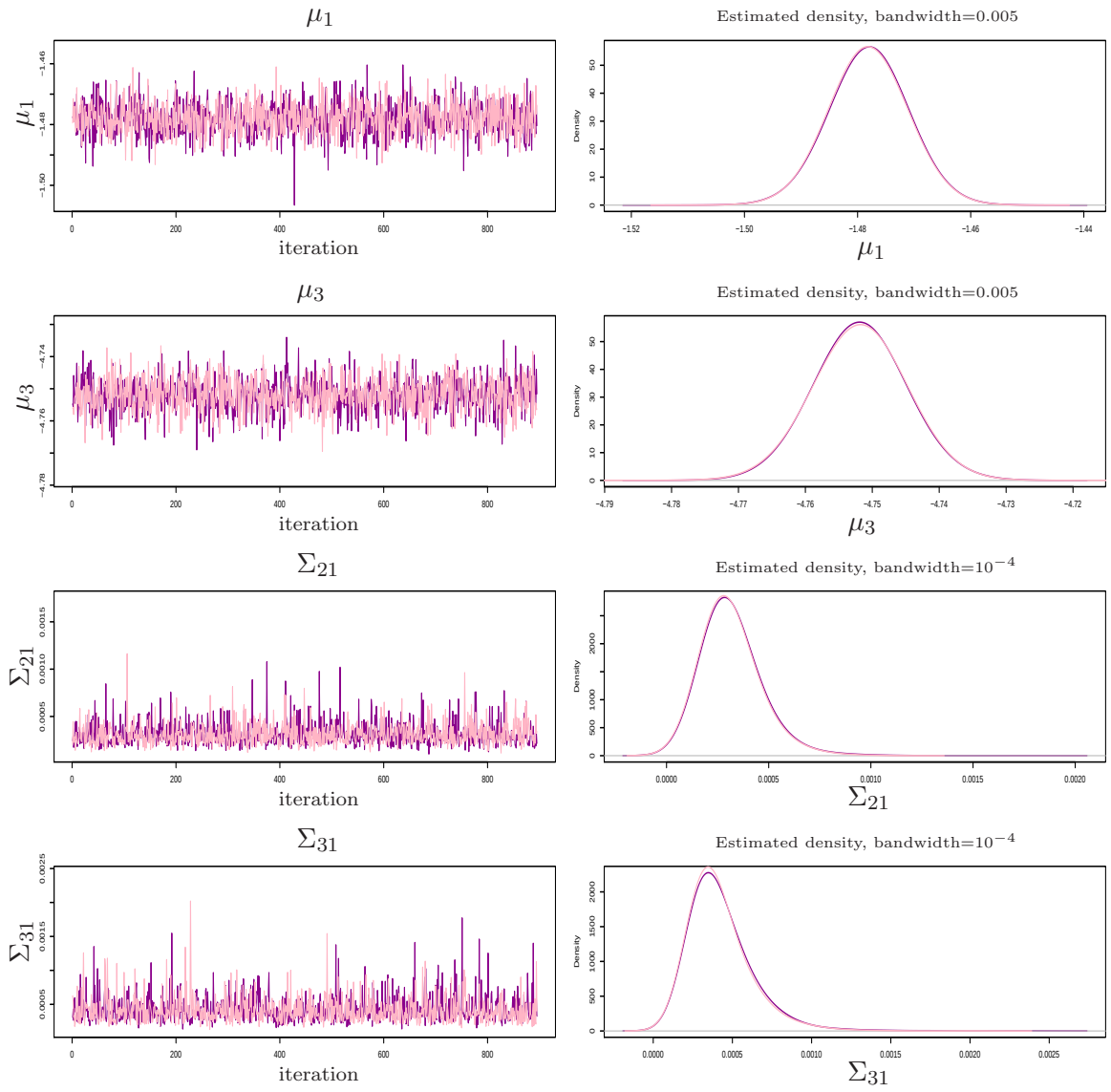


Figure 2.18: *Trace and density plots for dataset S4, showing excellent mixing. The trace plots are thinned by a factor of 8.*

2.4 Analysis for an unknown number of clusters

In practice, we almost never know the true number K of underlying significant edges or migrating haplotypes. We therefore have to draw inference about the number of significant clusters we are trying to split our data into. To this end, we use a Reversible-Jump MCMC method similar to the one described in Richardson and Green (1997), which allows moving between parameter spaces with different sizes.

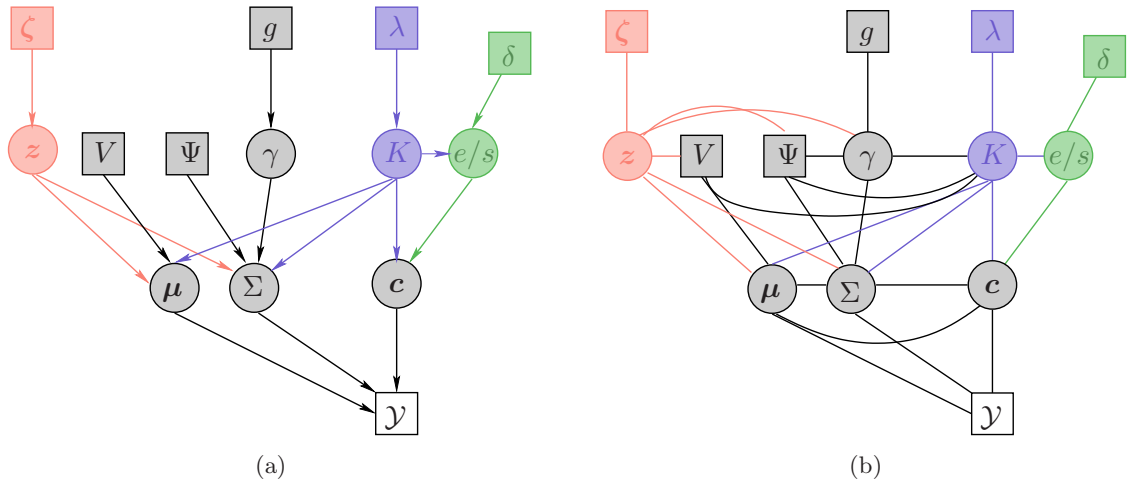


Figure 2.19: *Finally, this DAG shows all the parameters introduced in this Chapter. The colours correspond to different types of analysis: the black represents the basic parameters (descibed in the current Section 2.1), pink represents parameters for analysis of multidimensional traits (Section 2.2), the green for phylogeographic analysis (Section 2.3) and the blue for analysis with a variable number of clusters (Section 2.4). A few parameters (λ, ζ, δ) have been added to illustrate how non-uniform priors may be included. Depending on whether the clustering is phenotypic or phylogeographic, the clustering will be seeded by $\boldsymbol{e}$ or $\boldsymbol{s}$ respectively. Note here that in the one-dimensional case, $V = \sigma_\mu^2$, $\Sigma_k = \sigma_k^2$, $\Psi = a$ and $\gamma = b$.*

The Models (2.1), (2.6) and (2.12) are augmented by adding a parameter K, which is assumed to have a uniform prior in $[0, K_{\max}]$, i.e.,

$$p(K) \propto 1, \quad 0 \leq K \leq K_{\max}.$$

The hierarchical structure of the parameters for both the phenotypic and phylogeographic case is shown in Figure 2.19.

Here the hyperparameter γ becomes important. Clearly, the number of clusters is heavily dependant upon the spread of each cluster. Thus, a prior for the covariance favouring small clusters will tend to result in a large K, and vice versa. The variable γ allows the joint posterior of the number of clusters and their spread to be inferred.

2.4.1 Phenotypic analysis

In the case of phenotypic analysis with a fixed number of components (i.e., dimensions, represented by $\boldsymbol{z}$), we want to construct an MCMC sampler with target distribution

$$\pi(K, \boldsymbol{e}, \gamma, \Sigma, \boldsymbol{\mu} \,|\, \mathcal{Y}) \propto f(\mathcal{Y} \,|\, K, \boldsymbol{e}, \boldsymbol{\Sigma}, \boldsymbol{\mu}) p(K) p(\boldsymbol{\mu}) p(\gamma) p(\boldsymbol{\Sigma} \,|\, \gamma) p(\boldsymbol{e}).$$

In phenotypic clustering, introducing a new cluster is equivalent to adding another edge to the vector of signifiant mutations, and similarly removing a cluster is equivalent to removing a mutation, thus merging two clusters which are genetically adjacent (i.e., are separated by a single mutation).

The chain is initialized by generating $K^{(0)}$, $\boldsymbol{e}^{(0)}$, $\gamma^{(0)}$, $\boldsymbol{\mu}^{(0)}$, $\boldsymbol{\Sigma}^{(0)}$ from the prior distributions. Subsequently iterate the steps described below.

D1 Carry out Steps A1-A3 (or B1-B5 if the data is multi-dimensional, keeping $\boldsymbol{z}$ constant) for a fixed K.

D2a With probability p_{split} split one of the existing clusters into two so that $K^{(t+1)} = K^{(t)}+1$, and uniformly select an edge to be added to $\boldsymbol{e}$. Otherwise, with probability $p_{merge} = 1 - p_{split}$ combine two of the existing clusters into one so that $K^{(t+1)} = K^{(t)} - 1$ and uniformly select one of the entries of $\boldsymbol{e}$ to be removed. Calculate the sample means of the new clusters.

D2b Propose values for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ for the new clusters formed.

(a) If we decide to merge two clusters k_1 and k_2 into k', we propose $\Sigma'_{k'} \,|\, \mathcal{Y}, \boldsymbol{e}', \gamma$ (see Equation (2.7)), and $\boldsymbol{\mu}'_{k'} \,|\, \mathcal{Y}, \boldsymbol{e}, \Sigma_{k'}$ (see Equation (2.8)). The remaining covariances of clusters which are not affected by the move are left unchanged.

(b) Similarly, if we decide to split one of the existing $K^{(t)} + 1$ clusters, we propose $\Sigma'_{k_1}, \Sigma'_{k_2}, \boldsymbol{\mu}'_{k_1}, \boldsymbol{\mu}'_{k_2}$ from the distributions given in Equations (2.7), (2.8). Again, the remaining covariances of clusters which are not affected by the move are left unchanged.

D2c The acceptance probability of a merging move becomes $\alpha = \min(1, A_D)$ where

$$\begin{aligned} A_D &= \frac{f(\mathcal{Y}\,|\,\boldsymbol{e}', \gamma, \boldsymbol{\mu}', \boldsymbol{\Sigma}')}{f(\mathcal{Y}\,|\,\boldsymbol{e}, \gamma, \boldsymbol{\mu}, \boldsymbol{\Sigma})} \frac{p(\boldsymbol{e}')}{p(\boldsymbol{e})} \frac{p(\boldsymbol{\mu}'_{\boldsymbol{k}'})}{p(\boldsymbol{\mu}'_{\boldsymbol{k_1}})p(\boldsymbol{\mu}'_{\boldsymbol{k_2}})} \frac{p(\boldsymbol{\Sigma}'_{\boldsymbol{k}'})}{p(\boldsymbol{\Sigma}'_{\boldsymbol{k_1}})p(\boldsymbol{\Sigma}'_{\boldsymbol{k_2}})} \frac{q(K-1 \to K)}{q(K \to K-1)} |J| \\ &\times \frac{q(\boldsymbol{e}' \to \boldsymbol{e})}{q(\boldsymbol{e} \to \boldsymbol{e}')} \frac{q(\boldsymbol{\mu}'_{k'} \to \boldsymbol{\mu}_{k_1}, \boldsymbol{\mu}_{k_2})}{q(\boldsymbol{\mu}_{k_1}, \boldsymbol{\mu}_{k_2} \to \boldsymbol{\mu}'_{k'})} \frac{q(\boldsymbol{\Sigma}'_{k'} \to \boldsymbol{\Sigma}_{k_1}, \boldsymbol{\Sigma}_{k_2})}{q(\boldsymbol{\Sigma}_{k_1}, \boldsymbol{\Sigma}_{k_2} \to \boldsymbol{\Sigma}'_{k'})} \frac{q(K-1 \to K)}{q(K \to K-1)} |J| \\ &= \frac{f(\mathcal{Y}\,|\,\boldsymbol{e}', \gamma, \boldsymbol{\mu}', \boldsymbol{\Sigma}')}{f(\mathcal{Y}\,|\,\boldsymbol{e}, \gamma, \boldsymbol{\mu}, \boldsymbol{\Sigma})} \frac{p(\boldsymbol{e}')}{p(\boldsymbol{e})} \frac{p(\boldsymbol{\mu}'_{\boldsymbol{k}'})}{p(\boldsymbol{\mu}'_{\boldsymbol{k_1}})p(\boldsymbol{\mu}'_{\boldsymbol{k_2}})} \frac{p(\boldsymbol{\Sigma}'_{\boldsymbol{k}'})}{p(\boldsymbol{\Sigma}'_{\boldsymbol{k_1}})p(\boldsymbol{\Sigma}'_{\boldsymbol{k_2}})} \\ &\times \frac{q(\boldsymbol{e}' \to \boldsymbol{e})}{q(\boldsymbol{e} \to \boldsymbol{e}')} \frac{q(\boldsymbol{\mu}_{k_1})q(\boldsymbol{\mu}_{k_2})}{q(\boldsymbol{\mu}'_{k'})} \frac{q(\boldsymbol{\Sigma}_{k_1})q(\boldsymbol{\Sigma}_{k_2})}{q(\boldsymbol{\Sigma}'_{k'})} \frac{p_{split}}{p_{merge}} |J|\,, \end{aligned}$$

using Equations (2.7), (2.8), and

$$\begin{aligned} p(\boldsymbol{e}) &= \frac{1}{\binom{N_h-1}{K^{(t)}}} \\ q(\boldsymbol{e} \to \boldsymbol{e}') &= \frac{1}{K^{(t)}} \qquad \text{for a merging move, and} \\ q(\boldsymbol{e} \to \boldsymbol{e}') &= \frac{1}{N_h - K^{(t)}} \text{for a splitting move,} \end{aligned}$$

We check here that, for a merging move $K^{(t+1)} \to K^{(t)} - 1$, the priors and proposals for the vector $\boldsymbol{e}$ cancel:

$$\frac{q(\boldsymbol{e}' \to \boldsymbol{e})}{q(\boldsymbol{e} \to \boldsymbol{e}')} \frac{p(\boldsymbol{e}')}{p(\boldsymbol{e})} = \frac{\frac{1}{N_h - K^{(t)}}}{\frac{1}{K^{(t)}+1}} \times \frac{\binom{N_h}{K^{(t)}-1}}{\binom{N_h-1}{K^{(t)}}} = 1. \tag{2.16}$$

Finally, as before (see Section 2.2) the determinant of the Jacobian in this case is equal to one, since the move can be expressed as a combination of independent moves, yielding a Jacobian with determinant $|J| = 1$.

Similarly, the acceptance probability of a splitting move becomes $\alpha = \min(1, A_D^{-1})$, with some terms replaced appropriately.

D2d If we accept, we set

$$(\boldsymbol{K}^{(t+1)}, \boldsymbol{e}^{(t+1)}, \boldsymbol{\mu}^{(t+1)}, \boldsymbol{\Sigma}^{(t+1)}) = (\boldsymbol{K}', \boldsymbol{e}', \boldsymbol{\mu}', \boldsymbol{\Sigma}'),$$

otherwise we set

$$(\boldsymbol{K}^{(t+1)}, \boldsymbol{e}^{(t+1)}, \boldsymbol{\mu}^{(t+1)}, \boldsymbol{\Sigma}^{(t+1)}) = (\boldsymbol{K}^{(t)}, \boldsymbol{e}^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\Sigma}^{(\boldsymbol{t})}),$$

A run through steps A1-A3 (or B1-B5) and D2 produces a chain with a stationary distribution

the desired posterior. Since splits and merges always have positive probability, the chain remains irreducible and aperiodic as in the previous sections.

Notice here that the prior distribution $p(\boldsymbol{e}) = \binom{N_h-1}{K}$ is only true if all possible clusterings, including ones involving empty clusters, are permissible. If empty clusters are not allowed in the analysis, the number of clustering possibilities which do not result in empty clusters has to be computed but is generally intractable. Although for a fixed K the normalization constant remains constant throughout the analysis and hence need not be calculated, allowing K to vary implies that empty clusters have to be included. Analogously, the same is true for phylogeographic clustering analyses described in the next subsection.

In the case of multi-dimensional phenotypic data, there is clearly a strong dependence between the parameter $\boldsymbol{z}$ indicating which phenotypic traits are informative and K. Specifically, if $|\boldsymbol{z}| = 0$, then $K = 0$ shows a perfect fit with the data since it is the empty model.

2.4.2 Phylogeographic analysis

In the case of phylogeographic data, our objective is to construct an MCMC sampler with target distribution

$$\pi(K, \boldsymbol{s}, \boldsymbol{c}, \gamma, \Sigma, \boldsymbol{\mu} \,|\, \mathcal{Y}) \propto f(\mathcal{Y} \,|\, K, \boldsymbol{s}, \boldsymbol{c}, \boldsymbol{\Sigma}, \boldsymbol{\mu}) p(K) p(\boldsymbol{\mu}) p(\gamma) p(\boldsymbol{\Sigma} \,|\, \gamma) p(\boldsymbol{s}, \boldsymbol{c}). \tag{2.17}$$

Given a new clustering, the proposal and probabilities of the means and covariances are analogous to the phenotypic clustering, but the clustering move represented by Step D1a above is slightly more complicated. The algorithm is given below.

The chain is initialized by generating $K^{(0)}, \boldsymbol{s}^{(0)}$, $\boldsymbol{c}^{(0)}$, $\gamma^{(0)}$, $\boldsymbol{\mu}^{(0)}$, $\boldsymbol{\Sigma}^{(0)}$ from the prior distributions. Subsequently iterate the following steps.

E1 Carry out Steps C1-C3 for a fixed K.

E2a Propose to add or subtract a migrating haplotype s_k with probabilities p_{split} and p_{merge} as before.

(a) For a merging move, select two of the clusters $\mathcal{C}(s_k)$ between which s_k is shared, say k_1 and k_2, and merge them into one cluster k'. The probability of this move becomes

$$q(\boldsymbol{s}, \boldsymbol{c} \to \boldsymbol{s}', \boldsymbol{c}') = \frac{1}{K^{(t)} \times \binom{|\mathcal{C}(s_k)|}{2}} \tag{2.18}$$

(b) For a splitting move, add one of the N_h haplotypes to the vector $\boldsymbol{s}$. Then all of the datapoints and adjacent haplotypes of the added node have to be inserted to one of the available clusters. Since this may heavily affect the distribution of all the clusters, we

start with $\boldsymbol{s}'$ and re-allocate all the datapoints of all the haplotypes to clusters according to Algorithm 2.3.2. The probability of this move is equal to

$$q(\boldsymbol{s}, \boldsymbol{c} \to \boldsymbol{s}', \boldsymbol{c}') = \frac{1}{N_h} q(\boldsymbol{c}'|\boldsymbol{s}'), \tag{2.19}$$

where $q(\boldsymbol{c}'|\boldsymbol{s}')$ is calculated through Algorithm 2.3.2.

E2b We propose values for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ for the new clusters formed.

(a) If we decide to merge two clusters k_1 and k_2 into k', we propose $\Sigma'_{k'} \,|\, \mathcal{Y}, \boldsymbol{e}', \gamma$ (see Equation (2.7)), and $\boldsymbol{\mu}'_{k'} \,|\, \mathcal{Y}, \boldsymbol{e}, \Sigma_{k'}$ (see Equation (2.8)). The remaining covariances of clusters which are not affected by the move are left unchanged.

(b) Similarly, if we decide to split one of the existing $K^{(t)} + 1$ clusters, we propose $\Sigma'_{k_1}, \Sigma'_{k_2}, \boldsymbol{\mu}'_{k_1}, \boldsymbol{\mu}'_{k_2}$ from the distributions given in Equations (2.7), (2.8). The remaining covariances of clusters which are not affected by the move are left unchanged.

E2c The acceptance probability of a merging move becomes $\alpha = \min(1, A_E)$ where

$$\begin{aligned} A_E \;=\; & \frac{f(\mathcal{Y} \,|\, \boldsymbol{s}', \boldsymbol{c}', \boldsymbol{\mu}', \boldsymbol{\Sigma}')}{f(\mathcal{Y} \,|\, \boldsymbol{s}, \boldsymbol{c}, \boldsymbol{\mu}, \boldsymbol{\Sigma})} \frac{p(\boldsymbol{s}', \boldsymbol{c}')}{p(\boldsymbol{s}, \boldsymbol{c})} \frac{p(\boldsymbol{\mu}'_{\boldsymbol{k'}})}{p(\boldsymbol{\mu}'_{\boldsymbol{k_1}}) p(\boldsymbol{\mu}'_{\boldsymbol{k_2}})} \frac{p(\boldsymbol{\Sigma}'_{\boldsymbol{k'}})}{p(\boldsymbol{\Sigma}'_{\boldsymbol{k_1}}) p(\boldsymbol{\Sigma}'_{\boldsymbol{k_2}})} \\ & \times \frac{q(\boldsymbol{s}', \boldsymbol{c}' \to \boldsymbol{s}, \boldsymbol{c})}{q(\boldsymbol{s}, \boldsymbol{c} \to \boldsymbol{s}', \boldsymbol{c}')} \frac{q(\boldsymbol{\mu}_{k_1}) q(\boldsymbol{\mu}_{k_2})}{q(\boldsymbol{\mu}'_{k'})} \frac{q(\boldsymbol{\Sigma}_{k_1}) q(\boldsymbol{\Sigma}_{k_2})}{q(\boldsymbol{\Sigma}'_{k'})} \frac{p_{split}}{p_{merge}} |J| \,, \end{aligned}$$

using Equations (2.7), (2.8), (2.18) and (2.19). As before, $|J| = 1$.

Similarly, the acceptance probability of a splitting move becomes $\alpha = \min(1, A_E^{-1})$. We decide to accept or reject the proposed move, with some terms replaced appropriately.

E2d If we accept, we set

$$(\boldsymbol{K}^{(t+1)}, \boldsymbol{s}^{(t+1)}, \boldsymbol{c}^{(t+1)}, \boldsymbol{\mu}^{(t+1)}, \boldsymbol{\Sigma}^{(t+1)}) = (\boldsymbol{K}', \boldsymbol{s}', \boldsymbol{c}', \boldsymbol{\mu}', \boldsymbol{\Sigma}'),$$

otherwise

$$(\boldsymbol{K}^{(t+1)}, \boldsymbol{s}^{(t+1)}, \boldsymbol{c}^{(t+1)}, \boldsymbol{\mu}^{(t+1)}, \boldsymbol{\Sigma}^{(t+1)}) = (\boldsymbol{K}^{(t)}, \boldsymbol{s}^{(t)}, \boldsymbol{c}^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\Sigma}^{(\boldsymbol{t})}),$$

Cycling through steps C1-C3 and E2 produces an irreducible chain with stationary distribution (2.17).

Example: simulated dataset Using the simulated datasets S1 (one-dimensional phenotype, two significant mutations), S3 (three-dimensional phenotype, two significant mutations)

K	0	1	2	3	4	5
S1	0.05	0.04	0.80	0.07	0.02	0.01
S3	0.00	0.05	0.86	0.06	0.03	0.01
S4	0.00	0.01	0.04	0.80	0.12	0.03

Table 2.2: *The posterior probabibilities for the number of clusters in each of the datasets S1, S2 and S4. Indeed, the RJMCMC sampler identified the correct number of clusters in each case.*

and S4 (phylogeographic dataset, three migrating haplotypes), we repeated the analysis allowing the number of clusters to vary. The posterior distribution of the number of clusters for the three different analyses are shown in Table 2.2.

Chapter 3

Inference about the haplotype tree

In this Chapter we provide a Bayesian alternative to inferring the rooted haplotype tree from sequence data, which corresponds to the first step of Nested Clade Analysis. Together with the clustering algorithms described in the previous Chapter, this completes our method of phenotypic and phylogeographic cluster analysis.

In Section 3.1 we develop a fully Bayesian model for the haplotype trees, using the coalescent (see Kingman, 1982) and the Generalized Time-Reversible (GTR) mutation process (see Tavaré, 1986). Given the mutation parameters and the root of the tree, we propose a feasible approximation of the probability of a haplotype tree in order to allow computationally tractable inferences.

To prepare inference on the tree, we gradually increase the number of unknown parameters and illustrate the methods through simulated data at each stage. In Sections 3.2 - 3.4 we keep the haplotype tree and its root fixed, and describe how to draw inferences about the nucleotide frequencies, mutation coefficients and site-specific mutation rates. In Section 3.5 we treat the root of a fixed haplotype tree as unknown, and describe an algorithm to estimate it.

In Section 3.6 we invert the previous conditioning and infer the haplotype tree given the root and mutation parameters. First we assume that homoplasy (meaning correspondence of nucleotides that are not derived from the same ancestor) is not present. Recall that the haplotype tree comprises a tree topology and sequences corresponding to each node. We describe a deterministic algorithm to construct the tree topology. This procedure usually involves inserting missing intermediate sequences. Even though in the absence of homoplasy the tree topology is unique, the sequences of intermediate nodes can only be determined up to permutation of mutations as described in Section 3.6. Inference about the unknown intermediate sequences is described in Section 3.7. This approach is then extended in Sections 3.8 - 3.9 for datasets where the tree topology derived from the data may not be unique,

i.e., when homoplasy may be present. We devise a method of representing and obtaining tree topologies which affords efficient inference, and illustrate it through an example.

In Sections 3.10 - 3.11 we describe how phenotypic and/or phylogeographic clustering can be combined with inference about the haplotype tree to obtain the joint posterior distribution of the tree and the phenotypic/phylogeographic data. Finally, in the last Section 3.12 we propose sequence data can be combined with both phenotypic and phylogeographic data *simultaneously*, by describing an integrated two-fold clustering construction on the haplotype tree.

3.1 The haplotype tree model

Because haplotype trees are based on mutations, we supplement the coalescent with the GTR mutation model to develop the hierarchical structure of the parameters. The coalescent and the GTR mutation model are well established tools in population genetics. In this section we employ them to develop a hierarchical model for inferences on haplotype trees under the Bayesian paradigm.

The coalescent has already been introduced in Section 1.2.1 of the Introduction. It describes the evolutionary history of a set of genes by tracing them back to their most recent common ancestor. Considering evolution backwards in time, in a sample of N genes, coalescence events occur at rate $\binom{N}{2}$ (assuming constant population size). As discussed previously (see Algorithm 1.2.1), given a mutation model with Poisson rate $\theta/2$ and transition matrix $P^{(t)}$, the coalescent can also be used to calculate the relative probability of mutation or split events on a tree (see Ethier and Griffiths, 1987). If k ancestral sequences are present in the population, the time t of the next event is exponentially distributed with parameter $\frac{k(k-1+\theta)}{2}$ and the probability of the next event being a split is

$$\frac{k-1}{k-1+\theta}; \tag{3.1}$$

otherwise a mutation occurs with probabilities according to the the transition matrix $P^{(t)}$.

We now specify the transition matrix $P^{(t)}$ by employing the GTR model (see Subsection 1.2.2). We assume that the generator matrix Q determines the transition matrix $P^{(t)} = \exp(Qt)$, where Q is given by

$$Q = \phi_j \begin{pmatrix} \cdot & v_1\pi_G & v_2\pi_C & v_3\pi_T \\ v_1\pi_A & \cdot & v_4\pi_C & v_5\pi_T \\ v_2\pi_A & v_4\pi_G & \cdot & v_6\pi_T \\ v_3\pi_A & v_5\pi_G & v_6\pi_C & \cdot \end{pmatrix}. \tag{3.2}$$

The GTR mutation model is the most general model which is statistically convenient to implement. Based on the mutation model (3.2), the time of mutations for the nucleotide X_l^i of sequence i at site l is exponentially distributed with parameter

$$\phi_l q_{X_l^i} := \phi_l \sum_j q_{X_l^i j}. \tag{3.3}$$

This allows us to *relate* $\theta/2$ to the mutation process generated by Q, because the overall mutation rate $\theta/2$ represents the probability that any nucleotide of a sequence mutates. Under GTR, the rate at which a nucleotide of sequence i mutates is given by

$$\sum_{l=1}^{L} \phi_l \sum_j q_{X_l^i j}.$$

This implies that the rate of a randomly selected sequence mutating equals

$$\frac{\theta}{2} = \frac{1}{N_t} \sum_{i=1}^{N_t} \sum_{l=1}^{L} \phi_l q_{X_l^i},$$

where the sum over i is taken over all N_t sequences present at time t. From now on (for notational simplicity) we replace $\frac{1}{N_t} \sum_{i=1}^{N_t} \sum_{l=1}^{L} \phi_l q_{X_l^i}$ by $\frac{\theta}{2}$.

In analogy to (3.1), we employ (3.3) to calculate the probability that a *specific* sequence splits or mutates *next*.

$$\begin{aligned} \mathbb{P}(\text{sequence } n \text{ splits}) &= \frac{1}{N_t} \frac{N_t - 1}{N_t - 1 + \theta} \\ \mathbb{P}(l\text{th site of sequence } n \text{ mutates}) &= \frac{1}{N_t} \frac{\phi_l q_{X_l^n}}{((N_t - 1)/2 + \theta/2)}. \end{aligned} \tag{3.4}$$

Based on (3.4) we calculate the probabilitiy of a haplotype tree in the next Subsection.

3.1.1 The probability of a haplotype tree

Haplotype trees are (naturally) composed of haplotypes, which implies that equations such as (3.4) must be calculated in terms of haplotypes. These are then extended from probabilities of single events to probabilities of a series of events. A haplotype tree determines the events which occurred in history, but does not specify the precise order in which these events occurred. Therefore, we derive the probability of a haplotype tree by summing over all the possible series of events which occurred.

Within the context of haplotypes, sequences which correspond to the same haplotype are not identifiable. Therefore, the probability that haplotype h mutates or splits is equal to the

sum over the $|h|$ probabilities (3.4) of all the sequences n corresponding to that haplotype, using the additive property of independent exponential rates. In other words,

$$\begin{aligned} \mathbb{P}(\text{haplotype } h \text{ splits}) &= |h| \times \mathbb{P}(\text{sequence } n \text{ splits}), \\ \mathbb{P}(l\text{th site of haplotype } h \text{ mutates}) &= |h| \times \mathbb{P}(l\text{th site of sequence } n \text{ mutates}). \end{aligned} \tag{3.5}$$

We can now calculate the probability of a *series* of mutation and split events, *conditional on the total number of events*. The event probabilities (3.5) are conditional on an event occurring, and thus similarly for a series of events, the probability can only be calculated conditional on the total number of events. Here we denote the temporal order of mutation and split events with $\mathcal{H}$, with $\mathcal{H}^t$ being the t^{th} event and H the total number of events. Conditional on the root r, the site-specific mutation rates $\boldsymbol{\phi}$, the nucleotide frequencies $\boldsymbol{\pi}$ and the mutation rates $\boldsymbol{v}$, we have

$$\mathbb{P}(\mathcal{H} \,|\, H, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}) \;=\; \prod_t \mathbb{P}\left(\mathcal{H}^t \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}\right), \tag{3.6}$$

where the probabilities are given by Equations (3.4), (3.5). Note here that subsequent events are independent by the memoryless property of the exponential distribution, which allows us to calculate the joint posterior of the history $\mathcal{H}$ as the product of individual events $\mathcal{H}^t$.

We can now calculate the probability of a haplotype tree, denoted by $\mathcal{T}$, given the root r and the mutation parameters $\boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}$, by summing over the probabilities (3.6) of all temporal orderings $\mathcal{H}_j$ which are consistent with the tree. In other words,

$$\mathbb{P}(\mathcal{T} \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}) \;=\; \sum_j \mathbb{P}(\mathcal{H}_j \,|\, H_j, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}), \tag{3.7}$$

where $\mathcal{H}_j$ is consistent with $\mathcal{T}$. Equation (3.7) is key to fully specify a model for inference about the haplotype tree.

3.1.2 The haplotype tree model

In this subsection we develop a Bayesian model for the haplotype tree $\mathcal{T}$, the root r and the mutation parameters $(\boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})$ under sequence data $\mathcal{S}$. The objective of the analysis is to estimate the rooted haplotype tree $(\mathcal{T}, r)$, implying that $(\boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})$ are nuisance parameters. We remark that the haplotype tree $\mathcal{T}$ contains all the information available in the sequence data, and hence, conditional on the tree, the data are independent of the mutation process parameters.[1]

[1] To allow flow of the text, the calculations of several expressions are not presented within this subsection, but may be found in Appendix B.

We begin by assuming that, in the absence of any information about the mutation process, any haplotype tree $\mathcal{T}$ is equiprobable. We may express the haplotype tree $\mathcal{T}$ as the tree topology T together with the nucleotide state τ of all nodes on the tree, i.e., $\mathcal{T} = (T, \tau)$. In the first few sections to follow, both T and τ (i.e., the haplotype tree) are known. Similarly to the haplotype tree, we assume that all tree topologies are equally likely a priori given the root, as are all nucleotide states of missing intermediates given the tree topology. Furthermore, we assume that a priori any sequence is equally likely of being the root, and that $(\boldsymbol{\phi}, \boldsymbol{v})$ are independent of the root and of each other. Finally, we assume that the nucleotides of the root follow a multinomial distribution given the nucleotide frequencies. All these priors can be summarized as

$$\begin{aligned}
\mathbb{P}(\mathcal{T} \,|\, r) &\propto 1 && (3.8)\\
\mathbb{P}(T \,|\, r) &\propto 1 && (3.9)\\
\mathbb{P}(\tau \,|\, T) &\propto 1 && (3.10)\\
p(r) &\propto 1, && (3.11)\\
p(r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}) &= p(r, \boldsymbol{\pi}) \times p(\boldsymbol{\phi}) \times p(\boldsymbol{v}) && (3.12)\\
p(r \,|\, \boldsymbol{\pi}) &\propto \prod_{i=1}^{4} \pi_i^{n_i^r}, && (3.13)
\end{aligned}$$

where n_i^r represents the number of nucleotides of type i which are found in the root r.

Lemma 3.1.1. *Using the specified priors (3.8)-(3.13), the joint posterior of the mutation parameters given the rooted haplotype tree is given by*

$$\mathbb{P}(\boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v} \,|\, \mathcal{S}, \mathcal{T}, r) \propto \mathbb{P}(\mathcal{T} \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}) \times p(r) \times p(\boldsymbol{\pi} \,|\, r) \times p(\boldsymbol{\phi}) \times p(\boldsymbol{v}) \qquad (3.14)$$

Proof. See Appendix B.

Lemma 3.1.2. *Similarly, we can calculate the posterior distribution for the root*

$$\mathbb{P}(r \,|\, \mathcal{S}, \mathcal{T}, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}) \propto \mathbb{P}(\mathcal{T} \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}) \times p(r \,|\, \boldsymbol{\pi}) \qquad (3.15)$$

Proof. See Appendix B.

The two Lemmas 3.1.1, 3.1.2 above, together with Equation (3.7), provide us with the posterior distributions of all the parameters except the haplotype tree $\mathcal{T}$.

Only a little remains to compute the posterior of the haplotype tree $\mathcal{T}$ given the sequence data. Recalling that the haplotype tree fully describes the data $\mathcal{S}$, the likelihood of the data

becomes

$$\mathcal{L}(\mathcal{S} \,|\, \mathcal{T}, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}) = \begin{cases} 1 & \text{if } \mathcal{T} \text{ consistent with } \mathcal{S} \\ 0 & \text{otherwise} \end{cases}$$

Note that although only one $\mathcal{S}$ is consistent with the haplotype tree $\mathcal{T}$, there are several haplotype trees represented by the set Ω consistent with the sequences. Applying Bayes' Theorem, the posterior distribution of the haplotype tree is

$$\mathbb{P}(\mathcal{T} \,|\, \mathcal{S}, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}) \propto \begin{cases} \mathbb{P}(\mathcal{T} \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}) & \text{if } \mathcal{T} \in \Omega \\ 0 & \text{otherwise,} \end{cases} \tag{3.16}$$

where the normalization constant may be calculated as

$$\sum_{\mathcal{T}_i \in \Omega} \mathbb{P}(\mathcal{T}_i \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}).$$

For a number of reasons the infinite state space Ω which is consistent with $\mathcal{S}$ is problematic. We contend that under an argument of relaxed parsimony (discussed in detail in Sections 3.6 - 3.8), it is possible to reduce the state space to a finite (but vast) set $\Omega := \Omega(\mathcal{S})$ of realistic haplotype trees.

Remember that the probability of a tree can only be calculated conditional on the *total number of mutation and split events*, here denoted by H. However, the set Ω may contain trees involving a different number of events. In order to calculate the probability of one of those trees, we require

$$\mathbb{P}(\mathcal{T} \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}) \propto \mathbb{P}(\mathcal{T} \,|\, H, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}) \times \mathbb{P}(H \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}).$$

Assuming a uniform prior on H such that

$$\mathbb{P}(H \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}) \propto 1, \tag{3.17}$$

we obtain that $\mathbb{P}(\mathcal{T} \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})$ can be calculated using (3.7) and simply multiplying over all the events for any size of tree H.

These developments allow us to formulate a model for the haplotype tree under the Bayesian paradigm. The broad structure is given in the Directed Acyclic Graph of Figure 3.1, and the details are as follows.

To complete the model, we collect Equations (3.8) to (3.17), and assume the following priors for the mutation parameters and the rooted haplotype tree:

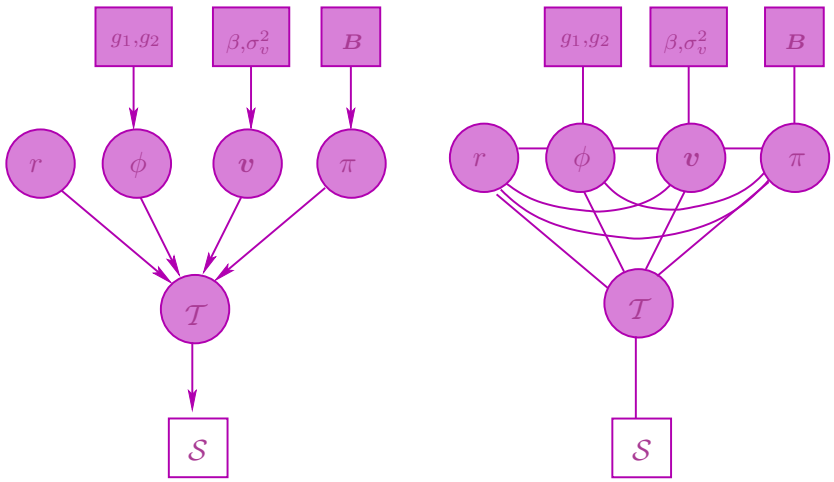


Figure 3.1: *(a) DAG specific to the model implemented; (b) Corresponding conditional independence graph.*

$$
\begin{aligned}
(\pi_1, \pi_2, \pi_3, \pi_4) &\sim \mathcal{D}(B_1, B_2, B_3, B_4) \\
v_1,\ v_6 &\sim \mathcal{G}(1/\sigma_v^2, 1/\sigma_v^2) \\
v_2,\ v_3,\ v_4,\ v_5 &\sim \mathcal{G}(1/(\beta\sigma_v^2), 1/\sigma_v^2) \\
\phi_l &\sim \mathcal{G}(g_1, g_2) \\
\phi_l \,|\, \mathcal{T} &\sim \mathcal{G}(g_1 \times m_i, g_2) \\
r &\sim \mathcal{U}\{1, \ldots, N_h\} \\
\mathcal{T} &\sim \mathcal{U}\{\mathcal{T}_i\}.
\end{aligned}
\tag{3.18}
$$

Here $\mathcal{T}_i \in \Omega$ and β represents the transition/transversion bias specific to each sample. Further, $\mathcal{D}$ denotes the Dirichlet distribution which is the conjugate prior for the multinomial distribution of $r \,|\, \boldsymbol{\pi}$ and $B_1, \ldots, B_4$ are taken small. We assume a priori that the number of mutations at each nucleotide site l given the tree topology is gamma distributed with parameter proportional to the number of mutations m_l at that site. Usually, we set σ_v^2 large, g_1, g_2 small to provide a vague prior, unless there is available information suggesting otherwise.

Model (3.18) provides several important contributions to inference on the rooted haplotype tree. Based on explicit distributions about haplotype trees, it supplies a rigorous mathematical framework for estimation. It is consistent with many of the empirical predictions raised by Crandall and Templeton (1993) and Posada and Crandall (2001). For example, older alleles have a greater probability of becoming interior haplotypes: this may be directly derived from

(3.7), since interior haplotypes naturally allow a much larger number of orderings in which events may have occurred. Furthermore, haplotypes of greater frequency are more likely to have a higher degree (i.e., more mutational connections in the tree): the probability of a mutation increases according to frequency of the haplotype; see (3.5). Perhaps the most important advantage of this model is that the posterior probability of a haplotype tree (3.16) can be explicitly expressed, which allows for *backward* rather than forward inference. In other words, we start from the data and reconstruct the tree, rather than fixing a tree and comparing with the data.

We remark that here there is no direct way of assessing the validity of the parsimony assumption. In the setting of our model, parsimony becomes invalid when one of the ϕ_is is too large compared to the rest. This would imply that sequences should not be collapsed onto haplotypes, since the large ϕ_i may have caused correspondence of whole sequences. Instead, that nucleotide site should be ignored. This could be incorporated in our model to account for such a possibility, by adding a binary parameter for each nucleotide site indicating whether it is parsimonious or not.

3.1.3 Approximating the probability of a haplotype tree

In order to draw inferences about the haplotype tree under the model presented in (3.18), calculation of $\mathbb{P}(\mathcal{T} \,|\, \mathcal{S}, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})$ is required. Remember that here

$$\mathbb{P}(\mathcal{T} \,|\, \mathcal{S}, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}) \quad \propto \quad \sum_j \mathbb{P}(\mathcal{H}_j \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}), \tag{3.19}$$

where $\mathcal{H}_j$ is any temporal ordering of events consistent with $\mathcal{T}$[2] The number of different temporal orderings which are consistent with a haplotype tree increases dramatically with the sample size N.

To overcome computationally intractable likelihoods when these involve multiple integrals or sums, a number of Approximate Bayesian Computation (ABC) approaches have been suggested (see Beaumont, 2003; Becquet and Przeworski, 2007; Beerli and Felsenstein, 2001; O'Neill et al., 2000). Here we follow the method presented by Beaumont (2003) to approximate distribution (3.7) in order to devise a computationally feasible MCMC algorithm. Specifically, to approximate (3.7), we perform importance sampling within MCMC (see Section 1.5). We draw J temporal orderings $\mathcal{H}_j$ from an importance distribution $q(\mathcal{H}_j)$ which allows a positive probability on all orderings consistent with the haplotype tree. We then calculate an

[2]In this subsection, for notational simplicity, we assume that $\mathcal{T} \in \Omega$.without loss of generality, which implies that $\mathbb{P}(\mathcal{T} \,|\, \mathcal{S}, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}) \propto \mathbb{P}(\mathcal{T} \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})$; see (3.16).

approximation to (3.7) by averaging over the importance weights

$$\hat{\mathbb{P}}_{\mathcal{H}}(\mathcal{T} \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}) = \frac{1}{J} \sum_{j} \frac{1}{q(\mathcal{H}_j)} \mathbb{P}(\mathcal{H}_j \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}), \tag{3.20}$$

In the following few sections we consider the simulated orderings $\boldsymbol{\mathcal{H}} = \{\mathcal{H}_1, \ldots, \mathcal{H}_J\}$ as an auxiliary variable within MCMC updates of individual parameters, and approximate the posterior distribution of the haplotype tree by (3.20). Beaumont (2003) shows that such a chain is time-reversible with target distribution $\hat{\mathbb{P}}_{\mathcal{H}}(\mathcal{T} \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})$ which is itself an unbiased approximation of the true posterior $\mathbb{P}(\mathcal{T} \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})$ given in Equation (3.7).

Notably the choice of the importance distribution is arbitrary, provided it allows a positive weight on all possible $\mathcal{H}_i$. Here we define $q(\mathcal{H} \,|\, \mathcal{T}, r)$ by the following Algorithm 3.1.3, which constructs a temporal ordering by starting at the MRCA of the sample (i.e., the root sequence) and iteratively picking the next mutation or split event. Before we move on to fully describe the algorithm, we present an example in order to discuss which mutation and split events are consistent with a haplotype tree.

Example

Suppose the haplotype tree is given by the top tree of Figure 3.2. For ease of exposition, the numbers on the nodes here represent the sample sizes of each haplotype rather than the label of each haplotype.

Simulating a temporal ordering implies that, starting with the ancestral sequence, we specify a series of split and mutation events which occurred by mimicking evolution, eventually resulting in the fixed haplotype tree. Here we represent each event by updating the numbers on each haplotype according to the number of times it is observed at each time-point in the sample. For example, the bottom panel of Figure 3.2 is a possible temporal ordering of the observed tree given in the top panel.

Observe now that, for example, the root node could not have split any further: this would result in three copies of the ancestral haplotype, which is inconsistent with the haplotype tree which specifies precisely two. In addition, it would not have been possible for the intermediate haplotype to mutate after Step 3 above, since then it would disappear from the ancestral sequences, and another mutation would not have been possible. In other words, consistent events are defined as follows.

- A split event is consistent with the haplotype tree, if it does not imply that the sample size of that haplotype will exceed the number of times it appears in the complete haplotype tree, plus the number of mutations that haplotype will be forced to undergo

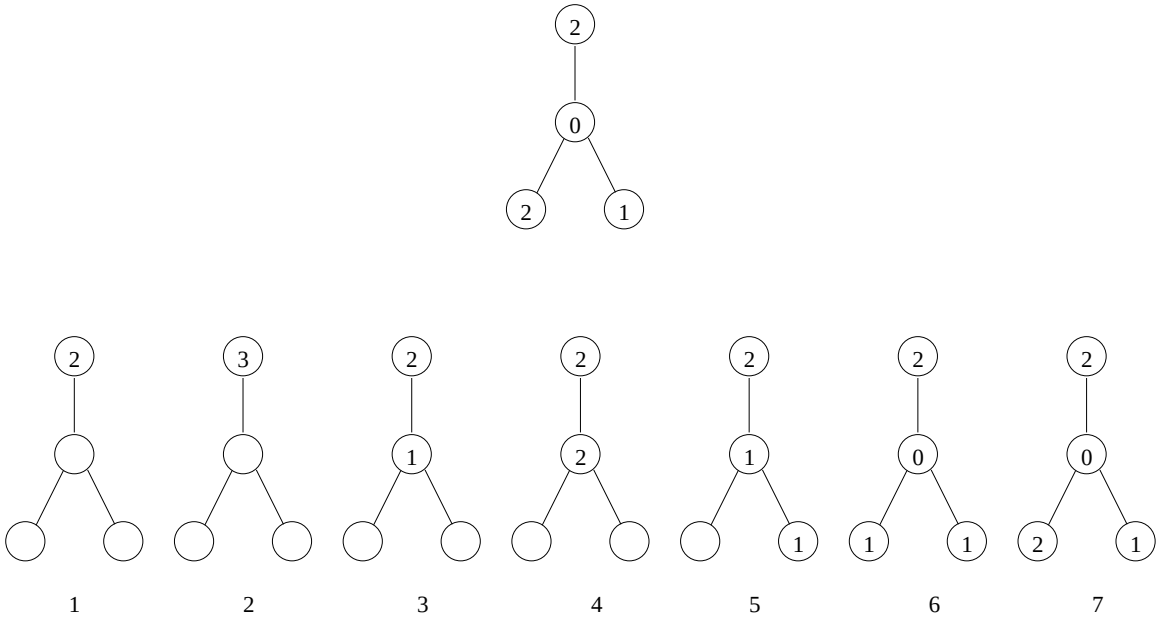


Figure 3.2: *Top panel: In this tree the mrca of the sample (the top haplotype) is observed twice in the sample. Note that one of the intermediate haplotypes in not observed in the sample (and hence has zero sample size). Bottom panel: nodes without a number represent haplotypes which have not arisen yet. At first one sequence is present, the ancestral sequence, which split into two (remember that the first event is always a split). Then one of those two identical sequences split again to give us a total of three. One of those three then mutate to give us the intermediate haplotype, which in turn splits and then mutates (and goes extinct) to give us the right-hand leaf. Finally, the intermediate haplotype mutates again to give us the left-hand leaf, which then also splits to give another copy of itself.*

in following steps (so, in the example, the intermediate haplotype after Step 5 will be forced to undergo exactly one more mutation).

- Similarly, a mutation is possible if (a) is true, and (b) OR (c) are true:

  (a) it is represented by an edge on the haplotype tree, where the ancestral sequence of the edge has already appeared in the ancestral sample

  (b) the ancestral sequence of the edge corresponding to that mutation does not go extinct

  (c) the ancestral sequence of the edge goes extinct, and there are not more events involving that sequence which have not yet occurred but are forced by the haplotype tree.

We can now describe Algorithm 3.1.3 which generates temporal orderings consistent with a fixed haplotype tree $\mathcal{T}$.

Algorithm 3.1.3.

This algorithm generates temporal orderings by mimicking the ancestral history of the sample, starting with the root and ending with the observed sequences.

1. Start at the root. Initially only one copy of the root haplotype is present. Split it into two copies and repeat the next step until all mutation or split events determined by the haplotype tree have occurred.

2. For all sequences present, consider all mutations and splits that are consistent with the haplotype tree.

   For each of those events calculate

$$\begin{aligned} \mathbb{P}(\text{sequence } n \text{ splits}) &\propto \frac{1}{N_{t_0}} \frac{N_{t_0}-1}{N_{t_0}-1+\theta} \\ \mathbb{P}(l\text{th site of sequence } n \text{ mutates}) &\propto \frac{1}{N_{t_0}} \frac{\phi_i q_{X_l^n}}{((N_{t_0}-1)/2+\theta/2)} \end{aligned} \tag{3.21}$$

   where N_t is the number of sequences present by that iteration of Algorithm 3.1.3. Note here that the probabilities are only proportional to the expressions above. This is because not all mutations and splits are possible on a given haplotype tree are possible, and hence the expressions do not sum to one.

   Select one of the available events to occur with probabilities proportional to (3.21), and repeat this Step.

For each temporal ordering $\mathcal{H}$ generated from this algorithm, the probability $q(\mathcal{H} \,|\, \mathcal{T}, r)$ can be calculated by first normalizing the terms in (3.21) at *each* step and then multiplying over them. By construction, any $\mathcal{H}$ consistent with $\mathcal{T}$ may be generated under Algorithm 3.1.3 because at all the steps, the consistent events have non-zero probability.

We have now completely specified the approximation of the posterior distribution (3.19). The approximation is also used to calculate the probabilities of the mutation process parameters and the root of the tree:

$$\hat{\mathbb{P}}_{\mathcal{H}}(\boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v} \,|\, T, r) = \hat{\mathbb{P}}_{\mathcal{H}}(\mathcal{T} \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}) \times p(r) \times p(\boldsymbol{\pi} \,|\, r) \times p(\boldsymbol{\phi}) \times p(\boldsymbol{v}) \tag{3.22}$$

$$\hat{\mathbb{P}}_{\mathcal{H}}(r \,|\, \mathcal{T}, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}) = \frac{\hat{\mathbb{P}}_{\mathcal{H}}(\mathcal{T} \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}) \times p(r \,|\, \boldsymbol{\pi})}{\mathbb{P}_{\mathcal{H}}(\mathcal{T} \,|\, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})}. \tag{3.23}$$

Using Equations (3.20), (3.22), (3.23), it is now possible to construct a Markov chain Monte Carlo sampler to simulate from the posterior distribution

$$\hat{\pi}_{\mathcal{H}}(\mathcal{T}, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v} \,|\, \mathcal{S}) \tag{3.24}$$

The chain is initialized by generating a tree $\mathcal{T}^{(0)}$, a root $r^{(0)}$ uniformly from $\mathcal{T}^{(0)}$, mutation rates $\boldsymbol{\phi}^{(0)}$ from their prior, nucleotide frequencies $\boldsymbol{\pi}^{(0)}$ from $\boldsymbol{\pi} \,|\, r$ and mutation coefficients $\boldsymbol{v}^{(0)}$ from their prior.

3.2 Updating the mutation rates

In this section we describe how to draw inferences about the mutation rates $\boldsymbol{\phi}$ given a known rooted haplotype tree, nucleotide frequencies $\boldsymbol{\pi}$, mutation coefficients $\boldsymbol{v}$ and the sequence data $\mathcal{S}$. All information contained in $\mathcal{S}$ is implicit in the haplotype tree $\mathcal{T}$, and thus given the tree, $\boldsymbol{\phi}$ is independent of the data.

E1a Propose to update one of $\phi_1, \ldots, \phi_L$ by using again a reflective proposal, so that

$$\phi_i' = \begin{cases} \phi_i + \epsilon & \text{if } v_i + \epsilon > 0 \\ -(\phi_i + \epsilon) & \text{otherwise} \end{cases}$$

where $\epsilon \sim \mathcal{U}[-E_\phi, E_\phi]$.

E1b Propose $\boldsymbol{\mathcal{H}}' = \{\mathcal{H}_1', \ldots, \mathcal{H}_J'\}$ according to $q(\mathcal{H} \,|\, \mathcal{T}, r)$ described above.

E1c This move is then accepted with probability $\min(A, 1)$, where A is given by

$$\begin{aligned} A &= \frac{q(\boldsymbol{\phi}' \to \boldsymbol{\phi})}{q(\boldsymbol{\phi} \to \boldsymbol{\phi}')} \times \frac{\hat{\mathbb{P}}_{\mathcal{H}'}(\boldsymbol{\phi}' \,|\, \mathcal{S}, \mathcal{T}, r, \boldsymbol{\pi}, \boldsymbol{v})}{\hat{\mathbb{P}}_{\mathcal{H}}(\boldsymbol{\phi} \,|\, \mathcal{S}, \mathcal{T}, r, \boldsymbol{\pi}, \boldsymbol{v})}. \\ &= \frac{\hat{\mathbb{P}}_{\mathcal{H}'}(\boldsymbol{\phi}' \,|\, \mathcal{T}, r, \boldsymbol{\pi}, \boldsymbol{v})}{\hat{\mathbb{P}}_{\mathcal{H}}(\boldsymbol{\phi} \,|\, \mathcal{T}, r, \boldsymbol{\pi}, \boldsymbol{v})}. \\ &= \frac{\hat{\mathbb{P}}_{\mathcal{H}'}(\mathcal{T} \,|\, r, \boldsymbol{\phi}', \boldsymbol{\pi}, \boldsymbol{v})}{\hat{\mathbb{P}}_{\mathcal{H}}(\mathcal{T} \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})} \times \frac{p(\phi_i')}{p(\phi_i)} \end{aligned} \tag{3.25}$$

using Equation (3.22).

If the move is accepted we set $(\boldsymbol{\phi}^{(t+1)}, \boldsymbol{\mathcal{H}}) = (\boldsymbol{\phi}', \boldsymbol{\mathcal{H}}')$, otherwise we set $(\boldsymbol{\phi}^{(t+1)}, \boldsymbol{\mathcal{H}}) = (\boldsymbol{\phi}^{(t)}, \boldsymbol{\mathcal{H}})$.

In practice, the number of sites it often large (more than 200), and calculation of the probability $\hat{\mathbb{P}}_{\mathcal{H}'}(\mathcal{T} \,|\, r, \boldsymbol{\phi}', \boldsymbol{\pi}, \boldsymbol{v})$ is computationally expensive. In addition, there is usually little information to draw inferences about all the ϕs, since very few mutate more than once, implying that accurate estimates are impossible. On average, the number of times a site mutates is proportional to the mutation rate of that site. Sites l that do not show any mutations in our sample naturally yield $\phi_l = 0$. As a result, it is often more efficient to reduce the number of parameters by using the prior $\phi_l \,|\, \mathcal{T}$ and letting $\phi_l = \phi(m_l)$. Here m_l is

the mutations number of mutations at site l on tree $\mathcal{T}$. In other words, all sites which mutate only once are assumed to have the same mutation rate, and so forth.

3.3 Updating the nucleotide frequencies

We now describe an update for the nucleotide frequencies $\boldsymbol{\pi}$.

E2a Propose to update $\boldsymbol{\pi}$ by proposing new values $\boldsymbol{\pi}$' from the distribution

$$\begin{aligned} p(\boldsymbol{\pi} \,|\, r) &\propto p(r \,|\, \boldsymbol{\pi}) \times p(\boldsymbol{\pi}) \\ &= \prod_{i=1}^{4} \pi_i^{n_i^r} \times \frac{\prod \pi_i^{B_i+i-1}}{Beta(\boldsymbol{\pi})}, \end{aligned} \tag{3.26}$$

which implies that $\boldsymbol{\pi} \,|\, r \sim \mathcal{D}(B_1 + n_1^r, B_2 + n_2^r, B_3 + n_3^r, B_4 + n_4^r)$. Here n_i^r represents the number of times nucleotide i is observed in the root haplotype r.

E2b Propose $\boldsymbol{\mathcal{H}'} = \{\mathcal{H}'_1, \ldots, \mathcal{H}'_J\}$ according to $q(\mathcal{H} \,|\, \mathcal{T}, r)$ described above.

E2c This move is then accepted with probability $\min(1, A)$, where A is given by

$$\begin{aligned} A &= \frac{q(\boldsymbol{\pi'} \to \boldsymbol{\pi})}{q(\boldsymbol{\pi} \to \boldsymbol{\pi'})} \times \frac{\hat{\mathbb{P}}_{\mathcal{H}'}(\boldsymbol{\pi}' \,|\, \mathcal{S}, \mathcal{T}, r, \boldsymbol{\phi}, \boldsymbol{v})}{\hat{\mathbb{P}}_{\mathcal{H}}(\boldsymbol{\pi} \,|\, \mathcal{S}, \mathcal{T}, r, \boldsymbol{\phi}, \boldsymbol{v})}. \\ &= \frac{\frac{\prod \pi_i^{B_i+n_i^r-1}}{Beta(\boldsymbol{\pi})}}{\frac{\prod \pi_i'^{B_i+n_i^r-1}}{Beta(\boldsymbol{\pi}')}} \times \frac{\hat{\mathbb{P}}_{\mathcal{H}'}(\mathcal{T} \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}', \boldsymbol{v})}{\hat{\mathbb{P}}_{\mathcal{H}}(\mathcal{T} \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})} \times \frac{p(\boldsymbol{\pi}' \,|\, r)}{p(\boldsymbol{\pi} \,|\, r)} \\ &= \frac{\hat{\mathbb{P}}_{\mathcal{H}'}(\mathcal{T} \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}', \boldsymbol{v})}{\hat{\mathbb{P}}_{\mathcal{H}}(\mathcal{T} \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})} \end{aligned} \tag{3.27}$$

using Equation (3.22).

If the move is accepted we set $(\boldsymbol{\pi}^{(t+1)}, \boldsymbol{\mathcal{H}}) = (\boldsymbol{\pi}', \boldsymbol{\mathcal{H}'})$, otherwise we set $(\boldsymbol{\pi}^{(t+1)}, \boldsymbol{\mathcal{H}}) = (\boldsymbol{\pi}^{(t)}, \boldsymbol{\mathcal{H}})$.

3.4 Updating the mutation coefficients

The mutation coefficients $\boldsymbol{v}$ are updated in a similar way to the mutation rates. As before, the mutation coefficients are independent of the sequence data $\mathcal{S}$ given the haplotype tree $\mathcal{T}$.

E3a Propose to change one of v_i randomly by a reflective proposal $v_i \to v_i'$ by generating

$\epsilon \sim \mathcal{U}[-E_v, E_v]$ and then setting:

$$v_i' = \begin{cases} v_i + \epsilon & \text{if } v_i + \epsilon > 0 \\ -(v_i + \epsilon) & \text{otherwise} \end{cases}$$

E3b Propose $\boldsymbol{\mathcal{H}}' = \{\mathcal{H}_1', \ldots, \mathcal{H}_J'\}$ according to $q(\mathcal{H} \,|\, \mathcal{T}, r)$ described above.

E3c The Hastings ratio becomes

$$\begin{aligned} A &= \frac{q(\boldsymbol{v}' \to \boldsymbol{v})}{q(\boldsymbol{v} \to \boldsymbol{v}')} \times \frac{\hat{\mathbb{P}}_{\mathcal{H}'}(\boldsymbol{v}' \,|\, \mathcal{S}, \mathcal{T}, r, \boldsymbol{\phi}, \boldsymbol{\pi})}{\hat{\mathbb{P}}_{\mathcal{H}}(\boldsymbol{v} \,|\, \mathcal{S}, \mathcal{T}, r, \boldsymbol{\phi}, \boldsymbol{v})}. \\ &= \frac{\hat{\mathbb{P}}_{\mathcal{H}'}(\boldsymbol{v}' \,|\, \mathcal{T}, r, \boldsymbol{\phi}, \boldsymbol{\pi})}{\hat{\mathbb{P}}_{\mathcal{H}}(\boldsymbol{v} \,|\, \mathcal{T}, r, \boldsymbol{\phi}, \boldsymbol{\pi})}. \\ &= \frac{\hat{\mathbb{P}}_{\mathcal{H}'}(\mathcal{T} \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}')}{\hat{\mathbb{P}}_{\mathcal{H}}(\mathcal{T} \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})} \times \frac{p(v_i')}{p(v_i)} \end{aligned} \tag{3.28}$$

using Equation (3.22).

If the move is accepted we set $(\boldsymbol{v}^{(t+1)}, \boldsymbol{\mathcal{H}}) = (\boldsymbol{v}', \boldsymbol{\mathcal{H}}')$, otherwise we set $(\boldsymbol{v}^{(t+1)}, \boldsymbol{\mathcal{H}}) = (\boldsymbol{v}^{(t)}, \boldsymbol{\mathcal{H}})$.

Example

The total algorithm to draw inferences about the mutation parameters then follows steps E1-E3. This means that we sequentially update the mutation rates, nucleotide frequencies and mutation coefficients.

We generated dataset S5 of 200 sequences of length 700 each from the prior distributions, taking the following parameters:

$$\begin{aligned} (\pi_1, \pi_2, \pi_3, \pi_4) &\sim \mathcal{D}(1,1,1,1) \\ v_1,\ v_6 &\sim \mathcal{N}(7, 0.5) \\ v_2,\ v_3,\ v_4,\ v_5 &\sim \mathcal{N}(1, 0.5) \\ \phi_i &\sim \mathcal{G}(1,1) \end{aligned}$$

We ran the described MCMC sampler assuming the known haplotype tree as fixed. We present the trace and density plots of a few representative parameters, as well as the corresponding Gelman-Rubin plots.

The same dataset was used in subsequent sections to demonstrate the results of our algorithm in the three cases where the rooted haplotype tree is known (current section),

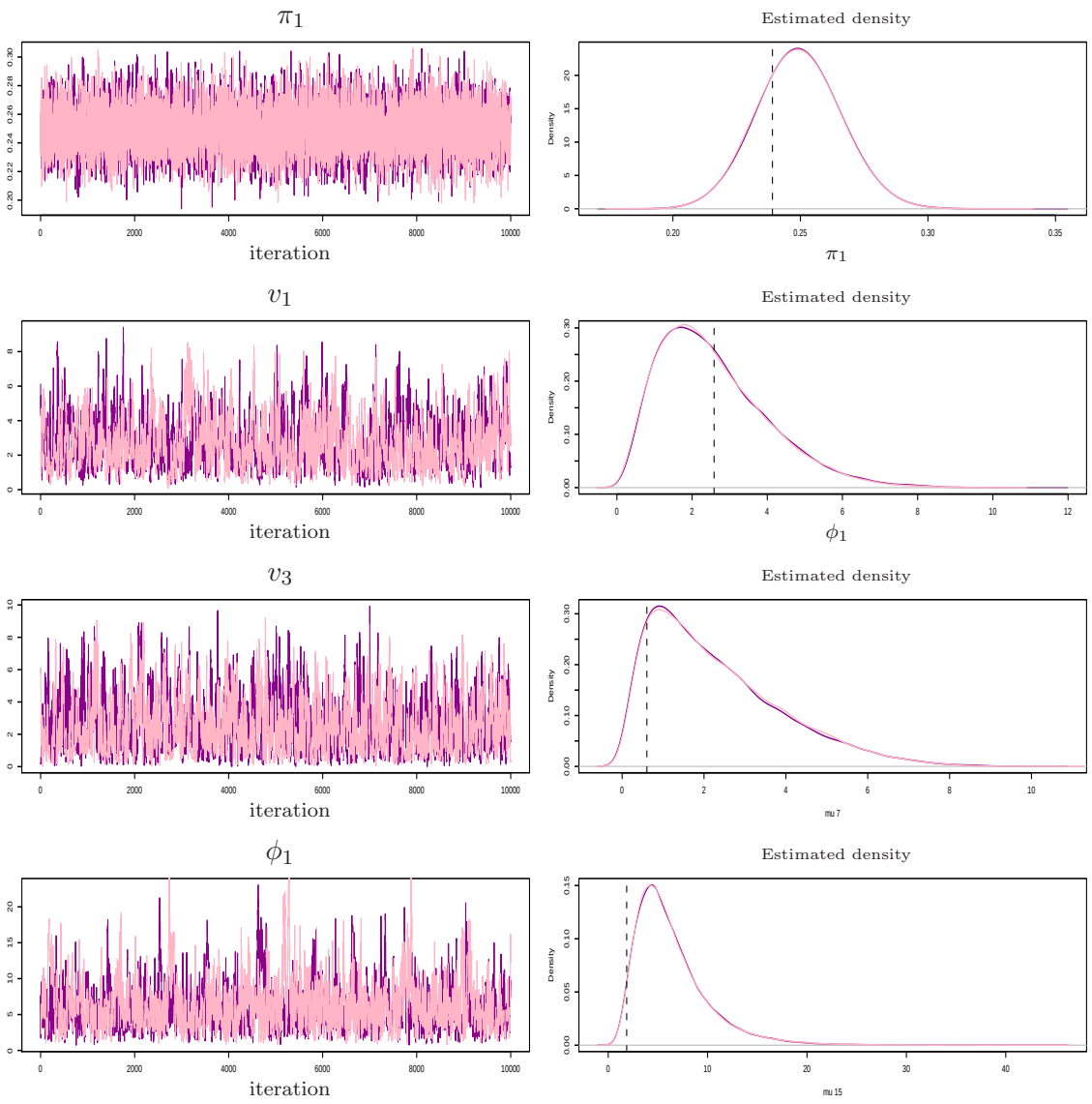


Figure 3.3: *Trace and density plots for simulated dataset S5 with a known root starting from two different seeds. The dashed lines represent the true values of the parameters. The trace plots are thinned by a factor of 10. All parameters show excellent mixing, and the posterior densities match for the two chains.*

where the haplotype tree is known but the root has to be inferred (next Section 3.5) and finally where the tree is unknown (Section 3.9). At the end we make a comparison of the estimates of the parameters in order to investigate how increasing the uncertainty of the model affects the reliability of the estimates.

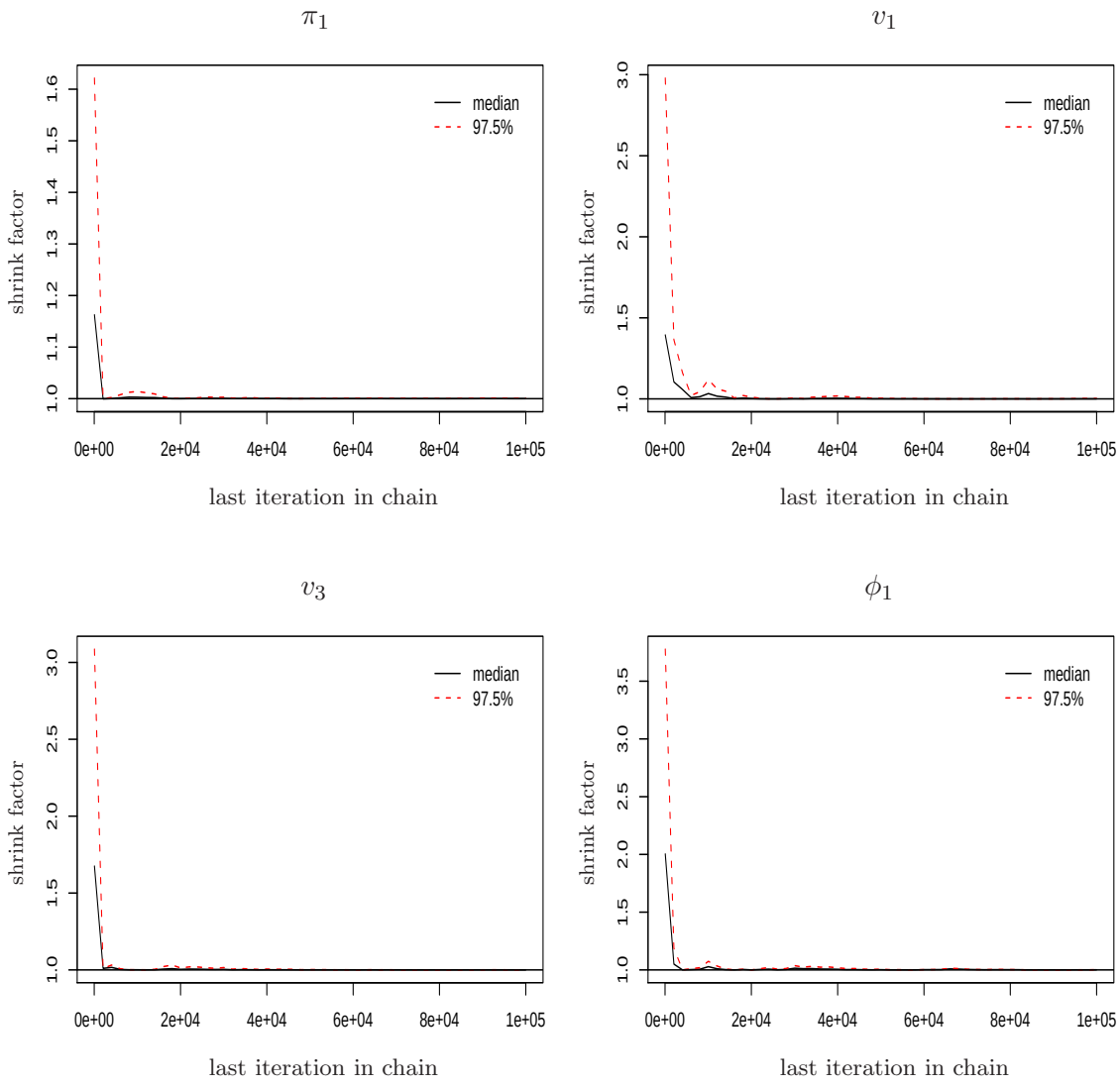


Figure 3.4: *Potential Scale Reduction Factor plots for simulated dataset S5 with a known root, suggesting that the chains have converged.*

3.5 Updating the root

We now consider the case where the root of the haplotype tree is unknown given a fixed haplotype tree and mutation process parameters. As with the mutation parameters, the root is independent of the sequence data $\mathcal{S}$ given the haplotype tree.

In order to update the root, we need to select a proposal kernel $q(r \to r')$. We present three proposal kernels below:

- Uniformly from all haplotypes, so that $q(r \to r') = q(r') \propto 1$

- Uniformly proportional to each node's degree, so that $q(r \to r') = q(r') \propto \text{degree}(r')$. Generally older haplotypes are more likely to have mutated more times, so it is more likely that haplotypes with a higher degree will be the root.
- Uniformly from adjacent nodes, so that $q(r \to r') = \frac{\mathbb{I}_{r' \text{ adjacent to } r}}{\text{degree}(r)}$. The oldest haplotype clearly has to be adjacent to the second oldest, implying that the oldest haplotypes are generally adjacent to each other.

Using one of the available proposal kernels, the root update becomes:

E4a Propose a new root according to the preferred proposal kernel q.

E4b Propose $\boldsymbol{\mathcal{H}}' = \{\mathcal{H}'_1, \ldots, \mathcal{H}'_J\}$ according to $q(\mathcal{H} \,|\, \mathcal{T}, r)$ described above.

E4c Using the expression (3.23), the probability of accepting the proposed move becomes $\min(1, A)$, where

$$\begin{aligned} A &= \frac{q(r' \to r)}{q(r \to r')} \frac{\hat{\mathbb{P}}_{\mathcal{H}'}(r' \,|\, \mathcal{T}, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})}{\hat{\mathbb{P}}_{\mathcal{H}}(r \,|\, \mathcal{T}, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})} \\ &= \frac{q(r' \to r)}{q(r \to r')} \frac{\hat{\mathbb{P}}_{\mathcal{H}'}(\mathcal{T} \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})}{\hat{\mathbb{P}}_{\mathcal{H}}(\mathcal{T} \,|\, r', \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})} \times \frac{p(r \,|\, \boldsymbol{\pi})}{p(r \,|\, \boldsymbol{\pi})} \end{aligned} \tag{3.29}$$

If the move is accepted we set $(r^{(t+1)}, \boldsymbol{\mathcal{H}}) = (r', \boldsymbol{\mathcal{H}}')$, otherwise we set $(r^{(t+1)}, \boldsymbol{\mathcal{H}}) = (r^{(t)}, \boldsymbol{\mathcal{H}})$.

Example

The algorithm to draw inferences about the root haplotype and mutation process parameters then follows steps E1-E4. This means that we sequentially update the root, nucleotide frequencies, mutation coefficients and mutation rates.

First we implement it on dataset S5 as before using a uniform proposal for the root, and obtain estimates for the mutation parameters which are no less accurate than the analysis with a known root in Example 3.4. Similarly to earlier, the mixing is excellent and the Gelman-Rubin plots suggest convergence. Here we do not show the plots since they do not show any significant differences to the previous Example.

We now generate 20 datasets of 200 sequences of length 100 (denoted by simulation S6) and investigate how frequently the algorithm estimates the correct root. The results of the algorithm are shown in Table 3.1. The correct haplotype is identified only five out of 20 times. This is not surprising: the evolutionary process assumed has a very large variance on the shape of haplotype trees, and hence the true posterior distribution of the root given the

Haplotype	1	2	3	4	5	6	7	8	9	10	11
Frequency	5	2	5	3	2	1	0	1	0	1	0

Table 3.1: *The results of the MCMC algorithm on the 20 simulated datasets S6. The haplotypes are labelled according to their temporal order (with haplotype 1 being the ancestral haplotype).*

data may deviate a lot from the true value of the root. For example, if the root haplotype does not go extinct, we expect the tree to have a root with lots of adjacent haplotypes. On the other hand, if the root goes extinct, we expect to see possibly deep divergence, yielding a very different shape.

Although the estimates for the root are inherently unreliable because of the variation in the model, we will see at the end of this Chapter 3.11 that when the sequence data $\mathcal{S}$ is combined with geographical data for each individual, ancestral locations may be estimated with a high probability of success.

3.6 Defining the tree space Ω

In the presence of homoplasy, the haplotype tree is unknown and the tree space is infinite. Here we describe how a *finite* set of realistic (in terms of a relaxed parsimony assumption) haplotype trees Ω may be obtained from the sequence data $\mathcal{S}$. Recall that in Subsection 3.1.2 we defined the posterior probability of a haplotype tree based on the set Ω. Homoplasy has two effects on the haplotype tree. Firstly, by allowing an arbitrary number of intermediate mutations, infinitely many haplotype trees are consistent with $\mathcal{S}$, most of which have a vanishing probability; this is illustrated in Figure 3.5. Secondly, homoplasy may lead to the presence of loops in the haplotype network, so that multiple parsimonious trees may be consistent with the sequence data; see Figure 3.6. These two effects combined with missing

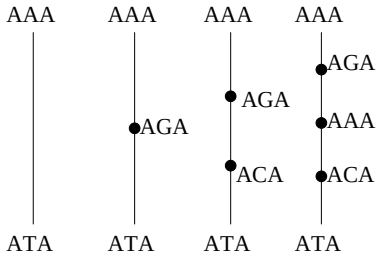


Figure 3.5: *Examples of possible haplotype trees consistent with the data AAA, ATA. Following the examples in this Figure, we can see how an infinite number of consistent trees may be constructed by adding homoplasious mutations. The parsimony assumption discards all but the first tree.*

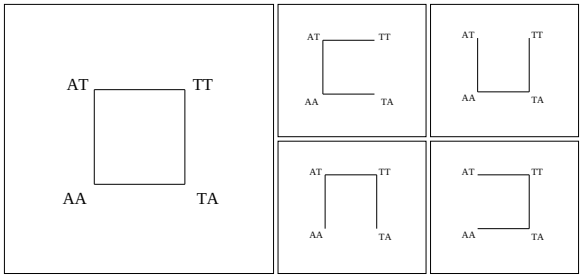


Figure 3.6: *Example of a haplotype network involving a loop. Following the parsimony assumption, sequences which are one SNP apart are assumed to be one mutation apart, and are connected. It is not possible to determine which of the three trees on the right is correct.*

intermediate sequences imply that there may be a vast space of possible haplotype trees which have a significant probability given the sequence data $\mathcal{S}$.

In cases such as Figure 3.5, it is reasonable to assume that the true haplotype tree is represented by the left-most tree, following a parsimonious argument for sequences which are one SNP apart (and hence do not require the insertion of missing intermediate sequences). In most cases, however, the available DNA sequence data result in more than one disconnected tree, requiring the insertion of unknown intermediate sequences. We construct a deterministic parsimony-based Algorithm 3.6.1 below to infer a set of realistic haplotype trees Ω. Algorithm 3.6.1 is quite technical, but is based on the intuitive idea that we set a mutational step limit d_s, and assume that any pair of disconnected sequences which is d_i SNPs apart will be a maximum of $d_i + d_s$ mutations apart. The set Ω is constructed by cumulatively adding intermediate sequences following the *relaxed* parsimony assumption defined by d_s, the mutational step limit.

Algorithm 3.6.1.

First we pick a number of steps d_s, which will be the number of mutations by which we relax the parsimony assumption for missing intermediate sequences. This means that we assume that if two sequence are k letters apart, then they are at most $k + d_s$ mutational steps apart. We then connect sequences of the sample which are one DNA change apart, thus forming a number of groups of connected nodes in a graph.

1. We connect any haplotypes which are one SNP apart, and count the number of disconnected groups of nodes. If the sequence data $\mathcal{S}$ form a connected tree, we assume that it is indeed the true haplotype tree $\mathcal{T}$ so that $\Omega = \{\mathcal{T}\}$ and the algorithm terminates. For every pair of groups, we find the closest distance between two nodes belonging to each group. We will refer to these pairs of nodes as the representatives between two groups (not always unique). When no homoplasy is present, these are unique for each

pair of groups. If the graph is connected (i.e., all sequences belong to the same group), the algorithm terminates.

2. Then we find the minimum of these minimum distances d_{min}.

3. We find all pairs of sequences (i, j) which belong to different groups and have distance (in terms of number of SNP mutations apart) $d(i, j) \leq d_{min} + d_s$. If no such pair can be found, go to Step 5 for the minimum pair of haplotypes.

4. For each pair (i, j) we then check if i has an adjacent node k which has $d(k, j) \leq d(i, j)$, and similarly for j. If either of these is true, we repeat this step for the next pair of edges. Else we go to the next step.

5. We then find all the pairs of groups which have the reference node as one of their two representatives. We store the separating mutation positions between each one of these representatives and the reference node.

6. Then we find the separating mutation(s) which occurs most frequently between those pairs, and we pick one of them, which we call the "reference mutation". This mutation has to be the one that occurred closest to the reference node, and so we create an extra node which is identical to the reference node except at the reference mutation position. When the reference mutation is not unique, without loss of generality we pick the first such nucleotide site. If any of these new nodes has already been created, clearly we do not add the same sequence twice. We then go back to step 3 and repeat for the next pair of sequences.

This algorithm results in a haplotype *network*, implying that loops may appear. The key assumption of our approach is that the true haplotype tree is assumed to be a subtree of the haplotype network obtained through the algorithm. The subtrees can be achieved by breaking the loops. Clearly, increasing d_s will generally result to disconnected groups of nodes being connected in more paths when homoplasy is present. This implies that we can allow more and more possible haplotype trees. However, that does not imply that letting $d_s \to \infty$ will ensure that the network formed will include any possible mutational path. In fact, after a value $d_s^{\max}$ is reached, increasing d_s has no effect on the set Ω.

The drawback of fixing the set Ω before the MCMC algorithm is that Ω may not include the true tree. Although it is generally true that evolution frequently follows the minimal path (see Sankoff, 1975), this is not always the case, especially when a multiple homoplasies are present. The parsimony assumption can alternatively be avoided by defininining the clustering algorithms directly on coalescent trees, but this approach proves computationally intractable; see Appendix C.

Example: simulated dataset

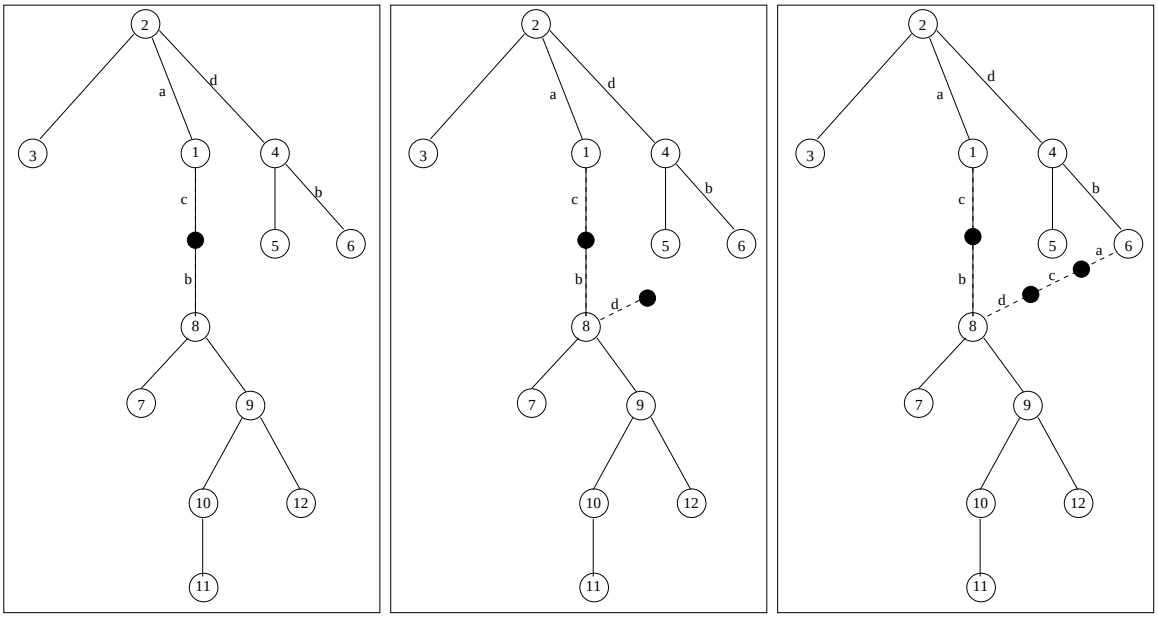


Figure 3.7: *The figure above shows the three iterations for finding the two missing nodes, using the tree in Figure 2.4 from the examples in Chapter 2. The letters on some of the edges represent the nucleotide position of each mutation. Here there is a back-mutation either at position a or position b.*

We use the haplotype tree from the previous chapter, removing haplotype 13, and follow the steps described above in Algorithm 3.6.1 in order to complete the missing nodes. We describe the algorithm for three cases: $d_s = 0$, $d_s = 1$ and $d_s > 1$.

A. Set $d_s = 0$. There are two disconnected groups of haplotypes: (1, 2, 3, 4, 5, 6), and (7, 8, 9, 10, 11, 12), with closest distance between nodes 1 and 8 which are two mutations apart (Step 1). Since there is only one pair of groups, immediately we obtain $d_{min} = 2$ (Step 2).

There are no other pairs of nodes from the two groups which are $d_{min} + d_s = d_{min}$ nucleotides apart (Step 3), so we only need to connect the two nodes 1 and 8. Since these are only two groups available, they are the only ones involving the two missing mutations (Step 4), so we insert the missing node 13 (referring to the original tree) and terminate (Step 5). This yields the true tree (of this Example). This single addition is shown in the left-hand panel of Figure 3.7.

B. Set $d_s = 1$. There are two disconnected groups of haplotypes: (1, 2, 3, 4, 5, 6), and (7, 8, 9, 10, 11, 12), with closest distance between nodes 1 and 8 which are two mutations

apart (Step 1). Since there is only one pair of groups, immediately we obtain $d_{min} = 2$ (Step 2).

In this case there, (1, 8) is the closest pair, but (2, 8) and (6, 8) are two nucleotides apart, which is indeed less than $\leq d_{min} + d_s$ apart (Step 3). Node 2 is adjacent to 1, which is closer to 8, so considering (2, 8) is implicit in the pair (1, 8), and hence redundant (Step 4). On the other hand, no such adjacent nodes exist for the pair (6, 8), which has to be taken into account (Step 4). For both pairs (1, 8) and (6, 8), there are only two groups involving the nucleotide changes (Step 5), so both pairs are connected through their quickest route (Step 5). In the case of (1, 8) this yields the same connection as before (left-hand panel of Figure 3.7), but an extra branch is added on the right through two missing nodes, as shown in the middle and right-hand panel of Figure 3.7.

C. Set $d_s > 1$. In this case the exact network is obtained as in the case $d_s = 1$. This is because any extra pairs of sequences (i, j) which are obtained in Step 3 actually have an adjacent node k which is closer to j, thus making the pair (i, j) redundant. The only pairs of sequences that reach Step 5 are, as before, (1, 8) and (6, 8).

Lemma 3.6.2. *When no homoplasy is present, the above Algorithm 3.6 results in a unique haplotype tree (the true tree) up to rearrangement of strands of missing intermediate sequences (explained below, see Figure 3.8) for any value of d_s.*

Proof. See Appendix B.

Even though the tree topology is uniquely determined when no homoplasy is present, it is often only possible to determine the order of mutations up to permutation. Specifically, consider the following example in Figure 3.8.

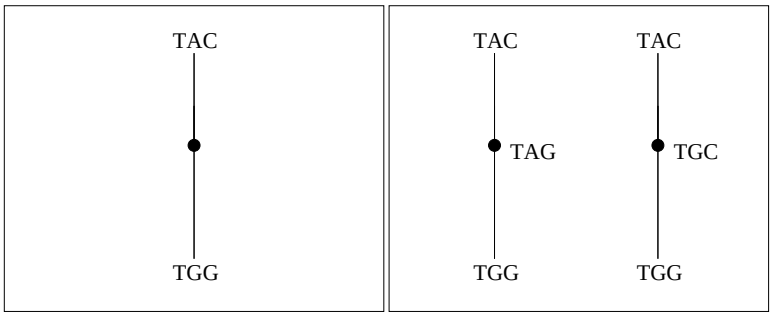


Figure 3.8: *The figure above shows an example of a missing intermediate node of degree two. In this case, it is not possible to determine which of the two possibilities on the right actually occurred.*

In the next section we assume that the tree topology is known, and describe how strands of missing intermediate sequences may be dealt with.

3.7 Updating the state of missing intermediate sequences

In order to calculate the likelihood of a haplotype tree, the series of mutation and split events has to be fixed. As described in the previous section and illustrated in Figure 3.8, when strands of missing nodes are present, it is not possible to uniquely determine the exact state of each of the missing sequences, and equally the exact mutations. Although this does not affect the tree topology (and hence the phenotypic or phylogeographic clustering), it affects the calculation of the likelihood of the tree. The state of missing sequences either has to be updated as an auxiliary parameter of our MCMC, or the likelihood would contain the sum over all the possibilities. For computational efficiency, we take the former approach.

For each strand of missing sequences of length l_s, there are $l_s!$ possibilities for the order in which the mutations occurred. The order of mutations is updated as follows.

E5a Uniformly pick one of the strands of missing nodes of length l_s and propose an order of mutations from the $l_s!$ possibilities, defining a new set of intermediate sequences, so that $\mathcal{T}' = (T, \tau')$.

E5b Propose $\boldsymbol{\mathcal{H}}' = \{\mathcal{H}'_1, \ldots, \mathcal{H}'_J\}$ according to $q(\mathcal{H} \,|\, \mathcal{T}, r)$.

E5c Accept the proposed move with probability $\min(1, A)$, where

$$\begin{aligned} A &= \frac{\hat{\mathbb{P}}_{\mathcal{H}'}(\tau' \,|\, \mathcal{S}, T, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})}{\hat{\mathbb{P}}_{\mathcal{H}}(\tau \,|\, \mathcal{S}, T, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})} \\ &= \frac{\hat{\mathbb{P}}_{\mathcal{H}'}(\tau', T \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})}{\hat{\mathbb{P}}_{\mathcal{H}'}(T \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})} \frac{\hat{\mathbb{P}}_{\mathcal{H}}(T \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})}{\hat{\mathbb{P}}_{\mathcal{H}}(\tau, T \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})} \\ &= \frac{\hat{\mathbb{P}}_{\mathcal{H}'}(\mathcal{T}' \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})}{\hat{\mathbb{P}}_{\mathcal{H}}(\mathcal{T} \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})} \end{aligned}$$

using Equation (3.9).

If we accept, set $(\mathcal{T}^{(t+1)}, \boldsymbol{\mathcal{H}}) = (\mathcal{T}', \boldsymbol{\mathcal{H}}')$, otherwise we set $(\mathcal{T}^{(t+1)}, \boldsymbol{\mathcal{H}}) = (\mathcal{T}^{(t)}, \boldsymbol{\mathcal{H}})$.

In practice, changing the state of intermediate sequences has an insignificant effect on the estimates of the mutation process parameters and the root, but it adds to the computational cost. Specifically, the probability the mutation events along a branch of known intermediate

sequences $X^j, j = 1, \ldots, l_s$ is equal to

$$\prod_{j=1}^{l_s} \mathbb{P}(l\text{th site of haplotype } h_{X^j} \text{ mutates}) = \frac{|h_{X^j}|}{N_t} \frac{\phi_l q_{X_l^j}}{\left((N_t - 1)/2 + \frac{1}{N_t}\sum_{i=1}^{N_t}\sum_{l=1}^{L} \phi_l q_{X_l^i}\right)},$$

using Equation (3.4). Allowing permutations of the mutations leaves the numerators of the sum unchanged, and only changes one of the terms of the sum over l in the denominator. In most datasets, the sequences are reasonably long, so that the sum consists of many terms, and the effect of changing one of them becomes insignificant. For this reason, it is often be preferable to fix an arbitrary choice of intermediate sequences throughout the MCMC iterations.

3.8 Representing the tree

In order to update the tree topology when homoplasy is present, consistent tree representations have to be defined. In this section we describe how subtrees of the haplotype network obtained by Algorithm 3.7 can be represented concisely based on the breaking of loops. We show that, based on an arbitrary set of loops, all possible trees can be obtained by breaking up each one of these loops consecutively, and that there is a one-to-one correspondence between tree topologies and loop breaks. This enables us to use a hashing algorithm to label and store trees (see Appendix D), and allows efficient local moves on the tree space. This approach proves to be much more efficient than generating tree topologies afresh and searching for loops at each iteration.

The first task here is to construct an algorithm that identifies loops in the haplotype network. Any tree of N_h nodes (including all internal nodes and leaves) has $N_h - 1$ edges. Therefore, a network of N_h nodes with $N_h + j - 1$ edges has j redundant edges, all of which are part of at least one loop. Our aim is to identify j loops, so that by removing one edge from each (i.e. breaking each loop), it is possible to obtain every single subtree of the original network. We show that this is always possible based on a set of j loops (which may not be unique). The algorithm we propose here for identifying loops is shown below.

Algorithm 3.8.1.

1. Set $i = 1$.

2. Starting with node i, check whether there is a path starting at i and returning to i so that no edge is covered more than once. If there is no such path, start again with node $i + 1$, else find the minimum of all such paths and go to Step 2.

3. Store this loop and break it by removing one of its edges. Return to Step 1.

Lemma 3.8.2. *The above Algorithm 3.8 identifies exactly j distinct loops. Furthemore, by removing one of the edges of each loop, we can obtain all subtrees of the original network.*

Proof. See Appendix B.

In fact, we can prove that there is a unique representation of deleted edges to loops to which they belong. In other words, there is a unique correspondence between the set of deleted edges and a respective set of loops which were broken by each edge. In order to prove this, we first prove the following result.

Theorem 3.8.3. *In order to obtain a tree topology by removing edges which belong to loops, at least one edge which only belongs to one loop has to be deleted.*

Proof. See Appendix B.

We can now prove the uniqueness of edge-loop correspondence.

Lemma 3.8.4. *For each tree topology, there is a one-to-one representation of edges deleted to loops to which they belong.*

Proof. See Appendix B.

Lemma 3.8.4 implies that all tree topologies can be represented uniquely by a vector of j integers, meaning the edges removed from each of the j loops. This allows us to define an efficient local update on the tree space and to use a hashing algorithm to keep track of the tree topologies. Note that, although all tree topologies can be represented by loop breaks, not all loop breaks define a tree topology, but some may result in two or more disconnected networks.

3.9 Updating the tree topology

We may update the tree topology locally, by picking one of the loops at random, and removing one of its edges at random. Removing an edge at random does not necessarily result in a tree, but may yield two disconnected networks. To avoid having to calculate the normalization constant associated with the number of possible tree topologies that can be obtained at each iteration, we allow all moves, and treat the resulting disconnected networks as having likelihood zero.

We now present the MCMC update for the tree topology.

E6a Propose a new tree topology T' by choosing one of the loops at random and proposing to change its deleted edge. The new haplotype tree becomes $\mathcal{T}' = (T', \tau)$.

E6b Propose $\boldsymbol{\mathcal{H}}' = \{\mathcal{H}'_1, \ldots, \mathcal{H}'_J\}$ according to $q(\mathcal{H} \,|\, \mathcal{T}, r)$.

E6c Accept the proposed move with probability $\min(A_T, 1)$, where

$$A_T \;=\; \frac{\hat{\mathbb{P}}_{\mathcal{H}'}(r, T', \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v} \,|\, \boldsymbol{X})}{\hat{\mathbb{P}}_{\mathcal{H}}(r, T, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v} \,|\, \boldsymbol{X})}.$$

If we accept, set $(\mathcal{T}^{(t+1)}, \boldsymbol{\mathcal{H}}) = (\mathcal{T}', \boldsymbol{\mathcal{H}}')$, otherwise we set $(\mathcal{T}^{(t+1)}, \boldsymbol{\mathcal{H}}) = (\mathcal{T}^{(t)}, \boldsymbol{\mathcal{H}})$.

Updating the tree topology by changing the breaking of one loop does not trivially guarantee irreducibility.

Lemma 3.9.1. *The local proposal described above is irreducible.*

Proof. See Appendix B.

We remark here that the tree topology can have a strong dependence on the mutation rates $\boldsymbol{\phi}$. Specifically, if all the ϕs are assumed to be equal, the algorithm will tend to yield a tree of minimum length. This is because, if the ϕs are equal, all mutations have approximately equal probabilities (depending, of course, on individual nucleotide sites). As a result, effectively all edges on the haplotype tree will have approximately similar weight on the total probability of the tree, and hence the minimum number of mutations will yield the highest posterior estimate.

Example

The complete algorithm to draw inferences about the haplotype tree, root haplotype and mutation parameters then follows steps E1-E6. We generated 25 datasets and applied the algorithm described, showing that in all cases, the posterior probabilities of all possible trees of the same size are approximately equal, and only one of the 25 yielding a Bayes factor > 2. We will see in the next section that the probabilities of trees are strongly dominated by corresponding phenotypic or geographical information. As a result, in those cases it is beneficial to assume a uniform distribution on trees irrespective of size, and allow the phenotypic/geographical measurements to determine the posterior probabilities of trees.

3.10 The complete clustering algorithm

We have now proposed updates for all parameters of the clustering and haplotype tree inference. The hierarchical structure of the parameters is summarized in the DAG of Figure 3.9. In order to infer the joint distribution of the clustering and the rooted haplotype tree, we construct a MCMC sampler with target distribution

$$\pi(K, \boldsymbol{e}, \boldsymbol{z}, \gamma, \Sigma, \boldsymbol{\mu}, \mathcal{T}, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v} \,|\, \mathcal{Y})$$

in the phenotypic case, and

$$\pi(K, \boldsymbol{s}, \boldsymbol{c}, \gamma, \Sigma, \boldsymbol{\mu}, \mathcal{T}, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v} \,|\, \mathcal{Y})$$

in the phylogeographic case.

Almost all parameters can be updated independently as described in previous sections, with only exception the tree topology. Clusterings are only defined within a tree topology and a certain clustering can be impossible under a different tree topology. There are a few options when updating the tree topology and clustering. The tree topology can either be updated locally or globally, as can the clustering. We describe a few different approaches.

- Loops and clustering are updated globally and simultaneously. This ensures no normalization constant, but leads to an inefficient chain.
- Loops are updated locally and the clustering globally, simultaneously. This implies that any tree topology is permissible, and the number of possible clusterings conditional on the tree remains constant (allowing empty clusters as described previously).
- Loops and clusterings are updated locally, simultaneously. In this case a normalization constant appears. The new tree has to contain all the edges which are assumed to be significant, otherwise one of the mutations assumed to be significant is invalid.
- Loops and significant edges are updated separately, either locally or globally. The disadvantage of this approach is that, for the same set of significant edges, alterations to the tree leads to a different clustering. This implies that, in effect, although the same mutations are assumed to be significant, the clustering is forced to be updated simultaneously.

Here we update tree topology together together with the edge set $\boldsymbol{e}$ or migrating haplotype construction $(\boldsymbol{s}, \boldsymbol{c})$. In other words, the clustering and tree topology updates are merged into one update in the total algorithm.

T1a Propose a new tree topology T' using the local proposal kernel described in Section 3.9.

T1b Propose a new clustering as well as cluster means and covariance matrices, following Steps B1-B5 or C1-C3 accordingly.

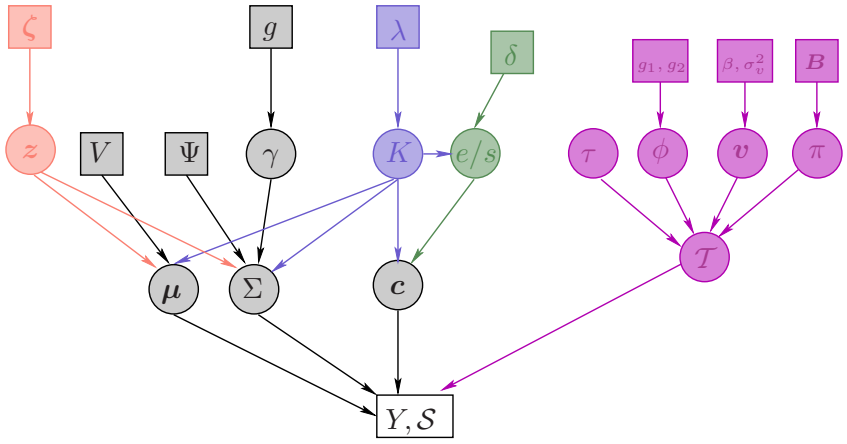


Figure 3.9: *The DAG showing the combined parameters of the phenotypic/phylogeographic clustering and the haplotype tree inference.*

T1c Accept the proposed move with probability $\alpha = \min(1, A)$, where

$$A = A_T \times A_B,$$

for phenotypic clustering, and

$$A = A_T \times A_C,$$

for phylogeographic clustering.

T2 Update each the parameters apart from the tree topology independently.

A full run through all the steps described is called a *sweep*. Using the same arguments as in the individual updates, it can be checked that despite the fact that in this case the acceptance probability is often equal to zero, the chain remains irreducible (and aperiodic).

The complete algorithm, both for phenotypic and phylogeographic clustering is implemented through an R package presented in Appendix E.

Phenotypic example

Using the simulated dataset S5, we generate a phenotypic effect with two significant mutations. The algorithm correctly identifies the number of significant mutations K and the MAP clustering. In addition, a significant improvement is achieved on tree topology inference. We discussed in an earlier example that the haplotype tree inference was frequently unable to identify the correct tree topology, because probabilities on trees all have very similar sizes. However, combining the tree inference with the phenotypic data, the accuracy of the

prediction was improved. Specifically, when the true tree was uniquely consistent with the phenotypic clustering, it was correctly identified. In cases where more than one tree were consistent with the phenotype, they all yielded a uniform posterior probability on trees, as expected.

3.11 Ancestral locations in phylogeographic analysis

One of the objectives of phylogeographic analysis is to identify the location where a population originated from. Although the analysis presented here does not assume a geographical model for the spread of populations in time, we are still able to calculate probabilities of root haplotypes belonging to specific locations. In addition, even if the root haplotype is extinct or has not been sampled, it is still possible to infer the oldest haplotypes within in our sample. This may be better understood through Figure 3.10.

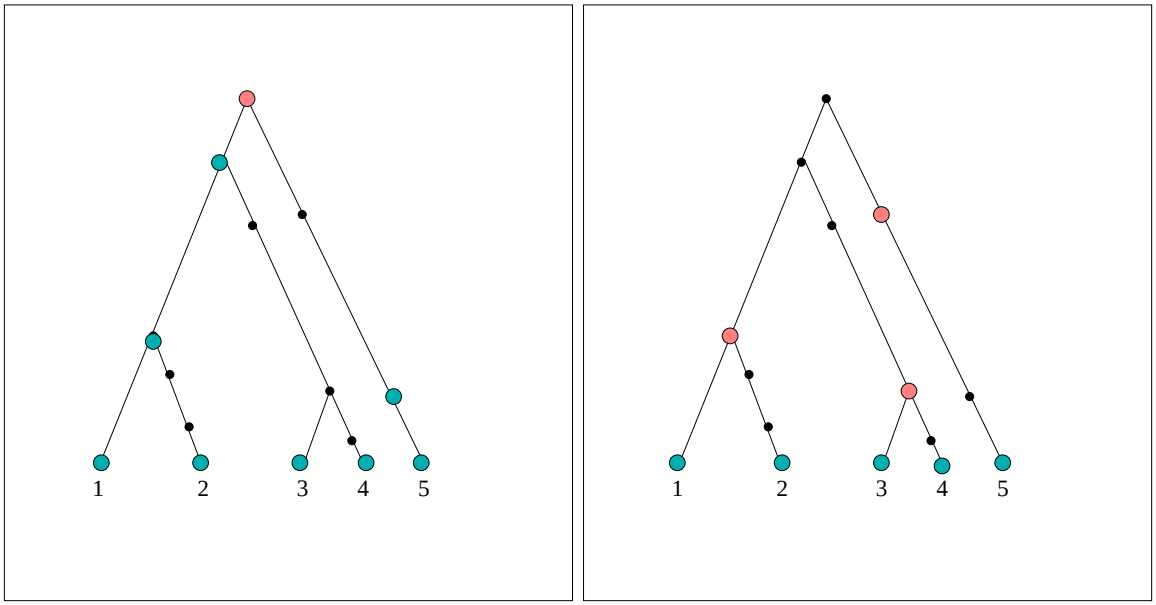


Figure 3.10: *Two possible genealogy scenarios, where coloured points represent observed haplotypes (with the colour representing the location), whereas small black circles are unsampled. In the figure on the left, the oldest haplotype is the pink one at the top. In the figure on the right, the oldest haplotype is missing, and the next possible descendants are the three pink haplotypes.*

Although the root haplotype may be missing like the right-hand panel of Figure 3.10, we know that the location where it originally belonged will, on average, contain haplotypes from all possible descendant branches. As a result, if a location contains all three pink haplotypes, it is more likely to be the ancestral than a location which only contains haplotype from one of

Root haplotype	1	2	3	4	5	6	7	8	9	10	11
Frequency	5	2	5	3	2	1	0	1	0	1	0
Ancestral location	1	2	3	4	5	6	7	8	9	10	11
Frequency	14	2	2	1	0	0	0	1	0	0	0

Table 3.2: *The results of the MCMC algorithm on the 20 simulated datasets S6. The haplotypes are labelled according to their temporal order (with haplotype 1 being the ancestral haplotype).*

the branches. This approach is consistent with many descriptive characteristics of an ancestral area, for example as presented by Emerson and Hewitt (2005).

Hence, at each iteration, we calculate a measure of the probability of each location being the ancestral location by finding the oldest haplotypes along each descendant branch of the root (if the root haplotype is observed in our sample, then we simply have the root only), and then for each location we add the contribution of each of those haplotypes, scaled by the number of times each haplotype is observed and by the number of total haplotypes found.

Example

Using the same simulated dataset S5, we generate locations for each sequence. We fix two population clusters, with means and covariances

$$\mu_1 = \begin{pmatrix} -2 \\ 0 \\ 0 \end{pmatrix} \quad , \quad \Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 10 \end{pmatrix},$$

$$\mu_2 = \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} \quad , \quad \Sigma_2 = \begin{pmatrix} 1 & 0 \\ 0 & 10 \end{pmatrix}.$$

For each split and mutation, the new sequence stays in the same location as its ancestor with probability 0.80, and creates a new location (with the same mean and covariance matrix, representing a local migration) with probability 0.20. We randomly pick one of the local migration events to define a migration which founds the second population cluster, and continue in the same manner.

The analysis identifies the ancestral location(s) in 15 out of 20 datasets. Indeed, this shows a significant improvement of the accuracy to the inference of the root haplotype of the same datasets.

3.12 Combining phenotypic and phylogeographic data

In some cases, both the geographical location as well as the phenotypic measurement of each individual are available. Although the geographical and phenotypic clustering may be different, since e.g. a colonisation event would be irrelevant to a mutation occurring, both can provide useful information about which mutation history is most likely.

In order to analyse such data, we combine the methods described above: we consider two separate clusterings, one corresponding to the geographical data, and one corresponding to the phenotypic data, both of which are consistent with a single mutation history at each iteration. The joint posterior distribution then provides us with the MAP estimate of the mutation history given both types of data.

This may be easily extended so that a separate clustering may be considered for, say, each dimension of multi-dimensional data. If a priori there is reason to believe that two dimensions would be uncorrelated in terms of which mutations would cause a significant effect, then two different clusterings can be inferred throughout the analysis.

Chapter 4

Data Analysis

We now implement the phenotypic and phylogeographic clustering methods on three datasets, and compare the results to existing analyses. In Section 4.1 we analyze a phylogeographic dataset of mitochondrial DNA taken from beetles on the island of La Palma in the Canaries. We first briefly present the results of standard NCPA as described by Emerson and Oromi (2005), and then apply the Bayesian approach developed in this thesis, comparing the outcomes. In Section 4.2, we repeat the above procedure for a weevil dataset taken from the Iberian peninsula. Finally, in Section 4.3 we use a phenotypic salmon sperm dataset where mitochondrial DNA is investigated for associations with several quantitative traits associated with sperm motility and longevity. We present the results of our clustering algorithm for multi-dimensional data, and discuss the challenges.

4.1 The beetle dataset

We use data from the geologically young and well-characterised island of La Palma from within the Canary Islands archipelago to generate phylogeographic predictions for *Brachyderes rugatus rugatus*, a flightless curculionid beetle species occurring throughout the island in the forests of *Pinus canariensis*. We have a sample of 135 beetles from 18 localities across the distribution of *B. R. Rugatus* for 570 base pairs (bp) of sequence data for the mtDNA cytochrome oxidase II (COII) gene. The data are summarised in Figure 4.1 which superimposes the sampling locations on a map of La Palma together with forest density. At each location the number of distinct haplotypes observed is recorded, with 69 distinct haplotypes observed in all. See Emerson *et al.* (2000,2006).

Geological studies of the island provide us with a fairly complete understanding of the island's geological history. The northern part of the island is mainly older volcanic terrain with the southern part comprising a ridge of more recent volcanic origin. It is a reasonable

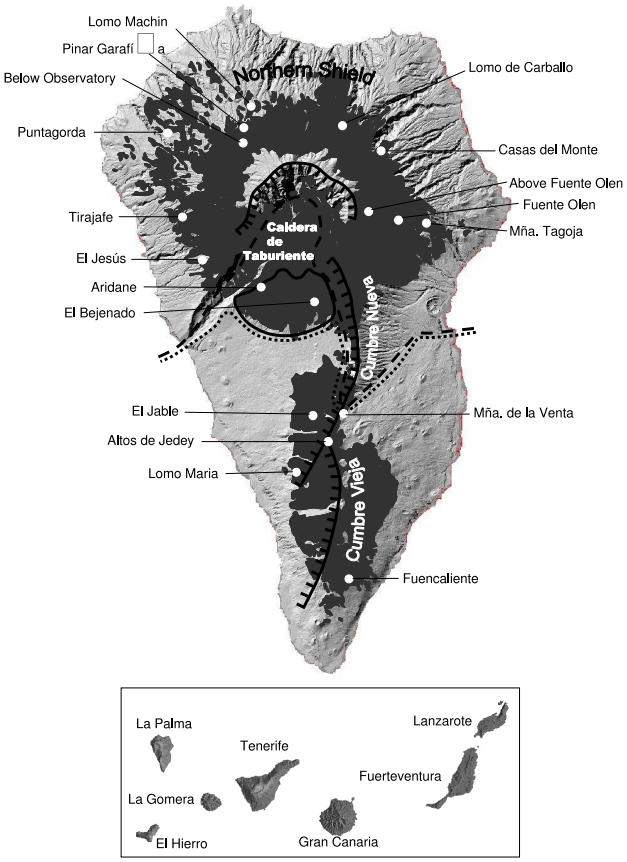


Figure 4.1: *Map of the sampling locations on the island of La Palma, taken from Emerson and Oromi (2005)*

assumption that the *Brachyderes* beetle population, with their limited mobility, would have been strongly influenced by La Palma's volcanic and erosional history and there is also strong evidence that the population was seeded by immigration from the nearby island of Tenerife to the east. Thus we might expect the oldest haplotypes to be concentrated in the northern part of the island and to observe evidence of a more recent range expansion to the southern tip. If we were to cluster the haplotypes geographically we should therefore find that those to the north should be more central to the haplotype network, and those to the south should be placed towards the tips of the network.

4.1.1 Nested Clade Phylogeographic Analysis

We present the results of the analysis using NCPA, as described by Emerson and Oromi (2005). Emerson and Oromi used TCS to infer the haplotype tree. Eight loops were formed, almost all of which were resolved by the criteria of Templeton et al. (1992) and Crandall and

Templeton (1993), resulting in three possible trees (see Figure 4.2) with equal probability based on the sequence data alone. Taking into account the geographical information as well, Emerson and Oromi (2005) point out that it is more feasible that the population around area 2 colonized into 1 and 3 respectively (corresponding to tree A in Figure 4.2), rather than that 3 colonized into 1 and 2 separately as in tree B of Figure 4.2 (and similarly for 1 colonizing into 2 and 3 in tree C). This is because, for example, tree B would imply that phylogroup 3 colonized into 1 through, but not including, the already inhabited area of 2. Combining all the criteria and information described, tree A of Figure 4.2 is chosen as the one which, based on the methods of Templeton et. al., explains the most phylogeographic information included in the data.

The chosen tree was then nested using the methods described by Templeton et al. (1987) and Templeton and Sing (1993), with the resulting nesting design shown in Figure 4.3.

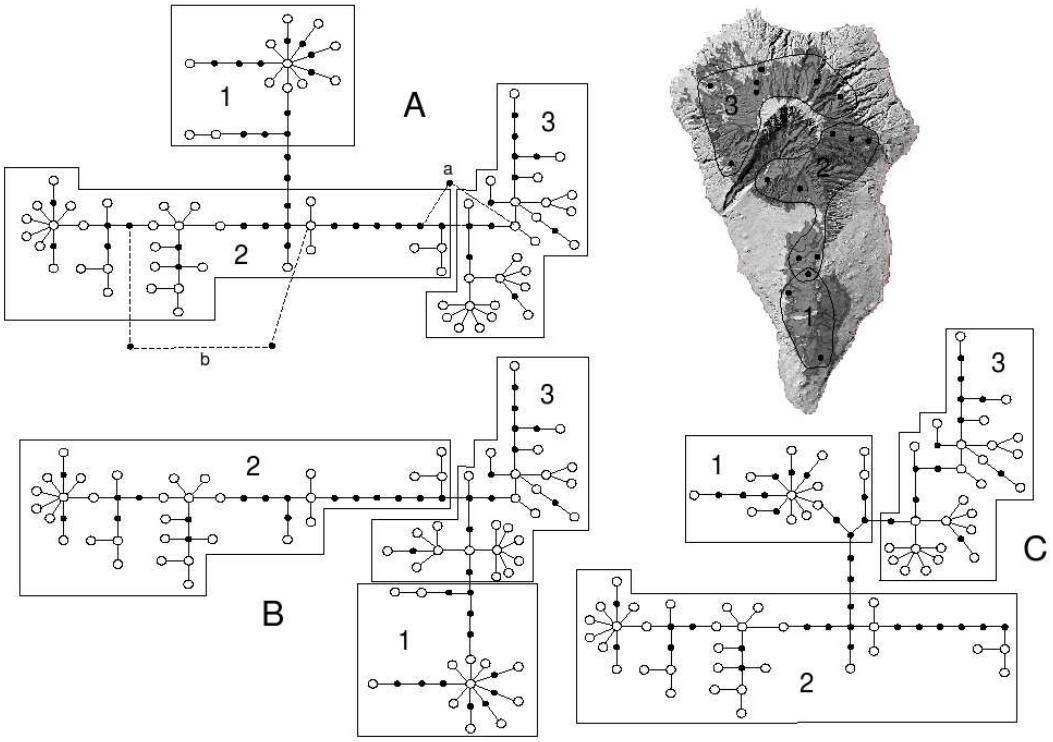


Figure 4.2: *The three possible haplotype trees for the beetle data, taken from Emerson and Oromi (2005), related to the map.*

The nested clades were then tested for significance using Nested Analysis of Variance (NANOVA). The results of the analysis are shown in Table 4.1.

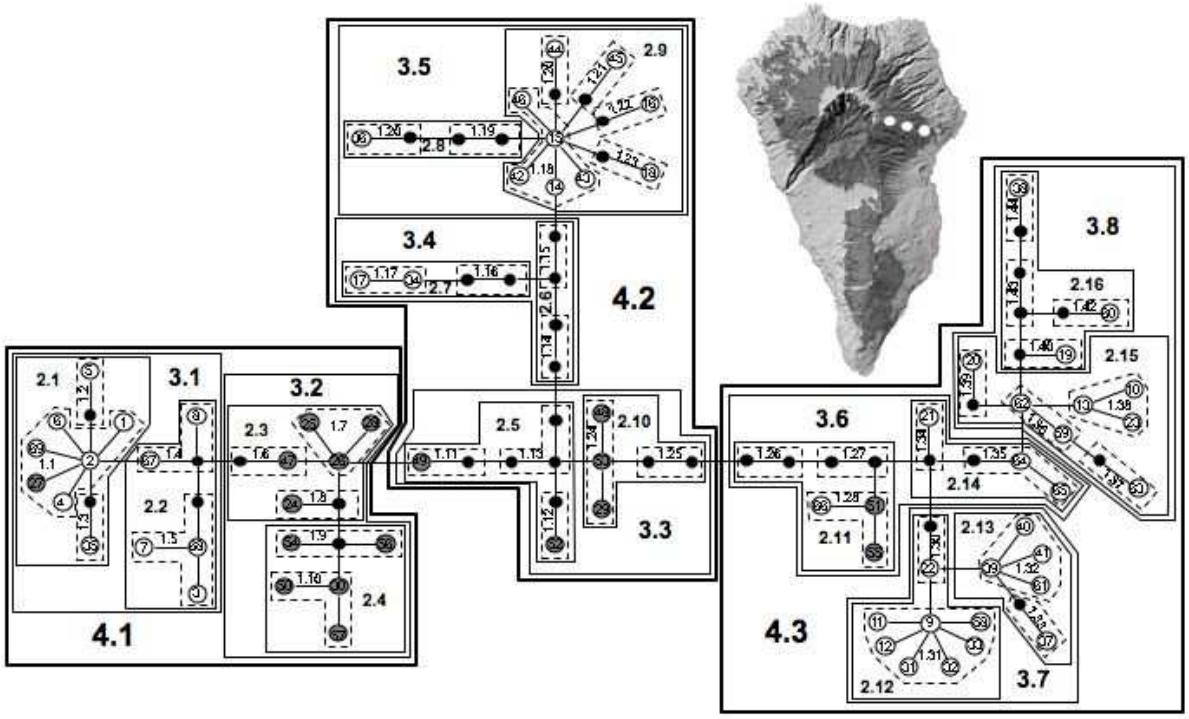


Figure 4.3: *The inferred haplotype tree of the beetle data using TCS together with the criteria described by Templeton (1998), taken from Emerson and Oromi (2005). It corresponds to tree A of Figure 4.2, and was nested using the algorithm and criteria described by Templeton et al. (1987) and Templeton and Sing (1993).*

Ancestral areas

In order to identify the root of the tree, Emerson and Oromi (2005) first investigated the existence of haplotypes which satisfy the empirical predictions of Crandall and Templeton (1993) and Posada and Crandall (2001) based on coalescent theory. Crandall and Templeton (1993) observed that root haplotypes on average appear at high frequency, occur in the greatest number of populations, have multiple connections with singletons (i.e., haplotypes which were observed only once), and are located at the interior of a network. However, none of the haplotypes in the inferred tree satisfied all of those criteria. Instead, a hypothesis involving regional population extinction and recolonization was considered. Although NCPA does not directly test this scenario, individual events which were deduced, together with a large percentage of missing intermediates around the centre of the tree, point towards the extinction of the root haplotype. This was further suggested by the fact that haplotypes occurring around the area Olen, which is predicted to be the ancestral area before the analysis, shows several haplotypes all around the central interior area of the tree. In fact, the three locations Montana Tagoja, Fuente Olen and above Fuente Olen collectively contain almost

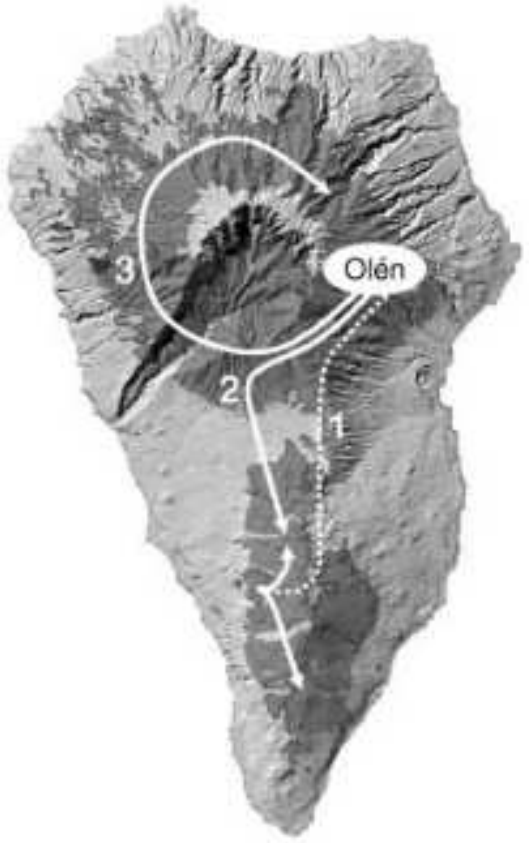


Figure 4.4: *The proposed colonization scenario obtained by interpreting the output of NCPA; the Figure is taken from Emerson and Oromi (2005). The island was colonized from the East. The North of the island was colonized in a clockwise fashion, and two independent migrations occurred to the South.*

all set of connections around clade 3.3 (shown in grey in Figure 4.3). As a result, the eastern flank of the north shield is suggested to be the ancestral area of the island.

4.1.2 Bayesian haplotype tree approach

We implemented the phylogeographic methods described in Chapters 2 and 3 for the beetle dataset. Specifically, we first ran the analysis for an unknown number of clusters in order to infer the MAP estimate for K, the number of migrating haplotypes. We then fixed the number of clusters and repeated the simulations, discussing the output in detail, both in terms of their statistical significance, as well as in relation to the results following the NCPA analysis presented in the previous subsection. For computational efficiency, we assume a uniform distribution on the haplotype tree temporal orderings $\mathcal{H}_j$. The MAP estimate of the haplotype tree, posterior cluster distributions and ancestral areas are discussed carefully. Finally, the

Clade	p-value	Chain of inference	Conclusion
1.1	0.0042*	1-2-11-12 No	contiguous range expansion
2.1	0.0532**	1-2-11-17 No	inconclusive outcome
2.3	0.0473*	1-2	inconclusive outcome
2.14	0.0219*	1-19-20-2	inconclusive outcome
3.2	0.0527**	1-2-11-12 No	contiguous range expansion
3.6	0.0010*	1-19 No	allopatric fragmentation
3.7	0.0000*	1-2-3-4 No	restricted gene flow with isolation by distance
4.1	0.0000*	1-2-11-12-13 Yes	past fragmentation followed by range expansion
4.2	0.0000*	1-2-11-12 No	contiguous range expansion
4.3	0.0000*	1-2-3-5-6-13 Yes	past fragmentation followed by range expansion
Total tree	0.0000*	1-2-11-12 No	contiguous range expansion

Table 4.1: *Results from NCPA NANOVA, taken from Emerson and Oromi (2005).*

# of clusters	1	2	3	4	5	6
post. model prob.	0.00	0.00	0.00	0.00	1.00	0.00

Table 4.2: *The posterior masses for the number of clusters. The existence of six clusters is suggested, showing the highest posterior mass of 1.00. Due to the different clustering construction used in order to describe phylogeographic predictions, these probabilities differ from the corresponding ones inferred by Brooks et al. (2007).*

convergence of the chains is assessed. A simplified version of the Bayesian approach described here is presented in Brooks et al. (2007), using phenotypic rather than phylogeographic clustering.

We ran the analysis on the phylogeographic data for an unknown number of clusters, taking $d_s = 2$; this resulted in nine loops. The posterior masses for the number of clusters are shown below in Table 4.2.

For the six-cluster model, the posterior mode for the tree is shown in Figure 4.5, with the posterior mean cluster of each haplotype superimposed as the colour. Figure 4.6 shows a geographical contour plot of the inferred posterior distributions of the clusters. Comparing the MAP estimate for the tree with the results of the analysis following TCS, we see that almost all connections are identical, with the exceptions of the branch 53-29-48 and branch 51-55-68. The former is placed around the same area within the tree as before, whereas the latter is moved from the interior of the tree to being a leaf. It is worth pointing out here that the tree deduced from TCS is indeed one of the trees in the set Ω fixed by the data using the Bayesian approach.

The ancestral locations predicted by Emerson and Oromi (2005) agree with the results of our analysis, shown in Table 4.3. Specifically, the most likely ancestral areas, all belonging to

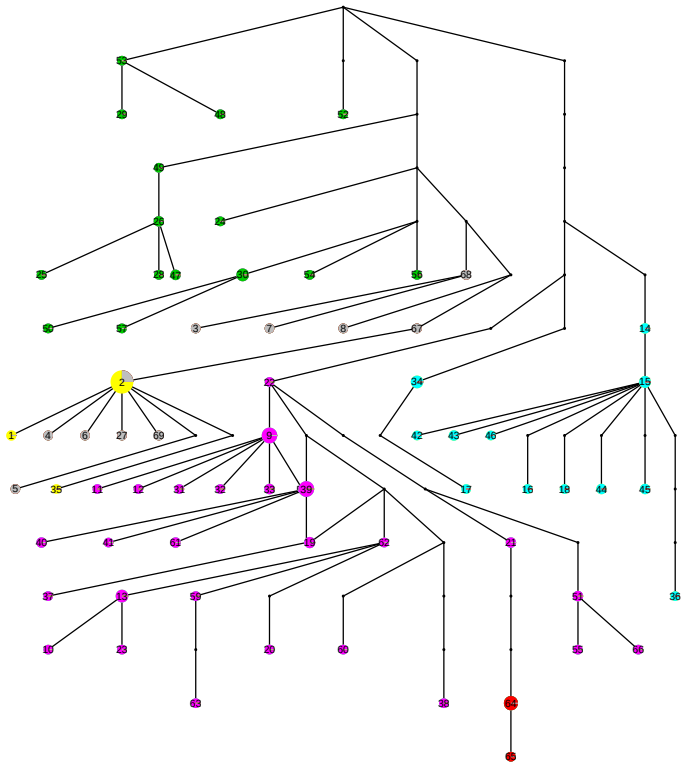


Figure 4.5: *The MAP estimate of the haplotype tree for the beetle dataset, where colour corresponds to cluster and size to the number of individuals sampled with each sequence. We notice that all migrating haplotypes appear to be extinct or unsampled.*

the green cluster, indeed agree with the prediction of La Palma having been colonized from the east. The fourth most likely location, with probability around 0.10, is located in the midwest of the island, representing the secondary colonization which occurred in the south of the island shown in Figure 4.4.

Convergence diagnostics

In this case, the clustering almost never changes after burn-in. As a result, both the number of clusters K and the clustering $\boldsymbol{c}$ show minimal mixing after convergence is reached. This is because, in phylogeographic clustering, local moves are not always possible. The "most local" move is usually one where only one of the datapoints of a migrating haplotype is moved to another cluster. In this case, we notice that almost all the migrating haplotypes are, in fact, inferred to be extinct. This implies that no datapoints can be moved, and the clustering is

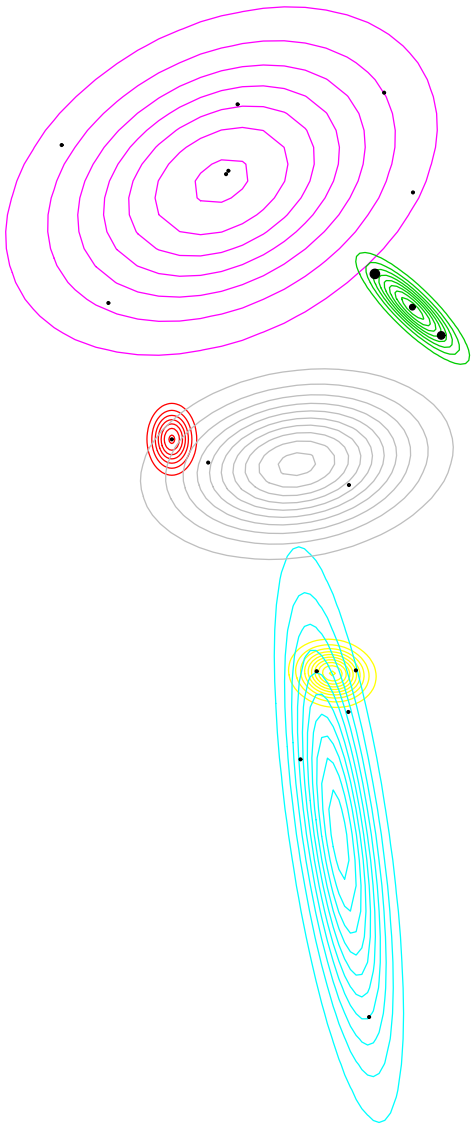


Figure 4.6: *Correponding bivariate normal contour plots for the beetle dataset evaluated at the posterior means. The haplotype numbers correspond to the previous Figure 4.3 obtained using NCA. The circles indicate sampling locations, with the larger circle indicating the location most likely to include the root node.*

not altered by such a local move. In order to assess the output of the MCMC algorithm, we start the chain from 10 different and over-dispersed clusterings, and confirm that the same unique clustering $\boldsymbol{c}$ is reached.

We show trace and density plots of a few representative clustering parameters (see Figure 4.7), suggesting convergence of the parameters.

Finally we investigate the convergence of the tree topology by comparing the posterior

location	posterior mass
Above Fuente de Olen	0.30
Montaña Tagoja	0.13
Fuente de Olen	0.13
Montaña de la Venta	0.11

Table 4.3: *Posterior ancestral probabilities of the top four sampling locations of the beetle data.*

probabilities of tree topologies for two different chains with different starting points, which all yield trivial rearrangements of the same tree. Although the MAP estimate of the clustering is unique (subject to trivial rearrangements), the MAP estimate for the tree topology is different depending on the starting point of the chain, implying that the tree topology has not converged. Instead, any of the topologies which allow the MAP estimate of the clustering to be achieved appear with a high probability. Since, however, both the clustering and the root estimates appear robust under all such topologies, it is sufficient to assume that they are all equiprobable a posteriori.

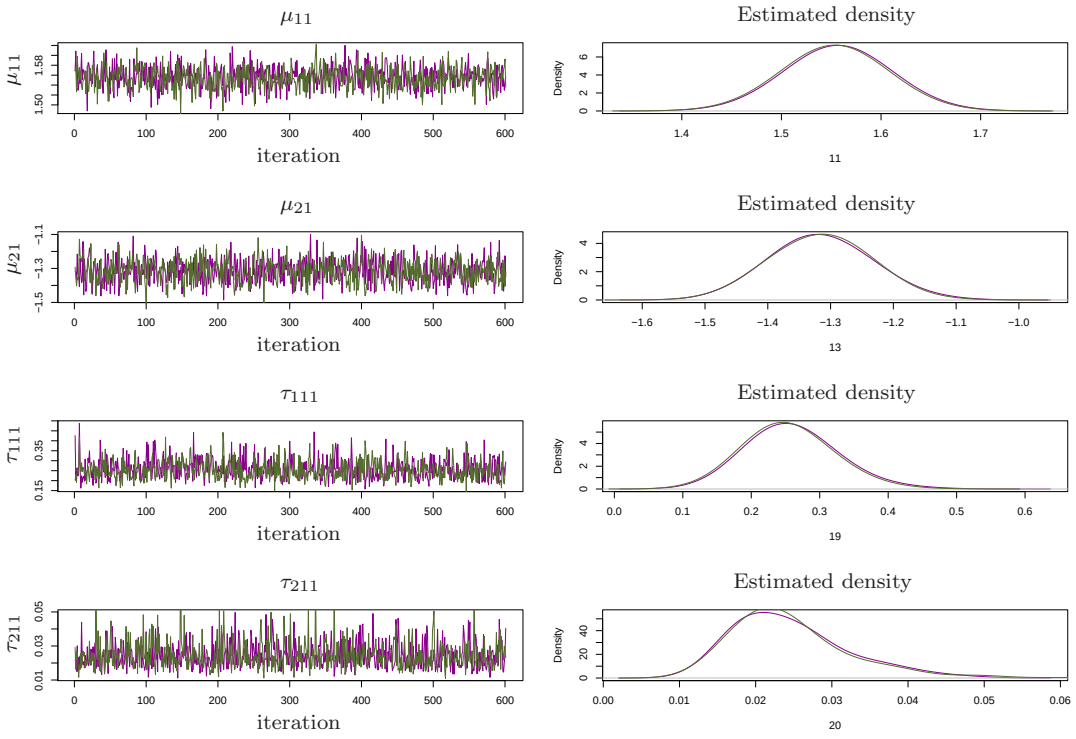


Figure 4.7: *Trace and density plots for μ and Σ for the beetle dataset, showing very good mixing and matching posterior densities. Here the parameters correspond to the same clustering throughout, since the clustering stays constant.*

4.2 The weevil dataset

Rhinusa vestita is a seed parasite weevil feeding and reproducing on snapdragons. It is believed to have been present in Portugal, Spain, France and Italy (see Legarreta et al., 2008). [1] The complete nucleotide sequence for the mitochondrial COII gene (722 bp) was obtained for 275 *Rhinusa vestita* individuals. Below is a map of the localities (see Figure 4.8).

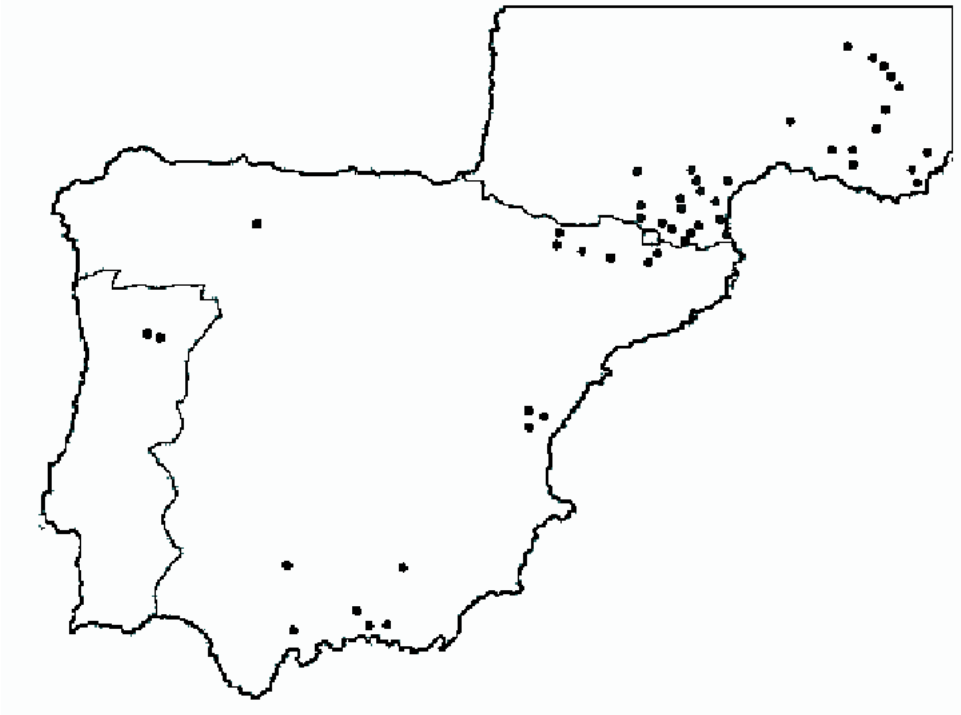


Figure 4.8: *Map of the sampling locations of the R. vestita dataset.*

Previous studies investigating the host association of weevils with three host plant species, combined with knowledge about the glaciation history of the iberian peninsula (see Hewitt, 2000), led to the biological prediction that the species originated from the Rhône valley to the east and west (see Legarreta et al., 2008).

4.2.1 Nested Clade Phylogeographic Analysis

A total of 74 haplotypes were revealed with 75 variable sites (10%), 46 of which were parsimony informative (61%). A combination of phylogeographic and population genetic analyses were used to understand the population history of *R. vestita*. The haplotype tree shown in Figure 4.9 was formed using TCS and the criteria of Crandall and Templeton (1993) to resolve

[1] This is a paper in preparation. The NCPA analysis and biological predictions were carried out by my collaborators, Dr L. Legarreta and Dr B. C. Emerson, whereas the MCMC approach was implemented by me.

loops. The NCPA results for the nesting clades where significant geographical association of haplotypes was found are shown in Table 4.4.

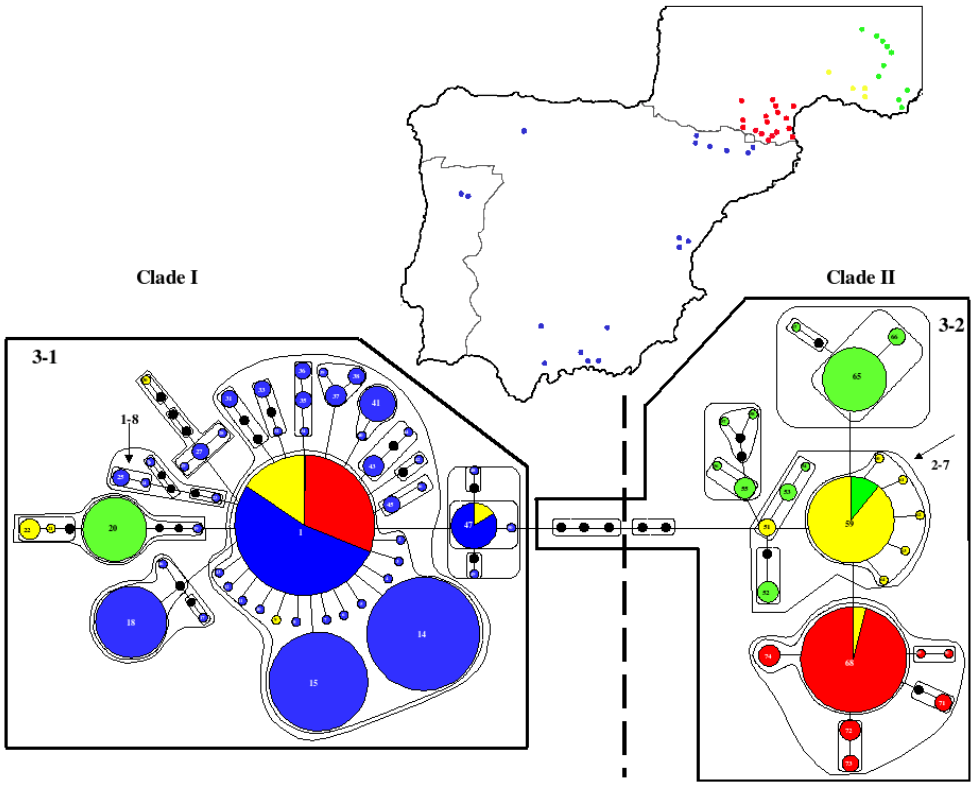


Figure 4.9: *The inferred haplotype network for the R. vestita data using TCS. Here the colours of haplotypes correpond to the locations shown in the map.*

The highly divergent sequences of related Rhinusa species meant that the tree could not be rooted using an outgroup. As with the beetle dataset, the empirical predictions of Crandall and Templeton (1993); Posada and Crandall (2001) from the coalescent were applied to infer the root haplotype. In this case, haplotype 2 satisfied the usual criteria of being the most frequent, broadly distributed geographically and with many mutational connections. However, as pointed out by Emerson and Oromi (2005), under a model of extinction and re-colonization this prediction may be false, but genealogical and geographical unity of haplotypes can be used to identify ancestral haplotypes and the likely ancestral area. In this case, the Rhône Valley area is consistent with a number of range expansions to the Alps, the Pyrenees and Iberia, placing the ancestral haplotypes as extinct and located around the center of the haplotype tree.

Specifically, haplotypes from the Rhône Valley form an almost continuous connection, and span the length of the network occurring in the two major clades, supporting the inference of

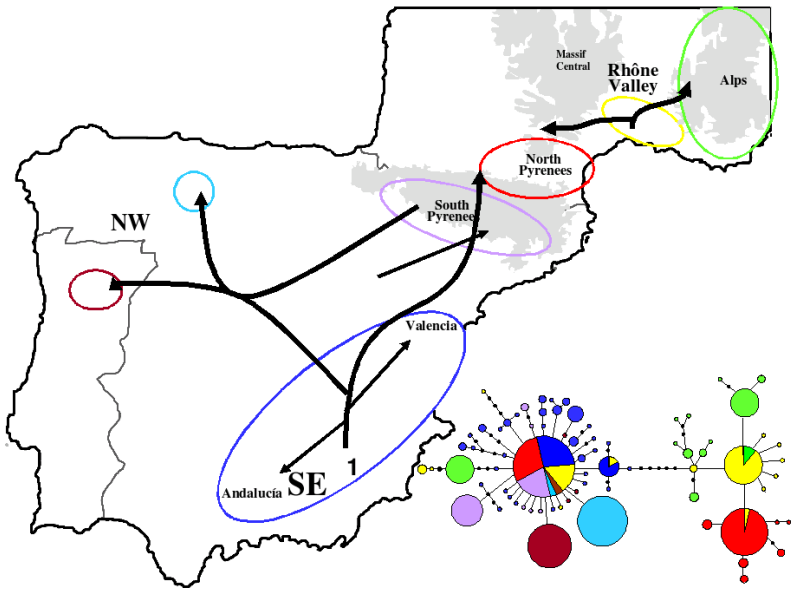


Figure 4.10: *The colonization scenario inferred using GeoDis and biological predictions. Two main ancestral locations were identified, one around the Rhône Valley in France to the East and West, and one around the South of Spain towards the North-West and back into the East.*

the Rhône Valley as an ancestral area.

Regardless of the lack of samples in this area, the star-like pattern derived from haplotype 1 is indicative of a population expansion. Moreover, all of the haplotypes sampled from the vast geographical area to the south and west of the Pyrenees are derived from, and closely related to, haplotype 1 (Figure 4.9), suggesting that mitotypes in the rest of Iberia are perhaps also closely related irrespective of geographical distance.

A refuge area is usually characterized by the fact that the genetic diversity is higher than in areas that were colonised by the refuge (see Avise, 2000), and the Rhône Valley area shows the highest nucleotide diversity (see Legarreta et al., 2008). Additionally, there is a low proportion of haplotypes around the Rhône descended from haplotypes outside the area. Finally, if Rhône were not the primary ancestral area, two long distance colonization events would need to be evoked in order to explain the current distribution, which is a less parsimonious scenario.

When considering the Rhône as the origin of a series of range expansions east and west to colonize the Alps, Pyrenees and Iberia, the distribution of haplotypes in the area of study is clearly understood. Therefore, based on both geographic and genetic arguments, the Rhône Valley is consistent with being a glacial refugial area for *Rhinusa vestita* with subsequent

Clade	Inference
1-1	Unable to discriminate between range expansion, long distance colonisation and past fragmentation
1-8	Restricted gene flow with isolation by distance
2-1	Unable to discriminate between range expansion, long distance colonisation and past fragmentation
2-7	Past fragmentation followed by range expansion
3-1	Unable to discriminate between isolation by distance (short distance movements) and long distance dispersal
3-2	Inconclusive outcome

Table 4.4: *The significant clades of the NANOVA for the weevil dataset.*

# of clusters	1	2	3	4	5	6
post. model prob.	0.00	0.00	0.00	1.00	0.00	0.00

Table 4.5: *The posterior masses for the number of clusters for the weevil dataset. The existence of five clusters is suggested, showing the highest posterior mass of 1.00.*

range expansions as the ice sheet retreated. Estimation of the timings of range expansions provides further support for this.

The two clades for which past processes could be inferred without uncertainty by following the inference key are marked with an arrow in Figure 4.9. Clade 1-8 was inferred to have restricted gene flow with isolation by distance, and Clade 2-7 showed past fragmentation followed by range expansion (Table 4.4).

In terms of individual migration events, haplotypes 20 and 11 are inferred to have migrated into the Alps, explained by past fragmentation and range expansion. In the Iberian peninsula, haplotypes 51, 1 and 2 migrated, with 2 having colonized in different routes to the SW and NW.

4.2.2 Bayesian haplotype tree approach

As with the beetle dataset, we implement our method on the weevil dataset, taking the maximum parsimony level at $d_s = 0$, yielding nine loops. As with the beetle dataset, here we assume a uniform distribution on the temporal orderings $\mathcal{H}_j$ for computational efficiency. In this case convergence proves to be harder to achieve, because of the structure of the data.

We ran the analysis on the phylogeographic data for an unknown number of clusters. The posterior masses for the number of clusters are shown below in Table 4.5.

The results of our method are shown below through a MAP estimate of the haplotype tree (see Figure 4.11) and a geographical contour plot (see Figure 4.12). A first implementation

of the MCMC approach described in this thesis is presented by Manolopoulou et al. (2008).

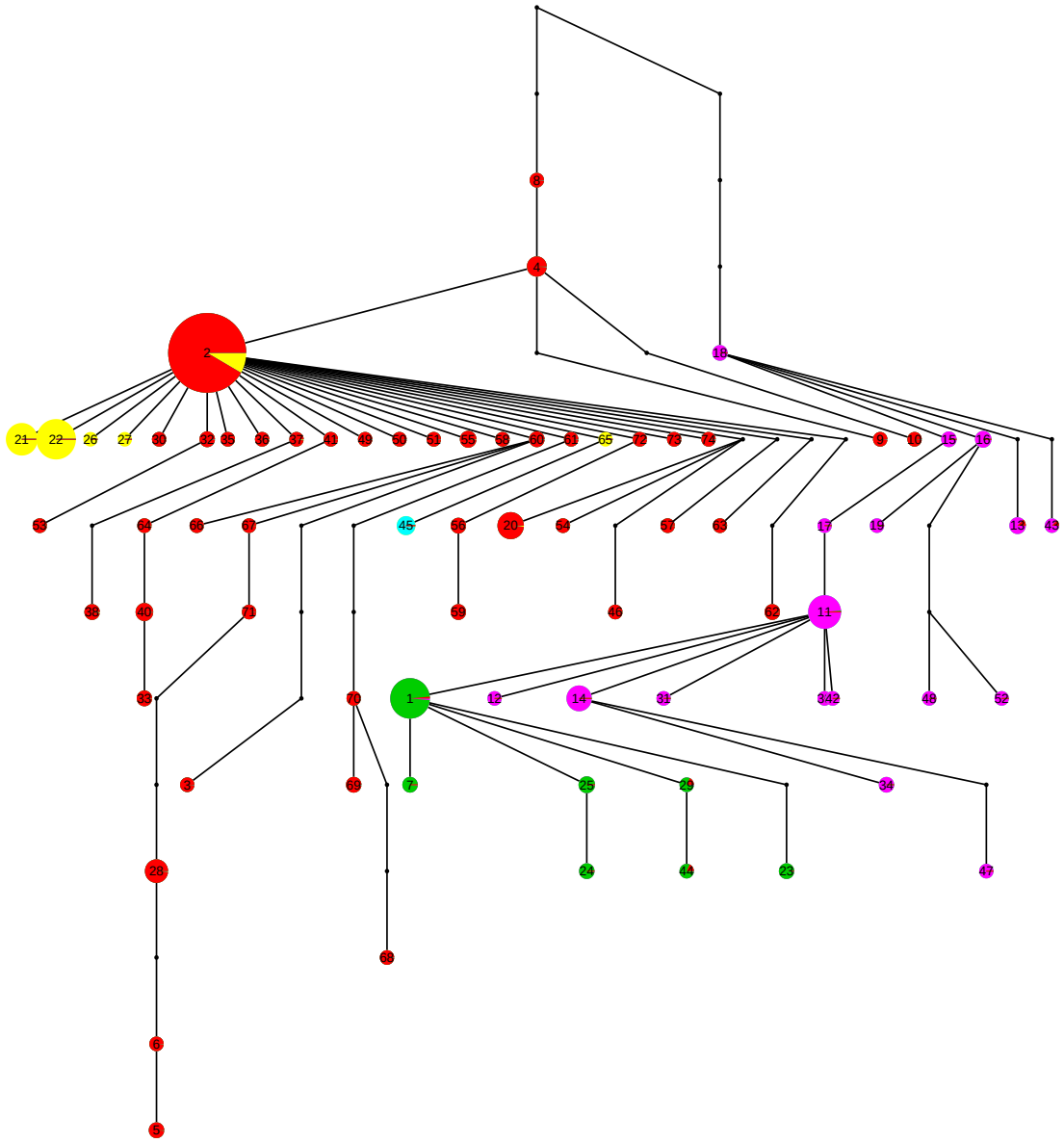


Figure 4.11: *The MAP estimate of the rooted haplotype tree using our approach, where colour corresponds to cluster and size to the number of individuals sampled with each sequence. Here the haplotype numbers correspond to those of figure 4.9. We notice that haplotype 2 is abundant, existing in almost all of the population clusters.*

In this case, the hyperparameter γ became important. Taking different values for Ψ yielded quite difference clusterings for a fixed γ, especially in regard to the NW locations. Allowing γ to vary ensured robustness of the method, increasing the posterior mean for larger

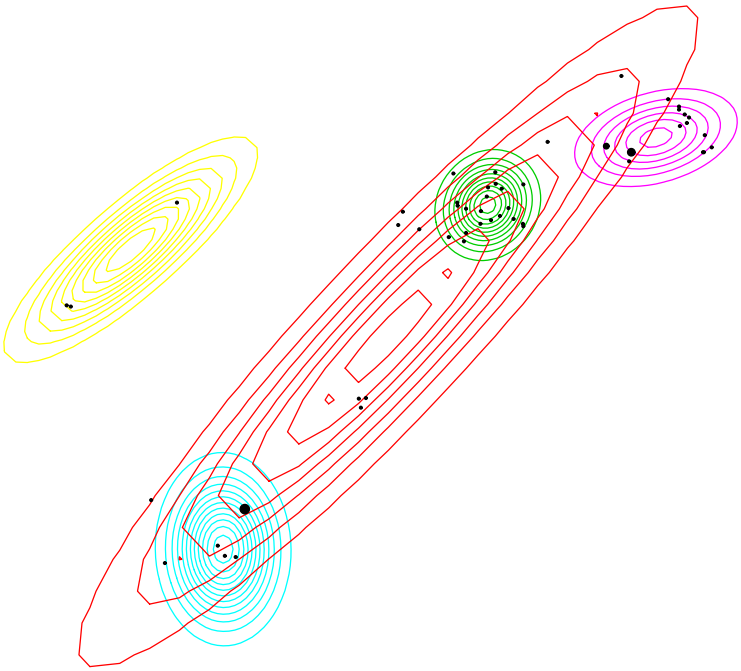


Figure 4.12: *Correponding bivariate normal contour plots evaluated at the posterior means for the weevil dataset. The circles indicate sampling locations, with the larger circle indicating the location most likely to include the root node. The colours correspond to the clusters shown in Figure 4.11.*

values of Ψ.

Two main ancestral locations were identified, as shown by the probabilities in Table 4.6. The first is in the south-west location of Torres, with probability 0.24, whereas the second, third and fourth around the Rhône valley, in Petit Luberon, Brissac and Menton. Taking a closer look at the tree, we observe that the Rhone area locations contain haplotypes from all descendant branches of the inferred root, whereas the south-east location of Torres contains only haplotypes from one of the branches. This indicates that the Rhône valley is the originial ancestral area, whereas Torres corresponds to a secondary ancestral area. It is interesting to see that the results of our analysis match the biological predictions very well. Indeed, the Rhone valley appears to be the original ancestral location, and Torres to be a secondary location of origin. Observing the tree, we confirm that there is evidence of some re-colonization from the French Alps back into the Iberian Peninsula, as demonstrated by haplotypes 1, 7, 23, 24, 25, 29, 44.

location	posterior mass
Torres	0.24
Petit Luberon	0.19
Brissac	0.12
Menton	0.11

Table 4.6: *Posterior ancestral probabilities of the top four sampling locations of the* R. vestita *data.*

Convergence assessment

As before, we present trace plots and convergence diagnostics for the various parameters. In this case the mixing of the clusterings is better: the migrating haplotypes are ones with several datapoints. This implies that it is possible to move between two different clusterings simply by moving one of the datapoints of a migrating haplotype from one cluster to another. Such a move is local enough to be accepted.

As with the beetle dataset, we investigate convergence of the means and covariances by observing the trace plots shown in Figure 4.13, which indicate good mixing. Although the MCMC sampler essentially stops moving once the maximum a posteriori clustering is reached, the same state is achieved from any starting point, indicating that the algorithm has indeed converged.

Finally we investigate the convergence of the tree topology by comparing the posterior probabilities of tree topologies for several different starting points, all yielding tree topologies which allow the MAP clustering shown above.

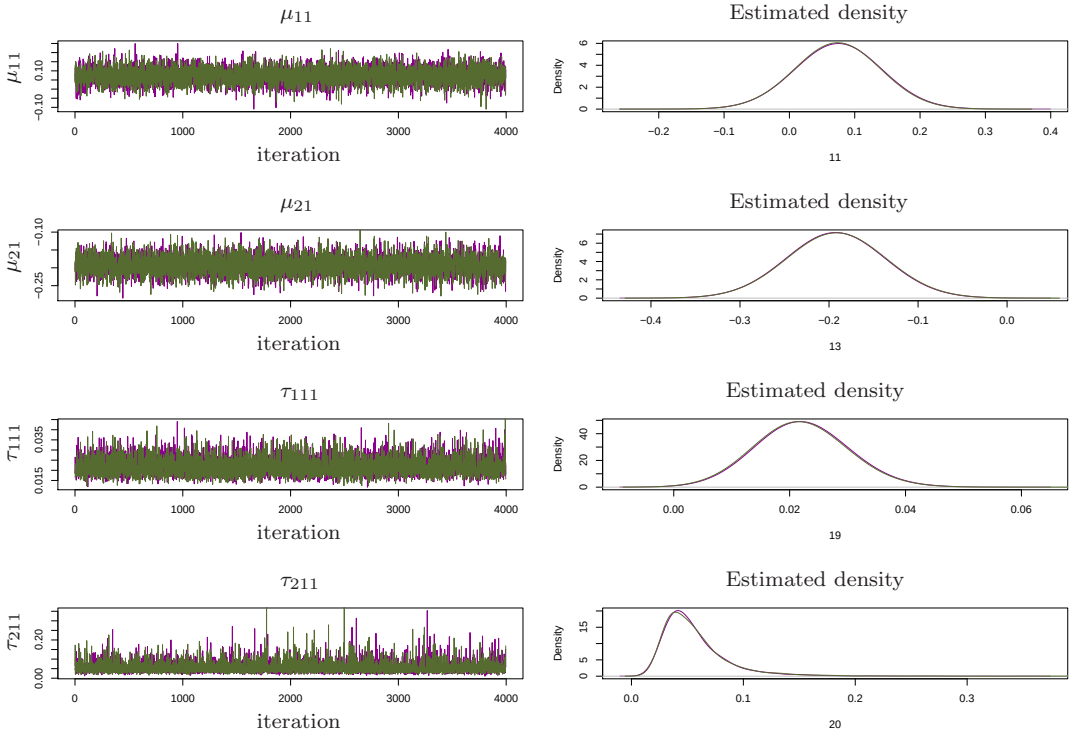


Figure 4.13: *Trace and density plots for μ and Σ of the clusters for the weevil dataset.*

4.3 The salmon dataset

We use a mitochondrial dataset of length 6397 from the NADH genes of 62 New Zealand chinook salmon (*Oncorhynchus tshawytscha*) individuals collected by Gemmell et al. (2006) in the period 2003-2004 in order to investigate associations between mtDNA SNPs and male fecundity. There has been substantial evidence that mtDNA plays a significant role in male fertility; (in humans, see Montiel-Sosa et al., 2002). Here we use a number of measurements associated with sperm longevity, velocity and other sperm characteristics to identify potential associations with nucleotide mutations. For each individual, 16 phenotypic measurements are available [2].

Previous analysis of the data by Gemmell et al. (2006) showed that there exist significant associations between haplotype groups and sperm velocity, and that sperm longevity does not appear to be correlated with mtDNA mutations.

We implemented our method in order to demonstrate the applicability of the algorithm. However, we illustrate a couple of limitations which deem our analysis insufficient, and describe how the algorithm may be extended to account for these shortcomings.

4.3.1 Bayesian haplotype tree approach

We apply the phenotypic clustering approach described in Section 2.2, aiming to identify if any of the measurements show a significant change with a SNP.

After implementing Algorithm 3.7 we immediately observe that there is a large number of back-mutations. In a total of 14 observed haplotypes, using a parsimonious level $d_s = 0$, 9 loops are formed, implying that phylogeny-based inference will inherently be unreliable. For this reason, we consider all possible trees purely on the basis of the clustering fitness, and ignoring the evolutionary model. In other words, we consider the haplotype tree to be a similarity-type tree, without any assumptions about the process generating it.

Referring back to the Model 2.6 for phenotypic data described in Section 2.2, remember that one of the properties of the Inverse Wishart distribution of the covariances is that the larger the degrees of freedom, the smaller the variance of their distribution. In this case, taking all 16 dimensions into account implies that the degrees of freedom are forced to be at least 18. However, such a precise prior is non-sensical: there is no prior information suggesting such strong support for a specific hypothesis for the covariance matrices. As a result, a different prior is required, and here we use the simplified model that the dimensions are independent

[2]The phenotypic measurements available are not described in detail here because the data are yet to be published.

of each other, with prior for the variances

$$\sigma_i^2 \sim \mathcal{IG}(a, b),$$

where $a = b = 10^{-4}$. In addition, we let the maximum number of significant mutations be $K_{\max} = 5$, and we fix the variance for μ to be 1000.

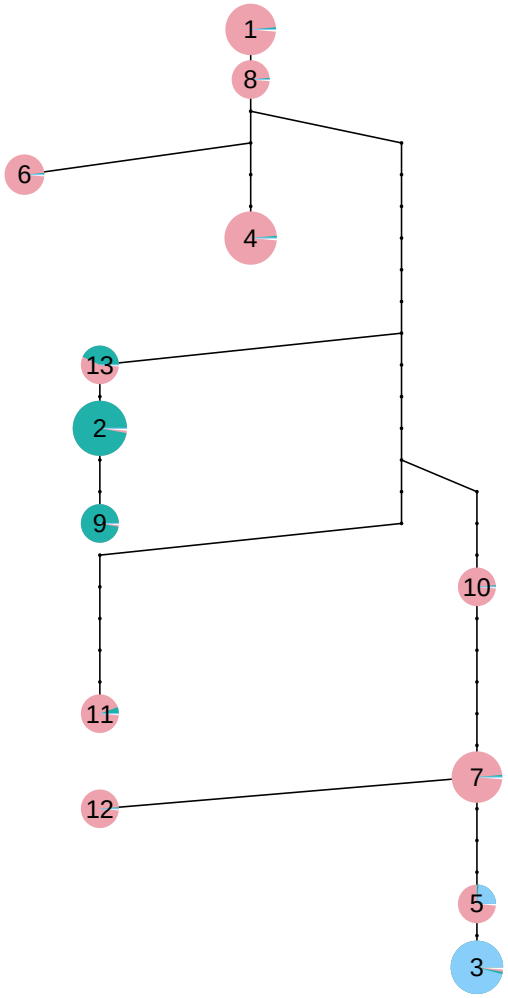


Figure 4.14: *The MAP estimate of the haplotype tree, where colour corresponds to cluster and size to the number of individuals sampled with each sequence*

We first allow the phenotypic measurements which are included in the clustering analysis to vary through $\boldsymbol{z}$ simultaneously with K. Implementing the Reversible-Jump algorithm for variable dimensions of Section 2.2 in combination with the RJMCMC for an unknown number of clusters described in Section 2.4, we obtain that there are only two measurements which

0	1	2	3	4	5
0.00	0.41	0.32	0.18	0.07	0.02

Table 4.7: *The posterior probabilities for the number of significant mutations K. Both the two and the three-cluster model show high posterior mass. Note tht they do not sum to one due to rounding.*

shows the most significant change with a mutation, namely GSI, the gonadosomatic index, and SLOW-PCT, the Slow Post-Coital test, with posterior mass > 0.90. The posterior model probabilities for the number of clusters are shown in Table 4.7.

We we-run the analysis, using only GSI and SLOW-PCT and fixing $K = 2$ so that three clusters are formed. The results are shown in the clustered haplotype tree and density plots below. We see that the three clusters are separated mainly because of a difference in the variance of the measurements within each cluster rather than the mean.

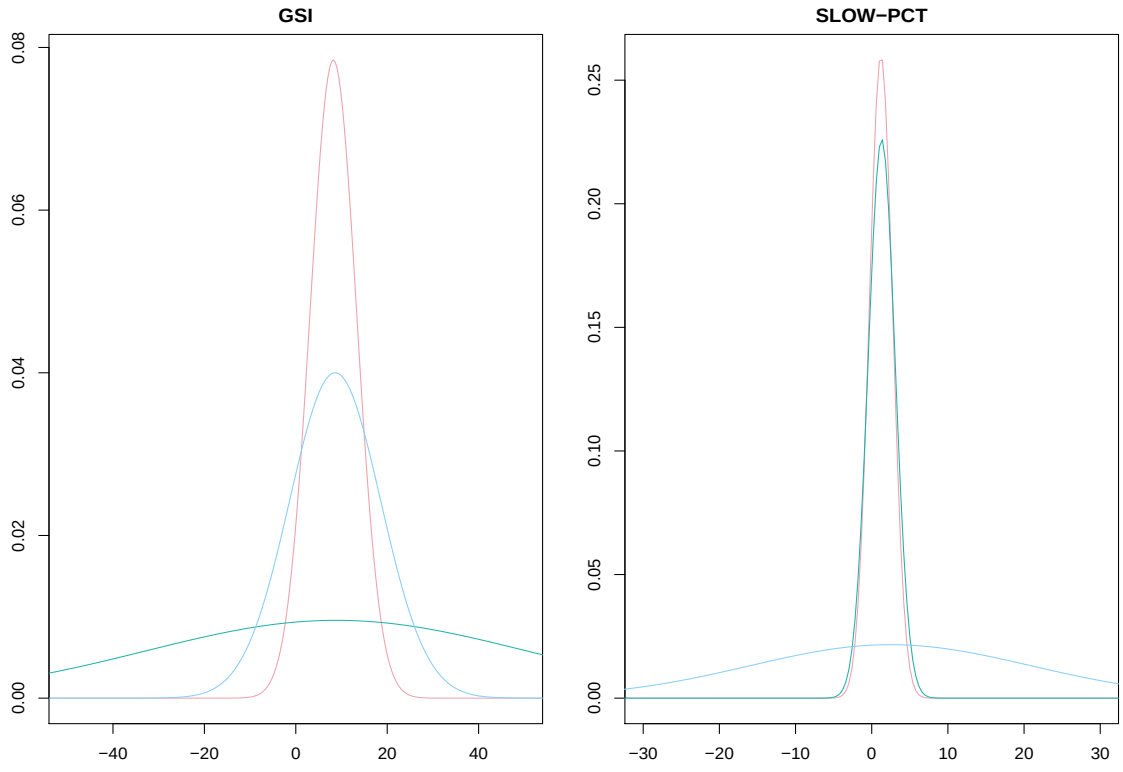


Figure 4.15: *Correponding estimates of the distribution of the significant measurements for each cluster, where colour corresponds to cluster.*

It is interesting to see that the clusters are separated due to the difference in the variance of the measurements within each cluster. This leads to the significance of sperm longevity characteristic, whereas previous analyses indicated sperm velocity. This discrepancy is probably because traditional tests assume a common underlying variance, and do not test for a

difference in the variance. It would be worthwhile to conduct a further analysis where the within-cluster variances are assumed equal, and investigate differences in the means.

We point out here that there are two serious drawbacks with our approach. Firstly, there were several unknown nucleotides in the dataset. Our analysis, however, assumed fixed DNA sequences. As a result, polymorphic sites with unknown nucleotide were ignored, collapsing the number of haplotypes to 14. In order to take unknown nucleotides into account, these may be added into the MCMC parameter set as a parameter which is updated at each iteration. A single change in a nucleotide position may have a serious impact on the topology of the haplotype tree, and hence this extension may greatly increase the complexity of the algorithms.

Secondly, we described in an earlier Section 2.2 how both the Inverse Wishart prior and the independent Gamma prior suffer from limitations in the model. This was directly observed in the analysis here: the posterior estimates were highly dependant upon the IW prior, and hence the Gamma prior was used, which assumed the unrealistic premise of independence of the phenotypes. The model should be extended to use the Generalised Inverse Wishart distribution in order to draw reliable conclusions.

Convergence assessment

As before, we present some representative trace plots (see Figure 4.16) and Gelman-Rubin plots (see Figure 4.17), showing that all clustering parameters indeed seem to have converged. In this case, we do not draw inferences about the tree based on an evolutionary model. As a result, the variability of the MAP estimate of the tree is great. However, for all different MAP trees, the MAP clustering is consistently the same with the same probability, indicating that although the marginal of the tree has not converged, the clustering distribution is not affected.

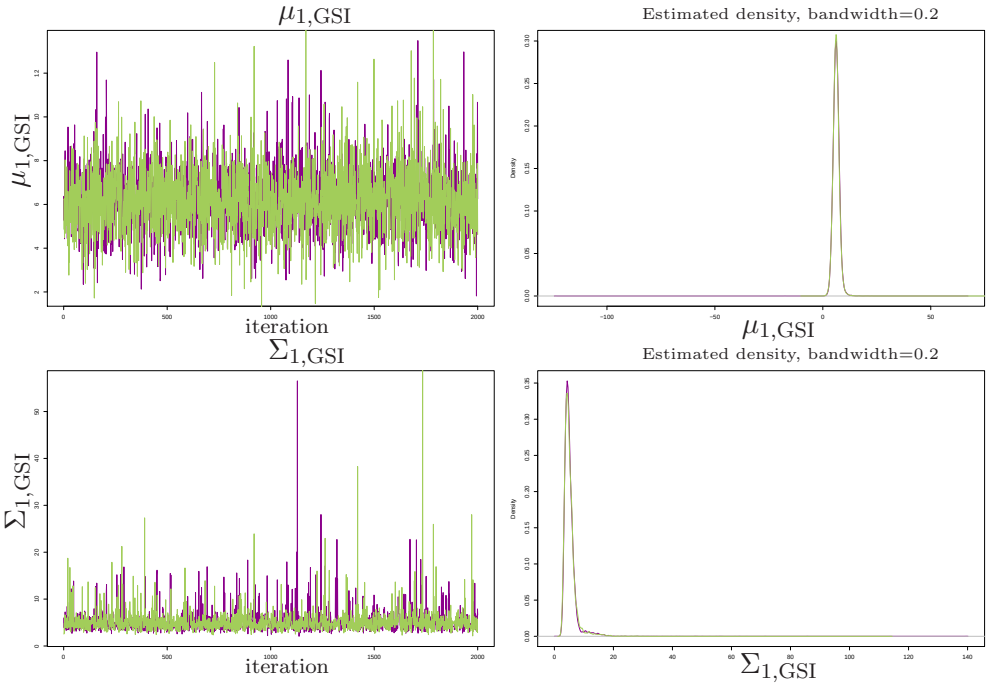


Figure 4.16: *Trace and density plots for cluster parameters of the salmon dataset, showing good mixing of the parameters and matching posterior densities from two different starting points.*

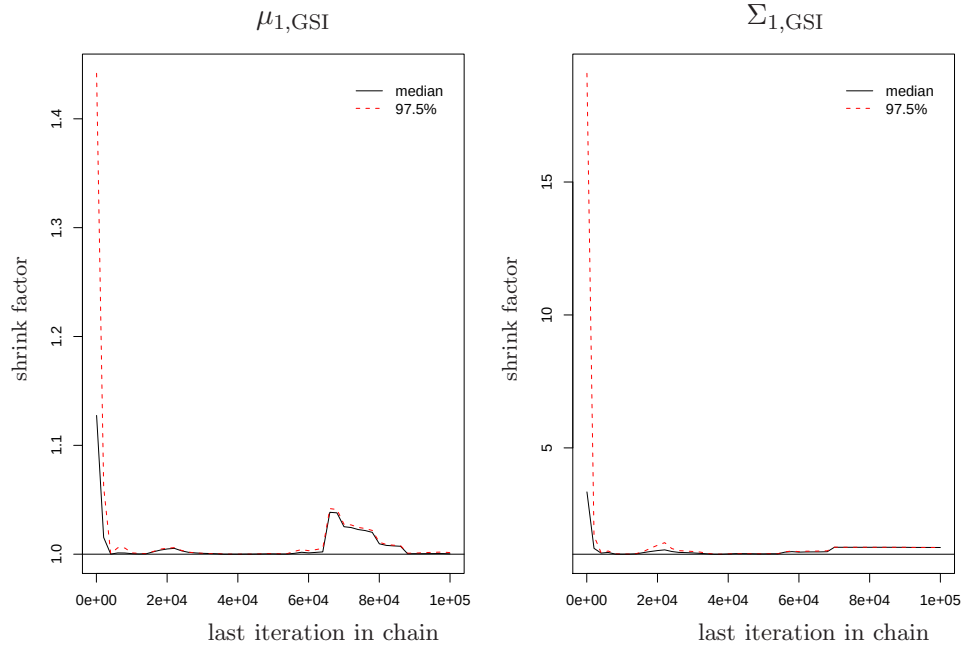


Figure 4.17: *Corresponding Gelman Rubin plot for the cluster parameters of the salmon dataset, suggesting that the tree parameters have indeed converged.*

Chapter 5

Conclusion

The objective of this thesis was to construct phenotypic and phylogeographic clustering methods based on DNA sequence data. We began by presenting existing approaches of inferring phylogenies and analyzing phenotypic and phylogeographic data in Chapter 1. The main methods used at the moment, Nested Clade Analysis (NCA) and Nested Clade Phylogeographic Analysis (NCPA) described in Subsections 1.2.6, 1.3.1 and 1.4.4, proceed by first deciding on a unique (or nearly unique) haplotype tree. The phenotypic/phylogeographic data in then analyzed based on consecutive analyses of variance (for phenotypic data) or permutation tests and application of a descriptive inference key (for phylogeographic data). The results are highly dependent on the initial tree inferred and the criteria proposed used are not always well-defined, and can be subjective. As a result, NCA and NCPA have been frequently criticized; see Subsection 1.4.4.

We then presented a coherent model-based Bayesian alternative to NCA and NCPA which shows significant improvement to phenotypic and phylogeographic clustering problems. We devised two clustering constructions based on a haplotype tree for NCA and NCPA respectively, so that the results of the analysis are consistent with phenotypic and phylogeographic effects. The haplotype tree was simultaneously inferred by assuming the coalescent model of evolution and using a general mutation model for the sequences, at the same time yielding results along Templeton's lines, allowing for direct biological interpretation.

Both inference about the clustering and about the haplotype tree involved a number of challenges. In clustering inference, designing sound clustering constructions based on the haplotype tree was crucial for our analysis, and the phenotypic and phylogeographic clusterings were devised based on the biological processes involved in the phenotypic and geographical distributions (see Sections 2.1-2.3). These allowed for a more natural interpretation of the output of the analysis which reduced the subjectivity of the results.

In phenotypic clustering, we presented an algorithm which allowed us to identify specific

phenotypic measurements which shows a significant change with mutations (see Section 2.2), allowing for large datasets to be easily narrowed down to characteristic traits which are "interesting" in terms of the significant mutations.

The parameter space of phylogeographic clusterings is vast and its construction highly complex. As a result, it was essential to employ adaptive techniques in order to explore the space of clustering possibilities efficiently. We described a number of tricks which improved the convergence and efficiency of the MCMC sampler.

In order to draw inferences about the haplotype tree, we developed an explicit probability model based on the coalescent and the GTR mutation model. This model provided a solid statistical framework which confirmed the empirical predictions of Crandall and Templeton (1993), Posada and Crandall (2001) and Emerson and Hewitt (2005) and allowed for backward inference. In other words, we were able to assess the probability of various ancestral scenarios based on the posterior probability rather than using forward simulations.

The haplotype tree model presented several issues. Intractable likelihoods and normalization constants were addressed by using Approximate Bayesian Computation techniques (see Subsection 3.1.3). The vast discrete parameter space of trees required the construction of efficient adaptive proposal distributions in order to achieve convergence. We described an efficient way of exploring the space of trees, by associating trees with deletions of edges, vastly reducing computational complexity (see Section 3.8). Although this approach was used here for haplotype trees, it can be applied to a number of tree inference problems (such as graphical network inference).

Finally, it was important to ensure that our algorithm was automated, so that the output was comprehensive and concise, allowing it to be used by non-statisticians. We created a software package described in Appendix E which yields biologically interpretable as well as statistically meaningful results.

One of the main advantages of our approach is that it allows for uncertainty to be propagated throughout the analysis, since the geographical distribution of a population often is a valuable source of information for drawing conclusions about a species' phylogeny. Although NCA and NCPA sometimes enable the intuitive inclusion of geographical information during the cladogram forming stage, their criteria are descriptive, here we give a more rigorous statistical basis to the connection between geography and phylogeny.

The coalescent model is a well-established tool in population genetics. As well as being a more accurate representation of the evolutionary process, it allowed us to implicitly incorporate many of the phylogeographic predictions, it affords direct inference of ancestral locations in phylogeographic data (see Section 3.11).

Taking a model-based approach to the phenotypic and phylogeographic clustering prob-

lems offers itself to modifications and additions. In all simulations as well as real datasets, we assumed that our data are Normaly distributed. However, any distribution may be used, with all relevant formulae replaced by the appropriate distribution functions. Similarly, a number of mutation and evolutionary models may be considered instead of the ones assumed in this study.

Perhaps the most important advantage of our method is that it provides quantitative measures of the statistics of interest. For example, we were able to associate probabilities with any one location as being the source population for later spread to all the areas sampled. The results of our analysis were confirmed both by simulations, as well as predictions of biologists regarding the beetle and weevil data.

In Chapter 4 we implemented our method on two phylogeographic datasets and a phenotypic dataset. In both geographical datasets, our results agreed with the NCPA, and confirmed many of the biologically intuitive predictions about ancestral locations population dispersal patterns. In the phenotypic dataset, we were able to identify characteristic traits which showed a significant change with mutations, and separated the data into phenotypically different haplotype clusters.

5.1 Future work

There are several ways in which this work can be further improved and extended, many of which have been mentioned in earlier chapters. We divide them into two categories, depending on whether they are related to the clustering algorithm or to the tree.

5.1.1 Improvements on clustering inference

Although adaptive techniques were employed in this study, the construction of more efficient proposals for the clustering is still a challenging and important issue which greatly affects the runtime of the algorithm. One of the ways in which this can be achieved is by using Population MCMC techniques, which essentially allow for multiple chains to be run simultaneously and the main chain to move around the possible available chains (see Laskey and Myers, 2003; Doucet et al., 2006)

In phenotypic clustering we described how the Inverse Wishart prior distribution imposes a number of constraints on prior belief which are often unrealistic (see Section 2.2). Using the Generalized Inverse Wishart offers a number of advantages which will allow for more reliable phenotypic clustering inference.

Within phylogeographic clustering (see Section 2.3) we discussed how the migrating haplotype clustering construction we implemented does not explicitly allow for fragmentation

events. The phylogeographic clustering can easily be extended to allow for a more general shared haplotype setting to account for a wider variety of phylogeographic events. Furthermore, specific models of phylogeographic hypotheses such as range expansion, restricted gene flow and fragmentation can be employed in order to asses particular scenarios. Dispersal patterns, however, are often too complex to model accurately, and hence such an extension is challenging.

In Subsection 2.3.2 we described how using both the migrating haplotype and the phenotypic clustering within a phenotypic clustering problem can allow us to investigate whether the significant mutation is unsampled, by comparing which of the two clusterings shows a better fit with the data. Of course, the migrating haplotype clusterings are more general than the significant mutation clusterings, which implies that they will always show a better fit in terms of the likelihood. However, they naturally have a much broader prior, implying that they will not always yield a higher posterior mass.

Finally, in Section 3.12 we explained how the two clustering constructions for phenotypic and phylogeographic data may be combined in order to analyze data which include both phenotypic measurements and geographical locations. Taking into account both sources of information can have a valuable effect on the inference about the haplotype tree, at very little computational cost. In the case of the beetle dataset, Emerson et al. (2006) hypothesised that the observed range expansions followed similar expansions in the host species Pinus canariensis and it would be interesting to conduct a complementary phylogeographic study to determine the extent of any such association.

5.1.2 Improvements on inference about the tree

The main drawback of our method is the inefficiency of tree inference. We discussed how inference about the tree is unreliable based only sequence data (see Section 3.9), at the same time heavily adding to computational complexity. The complexity can be hugely improved by increasing the number of importance samples taken to approximate the likelihood of haplotype trees, all of which may be calculated in parallel. In addition, better adaptive proposals may be devised in order to allow full exploration of the tree space.

Another important disadvantage of the tree algorithm described in this thesis is that it is inherently based on parsimony, which is known to often lead to false conclusions. As a result, many of the potential improvements to the methods aim to address the accuracy of the assumptions.

In Section 3.6 we described how allowing the parsimony level d_s is computationally very expensive. However, it is essential that we set it to be reasonably large in order to ensure that the true haplotype tree is indeed contained in the set Ω. More efficient tree moves should

be devised to improve the efficiency of the MCMC sampler so that d_s can be set arbitrarily large.

We discussed how extensive homoplasy can imply that sequences cannot be collapsed onto haplotypes in Subsection 3.1.2. In those cases, even when d_s is assumed as large as desired, the set Ω will never contain the true tree. This could be addressed by introducing a parsimonious index for each nucleotide site, indicating whether that site is assume to be parsimonious or not. Such an approach is similar to testing the parsimony assumption for each site, as described by Templeton et al. (1987). In our analysis, generally a site l will analogously breach the parsimony assumption if it has a very large mutation rate ϕ_l.

One of the assumptions of the haplotype tree model was a uniform prior on the size of the trees (see Subsection 3.1.2). This has an immediate consequence: it implies that, in effect, any non-minimal tree has a significantly smaller probability, because the probability of a mutation is very small. Although the MAP estimate of the tree may still be non-minimal because of, for example, geographical information which dominates the probability of the tree, it is important to investigate alternative prior distributions for the size of the haplotype tree.

Both limitations above could be overcome by constructing an efficient MCMC sampler which will implement the clustering algorithm on coalescent rather than haplotype trees, as described in the Appendix C. Although such an adaptation is potentially valuable, it is a challenging problem.

In this thesis we used the simplest form of the coalescent model (see Section 3.1), assuming a constant population size. This could be extended by allowing population growth (see Slatkin, 2001), and also allowing the population size of each population cluster to vary.

Finally, the methods described here do not allow for unknown nucleotides. In real datasets, such as the salmon dataset used in Section 4.3, there are often unknown nucleotides. These may be completely unknown, or there may be uncertainty between, say A or G. In our analysis, the different possibilities may yield very different trees, adding to the complexity of the analysis. The state of each nucleotide in the data (represented by different nodes in the haplotype tree) would be considered a parameter within the model which would be updated in a separate MCMC update together with the tree.

Appendix A

The label-switching problem

As is common with mixture modelling, our model suffers from the so-called label-switching problem caused by symmetry in the joint probability distribution of the data given the model parameters. This has the practical consequence that the collection of nodes labelled in one iteration of the MCMC algorithm as group k (with associated mean and covariance μ_k and Σ_k) may be labelled as group $j \neq k$ in the next iteration. The standard approach to overcoming such difficulties is to introduce an essentially arbitrary identifiability constraint such as ordering the components in terms of the associated component parameters e.g., the mean. This sort of approach was unsatisfactory in the range of examples we considered due, mainly, to the multidimensional nature of the mixture distributions and the tendency for there to be considerable overlap between the marginal distributions associated with the bivariate normal components. Ordering based upon distance of the mean vector from a fixed point as well as ordering based upon the properties of the tree (e.g., start at the left of the tree and label the components as you visit them) all failed to produce sensible results (see Brooks et al., 2007).

We present two approaches below which proved reliable at addressing label-switching issues. The first one, by Stephens (2000), described a method where the labels are chosen post-simulation based on the clusterings of each iteration. The second approach by Scott and Wang (2006) assigns labels at each iteration during the MCMC algorithm, and is more efficient when computational memory storage is limited.

A Method described by Stephens (2000)

We adapt the algorithm described in Stephens (2000) in order to draw inferences about component parameters of the phenotypic/phylogeographic clustering problems described in this thesis. The method is based upon choosing the labels $\nu^{(t)}$ post-simulation. Suppose we have a sample of T MCMC sample observations. For any observation t we require a

permutation ν_t of the the associated component labels that provides us with a consistent labelling for all $t = 1, \ldots, T$. Stephens suggests the following iterative algorithm.

Starting with some initial values for the permutations $\nu_1, \ldots, \nu_N$ of the first N steps (setting them all to the identity permutation for example), iterate the following steps until a fixed point is reached:

Step 1: Choose $\hat{a}$ to minimise $\sum_{t=1}^{N} \mathcal{L}_0\left(\hat{a}; \nu^{(t)}\left(\theta^{(t)}\right)\right)$.

Step 2: For $t = 1, \ldots, N$ choose ν_t to minimise $\sum_{t=1}^{N} \mathcal{L}_0\left(\hat{a}; \nu^{(t)}\left(\theta^{(t)}\right)\right)$.

Stephens (2000) suggest the loss function

$$\mathcal{L}_0(Q;\theta) = \sum_{z_1=1}^{k} \cdots \sum_{z_n=1}^{k} p_{1z_1}(\theta) \ldots p_{nz_n}(\theta) \log \frac{p_{1z_1}(\theta) \ldots p_{nz_n}(\theta)}{q_{1z_1} \cdots q_{nz_n}} = \sum_{i=1}^{n} \sum_{j=1}^{k} p_{ij}(\theta) \log \frac{p_{ij}(\theta)}{q_{ij}}.$$

It is easy to check that in step 1 the $\hat{q}_{ij}$ which minimizes $\mathcal{L}$ is given by

$$\hat{q}_{ij} = \frac{1}{N} \sum_{t=1}^{N} p_{ij}\left(\nu_t\left(\theta^{(t)}\right)\right).$$

Using the loss function suggested in Stephens (2000), starting with the identity permutation, the algorithm becomes a single step:

Pick a labelling to maximise

$$\sum_{j,l} \log \frac{1}{t} \left(\sum_{i=1}^{t} \mathbb{I}_{c_{jl}^{(i)} = c_{jl}^{(t)}} \right)$$

where c_{jl} denotes the cluster of the lth observation of haplotype j. In other words, this algorithm picks labels for each group so that "as many datapoints as possible belong to their favourite cluster".

The disadvantage of the above algorithm is that the labelling has to be done post-simulation, so all the values of the cluster parameters of interest have to be stored, which often proves computationally inefficient. An alternative approach which overcomes memory storage issues is described below.

B Method described by Scott and Wang (2006)

Scott and Wang (2006) suggest an inference-based method of choosing the clustering labels at each iteration. It is based upon finding the parameters which maximize the likelihood

during burn-in, and then choosing all subsequent labels by relating the parameters to the maximizing ones. This method works well with our model, and requires little computational memory. Specifically, the algorithm works in the following way.

During burn-in

1. Calculate $\ell^{(g)} = p(y \mid \theta^{(g)})p(\theta^{(g)})$.
2. Let z^* be the $z^{(g)}$ with the largest $\ell^{(g)}$.

Thereafter

1. At each iteration draw $\nu^{(g)} \mid y, \theta^{(g)}, z^*$.

The disadvantage of the algorithm is that if the chain has not converged by the time burn-in finishes, the value of the maximizing parameters will not ensure that the labelling is assigned efficiently, and it is sometimes difficult to distinghuish between poor convergence and poor labelling.

Appendix B

Proofs

Lemma 3.1.1 *Using the specified priors (3.8)-(3.11), the joint posterior of the mutation parameters given the rooted haplotype tree is given by*

$$\mathbb{P}(\boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v} \,|\, \mathcal{S}, \mathcal{T}, r) \quad \propto \quad \mathbb{P}(\mathcal{T} \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}) \times p(r) \times p(\boldsymbol{\pi} \,|\, r) \times p(\boldsymbol{\phi}) \times p(\boldsymbol{v})$$

Proof.

$$\begin{aligned}
P(\boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v} \,|\, \mathcal{S}, \mathcal{T}, r) &= P(\boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}, \mathcal{T}, r \,|\, \mathcal{S}) \times P(\mathcal{T}, r \,|\, \mathcal{S}) \\
&= P(\mathcal{T} \,|\, \mathcal{S}, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}) \times P(r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v} \,|\, \mathcal{S}) \times P(r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v} \,|\, \mathcal{S}) \times P(\mathcal{T}, r \,|\, \mathcal{S}) \\
&= \frac{P(\mathcal{T} \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})}{\sum_{\mathcal{T}_i \in \Omega} P(\mathcal{T}_i \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})} \times P(r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v} \,|\, \cup \mathcal{T}_i \in \Omega) \times P(\mathcal{T}, r \,|\, \mathcal{S}) \\
&= \frac{P(\mathcal{T} \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})}{\sum_{\mathcal{T}_i \in \Omega} P(\mathcal{T}_i \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})} \frac{\sum_{\mathcal{T}_i \in \Omega} P(\mathcal{T}_i \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}) \times P(r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})}{P(\cup \mathcal{T}_i \in \Omega)} \times P(\mathcal{T}, r \,|\, \mathcal{S}) \\
&= P(\mathcal{T} \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}) \times \frac{P(r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})}{P(\cup \mathcal{T}_i \in \Omega)} \times P(\mathcal{T} \,|\, \mathcal{S}, r) \times P(r \,|\, \mathcal{S}) \\
&= P(\mathcal{T} \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}) \times \frac{P(\pi \,|\, r) \times p(\boldsymbol{v}) \times p(\boldsymbol{\phi})}{P(\cup \mathcal{T}_i \in \Omega)} \times P(\mathcal{T} \,|\, \mathcal{S}, r) \times P(r \,|\, \mathcal{S}) \\
&\propto \mathbb{P}(\mathcal{T} \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}) \times p(r) \times p(\boldsymbol{\pi} \,|\, r) \times p(\boldsymbol{\phi}) \times p(\boldsymbol{v})
\end{aligned}$$

□

Lemma 3.1.2 *Similarly, using (3.11)-(3.12) we can calculate the posterior distribution for the root*

$$\mathbb{P}(r \,|\, \mathcal{S}, \mathcal{T}, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}) \quad \propto \quad \mathbb{P}(\mathcal{T} \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}) \times p(r \,|\, \boldsymbol{\pi})$$

Proof.

$$\begin{aligned}
\mathbb{P}(r \,|\, \mathcal{S}, \mathcal{T}, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}) &= \frac{P(\mathcal{T}|\mathcal{S}, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}) \times P(r|\mathcal{S}, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})}{P(\mathcal{T}|\mathcal{S}, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})} \\
&= \frac{P(\mathcal{T} \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})}{\sum_{\mathcal{T}_i \in \Omega} P(\mathcal{T}_i \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})} \frac{P(r| \cup \mathcal{T}_i \in \Omega, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})}{P(\mathcal{T}|\mathcal{S}, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})} \\
&= \frac{P(\mathcal{T} \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})}{\sum_{\mathcal{T}_i \in \Omega} P(\mathcal{T}_i \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})} \frac{P(\cup \mathcal{T}_i \in \Omega | r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}) \times P(r|\boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})}{P(\cup \mathcal{T}_i \in \Omega | \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}) \times P(\mathcal{T}|\mathcal{S}, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v})} \\
&\propto \mathbb{P}(\mathcal{T} \,|\, r, \boldsymbol{\phi}, \boldsymbol{\pi}, \boldsymbol{v}) \times p(r \,|\, \boldsymbol{\pi})
\end{aligned}$$

□

Lemma 3.6.2 *When no homoplasy is present, algorithm 3.6 results in a unique tree.*

Proof. In the absence of homoplasy, it is easy to check the following facts:

- The effective representatives of two groups are unique. This is true because in the absence of homoplasy, the mutational distance on the tree is always equal to the distance of the two sequences in terms of SNP mutations. If this were not the case, i.e., there exist two haplotypes which are closer in terms of their SNP distance than their tree distance, then at least one mutation would have had to be reversed, which contradicts the assumption of no homoplasy. This implies that there is only 1 pair of haplotypes which satisfies the condition in Step 4 of the algorithm.

- Each SNP mutation uniquely dichotomises the sequences even in the absence of the tree. This means that it is not possible for two different pairs of groups which have the same minimum distance to involve the same mutation.

- If two pairs of groups with different minimum distances involve a common mutation, then the inferred mutations of both will coincide on the shorter branch.

Now assume that the inferred tree is not unique. This is possible in two ways: either two groups yield two effective pairs of representatives, or two different pairs of groups which involve a common mutation yield different intermediate sequences. Clearly, none of these is possible, using the facts above. □

Lemma 3.8.2 *Algorithm 3.8 identifies exactly j distinct loops. Furthemore, by removing one of the edges of each loop, we can obtain all subtrees of the original network.*

Proof. (a) First we show that it is not possible for the above algorithm to identify less than j loops. If less than j edges have been removed, there exists at least one loop in the

graph. Since we repeat the steps in the algorithm until each node does not belong to a loop, that's not possible, since that loop would have been broken.

(b) It is not possible for the above algorithm to identify more than j loops. This is not possible, since after removing the jth loop, no more loops are present, so the algorithm terminates.

(c) We now assume that there exists a subtree of the original network which cannot be obtained by deleting one of the edges of each loop. However, it is clearly necessary that each loop has at least 1 deleted edge, otherwise that loop would remain unbroken. Hence, by the pigeonhole principle, it is always possible to find a correspondence of deleted edges and loops so as to obtain every possible subtree.

□

Theorem 3.8.3 *In order to obtain a tree topology by removing edges which belong to loops, at least one edge which only belongs to one loop has to be deleted.*

Proof. First consider two loops named 1 and 2 sharing one or more edges. If they only share one edge, and since we know that exactly two edges have to be deleted, at least one of the two will not be shared and the lemma trivially holds. If there is more than one shared edge, there are two cases, shown in Figure B.1. Either they are all adjacent (an example is shown in the left hand panel of Figure B.1), or they are not (shown in the right hand panel of Figure B.1).

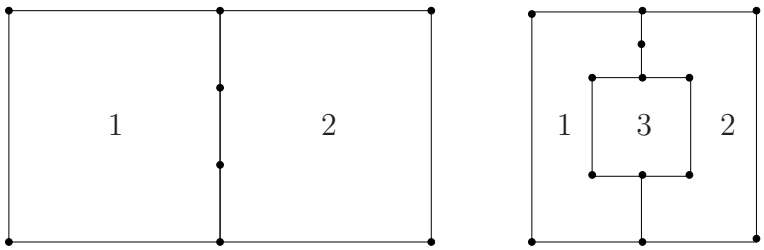


Figure B.1: *Two possibilities for two loops sharing edges. In the left hand figure, there are two loops present, 1 and 2. They share three edges, all of which are adjacent. Clearly, removing any two of the shared edges will not break up the outer loop of 1 and 2. In the right-hand figure, there are three loops present, but we focus on only two of them, namely loops 1 and 2. The shared edges are not adjacent in this case, but two are at the top and one at the bottom. Clearly, even if we remove all of the shared edges, the outer loop will remain.*

From the figure it is easy to see that, between two loops, it is necessary to remove at least one edge in order to break up both loops.

Now consider the case of a third loop named 3, sharing edges with one or both of 1 and 2 considered above. There are three cases: either loops only share an edge with only one of the other two loop, or there exists an edge shared between all three loops, or loops share edges with both other loops pairwse. These three cases are shown in the Figure B.2 below.

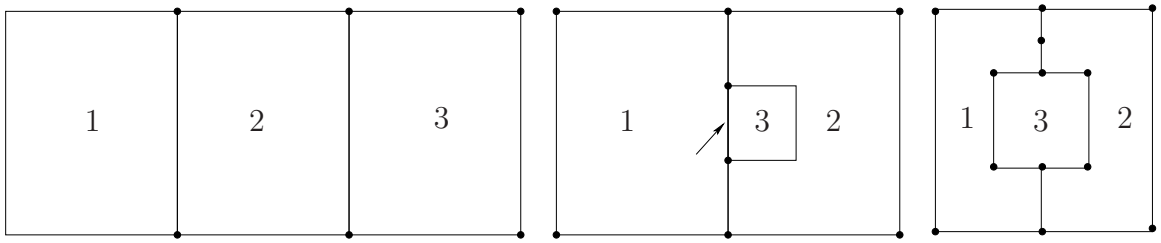


Figure B.2: *The three representative possibilities for three loops sharing edges. In the left-hand figure, loops only share an edge with one other loop. Clearly, the outer loop cannot be broken unless one non-shared edge is removed. In the middle figure, there exists an edge which is shared between all three loops, shown by the arrow. As before, clearly the outer loop cannot be broken unless a non-shared edge is broken. Lastly, in the figure on the right, loops pairwise share an edge, but again, the outer loop cannot be broken.*

Note here that the third case can also be achieved with a similar cyclic structure as the right-hand panel of Figure B.1. However, the same argument as before applies.

Inductively, we see that it is indeed not possible to break up all the loops unless at least one non-shared edge is deleted. Aside from examples, the intuitive reason why this theorem is true is that when removing an edge which is shared between two loops, there are always two paths to get from one endpoint of that edge to the other, moving along the first or second loop, respectively.

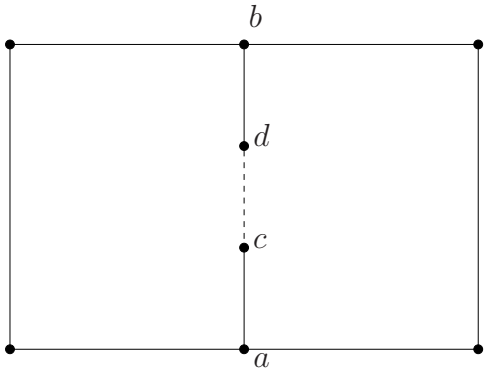


Figure B.3: *Two loops, sharing the three edges in the middle. Clearly, the middle one (or any of the other two) implies that there are two paths between points a and b. Although these two paths do not necessarily form a loop, it will always be possible to find points c and d such that a loop is forms, otherwise the two original loops would be identical.*

□

Lemma 3.8.4 *For each tree topology, there is a one-to-one representation of edges deleted to loops to which they belong.*

Proof. Using Theorem 3.8.3, for every tree topology, at least one deleted edge must belong to precisely one loop. Hence, the correspondence for that edge is unique to that loop. Returning to the original graph, and removing that edge, at least one of the remaining loop edges must belong to exactly one loop. As before, the correspondence is unique, we remove that edge and continue.

Clearly, the above process leads to a one-to-one representation. □

Lemma 3.9.1 *A local update of the tree topology which changes the edges removed from a single loop preserves irreducibility.*

Proof. As we proved previously, in every group of connected loops, there is at least one edge which only belongs to one loop. This means that there exists always a local move which will remove that specific edge. In addition, removing that edge is guaranteed to break that loop, so if the previous topology $\mathcal{T}$ was a tree, so will $\mathcal{T}'$. Subsequently, considering all but this last loop, we may repeat the process inductively, and will always be able to reach a specific tree topology which is obtained by removing a specific set of edges belonging to only one loop at a time. This tree topology can clearly be reached from any state of $\mathcal{T}$ after at most the number of steps as loops existing, and hence a local proposal preserves irreducibility. □

Appendix C

Clustering on the coalescent

Haplotype trees are inherently based on parsimony and thus impose assumptions on the coalescent model as described in Chapter 3. As a result, coalescent trees are considered to provide a more reliable basis for inference in population genetics. Here we implement our method on coalescent rather than haplotype trees, and describe why a direct adaptation of our method becomes computationally infeasible.

Remember that in order to draw inferences about the haplotype tree, we essentially calculated probabilities on coalescent trees with mutations which were collapsed onto haplotype trees. Our method is easily modified so that the clustering is applied directly on the coalescent.

A Phenotypic clustering

In order to apply our phenotypic clustering method on a coalescent tree, it is required that individual mutations can be identified on the tree. Standard coalescent trees do not specify mutations; the usual peeling algorithm essentially integrates over all possible mutational paths that may have led to the observed sample. Instead, we can use coalescent trees with mutations to specify the nucleotide state of the ancestral sequences at each coalescence event. In other words, rather than summing over all possibilities using the peeling algorithm, we insert the mutations as a parameter within the MCMC algorithm and update it at each iteration, similarly to Li et al. (2000). This allows for all mutations to be identified and thus the phenotypic clustering algorithm to be applied.

In order to distinguish between edges of the coalescent tree which represent mutations and edges which represent only coalescence events, we apply the clustering algorithm so that only edges which connect two different sequences may be associated with a significant phenotypic effect. This can be understood through the top panel of Figure C.1.

Implementing this approach proves computationally infeasible. The space of coalescent trees with specified mutations is vast and requires extremely long run times in order to converge. Usual population genetics datasets contain at least 100 datapoints, but the algorithm here doesn't converge within a realistic time-frame for datasets containing over 30 different haplotypes.

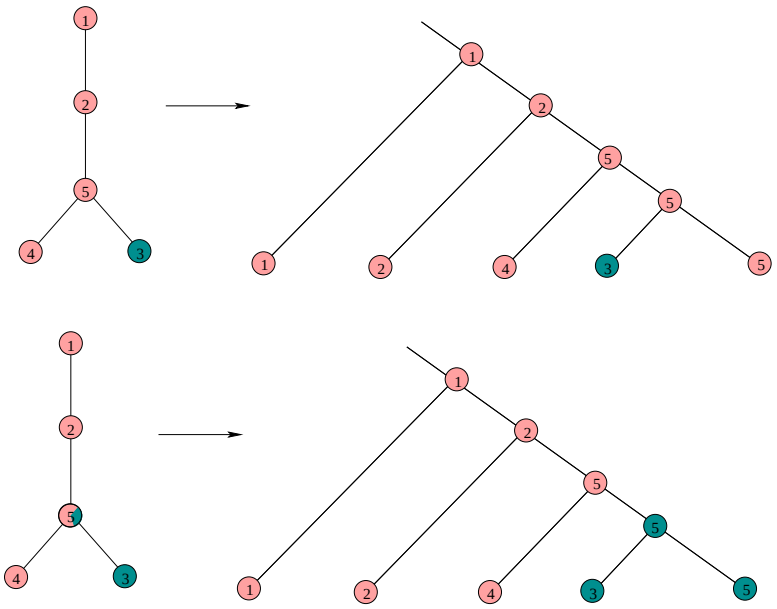


Figure C.1: *Examples of phenotypic (top) and phylogeographic (bottom) clusterings on the same haplotype tree. On the left are shown the clusterings as described in Chapter 2, and on the right are the equivalent clusterings defined on the corresponding coalescent tree. In the phenotypic case, it is necessary that the cluster-defining edge of the coalescent tree involves a mutation i.e., the two sequences at each end are different. In the phylogeographic case, this is not essential, as shown in the example.*

B Phylogeographic clustering

The phylogeographic clustering algorithm we described may be directly adapted to coalescent trees without requiring the nucleotide state of intermediate sequences to be determined. In the case of phylogeographic clustering, mutations are not important. As a result, standard coalescent trees may be used to apply a simple clustering construction. The clustering is equivalent to the approach used by De Iorio and Griffiths (2004b) described in Section 1.4.3. Through the bottom panel of Figure C.1 we see that the clustering construction we defined in Section 2.3 is equivalent to choosing any of the edges of the coalescent tree to separate the datapoints of the sample.

Implementing our approach proves, again, computationally infeasible. Within the coa-

lescent, any re-arrangement of identical sequences will have exactly the same probability. However, the geographical data often provides a useful source of information about the order in which identical sequences coalesce. In the haplotype tree approach, we used this information to construct efficient proposals about the clustering. These proposals cannot be easily adapted for the coalescent. For a haplotype with 50 datapoints, there are 50! rearrangements of the order in which they may have coalesced (if they are identical by descent), and the algorithm requires infeasibly long run times to explore the space. The method becomes inapplicable for datasets over 20-30 haplotypes.

Appendix D

The hashing algorithm for labelling trees

We have already described how tree topologies in our analysis can be represented by a finite-dimensional integer vector (corresponding to the deleted edges of the original network). In order to draw inferences about the tree topology, a way of efficiently keeping track of this vector is required. Storing multi-dimensional vectors directly becomes computationally infeasible when they consist of more than a few components (representing the number of loops in the graph). For example, having 20 or more loops would require storing a 20-dimensional array for each iteration.

Hashing algorithms provide an algorithm by which data can be represented by an integer. For example, they can be used to create an efficient phonebook, by turning names into integers which are then inserted into (and may subsequently be retrieved from) a table. This is analogous to what we require here: we want our integer vectors to be stored into a table so that it is possible to keep track of how many times each tree topology is visited during the MCMC algorithm.

First of all, it is straight-forward to construct a one-to-one representation of tree topologies with an integer. For example, if there are 100 edges and 3 loops in total, and edges 34, 67 and 88 have been removed, then

$$M = 34 \times 101^2 + 67 \times 101^1 + 88 \times 101^0$$

gives us a unique integer which is not possible to obtain in any other combination of edges, since none of the coefficients will ever be larger than 101. This implies that each tree topology can be denoted by a unique integer.

Although this approach solves the problem of representations, manipulating these repre-

sentations is still complex. Having a large set of such numbers, many of which can be of the order 10^{20} or higher for large datasets, makes it computationally challenging to identify the most frequently occurring number. It is essential to construct a method by which these numbers are stored efficiently, allowing us to access quickly.

Hash functions create a short (as short as possible) address book where each of these numbers is stored in a specific page, in such a way that it can easily be retrieved (see Knuth, 1998). As numbers M appear in our MCMC simulations, we use a hash function which takes M along with the number of times M has appeared so far to a page of the address book. If that page already contains M in it, we simply update v_M, the number of times it has appeared, to be $v_M + 1$. If that page contains some other number, we re-apply the function and move to a different page, until an empty one is found. A good hash function is one which minimises clashes, i.e., having to re-apply the address function, at the same time keeping the total number of pages required as low as possible.

At the end, the address book practically has the following form shown in Table D.1, where the first columns corresponds to the numbers M, and the second represents the number of times each of these Ms occurred.

M	v_M
247659	385
N/A	N/A
285372	294
824832	103
N/A	N/A

Table D.1: *An example of a hashing table. The left-hand column shows the number M, and the right-hand column shows v_M, the number of times the topology represented by M has appeared. The row number corresponds to the address obtained by the hashing algorithm. Here some of the rows have not been filled, shown by N/A.*

Here we let the number of rows of the hash matrix be N use the following hash algorithm:

Algorithm

1. Calculate $P_0 = M \mod N$ and go that row. If M is already in the first column, add 1 to the second column of that row. If that page is empty, insert M and 1 in the two columns. If the page is already taken by another number, go to step 2.

2. Calculate $(N-2) - (M \mod (N-2))$ and go to row $P_{i+1} = (P_i + (N-2) - (M \mod (N-2))) \mod N$. If M is already in that page, add 1 to the second column and

stop. If that page is empty, insert M and 1 in the two columns and then stop. If the page is already taken by another number, repeat this step.

It is easy to check that if N is prime, step 2 will eventually have to go through all possible addresses if no empty one is found. At the end of the MCMC simulation, a simple scan through the hash matrix can find the entry with the highest number of occurrences, in order to find the MAP M, which can be easily reduced to the vector of removed edges.

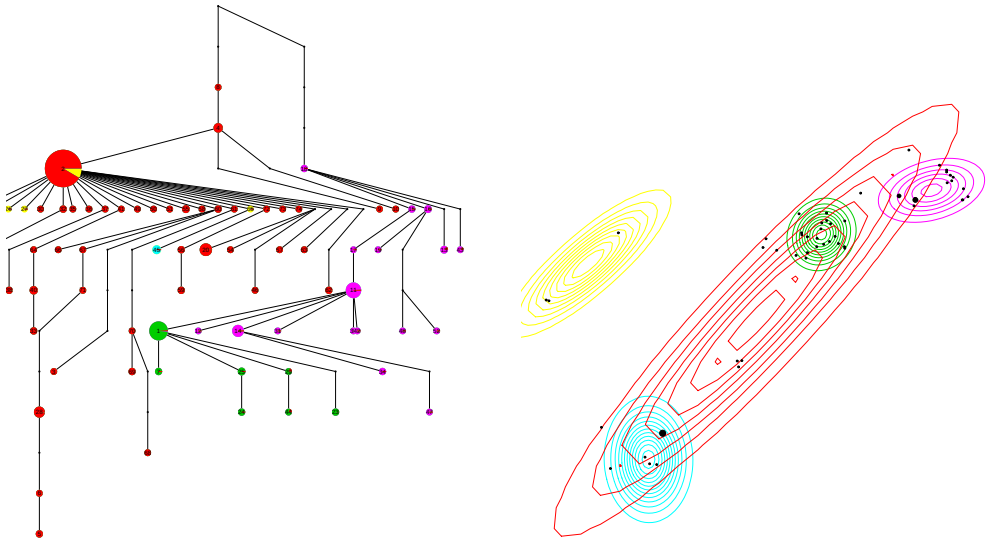
Appendix E

R Package

Many of the methods described have been implemented through an R package. The package allows for phenotypic and phylogeographic datasets to be analyzed, and the output presented in several ways.

In the case of phenotypic datasets, the output is presented graphically through the MAP haplotype tree estimate with clusters shown on each node by a different colour. Using the same colours, the posterior mean estimates for the distribution of each cluster are shown through density plots. An example is shown below in Figure E.1. In addition, a number of output files are created which may be analyzed in directly. An mcmc object is created which may be directly imported into CODA in order to assess the convergence of the parameters. Finally, the set of inferred intermediate sequences is stored in a .nex file so that it can be imported into tree-drawing software.

For phylogeographic clustering, instead of the density plots the clustering is presented graphically through a contour plot as in Figure E.2, also showing the most likely ancestral locations. Output files similar to the phenotypic ones as well as CODA objects are created.



(a) (b)

Figure E.1: *(a) The MAP estimate of the haplotype tree, where colour corresponds to cluster and size to the number of individuals sampled with each sequence; (b) Corresponding density plots for the posterior distribution of each phenotypic component.*

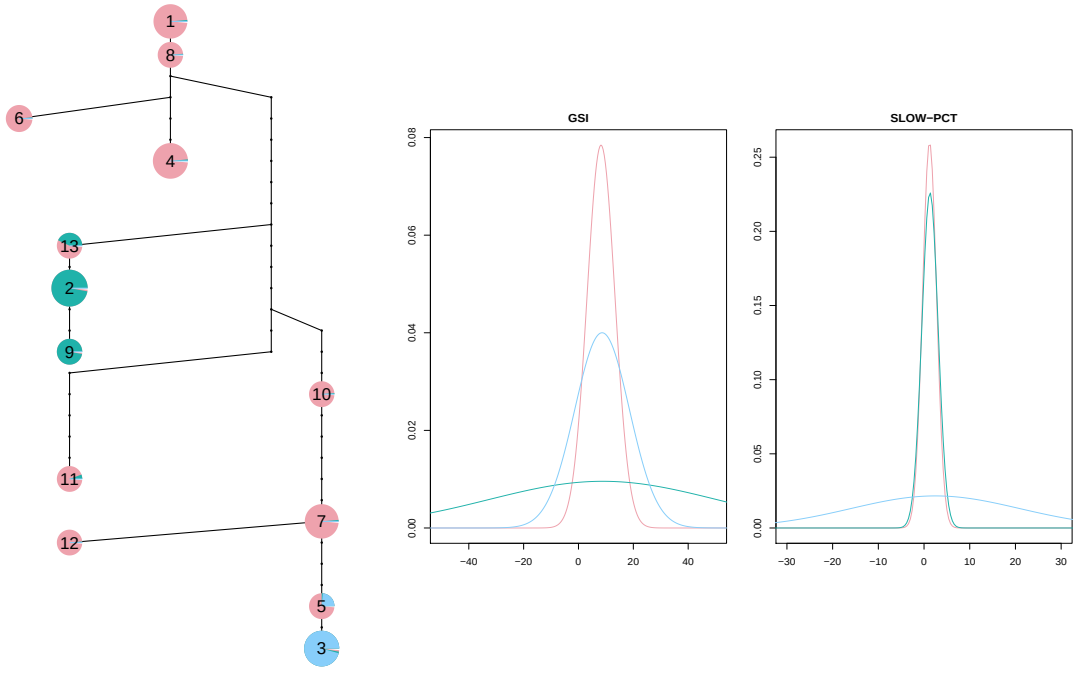


Figure E.2: *(a) The MAP estimate of the haplotype tree, where colour corresponds to cluster and size to the number of individuals sampled with each sequence; (b) Correponding bivariate normal contour plots and evaluated at the posterior means. The circles indicate sampling locations, with the larger circle indicating the location most likely to include the root node.*

Bibliography

Abecasis, G., Tam, P., Bustamante, C., Ostrander, E., Scherer, S., Chanock, S. J., Kwok, P., and Brookes, A. Human Genome Variation 2006: emerging views on structural variation and large-scale SNP analysis. *Nature Genetics*, 39:153–155, 2007.

Altekar, G., Dwarkadas, S., Huelsenbeck, J., and Ronquist, F. Parallel metropolis coupled Markov chain Monte Carlo for Bayesian phylogenetic inference. *Bioinformatics*, 20:407–415, 2004.

Atteson, K. The performance of neighbor-joining algorithms of phylogeny reconstruction. In *COCOON '97: Proceedings of the Third Annual International Conference on Computing and Combinatorics*, pages 101–110. Springer-Verlag, London, UK, 1997.

Atteson, K. The performance of neighbor-joining methods of phylogenetic recronstruction. *Algorithmica*, 25:251–278, 1999.

Avise, J. *Phylogeography: The History and Formation of Species*. Harvard University Press, 2000.

Avise, J., Arnold, J., Ball, R., Bermingham, E., Lamb, T., Nigel, J., Reeb, C., and Saunders, N. Intraspecific phylogeography: The mitochondrial DNA bridge between population genetics and systematics. *Annual Review of Ecology and Systematics*, 18:489–522, 1987.

Bahlo, M. and Griffiths, R. Inference from gene trees in a subdivided population. *Theoretical Population Biology*, 57:79–95, 2000.

Balding, D. *Handbook of Statistical Genetics*. Wiley, 2003.

Bandelt, H., Forster, P., and Rohl, A. Median-joining networks for inferring intraspecific phylogenies. *Molecular Biology Evolution*, 16:37–48, 1999.

Bandelt, H., Forster, P., Sykes, B., and Richards, M. Mitochondrial portraits of human populations using median networks. *Genetics*, 141(2):743–753, 1995.

Beaumont, M. Estimation of population growth or decline in genetically monitored populations. *Genetics*, 164:1139–1160, 2003.

Beaumont, M., Zhang, W., and Balding, D. Approximate Bayesian computation in population genetics. *Genetics*, 162:2025–2035, 2002.

Becquet, C. and Przeworski, M. A new approach to estimate parameters of speciation models with application to apes. *Genome Research*, page gr.6409707, 2007.

Beerli, P. and Felsenstein, J. Maximum likelihood estimation of a migration matrix and effective population sizes in n subpopulations by using a coalescent approach. *Proceedings of the National Academy of Sciences*, 98:4563–4568, 2001.

Brooks, S. Markov chain Monte Carlo method and its application. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 47:69–100, 1998.

Brooks, S. and Gelman, A. General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, 7:434–455, 1998.

Brooks, S., Manolopoulou, I., and Emerson, B. Assessing the effect of genetic mutation: A Bayesian framework for determining population history from DNA sequence data. In *Bayesian Statistics 7*. Oxford University Press, 2007.

Brooks, S. and Roberts, G. Assessing convergence of Markov chain Monte Carlo algorithms. *Statistics and Computing*, 8:319–335, 1998.

Castelloe, J. and Templeton, A. Root probabilities for intraspecific gene trees under neutral coalescent theory. *Molecular Phylogenetics and Evolution*, 3:102–113, 1994.

Clement, M., Posada, D., and Crandall, K. TCS: A computer program to estimate gene genealogies. *Molecular Ecology*, 9:1657–1659, 2000.

Clement, M., Snell, Q., Walker, P., Posada, D., and Crandall, K. Tcs: Estimating gene genealogies. In *International Workshop on High Performance Computational Biology*. 2002.

Cowles, M. and Carlin, B. Markov chain Monte Carlo convergence diagnostics: A comparative review. *Journal of the American Statistical Association*, 91:883–904, 1996.

Crandall, K. and Templeton, A. Empirical tests of some predictions from coalescent theory with applications to intraspecific phylogeny reconstruction. *Genetics*, 134:959–969, 1993.

De Iorio, M. and Griffiths, R. Importance sampling on coalescent histories. I. *Advances in Applied Probability*, 36:417–433, 2004a.

De Iorio, M. and Griffiths, R. Importance sampling on coalescent histories. II: Subdivided population models. *Advances in Applied Probability*, 36:434–454, 2004b.

Desper, R. and Gascuel, O. Fast and accurate phylogeny reconstruction algorithms based on the minimum-evolution principle. In *WABI '02: Proceedings of the Second International Workshop on Algorithms in Bioinformatics*, pages 357–374. Springer-Verlag, 2002.

Doucet, A., Jasra, A., and Del Moral, P. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68:411–436(26), 2006.

Emerson, B. and Hewitt, G. Phylogeography. *Current Biology*, 15:367–371, 2005.

Emerson, B. and Oromi, P. Diversification of the forest beetle genus tarphius in the Canary Islands, and the evolutionary origins of island endemics. *Evolution*, 59:586–598, 2005.

Emerson, B., Paradis, E., and Thébaud, C. Revealing the demographic histories of species using DNA sequences. *Trends in Ecology and Evolution*, 16, 2001.

Ethier, S. N. and Griffiths, R. C. The infinitely-many-sites model as a measure-valued diffusion. *The Annals of Probability*, 15:515–545, 1987.

Falush, D., Stephens, M., and Pritchard, J. Inference of population structure using multilocus genotype data: Linked loci and correlated allele frequencies. *Genetics*, 164:1567–1587, 2003.

Felsenstein, J. Cases in which parsimony or compatibility methods will be positively misleading. *Systematic Zoology*, 27:401–410, 1978.

Felsenstein, J. Statistical inference of phylogenies. *Journal Of The Royal Statistical Society Series A*, 146:246–272, 1983.

Felsenstein, J. The troubled growth of statistical phylogenetics. *Systematic Biology*, 50:465–467, 2001.

Felsenstein, J. *Inferring phylogenies*. Sinauer Associates, 2003.

Garthwaite, P. and Al-Awadhi, S. Non-conjugate prior distribution assessment for multivariate normal sampling. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 63:95–110, 2001.

Gascuel, O. and Steel, M. Neighbor-joining revealed. *Molecular Biology and Evolution*, 23:1997–2000, 2006.

Gelman, A. and Rubin, D. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7:457–511, 1992.

Gemmell, N., Metcalf, V., Irvine, P., Jones, F., McBride, K., and Allendorf, F. Do mitochondrial mutations affect population viability?: The effect of mitochondrial DNA mutations on sperm function. 11th International Congress of Human Genetics, 2006.

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

Geyer, C. Practical Markov chain Monte Carlo. *Statistical Science*, 7:473–483, 1992.

Gilks, W., Richardson, S., and Spiegelhalter, D. *Markov chain Monte Carlo in Practice (Interdisciplinary Statistics)*. Chapman & Hall, 1995.

Green, P. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82, 1995.

Green, P. A primer on Markov Chain Monte Carlo. In O. E. Barndorff-Nielsen, D. R. Cox, and C. Kluppelberg, editors, *Complex Stochastic Systems*. Chapman & Hall, 2000.

Green, P., Hjort, N., and Richardson, S. *Highly structured stochastic systems*. Oxford University Press, 2003.

Griffiths, R. and Tavaré, S. Ancestral inference in population genetics. *Statistical Science*, 9:307–319, 1994a.

Griffiths, R. and Tavaré, S. Simulating probability distributions in the coalescent. *Theoretical Population Biology*, 46:131–159, 1994b.

Griffiths, R. and Tavaré, S. Unrooted genealogical tree probabilities in the infinitely-many-sites model. *Mathematical Biosciences*, 127:77–98, 1995.

Gu, X. and Li, W. Estimation of evolutionary distances under stationary and nonstationary models of nucleotide substitution. *Proceedings of the National Academy of Sciences*, 95:5899–5905, 1998.

Handley, L., Manica, A., Goudet, J., and Balloux, F. Going the distance: human population genetics in a clinal world. *Trends in Genetics*, 23:432–439, 2007.

Hein, J., Schierup, M., and Wiuf, C. *Gene genealogies, variation and evolution*. Oxford University Press, 2005.

Hewitt, G. The genetic legacy of the Quaternary ice ages. *Nature*, 405:907–913, 2000.

Holder, M. and Lewis, P. Phylogeny estimation: traditional and Bayesian approaches. *Nature Genetics review*, 4:275–284, 2003.

Hudson, R. Properties of a neutral allele model with intragenic recombination. *Theoretical Population Biology*, 23:183–201, 1983.

Huelsenbeck, J., Larget, B., Miller, R., and Ronquist, F. Potential applications and pitfalls of Bayesian inference of phylogeny. *Systems Biology*, 51:673–688, 2002.

Huelsenbeck, J. and Ronquist, F. MrBayes: Bayesian inference on phylogenetic trees. *Bioinformatics*, 17:754–755, 2001.

Ibrahim, K., R., N., and Hewitt, G. Spatial patterns of genetic variation generated by different forms of dispersal during range expansion. *Heredity*, 77:282–291, 1996.

Jow, H., Amos, W., Luo, H., Zhang, Y., and Burroughs, N. A Markov chain Monte Carlo method for estimating population mixing using Y-chromosome markers: mixing of the Han people in China. *Annals of Human Genetics*, 71:407–420, 2007.

Kimura, M. and Weiss, G. The stepping stone model of population structure and the decrease of genetic correlation with distance. *Genetics*, 49:561–576, 1964.

Kingman, J. The coalescent. *Stochastic Processes and their Application*, 1982.

Knowles, L. The burgeoning field of statistical phylogeography. *Journal of Evolutionary Biology*, 17:1–10, 2004.

Knowles, L. and Maddison, W. Statistical phylogeography. *Molecular Ecology*, 11:2623–2635, 2002.

Knuth, D. Sorting and searching. In *The art of Computer programming*, volume 3. Addison-Wesley, 1998.

Kuhner, M. and Felsenstein, J. A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. *Molecular Biology and Evolution*, 11:459–468, 1994.

Larget, B. and Simon, D. Markov chain Monte Carlo algorithms for the Bayesian analysis of phylogenetic trees. *Molecular Biology and Evolution*, 16:750–759, 1999.

Laskey, K. and Myers, J. Population Markov chain Monte Carlo. *Machine Learning*, 50:175–196, 2003.

Latter, B. The island model of population differentiation: A general solution. *Genetics*, 73:147–157, 1973.

Legarreta, L., Manolopoulou, I., Thébaud, C., and Emerson, B. Phylogeography of *Rhinusa vestita* (Coleoptera: Curculionidae) in the Iberian Peninsula: a Bayesian approach. In preparation, 2008.

Li, S., Pearl, D., and Doss, H. Phylogenetic tree construction using Markov chain Monte Carlo. *Journal of the American Statistical Association*, 95:493–508, 2000.

Linz, B., Balloux, F., Moodley, Y., Manica, A., Liu, H., Roumagnac, P., Falush, D., Stamer, C., Prugnolle, F., van der Merwe, S., Yamaoka, Y., Graham, D., Perez-Trallero, E., Wadstrom, T., Suerbaum, S., and Achtman, M. An African origin for the intimate association between humans and *Helicobacter pylori*. *Nature*, 445:915–918, 2007.

Liu, H., Prugnolle, F., Manica, A., and Balloux, F. A geographically explicit genetic model of worldwide human-settlement history. *American Journal of Human Genetics*, 79:230–237, 2006.

Lloyd, D. and Calder, V. Multi-residue gaps, a class of molecular characters with exceptional reliability for phylogenetic analyses. *Journal of Evolutionary Biology*, 4:9–21, 1991.

Maddison, W., Donoghue, M., and Maddison, D. Outgroup analysis and parsimony. *Systematic Zoology*, 33:83–103, 1984.

Makarenkov, V., Kevorkov, D., and Legendre, P. Phylogenetic network construction approaches. *Applied Mycology and Biotechnology*, 6, 2006.

Manolopoulou, I., Brooks, S., and Legarreta, L. A Bayesian framework for analyses of demographic DNA sequence data. In *Proceedings of the 20th Panhellenic Statistics Conference 2007: Statistics and Society*. 2008.

Marjoram, P., Molitor, J., Plagnol, V., and Tavaré, S. Markov chain Monte Carlo without likelihoods. *Proceeding of the National Academy of Science*, 100:15324–15328, 2003.

Markovtsova, L., Marjoram, P., and Tavaré, S. The age of a unique event polymorphism. *Genetics*, 156:401–409, 2000.

Matsuda, H., Ogita, N., Sasaki, A., and Sato, K. Statistical mechanics of population. The lattice Lotka-Volterra model. *Progress of theoretical physics*, 88, 1992.

Mau, B., Newton, M., and Larget, B. Bayesian phylogenetic inference via Markov chain Monte Carlo methods. *Biometrics*, 55:1–12, 1999.

Meligkotsidou, L. and Fearnhead, P. Maximum-likelihood estimation of coalescence times in genealogical trees. *Genetics*, 171:2073–2084, 2005.

Moler, C. and Van Loan, C. Nineteen dubious ways to computer the exponential of a matrix, twenty-five years later. *SIAM Review*, 45:3–49, 2003.

Montiel-Sosa, J., Enriquez, J., and Lopez-Perez, M. Research of single mitochondrial nucleotide substitutions in male infertility should consider human mitochondrial haplogroups. *International Journal of Andrology*, 25:372–373, 2002.

Nei, M. and Kumar, S. *Molecular Evolution and Phylogenetics*. Oxford University Press, 2000.

Nelson, D. SNPs, linkage disequilibrium, human genetic variation and native american culture. *Trends in Genetics*, 17:15–16, 2001.

Neuhauser, C. and Krone, S. The genealogy of samples in models with selection. *Genetics*, 145, 1997.

Newton, M., Mau, B., and Larget, B. Markov chain Monte Carlo for the Bayesian analysis of evolutionary trees from aligned molecular sequences. *Statistics in molecular biology and genetics*, 33:143–162, 1999.

Nielsen, R. and Wakeley, J. Distinguishing migration from isolation: A Markov chain Monte Carlo approach. *Genetics*, 158:885–896, 2001.

Norris, J. *Markov Chains*. Cambridge University Press, 1997.

O'Neill, P., Balding, D., Becker, N., Eerola, M., and Mollison, D. Analyses of infectious disease data from household outbreaks by Markov chain Monte Carlo methods. *Journal Of The Royal Statistical Society Series C*, 49:517–542, 2000.

Panchal, M. The automation of Nested Clade Phylogeographic Analysis. *Bioinformatics*, 23:509–510, 2007.

Panchal, M. and Beaumont, M. The automation and evaluation of nested clade phylogeographic analysis. *Evolution*, 61:1466–1480, 2007.

Pearse, D. and Crandall, K. Beyond f_{ST}: Analysis of population genetic data for conservation. *Conservation Genetics*, 5, 2004.

Petit, R. The coup de grâce for the nested clade phylogeographic analysis? *Molecular Ecology*, 17:516–518, 2008.

Petit, R. and Grivet, D. Optimal randomization strategies when testing the existence of a phylogeographic structure. *Genetics*, 161:469–471, 2002.

Posada, D. and Crandall, K. Intraspecific gene genealogies: trees grafting into networks. *Trends in Ecology and Evolution*, 16:37–45, 2001.

Posada, D., Crandall, K., and Templeton, A. GeoDis: A program for the cladistic nested analysis of the geographical distribution of genetic haplotypes. *Molecular Ecology*, 9:487–488, 2000.

Posada, D., Crandall, K., and Templeton, A. Nested clade analysis statistics. *Molecular Ecology Notes*, 6:590–593, 2006.

Posada, D., Maxwell, T., and Templeton, A. TreeScan: a bioinformatic application to search for genotype/phenotype associations using haplotype trees. *Bioinformatics*, 21:2130–2132, 2005.

Ray, N., Currat, M., Berthier, P., and Excoffier, L. Recovering the geographic origin of early modern humans by realistic and spatially explicit simulations. *Genome Research*, 15:1161–1167, 2005.

Richardson, S. and Green, P. On Bayesian analysis of mixtures with an unknown number of components. *Journal of the Royal Statistical Society. Series B (Methodological)*, 59:731–792, 1997.

Ripley, B. *Stochastic Simulation*. Wiley & Sons, 1987.

Robert, C. and Casella, G. *Monte Carlo Statistical Methods*. Springer, 2004.

Rosenberg, N. and Nordborg, M. Genealogical trees, coalescent theory and the analysis of genetic polymorphisms. *Nature Genetics*, 3:380–390, 2002.

Rzhetsky, A. and Nei, M. Theoretical foundation of the minimum-evolution method of phylogenetic inference. *Molecular Biology and Evolution*, 10:1073–1095, 1993.

Sankoff, D. Minimal mutation trees of sequences. *SIAM Journal of Applied Mathematics*, 28:35–42, 1975.

Scott, S. and Wang, T. Label switching in finite mixture models: When to worry and what to do. Presentation for Valencia 8 meeting, 2006.

Semple, C. and Steel, M. *Phylogenetics*. Oxford University Press, 2003.

Slatkin, M. Simulating genealogies of selected alleles in a population of variable size. *Genetic Research*, 78:49–57, 2001.

Stamatakis, A. *Distributed and Parallel Algorithms and Systems for Inference of Huge Phylogenetic Trees based on the Maximum Likelihood Method*. Ph.D. thesis, Technische Universität München, Germany, 2004.

Stephens, M. Dealing with label-switching in mixture models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 62:795–809, 2000.

Stephens, M. and Donnelly, P. Inference in molecular population genetics. *Journal of the Royal Statistical Society. Series B (Methodological)*, 62:605–655, 2000.

Tadesse, M. G., Sha, N., and Vannucci, M. Bayesian variable selection in clustering high-dimensional data. *Journal of the American Statistical Association*, 100:602–617(16), 2005.

Tavaré, S. Some probabilistic and statistical problems in the analysis of DNA sequences. *Lectures on Mathematics in the Life Sciences,*, 17:57–86, 1986.

Tavaré, S. *Coalescent Theory*, volume 1. Nature Publishing Group, 2003.

Templeton, A. Nested clade analysis of phylogeographic data: testing hypotheses about gene flow and population history. *Molecular Ecology*, 7:381–397, 1998.

Templeton, A. Statistical phylogeography: methods of evaluating and minimizing inference errors. *Molecular Ecology*, 13:789–809, 2004.

Templeton, A., Boerwinkle, E., and Sing, C. 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*, 117:343–351, 1987.

Templeton, A., Crandall, K., and Sing, C. A cladistic analysis of phenotypic associations with haplotypes inferred from restriction endonuclease mapping. III. Cladogram estimation. *Genetics*, 132:619–633, 1992.

Templeton, A., Maxwell, T., Posada, D., Stengård, J., Boerwinkle, E., and Sing, C. Tree scanning: a method for using haplotype trees in phenotype/genotype association studies. *Genetics*, 169:441–453, 2005.

Templeton, A. and Sing, C. A cladistic analysis of phenotypic associations with haplotypes inferred from restriction endonuclease mapping. IV. Nested analyses with cladogram uncertainty and recombination. *Genetics*, 134:659–669, 1993.

Tuffley, C. and Steel, M. Links between maximum likelihood and maximum parsimony under a simple model of site substitution. *Bulletin for Mathematical Biology*, pages 582–607, 1997.

Van Loan, C. Nineteen dubious ways to computer the exponential of a matrix. *SIAM Review*, 20:801–836, 1978.

Wakeley, J. *Coalescent Theory: an introduction*. Roberts and Company Publishers, 2008.

Wakeley, J. and Hey, J. Estimating ancestral population parameters. *Genetics*, 145:847–855, 1997.

Wilson, I., Weale, M., and Balding, D. Inferences from DNA data: population histories, evolutionary processes and forensic match probabilities. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 166:155–188, 2003.

Wright, S. The genetical structure of populations. *Annals of Eugenics*, 15:323–354, 1951.

Xu, S. Phylogenetic analysis under reticulate evolution. *Molecular Biology and Evolution*, 17:897–907, 2000.

Yang, Z. and Rannala, B. Bayesian phylogenetic inference using DNA sequences: a Markov chain Monte Carlo Method. *Molecular Biology and Evolution*, 14:717–724, 1997.