Inverse problems in evolutionary biology

Gordon Hiscott

a thesis submitted for the degree of

Doctor of Philosophy

at the University of Otago, Dunedin,

New Zealand.

July 2017

Abstract

In this thesis, we explore three techniques which could be used to increase the efficiency of analyses in evolutionary genetics while still producing reasonably accurate results. The first of these methods improves the efficiency of analyses based on Markov chain Monte Carlo (MCMC) through the application of delayed acceptance sampling, an MCMC method with an additional proposal step in which an acceptance probability is computed from computationally less expensive approximate likelihoods. Rejection at the additional decision step should allow software like SNAPP ("SNP and AFLP Phylogenies") to avoid unnecessary computation of full likelihoods and, therefore, run more efficiently. The second method we discuss combines dynamic programming with classical numerical integration methods to compute likelihoods with respect to continuous trait models on trees. This method assumes explicitly known transition densities, but is efficient and has a relatively fast convergence rate. We apply the method to a threshold model which combines continuous traits with discrete observations. The third method we look at is another dynamic programming integration algorithm, except that this algorithm takes advantage of a basis function approximation of likelihood functions. This method allows for numerical solutions to PDEs to be applied directly and the use of Chebyshev polynomials as the basis functions make the method easy to implement. We apply this method to the computation of the likelihood given a genetic data set generated by diffusion processes.

Contributions

The research reported in Chapter 2 was done in collaboration with Prof. David Bryant, with input from Assoc. Prof. Colin Fox. I designed the algorithms, implemented multiple simulation experiments and am responsible for the final text.

The research reported in Chapter 3 was published in GBE, co-authored by Prof. David Bryant, Dr. Matthew Parry, and Assoc. Prof. Colin Fox; I was first author of the published paper. I helped design the algorithms, implemented and applied techniques, carried out the data analysis, and helped write the text of the paper. I am responsible for the final version of the text in the thesis.

The research reported in Chapter 4 was done in collaboration with Prof. David Bryant, with input from Assoc. Prof. Colin Fox. I helped design the algorithms, implemented the algorithms and am responsible for the final text.

Acknowledgements

I would like to thank my supervisor Prof. David Bryant for the undying support and invaluable advice that he has given me throughout my PhD program and for offering me a PhD position at the University of Otago. I have gained immensely from working with him and his patience and friendliness have made him a pleasure to work with. I also thank my co-supervisors Dr. Matthew Parry and Assoc. Prof. Colin Fox for their input and support.

I was supported by a PhD scholarship from the Allan Wilson Centre of Molecular Ecology and Evolution. My involvement with the centre allowed me to meet with several talented and kind-hearted people who work or have worked in phylogenetics. My experience with the centre also immensely increased my knowledge of and experience in phylogenetics and applications of mathematics in the field. It deeply saddens me that the centre was shut down in 2015; its closure has been a tremendous loss to the field of phylogenetics.

I thank my mother Janice, my father Michael, my brother Andrew, his husband Brian, and everyone else in my family for their support and encouragement from my home country of Canada. I also thank the friends that I met prior to my PhD program and the friends I met throughout my program for their support and friendship throughout this long and difficult task.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

An organism can be analyzed by its phenotype, which is the set of observable physical characteristics. Another way to analyze an organism is by its genotype, which is the genetic makeup of the organism and the component which determines the phenotype. In evolutionary biology, there are analyses which determine or infer the evolutionary history of a group of related species based on the known phenotypes and/or genotypes of organisms. Such analyses can help give a better idea of how or when certain biological traits or organisms originated.

Generally, the evolutionary history of a set of species is mapped in the form of an evolutionary tree, or dendrogram, as shown in Figure 1.1. While a dendrogram can show the evolutionary history of species (in which case it is a species tree), it can also show the evolutionary history of individual genes in separate individuals (in which case it is a gene tree). Many models which describe changes in evolutionary history incorporate the coalescent model, a reverse-time death process in which the state space is the set of ancestral lineages (lines of descent from one species to a descendant species) in the sample of organisms and in which lineages meet and merge, or coalesce, at common ancestors (Kingman, 1982a,b). The evolutionary models also incorporate mutation models to describe the mutation of organisms over time (see pages 178-182 of Hamilton (2009) and Bryant *et al.* (2012)). There are other factors that affect the evolution and prevalence of organisms such as geographical location (Notohara, 1990); however, the models discussed in this thesis mainly focus on coalescence and mutation.

There are algorithms which take genetic data as input and find the likeliest structure(s) of the evolutionary tree dendrogram connecting the organisms the data came from. One of these programs is SNAPP ("SNP and AFLP Phylogenies") (Bryant *et al.*, 2012), which takes in binary genetic data obtained from sites and produces samples

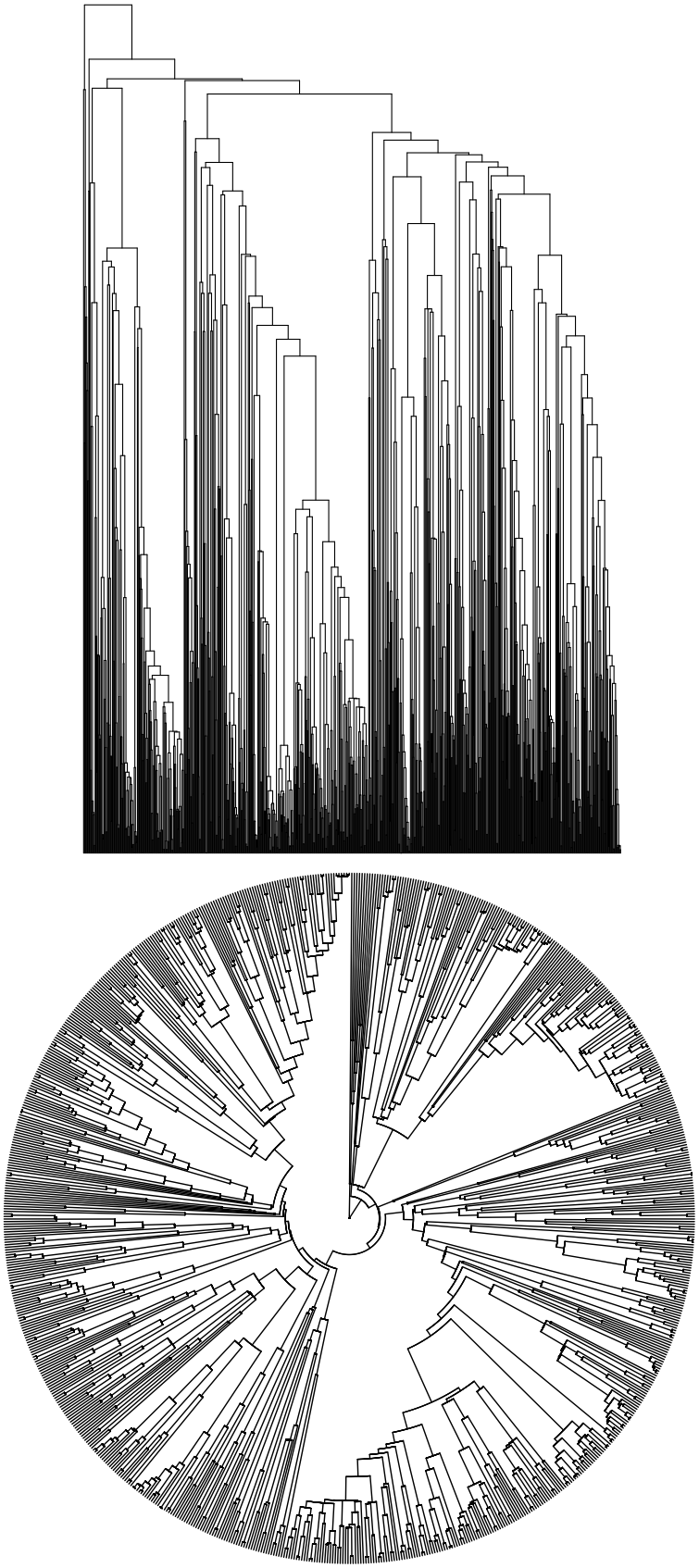

Figure 1.1: A basic dendrogram (top) and a circular dendrogram (bottom). Both are based on a phylogeny from Simon *et al.* (2009).

from the posterior distribution on evolutionary tree structures given that data, through the use of MCMC (Markov Chain Monte Carlo). SNAPP computes a conditional likelihood of a species tree and relevant parameters given genetic data while assuming each site's data come from an independent tree. This likelihood integrates out the gene trees thus removing a parameter to sample and increasing efficiency. Even with this increase in efficiency, the likelihood computations in SNAPP can still cost a large amount of computation time. In this thesis, we look at methods which could improve the efficiency of SNAPP while still yielding reasonably accurate results.

The first method we explore is the modification of the sampling procedure in SNAPP to reduce computation time. In particular, we look at a modification of a commonly used MCMC method called the Metropolis-Hasting algorithm (see Gilks *et al.* (1996)). The Metropolis-Hastings algorithm approximates the posterior probability distribution $\pi(\theta)$, where θ is a set of parameters, by proposing a series of samples of θ; each sample is either stored or rejected depending on an acceptance probability. If we let θ_i denote the state of θ at iteration i, let $q(.|\theta)$ denote the proposal distribution from which samples of θ are proposed, and let $f(\theta)$ denote a distribution proportional to $\pi(\theta)$, the algorithm can be written as follows:

- Choose an initial sample θ_1.
- At each iteration $i \in \{1, ..., N\}$,
  1. Sample proposal θ' from $q(.|\theta_i)$.
  2. Compute acceptance probability
  $$\alpha(\theta_i, \theta') = \min\left\{1, \frac{f(\theta')q(\theta_i|\theta')}{f(\theta_i)q(\theta'|\theta_i)}\right\}$$
  3. With probability $\alpha(\theta_i, \theta')$, accept the proposal and set $\theta_{i+1} = \theta'$. Otherwise, reject the sample and set $\theta_{i+1} = \theta_i$.

The modified Metropolis-Hastings algorithm we explore is called the delayed acceptance Metropolis-Hastings algorithm, which was developed by Christen and Fox (2005) and is also discussed in Fox *et al.* (2013). In the delayed acceptance Metropolis-Hastings algorithm, each iteration has an additional proposal step where an approximate acceptance probability is calculated from approximate likelihoods to determine if a sample is accepted or rejected; if the sample is rejected at this step, it is rejected entirely and the next iteration begins. Otherwise, the sample is subject to a decision step wherein an acceptance probability is computed based on full likelihoods; this

step determines whether or not the sample is ultimately accepted. The approximate likelihoods we use are computationally less expensive to compute than the exact likelihoods. Therefore, discarding samples that do not pass the additional proposal step reduces the computation time significantly by avoiding unnecessary calculations of the actual non-approximate likelihood. To approximate the likelihood for our particular implementation of delayed acceptance, we compute the genetic distance (the expected number of pairwise genetic differences) between species and approximate the likelihood via a multivariate Gaussian density.

The problem with our Gaussian approximation is that it assumes that the available data consists of both polymorphic sites (sites where the individuals are not all in the same state) and non-polymorphic sites. However, SNAPP only takes in polymorphic site data, which prevents accurate approximation of genetic distances. We therefore modify the delayed acceptance algorithm to correct the likelihood for lack of non-polymorphic sites. We implemented delayed acceptance sampling into SNAPP and ran experiments with SNAPP with delayed acceptance and without delayed acceptance. In Chapter 2, we present analyses of the results we obtained from our experiments. Upon analyzing the results, we concluded that the Gaussian approximation leads to poor mixing and, therefore, is not suitable for SNAPP. The difficulties in using genetic distances to approximate likelihoods motivated the work we accomplished on the methods that follow.

The second method we look at is the efficient computation of likelihoods over trees through the use of numerical integration and dynamic programming. We originally developed this algorithm to obtain improved approximate likelihoods for delayed acceptance, however we soon found that the method had far wider application. Numerical integration refers to a wide range of techniques for integral approximation including methods of quadrature, or methods for approximation via weighted sums of a finite number of discrete values (Davis and Rabinowitz, 1984; Dahlquist and Björck, 2008; Cheney and Kincaid, 2012). These sums are calculated by first dividing the interval of integration into N smaller sub-intervals, then approximating the area within each sub-interval. Let $f(x)$ be a function on $x \in [a, b]$, let n be the number of integration points, and let x_i denote an integration point for $i \in \{1, ..., n\}$. Quadrature approximates the integral of $f(x)$ over $[a, b]$ as

$$\int_a^b f(x)dx \approx \sum_{n=1}^{N} w_i f(x_i) \tag{1.1}$$

where x_i and the weights w_i depend on the quadrature rule used.

In Chapter 3, we apply two quadrature rules to a dynamic programming algorithm which computes likelihood values at specific points and stores the values to avoid unnecessary recalculation. One of these quadrature rules is the commonly used Simpson's rule, of which the formula is

$$\int_a^b f(x)dx \approx \sum_{n=1}^{N} \frac{h}{6}\left(f\left(a+(n-1)h\right)+4f\left(a+\left(n-\frac{1}{2}\right)h\right)+f\left(a+nh\right)\right) \quad (1.2)$$

where N is the number of evenly-spaced subintervals of $[a,b]$ and where $h = \frac{b-a}{N}$. The other integration rule we discuss in Chapter 3 is a rule in which the weights are tailored to better approximate an integral involving a Gaussian function. The method we discuss is efficient and has a relatively fast convergence rate; however, it assumes that the integrands are explicitly available and is capable of generating large integration errors when N is small.

We applied the dynamic programming algorithm we describe in Chapter 3 to compute the likelihood of a species tree given observed physical traits (or phenotype) under the threshold model of Wright (1934) and Felsenstein (2005, 2012). We used this algorithm to analyze datasets on extrafloral nectaries on an 839-taxon phylogeny (Marazzi *et al.*, 2012). We compare the results of our algorithm to results of likelihood computations under a precursor model published in Marazzi *et al.* (2012); this comparison has been published in Hiscott *et al.* (2016).

For the method described in Chapter 4, we once again use dynamic programming and integration. However, for each function $f(x)$ that we integrate, we express $f(x)$ as a weighted sum of basis functions and then compute the integral of $f(x)$ based on the basis functions. This method is a generalization of the numerical integration method in Chapter 3 and numerical solutions to PDEs can be applied directly to this method. In Chapter 4, we describe a dynamic programming algorithm which interpolates each function as a linear combination of Chebyshev polynomials and stores integrals of a particular form instead of point-specific data. Chebyshev polynomials are a family of commonly used orthogonal polynomials which are of the form $T_n(x) = \cos\left(n \arccos(x)\right)$ (Mason and Handscomb, 2003). We use Chebyshev polynomials because their properites and identities make them relatively easy to apply to likelihood calculations.

We apply our Chebyshev polynomial approach to diffusion processes, stochastic processes which are defined by PDEs called the Kolmogorov forward equation and the Kolmogorov backward equation (see pages 137-138 of Ewens (2004)). In particular, we compute the likelihood of a species tree and related parameters given binary genetic

data when gene frequency changes according to a diffusion process. We ran experiments on different trees with data simulated via diffusion processes; in these experiments, we compute likelihoods through a combination of Chebyshev expansion, the fourth order Runge-Kutta method, and other integration approximation techniques. Our results are consistent with spectral convergence with respect to the number of basis functons and fourth-order convergence with respect to the number of time steps of the Runge Kutta method.

Chapter 2

Delayed acceptance strategies for inferring evolutionary parameters

There exist several methods for inferring the evolutionary history of a sample of populations based on the multispecies coalescent model and given a sample of genetic data. Early inference methods used genetic distances; more recent inference methods have used the full likelihood of the tree and relevant parameters given the data. SNAPP ("SNP and AFLP Phylogenies") is an algorithm which computes a likelihood by integrating over gene trees analytically with an efficient (polynomial time) integration algorithm (Bryant *et al.*, 2012).

The problem with the likelihood calculations employed by SNAPP is that they can be very expensive in terms of computation time, which is especially a problem when the chain is slow to converge. In this chapter, we explore a newly developed technique for making MCMC computations more efficient. The technique is called *delayed acceptance sampling* and was developed by Fox *et al.* (2013). It is related to the surrogate transition method (see Liu and Chen (1998) and Liu (2013)). The application of the method relies on the availability of a rapidly computed approximate likelihood function; we implement an approximation based on classical genetic distance-based methods.

As an intermediate step, we develop the delayed acceptance sampler for the case that all sites, polymorphic and non-polymorphic, are present in the data. In general, however, SNAPP is only provided with polymorphic sites, a fact which makes it impossible to accurately estimate the genetic distances used in the approximate likelihoods. To counter this, we modify the approximate likelihood so that it integrates out the total number of sites. We ran SNAPP with and without delayed acceptance sampling

and our results from both chains are described in Section 2.6.

2.1 Introduction to delayed acceptance sampling

The delayed acceptance Metropolis-Hastings algorithm (Fox *et al.*, 2013) adds an additional proposal step to the standard Metropolis-Hastings algorithm. In this step, an acceptance probability based on an approximate and computationally less expensive likelihood is used in place of an acceptance probability based on the full likelihood. If the proposal is rejected using the approximate likelihood, then the proposed state is rejected and unnecessary computation of the true likelihood is avoided. If the proposal is accepted then a modified acceptance test is conducted using the full likelihood. Samples obtained using the delayed acceptance algorithm are still from the same full posterior distribution as samples obtained from MCMC without delayed acceptance.

Let D denote the observed data, let θ denote the set of parameters, and let θ_i denote the state of θ at iteration i of the algorithm. In addition, let $P(\theta|D)$ denote the posterior density of θ given D, let $P^*_{\theta^*}(\theta)$ denote an approximation of $P(\theta|D)$ which could depend on state θ^*, and let $q(\cdot|\theta)$ denote the distribution of proposals given the current state of θ. The delayed acceptance algorithm described in Fox *et al.* (2013) is then written as follows:

- Choose an initial sample θ_1.
- At each iteration $i \in \{1, ..., N\}$,
  1. Sample proposal θ' from $q(\cdot|\theta_i)$.
  2. With acceptance probability
  $$\alpha\left(\theta'|\theta_i\right) = \min\left\{1, \frac{P^*_{\theta_i}(\theta')q\left(\theta_i|\theta'\right)}{P^*_{\theta_i}(\theta_i)q\left(\theta'|\theta_i\right)}\right\}, \tag{2.1}$$
  proceed to the next step. Otherwise, reject the proposal, set $\theta_{i+1} = \theta_i$, and end iteration.
  3. With acceptance probability
  $$\beta\left(\theta'|\theta_i\right) = \min\left\{1, \frac{P(\theta'|D)\alpha\left(\theta_i|\theta'\right)q\left(\theta_i|\theta'\right)}{P(\theta_i|D)\alpha\left(\theta'|\theta_i\right)q\left(\theta'|\theta_i\right)}\right\}, \tag{2.2}$$
  accept the proposal and set $\theta_{i+1} = \theta'$. Otherwise, reject the sample and set $\theta_{i+1} = \theta_i$.

Christen and Fox (2005) show that θ_i has the correct equilibrium distribution (given standard restrictions on q). The intuition behind the method is that steps 1 and 2 constitute a proposal given by a step in approximate MCMC chain, and this proposal forms the basis for the full likelihood MCMC step.

Fox *et al.* (2013) state that if the approximations $P^*(\theta|D)$ are not state dependent, then (2.2) simplifies to

$$\beta\left(\theta'|\theta_i\right) = \min\left\{1, \frac{P(\theta'|D)P^*\left(\theta_i|D\right)}{P(\theta_i|D)P^*\left(\theta'|D\right)}\right\}. \tag{2.3}$$

Delayed acceptance MCMC generates a different chain than standard MCMC, but the two chains have the same stationary distribution. There will be situations or iterations where delayed acceptance MCMC rejects a sample where standard MCMC would accept the sample. The idea, however, is that these rejections are cheap, or have little effect on the total computation time. The closer the approximate density is to the original density, the more efficient delayed acceptance MCMC is.

2.2 SNAPP and the multispecies coalescent

2.2.1 The coalescent

One of the most important and most influential models in population genetics is the coalescent, a model which describes the genealogy, or genetic history, of a sample from a population from which genetic data is collected (Kingman, 1982a,b). The coalescent is a reverse time stochastic model in which the state space is the set of ancestral lineages (where a lineage is line of descent from an individual back to an ancestor) and in which lineages coalesce (in other words, meet and merge) at common ancestors, thus decreasing the number of ancestral lineages (Kingman, 1982a,b). Two other factors of the genealogy modeled by the coalescent are mutation, which is the change in genotype (genetic makeup), and recombination (when a lineage has two parental lineages); however, the models we look at in this chapter only focus on coalescence and mutation.

2.2.2 Multispecies coalescent

Although the coalescent was originally developed for a single population (Kingman, 1982a,b), it has since been extended to multiple species (Pamilo and Nei, 1988; Maddison, 1997; Rosenberg and Nordborg, 2002; Degnan and Rosenberg, 2009). In the multispecies coalescent model, individuals are separated into multiple populations (contem-

porary and ancestral), which are connected to each other in a phylogeny commonly referred to as a species tree. Within each population, the individuals/genes change according to the single population coalescent model; the phylogeny which shows the evolutionary history of a single gene is commonly referred to as a gene tree.

2.2.3 Inference under the multispecies coalescent

The first inference methods based on multispecies coalescent models used estimates of genetic distances, or the average number of pairwise allele or nucleotide differences between individuals, with different estimators explored including those discussed in Cavalli-Sforza and Edwards (1967); Nei (1972, 1987); Reynolds *et al.* (1983). Examples of methods used to infer phylogenies from genetic distance include UPGMA (Sokal and Michener, 1958), neighbour joining (Saitou and Nei, 1987), and the distance Wagner method (Farris, 1972). These methods are applied under the assumption that genetic distance is proportional to time of separation.

More recently, inference methods which use the full likelihood of the tree and the associated parameters given the data have been developed. An advantage to using the full likelihood is that the effective population size of a population can be inferred more accurately through a maximum likelihood method using the full likelihood than by using genetic distance (Felsenstein, 2009). The maximum likelihood estimate of the effective population size in Felsenstein (2009) also uses the information from all coalescence events whereas more than half of the information of the genetic distance based estimate is obtained from the earliest coalescence events. As Nielsen *et al.* (1998) show, a maximum likelihood estimation of population divergence times has lower variance than F_{ST}-based estimates computed by a formula in Reynolds *et al.* (1983), a formula commonly used to estimate F_{ST}, which is defined in Wright (1951) as "the correlation between random gametes, drawn from the same subpopulation, relative to the total." In addition, Nielsen *et al.* (1998) show that two samples can generate the same F_{ST} estimates but different likelihoods, indicating that the maximum likelihood approach uses more information than the approach using the formula in Reynolds *et al.* (1983). Another advantage of full likelihood algorithms is that one can incorporate ascertainment bias, or a bias in the data resulting from the ascertainment or collection method used (Kuhner *et al.*, 2000; Gelman, 2004; RoyChoudhury and Thompson, 2012). Later in this chapter, we discuss an application of ascertainment bias correction methods to our algorithm for the case where the observed data consists only of polymorphic sites, or sites where the individuals are not all in the same state.

Full likelihood approaches which incorporate finite site models for sequence mutations include an approach by Rannala and Yang (2003) and software named *BEAST (Heled and Drummond, 2010). The approach in Rannala and Yang (2003) assumes that the species tree topology is known and uses MCMC sampling to infer ancestral population sizes and species divergence times by integrating over the set of possible gene tree topologies and branch lengths. *BEAST (Heled and Drummond, 2010) uses MCMC sampling to infer species trees and relevant parameters from genetic sequence data through an evolutionary model composed of three components: the multispecies coalescent model, a model for changes in sequence alignment over a gene tree, and a prior density for the species tree. The prior for the species tree is the product of a prior on the divergence times (or branch lengths) and a prior on population sizes. The approach in Rannala and Yang (2003) and *BEAST (Heled and Drummond, 2010) requires simultaneous sampling of gene trees at every locus, severely limiting the number of individuals and genes that can be analyzed. An alternative is integrating over the entire set of gene trees given the species tree topology and the branch lengths. Each likelihood calculation is therefore computationally intensive, making this type of method a good candidate for optimizations such as delayed acceptance. SNAPP is an inference algorithm which uses an efficient algorithm to integrate gene trees in the case of unlinked binary alleles (Bryant *et al.*, 2012); SNAPP is the algorithm we focus on for the remainder of this chapter.

2.3 Applying delayed acceptance in the absence of acquisition error

2.3.1 Acquisition error

The genetic data analyzed by SNAPP is acquired from biallelic markers restricted to polymorphic sites, or sites where the individuals are not all in the same state (Bryant *et al.*, 2012). This causes acquisition error, error which causes bias in the parameter estimates and prevents calculation of genetic distances. In this section, we ignore ascertainment bias and describe methods and computations which assume that the data is complete, including both polymorphic and non-polymorphic sites. In Section 2.4, we discuss a method which includes methods that correct for acquisition error.

2.3.2 Approximate likelihood of parameters given complete data

The key step when implementing a delayed acceptance algorithm is the choice of approximation for the target density. A good choice can make the chain run much faster, while a poor choice can make it run very much slower.

For this problem, we want to obtain an approximation $P^*(\theta|D)$ for the posterior density of θ, the parameter set which consists of the species tree and relevant parameters. To find this approximation, we first make the approximation

$$P(\theta|D) \approx P(\theta|\widehat{d}(D))$$

so that we sample from the posterior given $\widehat{d}(D)$, the vector of pairwise distances computed from the data D, instead of sampling given the whole data. By Bayes theorem, we obtain

$$P\left(\theta \left| \widehat{d}(D) \right.\right) = P\left(\widehat{d}(D) \left| \theta \right.\right) \frac{P(\theta)}{P\left(\widehat{d}(D)\right)}. \tag{2.4}$$

We then make the approximation

$$P^*\left(\widehat{d}(D) \left| \theta \right.\right) = \phi\left(\widehat{d}(D) \left| \mu, \hat{\Sigma} \right.\right) \tag{2.5}$$

where

- the vector $\widehat{d} = \widehat{d}(D)$ contains the $\frac{1}{2}m(m+1)$ genetic distances estimates computed from the data D, where m is the number of species or populations.
- the vector $\mu = d(\theta)$ contains the expected genetic distances given the set of parameters θ.
- the matrix $\hat{\Sigma}$ is a sample covariance matrix, computed from the data.
- $\phi(x|\mu, \Sigma)$ denotes the density of the multivariate Gaussian with mean μ and covariance matrix Σ.

The computation of $\widehat{d}(D)$, $\mu = d(\theta)$ and $\hat{\Sigma}$ is described below.

Empirical distances: Let $d_{(i,i)}$ denote the probability that two individuals sampled with replacement from population i have different states at a randomly chosen site. In addition, let n_i denote the total number of individuals in population i, let a_{0i}^k denote the number of individuals in population i in state 0 at site k, and let a_{1i}^k denote

the number of individuals in population i in state 1 at site k. We can estimate $d_{(i,i)}$ from data D by

$$\widehat{d}_{(i,i)}(D) = \frac{1}{K}\sum_{k=1}^{K}\frac{a_{0i}^k a_{1i}^k}{\frac{1}{2}n_i^2} = \frac{2}{K}\sum_{k=1}^{K}\left(\frac{a_{0i}^k}{n_i}\right)\left(\frac{a_{1i}^k}{n_i}\right) = \frac{2}{K}\sum_{k=1}^{K} g_{0i}^k g_{1i}^k \tag{2.6}$$

where K is the number of sites, g_{0i}^k is the proportion of alleles in population i in state 0 at site k, and g_{1i}^k is the proportion of alleles in population i in state 1 at site k.

Let $d_{(i,j)}$ denote the probability that an individual sampled from population i has a different state than an individual sampled from population j at a randomly chosen site. When $i \neq j$, $d_{(i,j)}$ can be estimated from D as

$$\begin{aligned}\widehat{d}_{(i,j)}(D) =& \frac{1}{K}\sum_{k=1}^{K}\frac{a_{0i}^k a_{1j}^k + a_{1i}^k a_{0j}^k}{n_i n_j} \\ =& \frac{1}{K}\sum_{k=1}^{K}\frac{g_{0i}^k a_{1j}^k + g_{1i}^k a_{0j}^k}{n_j} \\ =& \frac{1}{K}\sum_{k=1}^{K}\left(g_{0i}^k g_{1j}^k + g_{0j}^k g_{1i}^k\right).\end{aligned} \tag{2.7}$$

Expected genetic distances: The multi-species coalescent model defines a random process for generating sequence data given a species tree and parameters. We can therefore define the expected values for $d_{(i,i)}$ and $d_{(i,j)}$ under this model. The computations here are similar to those in Heled (2012), except that we compute expectations with respect to a fixed species tree and we also deal with variation in effective population size along different branches of that tree.

To compute the expected genetic distances $d_{(i,j)}(\theta)$, we adopt a two state mutation model with mutation rates u and v, where u is the mutation rate from state 0 to state 1 and v is the mutation rate from state 1 to state 0; both of these rates are estimated from the data. Using an eigendecomposition of the rate matrix (Rodriguez *et al.*, 1990), it can be shown that the expected genetic distance $g(t)$ between two randomly selected individuals given coalescence time τ is

$$g(\tau) = 2\pi_0\pi_1\left(1 - e^{-2(u+v)\tau}\right),$$

where $\pi_0 = \frac{v}{u+v}$ and $\pi_1 = \frac{u}{u+v}$ are the stationary frequencies of alleles 0 and 1 respectively. To compute $d_{(i,j)}(\theta)$, we take expectations over τ, giving

$$d_{(i,j)}(\theta) = \int_0^\infty g(\tau) f_{(i,j)}(\tau)\,\mathrm{d}\tau, \tag{2.8}$$

where $f_{(i,j)}(\tau)$ is the probability density for the coalescence time between any individual from population i and any individual from population j. Note that $d_{(i,j)}(\theta)$ will always lie between 0 and 1.

For each node k on the population tree, let t_k denote the time where the node occurs. If k is a leaf, then $t_k = 0$; otherwise, t_k is the time when the speciation event occurs. In addition, for each node k below the root (we denote the root r), let λ_k denote the coalescence rate within the population along the branch between nodes k and p_k, where p_k is the parent node of k. Given two leaves i and j, let $w(i,j)$ denote the most recent common ancestor of i and j; if $i = j$, then $w(i,j) = i$. For an example of the process for determining each t_k and each $w(i,j)$, see Figure 2.1.

Let τ denote the random variable for the coalescence time between an individual sampled from population i and an individual sampled from population j, where i and j can be the same population or two different populations. For any branch (k, p_k) along the path from $w(i,j)$ and the root r, we have the conditional densities

$$f\left(\tau | t_k \leq \tau < t_{p_k}\right) = \frac{\lambda_k e^{-\lambda_k(\tau - t_k)}}{1 - e^{-\lambda_k(t_{p_k} - t_k)}} \tag{2.9}$$

and the conditional probability

$$P\left(\tau \geq t_{p_k} | \tau \geq t_k\right) = e^{-\lambda_k\left(t_{p_k} - t_k\right)}. \tag{2.10}$$

In addition, for any i and j, we have the conditional density for coalescence time above the root

$$f\left(\tau | \tau \geq t_r\right) = \lambda_r e^{-\lambda_r(\tau - t_r)}. \tag{2.11}$$

Since $w(i,j)$ is the most recent common ancestor that leaf i shares with leaf j,

$$P\left(\tau < t_{w(i,j)}\right) = 0. \tag{2.12}$$

Equations (2.9) to (2.12) determine the conditional density

$$f_{(i,j)}(\tau) = f\left(\tau | \tau \geq t_{w(i,j)}\right)$$

and its mgf (moment generating function)

$$M_{w(i,j)}(x) = \int_0^\infty e^{x\left(\tau - t_{w(i,j)}\right)} f\left(\tau | \tau \geq t_{w(i,j)}\right) \, \mathrm{d}\tau \tag{2.13}$$

for $x \leq 0$. The mgf is calculated as follows:

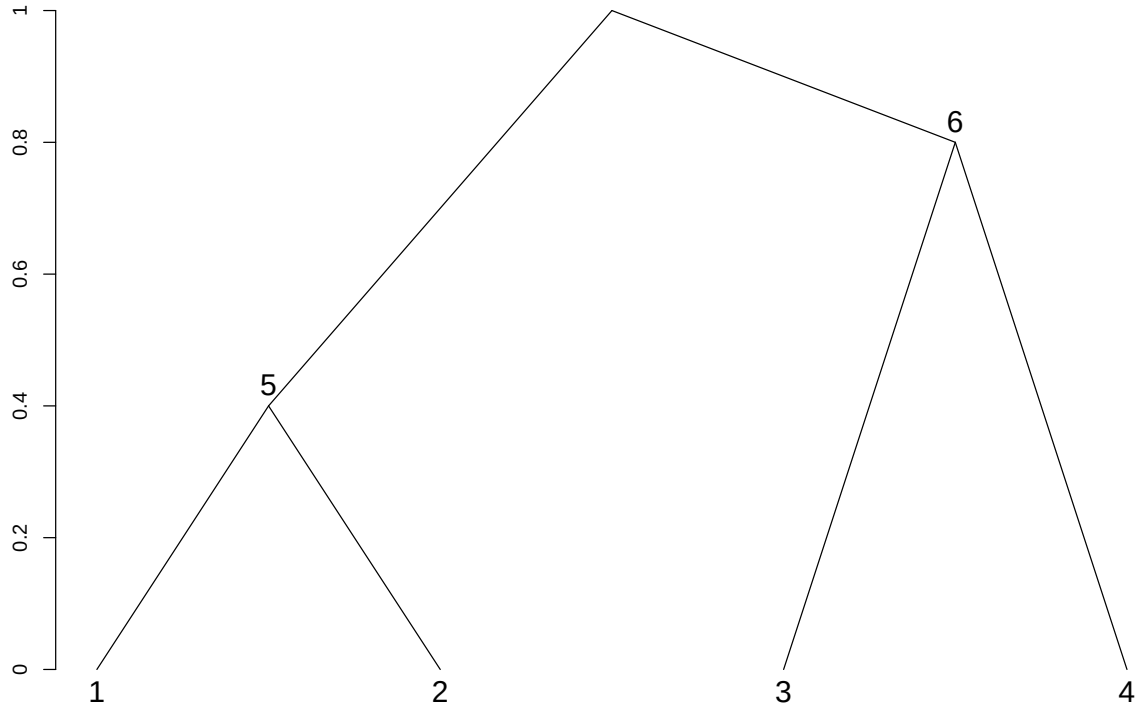

Figure 2.1: The axis on the left represents the amount of time before the present. Nodes 1 through 4 are all leaves, so $t_1 = t_2 = t_3 = t_4 = 0$. Node 5 is at 0.4 units of time in the past, so $t_5 = 0.4$; in addition, $t_6 = 0.8$ and $t_r = 1$. Since leaves 1 and 2 are child nodes of node 5, $w(1, 2) = 5$; similarly, since leaves 3 and 4 are child nodes of node 6, $w(3, 4) = 6$. Since the root is the only common ancestor leaf 3 shares with leaves 1 and 2, $w(1, 3) = r$ and $w(2, 3) = r$; for similar reasons, $w(1, 4) = r$ and $w(2, 4) = r$.

- If $w(i,j) = r$, then

$$\begin{aligned} M_{w(i,j)}(x) =& \int_0^\infty e^{x(\tau - t_r)} f\left(\tau | \tau \geq t_r\right) \, \mathrm{d}\tau \\ =& \int_{t_r}^\infty e^{x(\tau - t_r)} f\left(\tau | \tau \geq t_r\right) \, \mathrm{d}\tau \\ =& \int_{t_r}^\infty e^{x(\tau - t_r)} \lambda_r e^{-\lambda_r(\tau - t_r)} \, \mathrm{d}\tau \\ =& \int_{t_r}^\infty \lambda_r e^{(x-\lambda_r)(\tau - t_r)} \, \mathrm{d}\tau \\ =& \frac{\lambda_r}{\lambda_r - x} \end{aligned} \tag{2.14}$$

- If $w(i,j) \neq r$, then, letting $w = w(i,j)$,

$$\begin{aligned} M_w(x) =& \int_0^\infty e^{x(\tau - t_w)} f\left(\tau | \tau \geq t_w\right) \, \mathrm{d}\tau = \int_{t_w}^\infty e^{x(\tau - t_w)} f\left(\tau | \tau \geq t_w\right) \, \mathrm{d}\tau \\ =& \int_{t_w}^{t_{p_w}} e^{x(\tau - t_w)} P\left(\tau < t_{p_w} | \, \tau \geq t_w\right) f\left(\tau | \, t_w \leq \tau < t_{p_w}\right) \, \mathrm{d}\tau \\ &+ \int_{t_{p_w}}^\infty e^{x(\tau - t_w)} P\left(\tau \geq t_{p_w} | \, \tau \geq t_w\right) f\left(\tau | \tau \geq t_{p_w}\right) \, \mathrm{d}\tau \\ =& \int_{t_w}^{t_{p_w}} e^{x(\tau - t_w)} \left(1 - e^{-\lambda_w(t_{p_w} - t_w)}\right) \frac{\lambda_w e^{-\lambda_w(\tau - t_w)}}{1 - e^{-\lambda_w(t_{p_w} - t_w)}} \, \mathrm{d}\tau \\ &+ \int_{t_{p_w}}^\infty e^{x(\tau - t_w)} e^{-\lambda_w(t_{p_w} - t_w)} f\left(\tau | \tau \geq t_{p_w}\right) \, \mathrm{d}\tau \\ =& \int_{t_w}^{t_{p_w}} \lambda_w e^{(x-\lambda_w)(\tau - t_w)} \, \mathrm{d}\tau \\ &+ e^{(x-\lambda_w)(t_{p_w} - t_w)} \int_{t_{p_w}}^\infty e^{x(\tau - t_{p_w})} f\left(\tau | \tau \geq t_{p_w}\right) \, \mathrm{d}\tau \\ =& \frac{\lambda_w}{\lambda_w - x} \left(1 - e^{(x-\lambda_w)(t_{p_w} - t_w)}\right) + e^{(x-\lambda_w)(t_{p_w} - t_w)} M_{p_w}(x) \end{aligned} \tag{2.15}$$

Therefore, the probability of difference in state between an individual sampled from population i and an individual sampled from population j where $i \neq j$ is

$$\begin{aligned} d_{(i,j)}\left(\theta\right) =& \int_0^\infty 2\pi_0\pi_1 \left(1 - e^{-2(u+v)\tau}\right) f\left(\tau | \tau \geq t_{w(i,j)}\right) \, \mathrm{d}\tau \\ =& 2\pi_0\pi_1 \left(1 - e^{-2(u+v)t_{w(i,j)}} \int_0^\infty e^{-2(u+v)\left(\tau - t_{w(i,j)}\right)} f\left(\tau | \tau \geq t_{w(i,j)}\right) \, \mathrm{d}\tau\right) \\ =& 2\pi_0\pi_1 \left(1 - e^{-2(u+v)t_{w(i,j)}} M_{w(i,j)}(-2(u+v))\right) \end{aligned} \tag{2.16}$$

When $i = j$, since we allow for repetition, we scale the probability of difference in state

by the probability of sampling two distinct individuals to obtain

$$\begin{aligned} d_{(i,i)}(\theta) =& \left(1 - \frac{1}{n_i}\right)\left(2\pi_0\pi_1\left(1 - e^{-2(u+v)t_i} M_{w(i,i)}(-2(u+v))\right)\right) \\ =& 2\left(1 - \frac{1}{n_i}\right)\pi_0\pi_1\left(1 - M_i(-2(u+v))\right) \end{aligned} \tag{2.17}$$

Algorithm 1 summarizes the calculation of the probabilities of difference in state.

Algorithm 1: Compute $d(\theta)$.

Input:

θ: the set of parameters of an m-leaf population tree.

Output:

$d(\theta)$: the vector of expected genetic distances between pairs of leaves given θ.

Initialize vector M_i where $i \in \{1, 2, ..., 2m-1\}$.
Initialize vector $d(\theta)$ with elements $d_{(i,j)}(\theta)$ indexed by leaves
$i, j \in \{m, m+1, ..., 2m-1\}$ such that $i \leq j$.
$x \leftarrow -2(u+v)$.
$M_1 \leftarrow \frac{\lambda_r}{\lambda_r - x}$.
For $i \in \{2, 3, ..., 2m-1\}$
$M_i \leftarrow \frac{\lambda_i}{\lambda_i - x}\left(1 - e^{(x-\lambda_i)\left(t_{p_i} - t_i\right)}\right) + e^{(x-\lambda_i)\left(t_{p_i} - t_i\right)} M_{p_i}$
For $i \in \{m, m+1, ..., 2m-1\}$
$d_{(i,i)}(\theta) \leftarrow 2\left(1 - \frac{1}{n_i}\right)\pi_0\pi_1\left(1 - M_i\right)$
For $j \in \{i+1, i+2, ..., 2m-1\}$
$d_{(i,j)}(\theta) \leftarrow 2\pi_0\pi_1\left(1 - e^{xt_{w(i,j)}} M_{w(i,j)}\right)$

Algorithm 1: Algorithm for the computation of expected genetic distances between populations in a population tree, given different coalescent rates along each branch. The nodes are numbered in order from the root to the leaves.

Covariances for genetic distances: To estimate covariances, we use the formula

$$\hat{\Sigma}_{(i,j),(i,j)} = \frac{1}{K^2}\sum_{k=1}^{K} v_{ij}^k \tag{2.18}$$

where v_{ij}^k is the empirical variance in genetic distance between populations i and j at site k. We compute the empirical variances according to Nei and Roychoudhury

(1974). If $i = j$, then v_{ii}^k can be calculated as

$$v_{ii}^k = \frac{2(n_i-1)}{n_i^3}\Big((3-2n_i)\left(\left(g_{0i}^k\right)^2+\left(g_{1i}^k\right)^2\right)^2+\left(g_{0i}^k\right)^2+\left(g_{1i}^k\right)^2 \\ +2(n_i-2)\left(\left(g_{0i}^k\right)^3+\left(g_{1i}^k\right)^3\right)\Big) \tag{2.19}$$

If i and j are two different populations, then v_{ij}^k can be calculated as

$$v_{ij}^k = \frac{1}{n_i n_j}\Big((1-n_i-n_j)\left(g_{0i}^k g_{0j}^k+g_{1i}^k g_{1j}^k\right)^2+(n_i-1)\left(\left(g_{0i}^k\right)^2 g_{0j}^k+\left(g_{1i}^k\right)^2 g_{1j}^k\right) \\ +(n_j-1)\left(\left(g_{0j}^k\right)^2 g_{0i}^k+\left(g_{1j}^k\right)^2 g_{1i}^k\right)+g_{0i}^k g_{0j}^k+g_{1i}^k g_{1j}^k\Big) \tag{2.20}$$

2.3.3 MCMC with delayed acceptance

The MCMC algorithm in SNAPP samples the topology of the species tree, the lengths of the branches in the tree, and the effective population size for each population. Instead of directly sampling effective population size, the algorithm samples a vector Θ. Each member of Θ is the expected number of mutations between two randomly selected individuals within a population, a quantity related to the computation of effective population size. If a node i on the species tree has two child nodes L and R, we refer to R as the sibling node of L and to L as the sibling node of R. The set of proposals is based on those described in Drummond *et al.* (2002):

- randomly choosing δ uniformly from $[\beta^{-1}, \beta]$ for some parameter $\beta > 1$, then rescaling internal branch lengths (as opposed to leaf lengths) and Θ by δ.
- moving a subtree to a new branch in a move based on a technique by Wilson and Balding (1998) originally designed for swapping branches. For this move, two nodes are chosen randomly and uniformly.
- exchanging subtrees rooted at nodes i and j, respectively, by first choosing a node i, then either choosing j such that i is a child node of the sibling node of j or randomly and uniformly choosing j.
- moving a uniformly selected node by changing the lengths of any branches immediately above or immediately below the node based on a uniformly selected number.
- changing Θ via random walks, or by uniformly selected numbers.

After the parameters for θ_1, the initial sample of parameter set θ, have been chosen, new samples are obtained via delayed acceptance as follows:

1. Compute $d(\theta_1)$ and $P(D|\theta_1)$.

2. For $i = 1, 2, \ldots, M$:

   (a) Propose θ' from $q(\cdot|\theta_i)$.

   (b) Compute $d(\theta')$.

   (c) Compute first acceptance probability

$$\alpha(\theta'|\theta_i) = \min\left\{1, \frac{\phi\left(\widehat{d}(D)\middle| d(\theta'), \hat{\Sigma}\right) P(\theta')\, q(\theta_i|\theta')}{\phi\left(\widehat{d}(D)\middle| d(\theta_i), \hat{\Sigma}\right) P(\theta_i)\, q(\theta'|\theta_i)}\right\}. \tag{2.21}$$

   (d) With probability $\alpha(\theta'|\theta_i)$, proceed to next step. Otherwise, set $\theta_{i+1} = \theta_i$ and end iteration.

   (e) Compute $\frac{P(D|\theta')}{P(D|\theta_i)}$.

   (f) Compute second acceptance probability

$$\beta(\theta'|\theta_i) = \min\left\{1, \frac{P(D|\theta')P(\theta')\,\alpha(\theta_i|\theta')\, q(\theta_i|\theta')}{P(D|\theta_i)P(\theta_i)\,\alpha(\theta'|\theta_i)\, q(\theta'|\theta_i)}\right\} \tag{2.22}$$

   (g) With probability $\beta(\theta'|\theta_i)$, accept proposal, then set $\theta_{i+1} = \theta'$. Otherwise, set $\theta_{i+1} = \theta_i$.

If we let n denote the total number of individuals, the time complexity of the likelihood calculation in SNAPP is $O(mn^2 \log n)$ (Bryant *et al.*, 2012). As implied by Algorithm 1, the computation of the mgfs takes $O(m)$ operations and the computation of $d(\theta)$ is $O(m^2)$. In addition, calculation of $\phi\left(\widehat{d}(D)\middle| d(\theta), \hat{\Sigma}\right)$ with diagonal matrix $\hat{\Sigma}$ is $O(m^2)$. This means the time complexity of the entire approximate likelihood calculation is $O(m^2)$, much faster than SNAPP's likelihood calculation since $n \geq m$. Due to the time complexity of the calculation of $\widehat{d}(D)$, we only calculate $\widehat{d}(D)$ once. In addition, to save time, we store each value of $P(D|\theta_i)$ to avoid recomputing $P(D|\theta_i)$ for the computation of $\beta(\theta'|\theta_i)$.

2.4 Delayed acceptance with correction for acquisition error

In Section 2.3, we showed how a delayed acceptance strategy could be applied in SNAPP, however only in the case that we have both polymorphic and non-polymorphic

sites. In many situations, the proportion of constant sites is not available, for example, in data from SNP chips, or AFLP data (Bryant *et al.*, 2012). In this section, we attempt to extend the delayed acceptance framework to cover this case, the problem being that with this information lost, estimation of genetic distances becomes far more difficult.

2.4.1 Correcting for missing sites

The data used by SNAPP is restricted to polymorphic sites, that is, loci where not all individuals have the same state (Bryant *et al.*, 2012). This means that the frequencies of non-polymorphic (constant) sites are lost and not available for the calculation. The approximate likelihoods in the previous section were all based on genetic distances and the calculation of these distances requires knowledge not just of polymorphic sites, but of the proportion of sites which are polymorphic. In order to implement our delayed acceptance strategy for this context, we apply a new strategy, and either sample the numbers of missing sites (in addition to the parameters of the model) or integrate out the number of missing sites. Furthermore, we do this in a way that gives the same posterior density for parameters as the conditional likelihood approach used in Bryant *et al.* (2012).

The problem we face is that we know the total number of polymorphic sites (which we denote z), but not the total number of sites (which we denote by N) including those which are non-polymorphic. We therefore treat N as a variable to be inferred, and derive its posterior distribution.

The first question is what prior to assign to N. There are several possible priors one could consider. Following Gelman (2004), we use an improper prior, namely $P(N|\theta) = \frac{1}{N}$. Under this prior, we can compute the posterior density of the data. To this end, let s_i denote the number of polymorphic sites of class i, where $i \in \{1, 2, ..., \rho\}$ for some integer ρ.

To find the posterior density, we consider a number of density equations. We first rewrite the posterior density $P(\theta|D)$ in terms of $P(\theta, N|D)$ to obtain

$$P(\theta|D) = \sum_{N=z}^{\infty} P(\theta, N|D). \tag{2.23}$$

Using Bayes' theorem, we rewrite $P(\theta, N|D)$ as

$$\begin{aligned} P(\theta, N|D) &= \frac{P(\theta, N)\, P(D|\theta, N)}{P(D)} \\ &= \frac{P(\theta)\, P(N|\theta)\, P(D|\theta, N)}{P(D)}. \end{aligned} \tag{2.24}$$

From (2.23) and (2.24), we obtain the equation,

$$P(\theta|D) = \sum_{N=z}^{\infty} \frac{P(\theta)\, P(N|\theta)\, P(D|\theta, N)}{P(D)} \tag{2.25}$$

or, alternatively, the relation

$$P(\theta|D) \propto \sum_{N=z}^{\infty} P(\theta)\, P(N|\theta)\, P(D|\theta, N)\,. \tag{2.26}$$

If we let $p_i(\theta)$ denote the probability that a site is a polymorphic site of class i, we can write the multinomial $P(D|\theta, N)$ as

$$P(D|\theta, N) = \frac{N!}{s_1! \ldots s_\rho!(N-z)!} \left(1 - \sum_{i=1}^{\rho} p_i(\theta)\right)^{N-z} \prod_{j=1}^{\rho} p_j(\theta)^{s_j}. \tag{2.27}$$

From (2.26) and (2.27), we obtain

$$\begin{aligned} P(\theta|D) &\propto \sum_{N=z}^{\infty} P(\theta) P(N|\theta) \frac{N!}{s_1! \ldots s_\rho!(N-z)!} \left(1 - \sum_{i=1}^{\rho} p_i(\theta)\right)^{N-z} \prod_{j=1}^{\rho} p_j(\theta)^{s_j} \\ &\propto \sum_{N=z}^{\infty} P(\theta) \frac{(N-1)!}{s_1! \ldots s_\rho!(N-z)!} \left(1 - \sum_{i=1}^{\rho} p_i(\theta)\right)^{N-z} \prod_{j=1}^{\rho} p_j(\theta)^{s_j} \\ &= P(\theta) \left(\prod_{i=1}^{\rho} p_i(\theta)^{s_i}\right) \sum_{N=z}^{\infty} \frac{(N-1)!}{s_1! \ldots s_\rho!(N-z)!} \left(1 - \sum_{j=1}^{\rho} p_j(\theta)\right)^{N-z} \\ &\propto P(\theta) \left(\prod_{i=1}^{\rho} p_i(\theta)^{s_i}\right) \sum_{N=z}^{\infty} \frac{(N-1)!}{(z-1)!(N-z)!} \left(1 - \sum_{j=1}^{\rho} p_j(\theta)\right)^{N-z} \end{aligned}$$

If we let

$$B\left(N - z; z, 1 - \sum_{j=1}^{\rho} p_j(\theta)\right)$$

denote the negative binomial pmf where $N - z$ is the number of "successes" and $1 -$

$\sum_{j=1}^{\rho} p_j(\theta)$ is the probability of "success," then

$$\begin{aligned} P(\theta|D) &\propto P(\theta)\left(\prod_{i=1}^{\rho} p_i(\theta)^{s_i}\right)\sum_{N=z}^{\infty}\frac{B\left(N-z; z, 1-\sum_{j=1}^{\rho} p_j(\theta)\right)}{\left(\sum_{j=1}^{\rho} p_j(\theta)\right)^z} \\ &= P(\theta)\left(\prod_{i=1}^{\rho} p_i(\theta)^{s_i}\right)\frac{1}{\left(\sum_{j=1}^{\rho} p_j(\theta)\right)^z} \\ &= P(\theta)\prod_{i=1}^{\rho}\left(\frac{p_i(\theta)}{\sum_{j=1}^{\rho} p_j(\theta)}\right)^{s_i} \end{aligned} \tag{2.28}$$

The term

$$\frac{p_i(\theta)}{\sum_{j=1}^{\rho} p_j(\theta)}$$

is a conditional probability; that is, it is the probability that a site is in class i given that it is in one of classes 1 through ρ (in other words, given that it is polymorphic). Therefore,

$$\prod_{i=1}^{\rho}\left(\frac{p_i(\theta)}{\sum_{j=1}^{\rho} p_j(\theta)}\right)^{s_i}$$

is the conditional likelihood, as introduced by Felsenstein (1981a) and others, and currently implemented in SNAPP (Bryant *et al.*, 2012). Thus, the prior $P(N|\theta) \propto \frac{1}{N}$ corresponds exactly to the current situation. We take advantage of this fact to design an approximate posterior.

First, however, we need to be able to sample the variable N. To sample N, we would need to specify $P(N|\theta, D)$. We first note that

$$P(N|\theta, D) = \frac{P(\theta, N|D)}{P(\theta|D)}. \tag{2.29}$$

From (2.24), we obtain

$$P(N|\theta, D) = \frac{P(\theta)\,P(N|\theta)\,P(D|\theta, N)}{P(D)P(\theta|D)} \propto \frac{P(\theta)\,P(N|\theta)\,P(D|\theta, N)}{P(\theta|D)}. \tag{2.30}$$

Let $\mathcal{P}_c(\theta)$ denote the probability of sites where all individuals are in the same state. From (2.27), (2.28), and (2.30), we obtain

$$\begin{aligned} P(N|\theta, D) &\propto \frac{N!}{s_1!\ldots s_\rho!(N-z)!}\frac{P(\theta)\,P(N|\theta)\,(1-\mathcal{P}_c(\theta))^{N-z}\prod_{j=1}^{\rho} p_j(\theta)^{s_j}}{P(\theta)\prod_{i=1}^{\rho}\left(\frac{p_i(\theta)}{\mathcal{P}_c(\theta)}\right)^{s_i}} \\ &= \frac{N!P(N|\theta)}{s_1!\ldots s_\rho!(N-z)!}(1-\mathcal{P}_c(\theta))^{N-z}(\mathcal{P}_c(\theta))^z \end{aligned}$$

Since $P\left(N|\theta, D\right) \propto \frac{1}{N}$, we obtain a negative binomial pmf

$$P\left(N|\theta, D\right) \propto \frac{(N-1)!}{(z-1)!(N-z)!}\mathcal{P}_c(\theta)^{N-z}\left(1-\mathcal{P}_c(\theta)\right)^z. \tag{2.31}$$

We will use this observation to integrate out the variable N in our approximation.

2.4.2 Approximating the likelihood

For delayed acceptance, the algorithm we use is the same as the algorithm in Section 2.3.3, modified for data with only polymorphic sites. We approximate the target density as

$$P(\theta|D) \approx P\left(\theta\left|\widehat{d}(D)\right.\right)$$

as in Section 2.3.2. Note that $\widehat{d}(D)$ is computed from (2.6) and (2.7), but only from polymorphic sites. Next, from (2.4), we write

$$P\left(\theta\left|\widehat{d}(D)\right.\right) = P\left(\widehat{d}(D)\right|\theta\right)\frac{P(\theta)}{P\left(\widehat{d}(D)\right)}.$$

For our approximation, we compute the expectation

$$E\left[\widehat{d}(D)|\theta\right] \tag{2.32}$$

and variance

$$\text{var}\left[\widehat{d}(D)|\theta\right] \tag{2.33}$$

of $\widehat{d}(D)$ given θ; these computations are described later in this section.

After we obtain (2.32) and (2.33) for each population i and each population j, we make the Gaussian approximation

$$P^*\left(\widehat{d}(D)\right|\theta\right) = \phi\left(\widehat{d}(D)\right| E\left[\widehat{d}(D)|\theta\right], \text{var}\left[\widehat{d}(D)|\theta\right]\right) \tag{2.34}$$

where

- $E\left[\widehat{d}(D)|\theta\right]$ is a vector in which each element is

  $$E\left[\widehat{d}_{(i,j)}(D)|\theta\right]$$

  for populations i and j.

- $\text{var}\left[\widehat{d}(D)|\theta\right]$ is a diagonal matrix in which each diagonal element is

  $$\text{var}\left[\widehat{d}_{(i,j)}(D)|\theta\right]$$

  for populations i and j.

Expectation of $\widehat{d}(D)$: Let $p_\theta = 1 - \mathcal{P}_c(\theta)$ denote the probability of polymorphic sites given θ. We compute (2.32) by computing

$$E\left[\widehat{d}_{(i,j)}(D)|\theta\right] = \sum_{N=z}^{\infty} E\left[\widehat{d}_{(i,j)}(D)|\theta, N\right] P(N|\theta, D)$$

for each population i and j and where $E\left[\widehat{d}_{(i,j)}(D)|\theta, N\right]$ satisfies the equation

$$\frac{z}{N} E\left[\widehat{d}_{(i,j)}(D)|\theta, N\right] = d_{(i,j)}(\theta). \tag{2.35}$$

From (2.35), we obtain

$$E\left[\widehat{d}_{(i,j)}(D)|\theta, N\right] = \frac{N}{z} d_{(i,j)}(\theta).$$

As stated in Section 2.4.1, N has a negative binomial distribution (2.31); Therefore, we obtain

$$E\left[\widehat{d}_{(i,j)}(D)|\theta\right] = \frac{d_{(i,j)}(\theta)}{p_\theta}. \tag{2.36}$$

Variance of $\widehat{d}(D)$: We compute (2.33) by computing the empirical variance

$$\text{var}\left[\widehat{d}_{(i,j)}(D)\middle|\theta\right] = E\left[\left(d^k_{(i,j)}\right)^2\middle|\theta\right] - \left(E\left[d^k_{(i,j)}\middle|\theta\right]\right)^2. \tag{2.37}$$

for each population i and j and where $d^k_{(i,j)}$ is the probability that, at site k, an individual sampled from population i has a different state than an individual sampled from population j. As in Section 2.3.2, we let g^k_{0i} and g^k_{1i} is the proportion of alleles at site k in population i in state 0 and 1, respectively. If $i = j$, then

$$d^k_{(i,i)} = 2g^k_{0i}g^k_{1i};$$

If $i \neq j$, then

$$d^k_{(i,j)} = g^k_{0i}g^k_{1j} + g^k_{1i}g^k_{0j}.$$

2.5 Delayed acceptance

For delayed acceptance sampling, we determine simplified formulae to compute the acceptance probabilities (2.1) and (2.2).

2.5.1 Acceptance probabilities

From (2.4) and (2.34), it follows that

$$\begin{aligned}\frac{P^*(\theta'|D)}{P^*(\theta|D)} =& \frac{P^*\left(\widehat{d}(D)\middle|\,\theta'\right)\frac{P(\theta')}{P\left(\widehat{d}(D)\right)}}{P^*\left(\widehat{d}(D)\middle|\,\theta\right)\frac{P(\theta)}{P\left(\widehat{d}(D)\right)}} \\ =& \frac{P^*\left(\widehat{d}(D)\middle|\,\theta'\right)P(\theta')}{P^*\left(\widehat{d}(D)\middle|\,\theta\right)P(\theta)} \\ =& \frac{\phi\left(\widehat{d}(D)\middle|\,E\left[\widehat{d}(D)|\theta'\right],\mathrm{var}\left[\widehat{d}(D)|\theta'\right]\right)P(\theta')}{\phi\left(\widehat{d}(D)\middle|\,E\left[\widehat{d}(D)|\theta\right],\mathrm{var}\left[\widehat{d}(D)|\theta\right]\right)P(\theta)}\end{aligned}$$

and that

$$\begin{aligned}\alpha(\theta'|\theta) =& \min\left\{1, \frac{P^*(\theta'|D)}{P^*(\theta|D)}\frac{q(\theta|\theta')}{q(\theta'|\theta)}\right\} \\ =& \min\left\{1, \frac{\phi\left(\widehat{d}(D)\middle|\,E\left[\widehat{d}(D)|\theta'\right],\mathrm{var}\left[\widehat{d}(D)|\theta'\right]\right)P(\theta')}{\phi\left(\widehat{d}(D)\middle|\,E\left[\widehat{d}(D)|\theta\right],\mathrm{var}\left[\widehat{d}(D)|\theta\right]\right)P(\theta)}\frac{q(\theta|\theta')}{q(\theta'|\theta)}\right\}. \qquad (2.38)\end{aligned}$$

If a sample of θ' passes the first acceptance step, it is subject to a second acceptance step at which the full likelihood is computed. At this same step, the acceptance probability is computed, based on (2.3) and (2.34), as

$$\begin{aligned}\beta(\theta'|\theta) =& \min\left[1, \frac{P(\theta'|D)}{P(\theta|D)}\frac{P^*(\theta|D)}{P^*(\theta'|D)}\right] \\ =& \min\left[1, \frac{P(D|\theta')}{P(D|\theta)}\frac{\phi\left(\widehat{d}(D)\middle|\,E\left[\widehat{d}(D)|\theta\right],\mathrm{var}\left[\widehat{d}(D)|\theta\right]\right)}{\phi\left(\widehat{d}(D)\middle|\,E\left[\widehat{d}(D)|\theta'\right],\mathrm{var}\left[\widehat{d}(D)|\theta'\right]\right)}\right] \qquad (2.39)\end{aligned}$$

2.5.2 Algorithm

Before we run the algorithm, we choose θ_1, the initial state of θ_i. We then run the following implementation of the delayed acceptance algorithm:

1. Compute $P(D|\theta_1)$.
2. For $i = 1, 2, \ldots, M$:
   (a) Propose θ' from $q(\cdot|\theta_i)$.
   (b) Compute first acceptance probability $\alpha(\theta'|\theta_i)$ based on (2.38).

(c) With probability $\alpha(\theta'|\theta_i)$, proceed to next step. Otherwise, reject proposal, set $\theta_{i+1} = \theta_i$, and end iteration.

(d) Compute $P(D|\theta')$ and $\alpha(\theta_i|\theta')$.

(e) Compute acceptance probability $\beta(\theta'|\theta_i)$ based on (2.39).

(f) With probability $\beta(\theta'|\theta_i)$, accept proposal and set $\theta_i = \theta'$. Otherwise, reject proposal and set $\theta_{i+1} = \theta_i$.

2.6 Experimental results

We ran simulations of SNAPP both with and without delayed acceptance by using data from populations of Ourisia plants in New Zealand; this data has previously been analyzed by Meudt *et al.* (2009) and Bryant *et al.* (2012). The dataset we use comes from 8 taxa, each with more than 1 individual, and contains data from 2576 polymorphic sites. The taxon labels we use and the number of individuals at each taxon is given in Table 2.1.

Taxon label	cae	cro	mcccal	mccmcc	sesses	mcplac	rem	sesspl	gla
# of individuals	8	3	2	2	2	2	2	2	2

Table 2.1: The taxon labels and the number of individuals at each taxon.

Letting Θ_i denote the expected number of mutations at node i, we choose an initial value of $\Theta_i = 0.2$ at each node i for our experiments. In addition, we choose $u = 1$ and $v = 1$ as our mutation rates and set the chain length to 500000 iterations.

Unfortunately, the delayed acceptance chain has not converged, as shown in the trace plots of the logarithm of the posterior of the two chains in Figure 2.2. The trace plot of the chain without delayed acceptance shows fluctuations in the logarithm that remain within a range, indicating convergence. Meanwhile, the chain with delayed acceptance shows fluctuations but the range of the fluctuations changes over time and the chain clearly does not converge. To understand why the delayed acceptance chain has not converged, we first analyze the acceptance rates of the two chains and the tendency of each chain to remain in a state. We then compare the full likelihood and approximate likelihood computed at each sample.

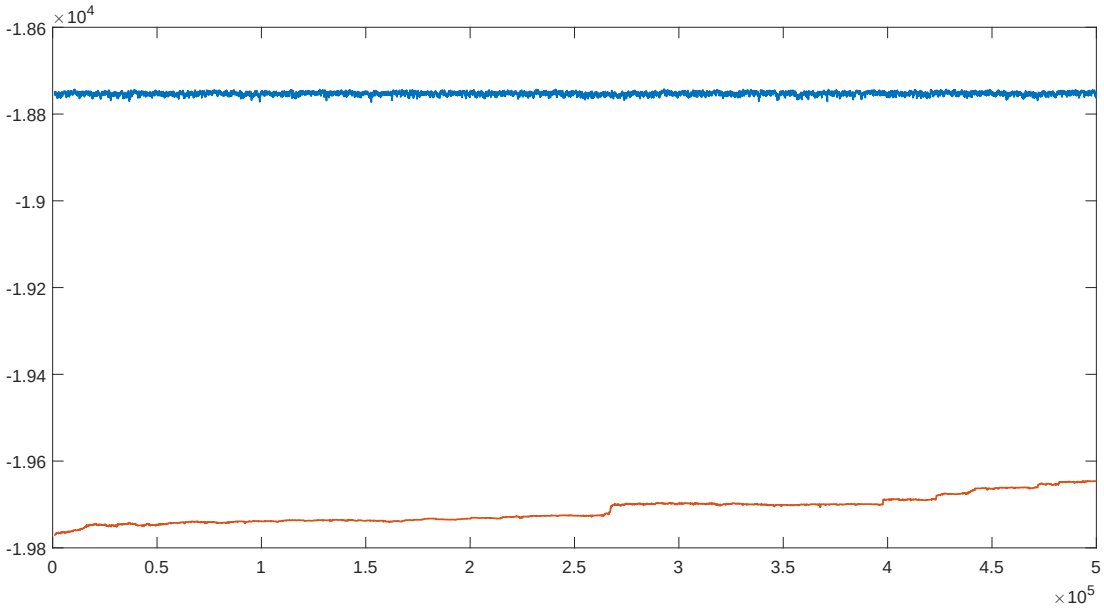

Figure 2.2: Trace plot of logarithm of posterior without delayed acceptance (blue) and with delayed acceptance (red).

2.6.1 Acceptance rates

To analyze the acceptance rates of each chain, we ignore the initial 1000 samples of each chain as burn-in. We then find the highest number of recurrences of a state and the mean number of state recurrences. Table 2.2 shows recurrence numbers of the entire parameter set. As expected, the acceptance rate is lower in the delayed acceptance case. In addition, each recurrence value in the delayed acceptance case is larger than its corresponding recurrence value in the case without delayed acceptance.

delayed acceptance?	acceptance rate	R_{max}	$\bar{R}$
no	0.2225	43	4.4953
yes	0.0124	1180	80.9539

Table 2.2: The acceptance rates, the values of R_{max} (maximum number of recurrences of a state), and the values of $\bar{R}$ (the average number of recurrences of a state) of each chain.

2.6.2 Likelihood comparisons

To analyze the likelihoods generated in the chains, we remove burn-in and make a scatter plot from the two chains individually. In this scatter plot, each point represents a sample, with the logarithm of the full likelihood as its horizontal coordinate and the logarithm of the approximate likelihood as its horizontal coordinate. The scatter plot of the two chains is shown in Figure 2.3 along with a line of slope 1; if the approximation is good, then the slope of the scatter plot would be roughly 1. However, as Figure 2.3 shows, not only does the scatter plot not have a slope of 1, the scatter plot is not even polynomial. We can conclude from these plots that $\widehat{d}(D)$ is not a good approximation of $\frac{d(\theta)}{p_\theta}$ for the true value of θ.

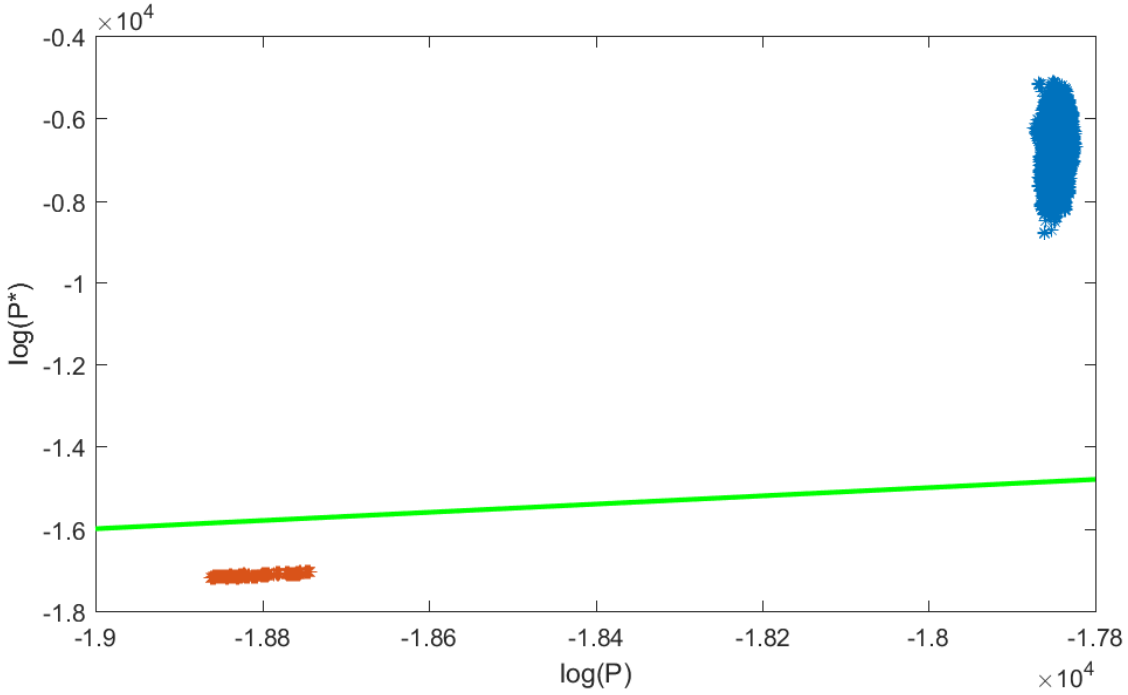

Figure 2.3: Scatter plot of logarithms of full likelihoods (horizontal axis) against logarithms of approximate likelihoods (vertical axis) along with a green line of slope 1. Blue points are from the chain without delayed acceptance and red are from the chain with delayed acceptance.

2.7 Conclusions

Our results and analyses in Section 2.6 show that the delayed acceptance chain does not mix well when a Gaussian approximation is used. As indicated by Figure 2.2 and

Table 2.2, delayed acceptance with a Gaussian approximation significantly decreased the acceptance rate and increased the tendency for a chain to remain in a state. These and Figure 2.3 make it clear that $\widehat{d}(D)$ is not a good approximation of $\frac{d(\theta)}{p_\theta}$ for the true value of θ. Indeed, the Gaussian approximation is not suitable for chains in which the total number of sites N is integrated out, let alone chains in which N is sampled.

The difficulties in approximating the likelihood from the genetic distances motivated us to explore other approximations that could be applied to SNAPP, which led to the work described in Chapter 3 (numerical integration) and Chapter 4 (basis functions). The results in Chapters 3 and 4 have proven to be very useful in terms of likelihood approximation. Although the methods themselves are applied in this thesis to contexts outside of SNAPP, they could still be applied to SNAPP, given their convergence and applicability.

Chapter 3

Dynamic programming algorithm for calculation of likelihoods

3.1 Background

3.1.1 Introduction to problem

Statistical models for nucleotide or amino acid mutations and substitutions, and the algorithms for computing with them, are fundamental to the study of molecular evolution and biology. As we widen our focus from the evolution of genes to the evolution of genomes, individuals, and populations, a whole new class of modelling challenges present themselves. These include the development of realistic *quantitative* models where a trait can take on a continuous range of values (O'Meara, 2012). Of course, the usefulness of any new model is contingent on the tools available to compute with them. The main contribution of this chapter is to show how, by combining ideas from statistical phylogenetics and numerical mathematics, we can compute efficiently with a far larger range of evolutionary models.

The algorithms we develop are for computation of the likelihood, that is the probability of the data given the phylogeny, evolutionary model and parameters. If we are working with an evolutionary model with only a small (finite) number of states, then likelihoods can be computed using the dynamic programming algorithm of Felsenstein (1981a). We will show how to extend this algorithm to also compute likelihoods for a wide range of continuous trait models.

There is already a wide range of evolutionary phenomena that are studied using continuous trait models. Much of comparative genomics relies on implicit or explicit

models for the evolution of morphology (Stevens, 1991; Felsenstein, 2002; Ronquist, 2004; Harmon *et al.*, 2010; O'Meara, 2012), many of which make gross simplifying assumptions about how traits vary over time. Continuous evolutionary models have been used in comparative transcriptomics to study heritable aspects of gene expression levels (Khaitovich *et al.*, 2005, 2006), an area with exceptional promise given recent improvements in accuracy and the ability to sample in situ (Voelckel *et al.*, 2012).

Continuous trait models will be of growing importance in evolutionary studies of whole genome single nucleotide polymorphism-databases. Inference methods based on the coalescent such as SNAPP (Bryant *et al.*, 2012) do not scale well as the number of individuals grows, while those based on continuous models of gene frequencies (Cavalli-Sforza and Edwards, 1967; Felsenstein, 1981b; Sirén *et al.*, 2011) depend only on proportions of individuals with each allele, so scale extremely well. In addition, it is often easier to model the effect of selection on continuous gene frequency models than with the coalescent. Continuous evolutionary models have also been applied successfully to the study of ancestral geography distributions (Lemey *et al.*, 2010).

Our interest is in developing techniques used to compute with these models, and to expand the range of models we can work with. Early work of Felsenstein (Felsenstein, 1968, 1973), revisited by Freckleton (2012) and FitzJohn (2012), demonstrated that if traits are evolving according to Brownian motion then we can compute likelihoods quickly and (up to numerical precision) exactly. Felsenstein's approach extends to other Gaussian processes, notably the Ornstein-Uhlenbeck (OU) process (Lande, 1976; Felsenstein, 1988; Hansen, 1997), and for several decades, Gaussian models were used almost exclusively to model the evolution of quantitative traits. Ho and Ané (2014) used clever algebraic techniques to develop an alternative algorithm for computing the likelihood and related quantities. They survey several other models which can be handled using the same approach.

These methods are very efficient, and when they can be used, they should be used. The drawback of these methods is that they are fundamentally restricted to models which are Gaussian processes or transforms of Gaussian processes, where the computational bottleneck lies in the computation of a quadratic form involving the covariance matrix Ho and Ané (2014). Many evolutionary models can not be handled within this framework (e.g. Ronquist, 2004; Landis *et al.*, 2013). Some of the properties of Gaussian processes are quite restrictive: Gaussian processes have single modes, so can only model adaptive landscapes with single peaks; Brownian motion has independent increments, so the rate of change is independent of the value of the trait. The standard

strategy for computing with non-Gaussian models is to resort to Monte Carlo strategies. Even when we *are* working with a model satisfying the assumptions of Ho and Ané (2014), the algorithms they describe do not give an efficient method for integrating over sets of trait values at the tips, as in the threshold models we discuss below.

Computing the probability of quantitative character evolution may be framed as a numerical integration (quadrature) problem. For most models, if we know the value of the trait at each ancestral node in the phylogeny we can quickly compute the various transition probabilities. Since we do not usually know these ancestral trait values we integrate them out. This is a multi-dimensional integration problem with one dimension for each ancestral node (or two dimensions for each node if we are modelling covarying traits); see Felsenstein (2004).

Methods for estimating or approximating integrals are usually judged by their "rate of convergence": how quickly the error of approximation decreases as the amount of work (function evaluations) increases. Consider the problem of computing a one-dimensional integral

$$\int_0^1 f(x)\,\mathrm{d}x \tag{3.1}$$

where f is a "nice" function with continuous and bounded derivatives. Simpson's rule, a simple textbook method reviewed below, can be shown to have an $O(N^{-4})$ rate of convergence (see pg. 531 of Dahlquist and Björck (2008)), meaning that, asymptotically in N, evaluating 10 times more points reduces the error by a factor of 10^4. In contrast, a standard Monte Carlo method has a rate of convergence of $O(N^{-\frac{1}{2}})$ (see pg. 600 of Dahlquist and Björck (2008)), meaning that evaluating 10 times more points will only reduced the error by a factor of around 3. For this reason, numerical analysis texts often refer to Monte Carlo approaches as "methods of last resort."

Despite this apparently lacklustre performance guarantee, Monte Carlo methods have revolutionised phylogenetics in general and the analysis of quantitative characters in particular. The reason is their partial immunity to the curse of dimensionality. Methods like Simpson's rule are not practical for a high number of dimensions as the asymptotic convergence rate, quoted above, is only achieved for an infeasibly large number of function evaluations N. The effective convergence rate for small N can be very poor, and typically worse than Monte-Carlo. In contrast, there are Monte Carlo approaches which achieve close to $O(N^{-\frac{1}{2}})$ convergence irrespective of dimension. This has been critical when computing the likelihoods of complex evolutionary models with as many dimensions as there are nodes in the phylogeny.

The main contribution of this chapter is to demonstrate how to efficiently and

accurately compute likelihoods on a phylogeny using a sequence of one-dimensional integrations. We obtain a fast algorithm with convergence guarantees that far exceed what can be obtained by Monte Carlo integration. Our approach combines two standard tools: classical numerical integrators and Felsenstein's pruning algorithm for *discrete characters* (Felsenstein, 1981a). Indeed, the only real difference between our approach and Felsenstein's discrete character algorithm is that we use numerical integration techniques to integrate states at ancestral nodes, instead of just carrying out a summation.

The running time of the algorithm is $O(N^2 n)$, where N is the number of points used in the numerical integration at each node and n is the number of taxa (leaves) in the tree. Using Simpson's method, we obtain a convergence rate of $O(nN^{-4})$, meaning that if we increase N by a factor of 10 we will obtain an estimate which is accurate to four more decimal places.

To illustrate the application of our general framework, we develop an efficient algorithm for computing the likelihood of a tree under the threshold model of Wright (1934) and Felsenstein (2005, 2012). We also show how to infer marginal trait densities at ancestral nodes. We have implemented these algorithms and used them to study evolution of extrafloral nectaries on an 839-taxon phylogeny (Marazzi *et al.*, 2012).

The combination of numerical integrators and the pruning algorithm opens up a large range of potential models and approaches which we have only just begun to explore. It may well be that Gaussian type models provide good approximations for many problems, however the extent to which this is true will be unknown until we have computational tools for handling richer models.

The results presented in this chapter were published in Hiscott *et al.* (2016), with the exception of those in Section 3.7.1.

3.1.2 Models

Phylogenetic models for continuous trait evolution, like those for discrete traits, are specified by the density of trait values at the root and the transition densities along the branches. We use $f(x_r|\theta_r)$ to denote the density for the trait value at the root, where θ_r is a set of relevant model parameters. We use $f(x_i|x_j, \theta_i)$ to denote the transitional density for the value at node i, conditional on the trait value at its parent node j. Here, θ_i represents a bundle of parameters related to node i such as branch length, population size, and mutation rate. All of these parameters could vary throughout the tree.

To see how the model works, consider how continuous traits might be simulated. A state X_r is sampled from the root density $f(X_r|\theta_r)$. We now proceed through the phylogeny from the root to the tips, each time visiting a node only after its parent has already been visited. For each node i, we generate the value at that node from the density $f(X_i|x_j, \theta_v)$, where x_j is the simulated trait value at node j, the parent of node i. In this way, we will eventually generate trait values for the tips.

We use $X_1, \ldots, X_n$ to denote the random trait values at the tips and $X_{n+1}, \ldots, X_{2n-1}$ to denote the random trait values at the internal nodes, ordered so that children come before parents. Hence X_{2n-1} is the state assigned to the root. Let

$$\mathcal{E}(T) = \{(i, j) : \text{node } i \text{ is a child of node } j\} \tag{3.2}$$

denote the set of branches in the tree. The joint density for all trait values, observed and ancestral, is given by multiplying the root density with all of the transition densities

$$f(x_1, \ldots, x_n, x_{n+1}, \ldots, x_{2n-1}|\theta) = f(x_{2n-1}|\theta) \prod_{(i,j)\in\mathcal{E}(T)} f(x_i|x_j, \theta_i). \tag{3.3}$$

The probability of the observed trait values $x_1, \ldots, x_n$ is now determined by integrating out all of the ancestral trait values:

$$\mathcal{L}(T) = f(x_1, \ldots, x_n|\theta) = \int \int \cdots \int f(x_{2n-1}|\theta_r) \prod_{(i,j)\in\mathcal{E}(T)} f(x_i|x_j, \theta_i) \, \mathrm{d}x_{n+1}, \ldots, \mathrm{d}x_{2n-1}. \tag{3.4}$$

In these integrals, the bounds of integration will vary according to the model.

The oldest, and most widely used, continuous trait models assume that traits (or transformed gene frequencies) evolve like Brownian motion (Cavalli-Sforza and Edwards, 1967; Felsenstein, 1973). For these models, the root density $f(x_r|\theta)$ is Gaussian (normal) with mean 0 and unknown variance σ_r^2. The transition densities $f(x_i|x_j, \theta_v)$ are also Gaussian, with mean x_j (the trait value of the parent) and variance proportional to branch length. Note that there are identifiability issues which arise with the inference of the root position under this model, necessitating a few tweaks in practice (see the discussion in Chapter 23 of Felsenstein (2004)).

It can be shown that when the root density and transitional densities are all Gaussian, the joint density (3.4) is multivariate Gaussian. Furthermore, the covariance matrix for this density has a special structure which methods such as the pruning techniques of Felsenstein (1968, 1973), Freckleton (2012), and FitzJohn (2012) exploit, as does the top-down approach of Ho and Ané (2014). This general approach continues to

work when Brownian motion is replaced by an OU process (Lande, 1976; Felsenstein, 1988; Hansen, 1997), or indeed to many linear or generalized linear models.

Gaussian models, and their relatives, are mathematically and computationally convenient, but rely on assumptions which are unrealistic and inappropriate in many contexts. Numerous researchers have implemented models which do not fit into the general Gaussian framework; most have resorted to Monte Carlo computation to carry out their analyses.

Landis *et al.* (2013) discuss a class of continuous trait models which are based on *Lévy processes* and include jumps. At particular times, as governed by a Poisson process, the trait value jumps to a value drawn from a given density. Examples include a *compound Poisson process* with Gaussian jumps and a *Variance Gamma* model given by Brownian motion with with time varying according to a gamma process. Both of these processes have analytical transition probabilities in some special cases.

Lepage *et al.* (2006) use the Cox-Ingersoll-Ross (CIR) process to model rate variation across a phylogeny. Like the OU process (but unlike Brownian motion), the CIR process is ergodic. It has a stationary Gamma density which can be used for the root density. The transition density is a particular non-central Chi-squared density and the process only assumes positive values.

Kutsukake and Innan (2013) examine a family of compound Poisson models, focusing particularly on a model where the trait values make exponentially distributed jumps upwards or downwards. In the case that the rates of upward and downward jumps are the same, the model has jumps that follow a double exponential distribution. Kutsukake and Innan (2013) use approximate Bayesian computation to carry out inference.

Sirén *et al.* (2011) propose a simple and elegant model for gene frequencies whereby the root value is drawn from a Beta distribution and each transitional density is Beta with appropriately chosen parameters.

Trait values at the tips are not always observed directly. A simple, but important, example of this is the threshold model of Wright (1934), explored by Felsenstein (2005). Under this model, the trait value itself is censored and we only observe whether or not the value is positive or negative. A similar complication arises when dealing with gene frequency data as we typically do not observe the actual gene frequency but instead a binomially distributed sample based on that frequency (Sirén *et al.*, 2011).

If the trait values at the tip are not directly observed we integrate over these values as well. Let $\pi(z_i|x_i)$ denote the probability of observing z_i given the trait value x_i. The

marginalised likelihood is then

$$\mathcal{L}(T|z_1,\ldots,z_n) = \int\int\cdots\int f(x_r|\theta) \prod_{(i,j)\in\mathcal{E}(T)} f(x_i|x_j,\theta_v) \prod_{i=1}^{n} \pi(z_i|x_i)\ \mathrm{d}x_1,\ldots,\ \mathrm{d}x_{2n-1}. \tag{3.5}$$

3.2 Algorithm

3.2.1 Numerical Integration

Analytical integration can be difficult or impossible. For the most part, it is unusual for an integral to have an analytical solution, and there is no general method for finding it when it does exist. In contrast, *numerical integration* techniques (also known as *numerical quadrature*) are remarkably effective and are often easy to implement. A numerical integration method computes an approximation of the integral from function values at a finite number of points. Hence we can obtain approximate integrals of functions even when we don't have an equation for the function itself. See Cheney and Kincaid (2012) for an introduction to numerical integration, and Dahlquist and Björck (2008) and Davis and Rabinowitz (1984) for more comprehensive technical surveys.

The idea behind most numerical integration techniques is to approximate the target function using a function which is easy to integrate. To begin, consider the popular *Simpson's method* which approximates the original function using piecewise quadratic functions. To approximate an integral $\int_a^b f(x)\,\mathrm{d}x$ we first determine $N+1$ equally spaced points

$$x_0 = a,\quad x_1 = a + \frac{b-a}{N},\quad x_2 = a + 2\frac{b-a}{N}, \ldots, x_k = a + k\frac{b-a}{N}, \ldots,\ x_N = b. \tag{3.6}$$

We now divide the integration into $N/2$ intervals

$$\int_a^b f(x)\ \mathrm{d}x = \sum_{\ell=1}^{N/2} \int_{x_{2\ell-2}}^{x_{2\ell}} f(x)\ \mathrm{d}x. \tag{3.7}$$

Within each interval $[x_{2\ell-2}, x_{2\ell}]$, there is a unique quadratic function which equals $f(x)$ at each the three points $x = x_{2\ell-2}$, $x = x_{2\ell-1}$ and $x = x_{2\ell}$. The integral of this quadratic on the interval $[x_{2\ell-2}, x_{2\ell}]$ is

$$\frac{(b-a)}{3N}\left(f(x_{2\ell-2}) + 4f(x_{2\ell-1}) + f(x_{2\ell})\right) \tag{3.8}$$

Summing over ℓ, we obtain the approximation

$$\int_a^b f(x)\ \mathrm{d}x \approx \sum_{\ell=1}^{N/2} \frac{(b-a)}{3N} \left(f(x_{2\ell-2}) + 4f(x_{2\ell-1}) + f(x_{2\ell})\right). \tag{3.9}$$

With a little rearrangement, the approximation can be written in the form

$$\int_a^b f(x)\ \mathrm{d}x \approx \frac{(b-a)}{N} \sum_{k=0}^{N} w_k f(x_k) \tag{3.10}$$

where $w_k = 4/3$ when k is odd and $w_k = 2/3$ when k is even, with the exception of w_0 and w_N which both equal $1/3$. Simpson's method is easy to implement and has a convergence rate of $O(N^{-4})$. Increasing the number of intervals by a factor of 10 decreases the error by a factor of 10^{-4}. See Dahlquist and Björck (2008) and Davis and Rabinowitz (1984) for further details.

It should be remembered, however, that the convergence rate is still only an asymptotic bound, and gives no guarantees on how well the method performs for a specific function and choice of N. Simpson's method, for example, can perform quite poorly when the function being integrated has rapid changes or soft peaks. We observed this behaviour when implementing threshold models, as described below. Our response was to better tailor the integration method for the functions appearing. We noted that the numerical integrations we carried out all had the form

$$\int_a^b e^{-\frac{(x-\mu)^2}{2\sigma^2}} f(x)\ \mathrm{d}x \tag{3.11}$$

where μ and σ varied. Using the same general approach as Simpson's rule, we approximated $f(x)$, rather than the whole function $e^{-\frac{(x-\mu)^2}{2\sigma^2}} f(x)$, by a piecewise quadratic function $p(x)$. We could then use standard techniques and tools to evaluate $\int_a^b e^{-\frac{(x-\mu)^2}{2\sigma^2}} p(x)\ \mathrm{d}x$ numerically. The resulting integration formula, which we call the *Gaussian kernel method*, gives a significant improvement in numerical accuracy.

A further complication is that, in models of continuous traits, the trait value often ranges over the whole real line, or at least over the set of positive reals. Hence, we need to approximate integrals of the form

$$\int_{-\infty}^{\infty} f(x)\ \mathrm{d}x \text{ or } \int_0^{\infty} f(x)\ \mathrm{d}x \tag{3.12}$$

though the methods discussed above only apply to integrals on finite intervals. We truncate these integrals, determining values U and L such that the difference

$$\int_{-\infty}^{\infty} f(x)\ \mathrm{d}x - \int_L^U f(x)\ \mathrm{d}x \tag{3.13}$$

between the full integral $\int_{-\infty}^{\infty} f(x)\ \mathrm{d}x$ and the truncated integral $\int_{L}^{U} f(x)\ \mathrm{d}x$ can be bounded analytically. Other strategies are possible; see Dahlquist and Björck (2008) for a comprehensive review.

3.2.2 Pruning algorithm

Felsenstein has developed pruning algorithms for both continuous and discrete characters (Felsenstein, 1981a,b). His algorithm for continuous characters works only for Gaussian processes. Our approach is to take his algorithm for discrete characters and adapt it to continuous characters.

The (discrete character) pruning algorithm is an application of dynamic programming. For each node i, and each state x, we compute the probability of observing the states for all tips which are descendants of node i, conditional on node i having ancestral state x. This probability is called the partial likelihood at node i given state x. Our algorithm follows the same scheme, with one major difference. Since traits are continuous, we cannot store all possible partial likelihoods. Instead, we store likelihoods for a finite set of values and plug these values into a numerical integration routine.

Let i be the index of a node in the tree not equal to the root, let node j be its parent node. We define the partial likelihood, $\mathcal{F}_i(x_j)$ to be the likelihood for the observed trait values at the tips which are descendants of node i, conditional on the parent node j having trait value x_j. If node i is a tip with observed trait value x_i we have

$$\mathcal{F}_i(x_j) = f(x_i|x_j, \theta_i) \tag{3.14}$$

recalling that $f(x_i|x_j, \theta_i)$ is the density for the value of the trait at node i conditional on the value of the trait for its parent. More generally, we may only observe some value z_i for which we have the conditional probability $\pi(z_i|x_i)$ conditional on the trait value x_i. In this case, the partial likelihood is given by

$$\mathcal{F}_i(x_j) = \int f(\tilde{x}_i|x_j, \theta_i)\pi(z_i|\tilde{x}_i)\ \mathrm{d}\tilde{x}_i. \tag{3.15}$$

Suppose node i is not the root and that it has two children u, v. Since trait evolution is conditionally independent on disjoint subtrees, we obtain the recursive formula

$$\mathcal{F}_i(x_j) = \int f(\tilde{x}_i|x_j, \theta_i)\mathcal{F}_u(\tilde{x}_i)\mathcal{F}_v(\tilde{x}_i)\, d\tilde{x}_i. \tag{3.16}$$

Finally, suppose that node i is the root and has two children u and v. We evaluate the complete tree likelihood using the density of the trait value at the root,

$$\mathcal{L}(T) = \int f(x|\theta_r)\mathcal{F}_u(x)\mathcal{F}_v(x)\ \mathrm{d}x. \tag{3.17}$$

The bounds of integration in (3.15)—(3.17) will vary according to the model.

We use numerical integration techniques to approximate (3.15)—(3.17) and dynamic programming to avoid an exponential explosion in the computation time. Let N denote the number of function evaluations for each node. In practice, this might vary over the tree, but for simplicity we assume that it is constant. For each node i, we select $N + 1$ trait values

$$X_i[0] < X_i[1] < \cdots < X_i[N]. \tag{3.18}$$

How we do this will depend on the trait model and the numerical integration technique. If, for example, the trait values vary between a and b and we are applying Simpson's method with N intervals we would use $X_i[k] = a + \frac{b-a}{N}k$ for $k = 0, 1, 2, \ldots, N$.

We traverse the tree starting at the tips and working towards the root. For each non-root node i and $k = 0, 1, \ldots, N$ we compute and store an approximation $F_i[k]$ of $\mathcal{F}_i(X_j[k])$, where node j is the parent of node i. Note that this is an approximation of $\mathcal{F}_i(X_j[k])$ rather than of $\mathcal{F}_i(X_i[k])$ since $\mathcal{F}_i(x)$ is the partial likelihood conditional on the trait value for the *parent* of node i. The value approximation $F_v[i]$ is computed by applying the numerical integration method to the appropriate integral (3.15)—(3.17), where we replace function evaluations with approximations previously computed. See below for a worked example of this general approach.

The numerical integration methods we use run in time linear in the number of points being evaluated. Hence if n is the number of tips in the tree, the algorithm will run in time $O(nN^2)$. For the integration techniques described above, the convergence rate (in N) for the likelihood on the entire tree had the same order as the convergence rate for the individual one-dimensional integrations (see below for a formal proof of a specific model). We have therefore avoided the computational blow-out typically associated with such high-dimensional integrations, and achieve this without sacrificing accuracy.

3.2.3 Posterior densities for ancestral states

The algorithms we have described compute the joint density of the states at the tips, given the tree, the branch lengths, and other parameters. As with discrete traits, the algorithms can be modified to infer ancestral states for internal nodes in the tree. Here we show how to carry out reconstruction of the marginal posterior density of a state at a particular node. The differences between marginal and joint reconstructions are reviewed in (Yang, 2006, pg 121).

First consider marginal reconstruction of ancestral states at the root. Let u and v be the children of the root. The product $\mathcal{F}_u(x)\mathcal{F}_v(x)$ equals the probability of the observed character conditional on the tree, branch lengths, parameters and a state of x at the root. The marginal probability of x, ignoring the data, is given by the root density $f(x|\theta_r)$. Integrating the product of $\mathcal{F}_u(x)\mathcal{F}_v(x)$ and $f(x|\theta_r)$ gives the likelihood $\mathcal{L}(T)$, as in (3.17). Plugging these into Bayes' rule, we obtain the *posterior density* of the state at the root:

$$f(x_r|z_1,\ldots,z_n) = \frac{\mathcal{F}_u(x_r)\mathcal{F}_v(x_r)f(x_r|\theta_r)}{\mathcal{L}(T)}. \tag{3.19}$$

With general time reversible models used in phylogenetics, the posterior distributions at other nodes can be found by changing the root of the tree. Unfortunately the same trick does not work for many quantitative trait models, including the threshold model we study here. Furthermore, recomputing likelihoods for each possible root entails a large amount of unneccessary computation.

Instead, we derive a second recursion, this one starting at the root and working towards the tips. A similar trick is used to compute derivatives of the likelihood function in Felsenstein and Churchill (1996). For a node i and state x we let $\mathcal{G}_i(x)$ denote the likelihood for the trait values at tips which are *not* descendants of node i, conditional on node i having trait value x. If node i is the root r, then $\mathcal{G}_r(x)$ is 1 for all x.

Let node i be any node apart from the root, let node j be its parent and let node u be the other child of j (that is, the sibling of node i). We let $\tilde{x}$ denote the trait value at node j. Then $\mathcal{G}_i(x)$ can be written

$$\mathcal{G}_i(x) = \int f(\tilde{x}|x,\theta_i)\mathcal{G}_j(\tilde{x})\mathcal{F}_u(\tilde{x})\ \mathrm{d}\tilde{x}. \tag{3.20}$$

This integral can be evaluated using the same numerical integrators used when computing likelihoods. Note that $f(\tilde{x}|x,\theta_i)$ is the conditional density of the parent state given the child state, which is the reverse of the transition densities used to formulate the model. It should be noted that while Brownian motion has reversible transition probabilities, the OU process does not. How $\mathcal{G}_i(x)$ is computed will depend on the model and its properties; see below for an implementation of this calculation in the threshold model.

Once $\mathcal{G}_i(x)$ has been computed for all nodes, the actual (marginal) posterior densities are computed from Bayes' rule. Letting u, v be the children of node i,

$$f(x_i|z_1,\ldots,z_n) = \frac{\mathcal{G}_i(x_i)\mathcal{F}_u(x_i)\mathcal{F}_v(x_i)f(x_i)}{\mathcal{L}(T)}. \tag{3.21}$$

3.3 Case study: threshold models

In this section we show how the general framework can be applied to the threshold model of Wright (1934) and Felsenstein (2005, 2012). Each trait is modelled by a continuously varying "liability" which evolves along branches according to a Brownian motion process. While the underlying liability is continuous, the observed data are discrete: at each tip, we observe only whether the liability is above or below some threshold.

We will use standard notation for Gaussian densities. Let $\phi(x|\mu, \sigma^2)$ denote the density of a Gaussian random variable x with mean μ and variance σ^2; let

$$\Phi(y|\mu, \sigma^2) = \int_{-\infty}^{y} \phi(x|\mu, \sigma^2) \tag{3.22}$$

denote its cumulative density function, with inverse $\Phi^{-1}(\alpha|\mu, \sigma^2)$.

Let $X_1, \ldots, X_{2n-1}$ denote the (unobserved) liability values at the n tips and $n-1$ internal nodes. As above we assume that the $i < j$ whenever node i is a child of node j, so that the root has index $2n-1$.

The liability value at the root has a Gaussian density with mean μ_r and variance σ_r^2:

$$f(x_{2n-1}|\theta_r) = \phi(x_{2n-1}|\mu_r, \sigma_r^2). \tag{3.23}$$

Consider any non-root node i and let j be the index of its parent. Let t_i denote the length of the branch connecting nodes i and j. Then X_i has a Gaussian density with mean x_j and variance $\sigma^2 t_v$:

$$f(x_i|x_j, \theta_i) = \phi(x_i|x_j, \sigma^2 t_i). \tag{3.24}$$

Following Felsenstein (2005), we assume thresholds for the tips are all set at zero. We observe 1 if the liability is positive, 0 if the liability is negative, and ? if data is missing. We can include the threshold step into our earlier framework by defining

$$\pi(z_i|x_i) = \begin{cases} 1 & \text{if } z_i = 1 \text{ and } x_i > 0, \text{ or } z_i = 0 \text{ and } x_i \leq 0, \text{ or } z_i = ? \\ 0 & \text{otherwise.} \end{cases} \tag{3.25}$$

The likelihood function for observed discrete values $z_1, \ldots, z_n$ is then given by integrating over liability values for all nodes on the tree:

$$\mathcal{L}(T|z_1, \ldots, z_n) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \phi(x_{2n-1}|\mu_r, \sigma_r^2) \prod_{(i,j)} \phi(x_i|x_j, \sigma^2 t_i) \prod_{i=1}^{n} \pi(z_i|x_i) \, \mathrm{d}x_1 \ldots \mathrm{d}x_{2n-1}. \tag{3.26}$$

3.3.1 Implementation of threshold model

The first step towards computing $\mathcal{L}(T|z_1, \ldots, z_n)$ is to bound the domain of integration so that we can apply Simpson's method. Ideally, we would like these bounds to be as tight as possible, for improved efficiency. For the moment we will just outline a general procedure which can be adapted to a wide range of evolutionary models.

The "marginal (prior) density" of a single liability or trait value at a single node is the density for that liability value marginalizing over all other values and data. With the threshold model, the marginal density for the liability at node i is Gaussian with mean μ_r (like the root) and variance v_i equal to the sum of the variance at the root and the transition variances on the path from the root to node i. If P_i is the set of nodes from the root to node i, then

$$v_i = \sigma_r^2 + \sigma^2 \sum_{j \in P_i} t_j. \tag{3.27}$$

The goal is to constrain the error introduced by truncating the integrals with infinite domain. Let ϵ be the desired bound on this truncation error. Recall that the number of internal nodes in the tree is $n - 1$. Define

$$L_i = \Phi^{-1}\left(\frac{\epsilon}{2(n-1)} \middle| \mu_r, v_i\right) \tag{3.28}$$

and

$$U_i = \Phi^{-1}\left(1 - \frac{\epsilon}{2(n-1)} \middle| \mu_r, v_i\right). \tag{3.29}$$

The bounds L_i and U_i are chosen so that the (marginal) probability X_i lies outside the interval $[L_i, U_i]$ is at most $\epsilon/(n-1)$. For this model, these are given by the inverse distribution function of a Gaussian; other models would involve different transition densities. By the inclusion-exclusion principle, the joint probability $X_i \notin [L_i, U_i]$ for *any* internal node i is at most ϵ. We use this fact to bound the contribution of the

regions outside these bounds.

$$\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} f(x_{2n-1}|\mu_r,\sigma_r^2)\prod_{(u,v)} f(x_v|x_u,\theta_v)\prod_{i=1}^{n}\pi(z_i|x_i)\ \mathrm{d}x_1\ldots\mathrm{d}x_{2n-1}$$

$$-\int_{a_{2n-1}}^{b_{2n-1}}\cdots\int_{a_{n+1}}^{b_{n+1}}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} f(x_{2n-1}|\mu_r,\sigma_r^2)\prod_{(u,v)} f(x_v|x_u,\theta_v)\prod_{i=1}^{n}\pi(z_i|x_i)\ \mathrm{d}x_1\ldots\mathrm{d}x_{2n-1} \tag{3.30}$$

$$\leq \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} f(x_{2n-1}|\mu_r,\sigma_r^2)\prod_{(u,v)} f(x_v|x_u,\theta_v)\ \mathrm{d}x_1\ldots\mathrm{d}x_{2n-1}$$

$$-\int_{a_{2n-1}}^{b_{2n-1}}\cdots\int_{a_{n+1}}^{b_{n+1}}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} f(x_{2n-1}|\mu_r,\sigma_r^2)\prod_{(u,v)} f(x_v|x_u,\theta_v)\ \mathrm{d}x_1\ldots\mathrm{d}x_{2n-1} \tag{3.31}$$

$$\leq P\Big(X_{n+1}\notin[L_{n+1},U_{n+1}]\text{ or }X_{n+2}\notin[L_{n+2},U_{n+2}]\text{ or }\cdots\text{ or }X_{2n-1}\notin[L_{2n-1},U_{2n-1}]\Big) \tag{3.32}$$

$$<\epsilon. \tag{3.33}$$

We therefore compute values L_i, U_i for $n+1\leq i\leq 2n-1$ using (3.28) and (3.29) and use these bounds when carrying out integration at the internal nodes. We define

$$X_i[k]=L_i+\frac{U_i-L_i}{N}k \tag{3.34}$$

for $k=0,1,\ldots,N$ for each internal node i.

The next step is to use dynamic programming and numerical integration to compute the approximate likelihood. Let node i be a tip of the tree, let node j be its parent and let z_i be the binary trait value at this tip. For each $k=0,1,\ldots,N$ we use standard error functions to compute

$$F_i[k] = \mathcal{F}_i(X_j[k]) \tag{3.35}$$

$$= \begin{cases} \int_0^\infty \phi(\tilde{x}|X_j[k],\sigma^2 t_i)\ \mathrm{d}\tilde{x} & \text{if } z_i=1\\ \int_{-\infty}^0 \phi(\tilde{x}|X_j[k],\sigma^2 t_i)\ \mathrm{d}\tilde{x} & \text{if } z_i=0\\ 1 & \text{if } z_i=?. \end{cases} \tag{3.36}$$

Here $\phi(x|\mu,\sigma^2)$ is the density of a Gaussian with mean μ and variance σ^2.

Now suppose that node i is an internal node with parent node j and children u and v. Applying Simpson's rule to the bounds L_i, U_i to (3.16) we have for each

$k = 0, 1, \ldots, N$:

$$F_i[k] = \frac{U_i - L_i}{N} \sum_{\ell=0}^{N} w_\ell \phi(X_i[\ell] | X_j[k], \sigma^2 t_i) F_u[\ell] F_v[\ell] \tag{3.37}$$

$$\approx \mathcal{F}_i(X_j[k]). \tag{3.38}$$

Suppose node i is the root, and u, v are its children. Applying Simpson's rule to (3.17) gives an approximate likelihood of

$$\frac{U_{2n-1} - L_{n-1}}{N} \sum_{\ell=0}^{N} w_\ell \phi(X_i[\ell] | \mu_r, \sigma_r^2) F_u[\ell] F_v[\ell]. \tag{3.39}$$

Pseudo-code for the algorithm appears in Algorithm 2.

We can estimate posterior densities using the recursion (3.20) followed by equation (3.21). The conditional density

$$f(\tilde{x}|x, \theta_i) = \phi\left(\tilde{x} \middle| \mu_r + \frac{v_j}{v_i}(x - \mu_r), \frac{\sigma^2 t_i v_j}{v_i}\right) \tag{3.40}$$

can be obtained by plugging the transitional density

$$f(x|\tilde{x}, \theta_i) = \phi(x|\tilde{x}, \sigma^2 t_i) \tag{3.41}$$

and the two marginal densities (3.27)

$$f(\tilde{x}) = \phi(\tilde{x}, v_j), \quad f(x) = \phi(x, v_i) \tag{3.42}$$

into the identity $f(\tilde{x}|x, \theta_i) = f(x|\tilde{x}, \theta_i)\frac{f(\tilde{x})}{f(x)}$. We thereby obtain the recursion

$$\mathcal{G}_i(x) = \int \phi\left(\tilde{x} \middle| \mu_r + \frac{v_j}{v_i}(x - \mu_r), \frac{\sigma^2 t_i v_j}{v_i}\right) \mathcal{G}_j(\tilde{x}) \mathcal{F}_u(\tilde{x})\, d\tilde{x} \tag{3.43}$$

which we estimate using Simpson's method. Algorithm 3 estimates values of the posterior densities at each node, evaluated using the same set of grid points as used in Algorithm 2. An additional round of numerical integration can be used to obtain posterior means and variances.

3.3.2 Validation of Algorithm 2

Theorem 1. *Algorithm 2 runs in $O(nN^2)$ time and approximates $L(T)$ with $O(nN^{-4})$ error.*

Algorithm 2: Compute probability of a threshold character.

Input:

N: Number of intervals in numerical integration.

$t_1, \ldots, t_{2n-2}$: branch lengths in tree.

μ_r, σ_r^2: mean and variance of root density.

σ^2: variance of transition densities (per unit branch length).

$z_1, \ldots, z_n$ observed character ($z_i \in \{+1, 0, ?\}$).

Output:

Probability L of observed character under the threshold model.

Construct the vector $\mathbf{x} = [0, 1, 2, \ldots, N]/N$.

Construct the vector $\mathbf{w} = [1, 4, 2, 4, 2, \ldots, 4, 2, 1]/3$ as in (3.10).

Compute the path length p_i from the root to each node i.

Initialize $F_i[k] \leftarrow 1$ for all nodes i and $0 \leq k \leq N$.

For all $i = n+1, n+2, \ldots, 2n-1$

$L_i \leftarrow \Phi^{-1}(\frac{nN^{-4}}{2(n-1)} | \mu_r, \sigma_r^2 + \sigma^2 p_i)$

$U_i \leftarrow \Phi^{-1}(1 - \frac{nN^{-4}}{2(n-1)} | \mu_r, \sigma_r^2 + \sigma^2 p_i)$

$X_i \leftarrow (U_i - L_i)\mathbf{x} + L_i$

For all tip nodes $i = 1, 2, \ldots, n$

Let j be the index of the parent of node i.

For $k = 0, \ldots, N$

If $z_i = 1$

$F_i[k] \leftarrow 1 - \Phi(0; X_j[k], \sigma^2 t_i)$

else if $z_i = 0$

$F_i[k] \leftarrow \Phi(0; X_j[k], \sigma^2 t_i)$

For all internal nodes $i = n+1, ..., 2n-2$, excluding the root

Let j be the index of the parent of node i.

Let u, v be the indices of the children of node i.

For $k = 0, 1, \ldots, N$

$$F_i[k] \leftarrow \frac{U_i - L_i}{N} \sum_{\ell=0}^{N} \mathbf{w}_\ell \phi(X_i[\ell]; X_j[k], \sigma^2 t_i) F_u[\ell] F_v[\ell]$$

Let u, v be indices of the the children of the root.

$$L \leftarrow \frac{U_{2n-1} - L_{n-1}}{N} \sum_{\ell=0}^{N} \mathbf{w}_\ell \phi(X_i[\ell]; \mu_r, \sigma_r^2) F_u[\ell] F_v[\ell]$$

Algorithm 2: Pseudo-code of the likelihood approximation algorithm for a single character, under the threshold model. The nodes are numbered in increasing order from tips to the root.

Algorithm 3: Compute posterior densities

Input:

N, $t_1, \ldots 2n-2$, μ_r, σ_r^2, and σ^2 as in Algorithm 2.

Vector p, likelihood L and arrays F_i computed in Algorithm 2.

Output:

Arrays H_i for each internal node i.

Construct the vectors $\mathbf{x}$, $\mathbf{w}$, $\{L_i : i \in \{n+1, \ldots, 2n-2\}\}$,

$\{U_i : i \in \{n+1, \ldots, 2n-2\}\}$, and path lengths p_i as in Algorithm 2.

$G_{2n-1}[k] \leftarrow 1$ for all k.

For all $i = 2n-2, 2n-3, \ldots, n+1$

Let j be the index of the parent of node i.

Let v be the index of the sibling of node i.

For $k = 0, 1, \ldots, N$

$$\mu \leftarrow \mu_r + \frac{\sigma_r^2 + \sigma^2 p_j}{\sigma_r^2 + \sigma^2 p_i} \left(X_i[k] - \mu_r\right)$$

$$V \leftarrow \frac{\sigma^2 t_i \left(\sigma_r^2 + \sigma^2 p_j\right)}{\sigma_r^2 + \sigma^2 p_i}$$

$$G_i[k] \leftarrow \frac{U_j - L_j}{N} \sum_{\ell=0}^{N} \mathbf{w}_\ell \phi(X_j[\ell]; \mu, V) G_j[\ell] F_v[\ell]$$

For all $i = n+1, \ldots, 2n-1$

Let u, v be the children of node i.

For all $k = 0, 1, \ldots, N$

$$H_i[k] \leftarrow \frac{1}{L} G_i[k] F_u[k] F_v[k] \phi(X_i[k] | \mu_r, \sigma_r^2 + \sigma^2 p_i)$$

Algorithm 3: Pseudo-code for the algorithm to efficiently compute ancestral posterior densities under the threshold model. At the termination of the algorithm, $H_i[k]$ is an estimate of the posterior density at internal node i, evaluated at $x = X_i[k]$.

Proof
The running time follows from the fact that for each of the $O(n)$ nodes in the tree we carry out $O(N)$ applications of Simpson's method.

Simpson's rule has $O(N^{-4})$ convergence on functions with bounded fourth derivatives (Dahlquist and Björck, 2008). For each node i, let n_i denote the number of tips which are descendents of the node. Using induction on (3.16), we see that for all nodes i, the fourth derivative of $\mathcal{F}_i(x)$ is $O(n_i)$.

If we use $\epsilon = nN^{-4}$ in (3.28) and (3.29) then replacing the infinite domain integrals with integrals on $[L_i, U_i]$ introduces at most nN^{-4} error. Using a second induction proof on (3.16) and (3.37) together with the bound on fourth derivatives, we have that $|\mathcal{F}_i(X_j[k]) - F_i[k]|$ is at most $O(n_i N^{-4})$ for all nodes i, where node j is the parent of node i. In this way, we obtain an error bound of $O(n_{2n-1}N^{-4}) = O(nN^{-4})$ on the approximation of $\mathcal{L}(T|z_1, \ldots, z_n, \theta)$. □

3.4 Evolutionary precursor model on extrafloral nectary data

To study the methods in practice, we reanalyse trait data published by Marazzi *et al.* (2012), using a fixed phylogeny. This trait data is for the absence/presence of extrafloral nectaries (EFNs), which are nectar-secreting glands that occur outside of the flower(s) of a plant.

3.4.1 Precursor model and motivation

Marazzi *et al.* (2012) introduce and apply a new discrete state model for morphological traits which, in addition to states for presence and absence, incorporates an intermediate "pre-cursor" state. Whenever the intermediate state is observed at the tips it is coded as "absent". The motivation behind the model is that the intermediate state represents evolutionary precursors, changes which are necessary for the evolution of a new state but which may not be directly observed. These pre-cursors could explain repeated parallel evolution of a trait in closely related traits (Marazzi *et al.*, 2012). They compiled a data set recording presence or absence of plant EFNs across a phylogeny of 839 species of Fabales, fitting their models to these data.

3.4.2 Application of the threshold model for this problem

The threshold model also involves evolutionary pre-cursors in terms of changes in ancestral liabilities. We use these models, and our new algorithms to analyse the EFN dataset. Our analysis also makes use of the time-calibrated phylogeny inferred by Simon *et al.* (2009), although unlike Marazzi *et al.* (2012), we ignore phylogenetic uncertainty.

3.5 Results

3.5.1 Protocol

We ran three experiments to assess the performance of the new algorithms. For the first experiment, we examine the rate of convergence of the likelihood algorithm as we increase N. This is done for the 'All' EFN character (Character 1 in Marazzi *et al.* (2012)) for a range of estimates for the liability variance at the root, σ_r^2. The interest in σ_r^2 stems from its use in determining bounds L_i, U_i for each node, with the expectation that as σ_r^2 increases, the convergence of the integration algorithm will slow. The mean liability at the root, μ_r, was determined from the data using Maximum Likelihood estimation.

We also examined convergence of the algorithm on randomly generated characters. We first evolved liabilities according to the threshold model, using the parameter settings obtained above. To examine the difference in performance for non-phylogenetic characters we also simulated binary characters by simulated coin flipping. Twenty replicates were carried out for each case.

The second experiment extends the model comparisons carried out in Marazzi *et al.* (2012) to include the threshold models. For this comparison, we fix the transitional variance σ^2 at one, since changing this values corresponds to a rescaling of the Brownian process, with no change in likelihood. With only one character, the maximum likelihood estimate of the root variance σ_r^2 is zero, irrespective of the data. This leaves a single parameter to infer: the value of the liability at the root state. We computed a maximum likelihood estimate for the state at the root, then applied our algorithm with a sufficiently large value of N to be sure of convergence. The Akaike Information Criterion (AIC, see Akaike (1974)) was determined and compared with those obtained for the model of Marazzi *et al.* (2012).

For the third experiment, we determine the marginal posterior densities for the

liabilities at internal nodes, using Algorithm 3.

These posterior probabilities are then mapped onto the phylogeny, using shading to denote the (marginal) posterior probability that a liability is larger than zero. We therefore obtain a figure analogous to Supplementary Figure 7 of Marazzi *et al.* (2012).

3.5.2 Convergence of numerical integration

To examine convergence, we compute the absolute error of each likelihood approximation; since the actual likelihood is not available we use the approximation when $N = 1000$. Plots of error versus N are given in Figure 3.1, both for Simpson's method (left) and for the modified Gaussian kernel method (right). For larger N, the error in a log-log plot decreases with slope at most -4 (as indicated), corresponding to N^{-4} convergence of the method. Log-log plots of error versus N for the simulated data are given in Figure 3.2. In each case, the method converges for by $N \approx 30$.

While the level of convergence for both algorithms is correct, the accuracy of the method based on Simpson's method is far worse. When a branch length is short, the transition density becomes highly peaked, as does the function being integrated. Such functions are difficult to approximate with piecewise quadratics, and Simpson's method can fail miserably. Indeed, for $N < 50$, we would often observe estimated likelihoods equal to 0, or estimates of probability greater than one (These were omitted from the plots). While we can always bound estimates computed by the algorithm, a sounder approach is to improve the integration technique. This we did using the Gaussian kernel method, and the result was far improved accuracy for little additional computation. For the remainder of the experiments in this chapter with this model we used the Gaussian kernel method when carrying out numerical integration.

3.5.3 Threshold model vs. Precursor model

Akaike Information Criterion (AIC)

Marazzi *et al.* (2012) describe AIC comparisons between their pre-cursor model and a conventional binary trait model. The AIC of a model is calculated as

$$AIC = -2 * \log\left(\max_{\theta} L(\text{data}|\theta)\right) + 2k$$

where θ is the set of parameters of the model and k is the number of parameters. After the AIC is calculated for multiple models, the model with the smallest AIC value is classified as the best fitting model in the set. We extend the AIC comparisons in

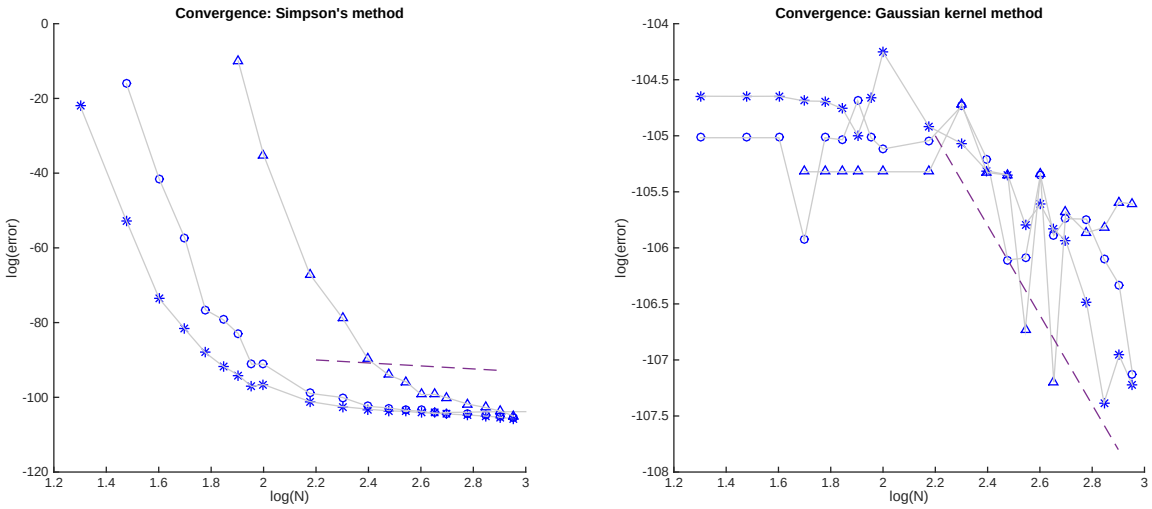

Figure 3.1: Log-log plots of error as a function of N for the dynamic programming algorithm with Simpson's method (left) and with the Gaussian kernel method (right). The likelihoods were computed under the threshold model on EFN trait data for an 839 taxon tree. Dotted lines have slope -4 (corresponding to convergence rate of N^{-4}. Note the difference in scale for the two methods.). Logarithms computed to base 10. Letting h be the height of the tree, the circles in both plots represent errors when $\sigma_r^2 = h$, the asterisks represent errors when $\sigma_r^2 = 0.1h$, and the triangles represent errors when $\sigma_r^2 = 10h$. As the plots show, the logarithms of the errors from the Gaussian kernel method are no larger than -104 while the logarithms of the errors from Simpson's method can be between 0 and -100.

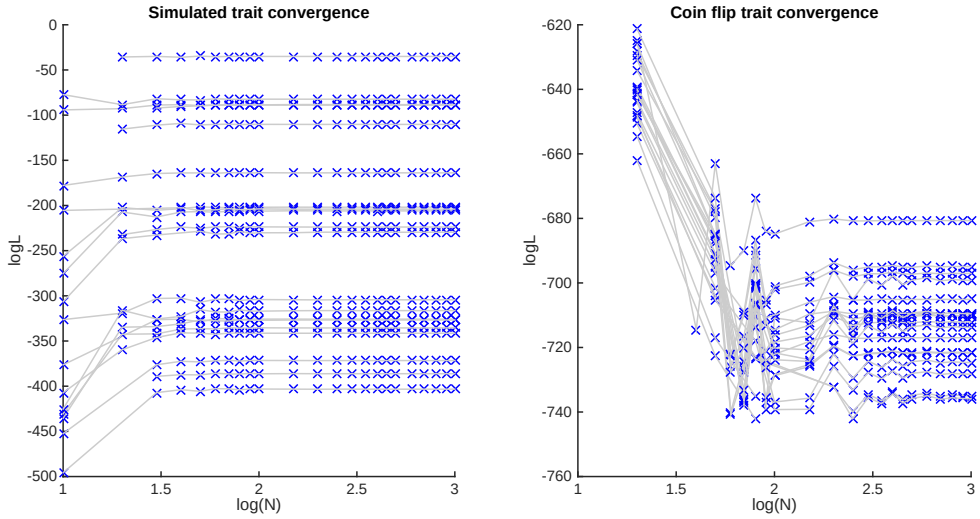

Figure 3.2: Plots of log-likelihood values as a function of $\log(N)$ for the two types of data simulated from the fixed EFN tree, computed using our algorithm together with the Gaussian kernel method. Logarithms computed to base 10.

Marazzi *et al.* (2012) to include the threshold model. This is a one parameter model, the parameter being the value of the liability at the root. We used the MATLAB command fminsearch with multiple starting points to compute the maximum likelihood estimate for this value. The resulting log-likelihood was $\log L = -240.6$, giving an AIC of 483.2. This compares to an AIC of 507.4 for the (two parameter) binary character model and an AIC of 495.4 for the (one parameter) precursor model of Marazzi *et al.* (2012).

We analyzed the five other EFN traits in the same way, and present the computed AIC values in Table 3.1, together with AIC values for the two parameter binary state model and one parameter precursor model computed by Marazzi *et al.* (2012) (and the 2 parameter precursor model for trait 6). We see that the threshold model fits better than either the binary or precursor models for all of the six traits.

It is not clear, *a priori*, why the threshold model would appear to fit some data better than the precursor model since they appear to capture similar evolutionary phenomena. It would be useful to explore this observation more thoroughly, given the new computational tools, perhaps incorporating phylogenetic error in a manner similar to Marazzi *et al.* (2012).

Trait	Model	k	$\log L$	AIC
1 (All)	Binary	2	-251.7	507.4
	Precursor	1	-246.7	495.4
	Threshold	1	-240.6	**483.2**
2 (Leaves)	Binary	2	-240.3	484.6
	Precursor	1	-234.5	470.9
	Threshold	1	-230.6	**463.1**
3 (Inflorescence)	Binary	2	-108.3	220.5
	Precursor	1	-110.9	223.9
	Threshold	1	-108.3	**218.5**
4 (Trichomes)	Binary	2	-86.7	177.3
	Precursor	1	-86.9	175.9
	Threshold	1	-85.8	**173.5**
5 (Substitutive)	Binary	2	-163.0	330.1
	Precursor	1	-161.6	325.3
	Threshold	1	-161.3	**324.6**
6 (True)	Binary	2	-132.3	268.7
	Precursor	1	-131.1	264.3
	Precursor	2	-126.7	257.3
	Threshold	1	-125.3	**252.6**

Table 3.1: Table of log-likelihood and AIC values for the binary character, pre-cursor, and threshold models on six EFN traits. Column k indicates numbers of parameters for each model. Data for the binary and precursor models copied from Table 1 in Marazzi *et al.* (2012). All likelihoods and AIC values rounded to 1 d.p. Boldface indicates the best fitting model for each trait. A pre-cursor model with one parameter was used for all experiments, except for trait 6 where it obtained a better AIC than the one-parameter model (see discussion in Marazzi *et al.* (2012)).

3.5.4 Marginal posterior probabilities

Figure 3.3 gives a representation of how the (marginal) posterior liabilities change over the tree. Branches are divided into three classes according to the posterior probability that the liability is positive, with lineages with posterior probability > 0.7 colored red, lineages with posterior probability < 0.3 colored white, and remaining lineages colored pink.

This diagram can be compared to Supplementary Figure 7, of Marazzi *et al.* (2012). The representations are, on the whole, directly comparable. A positive liability corresponds, roughly, to an ancestral precursor state. Both analyses suggest multiple origins of a precursor state, for example for a large clade of Mimosoidae. Interestingly, there are several clades where the analysis of Marazzi *et al.* (2012) suggests widespread ancestral distribution of the precursor state whereas our analysis indicates a negative liability at the same nodes.

Once again, our analysis is only preliminary, our goal here simply being to demonstrate what calculations can now be carried out.

3.6 Discussion

We have introduced a new framework for the computation of likelihoods from continuous characters, and illustrated the framework using an efficient algorithm for evaluating (approximate) likelihoods under Wright and Felsenstein's threshold model.

This framework opens up possibilities in several directions. The numerical integration, or numerical quadrature, literature is vast. In this article, we have focused in on a popular and simple numerical integration method, and our algorithm should be seen as a proof of principle rather than a definitive threshold likelihood method. There is no question that the numerical efficiency of Algorithm 1 could be improved significantly through the use of more sophisticated techniques: better basis functions or adaptive quadrature methods for a start.

The connection with Felsenstein's (discrete character) pruning algorithm also opens up opportunities for efficiency gains. Techniques such as storing partial likelihoods, or approximating local neighborhoods, are fundamental to efficient phylogenetic computations on sequence data (Felsenstein, 1981b; Larget and Simon, 1998; Swofford, 2002; Pond and Muse, 2004; Stamatakis, 2006). These tricks could all now be applied to the calculation of likelihoods from continuous traits. Finally, we stress that the algorithm does not depend on special characteristics of the continuous trait model, beyond

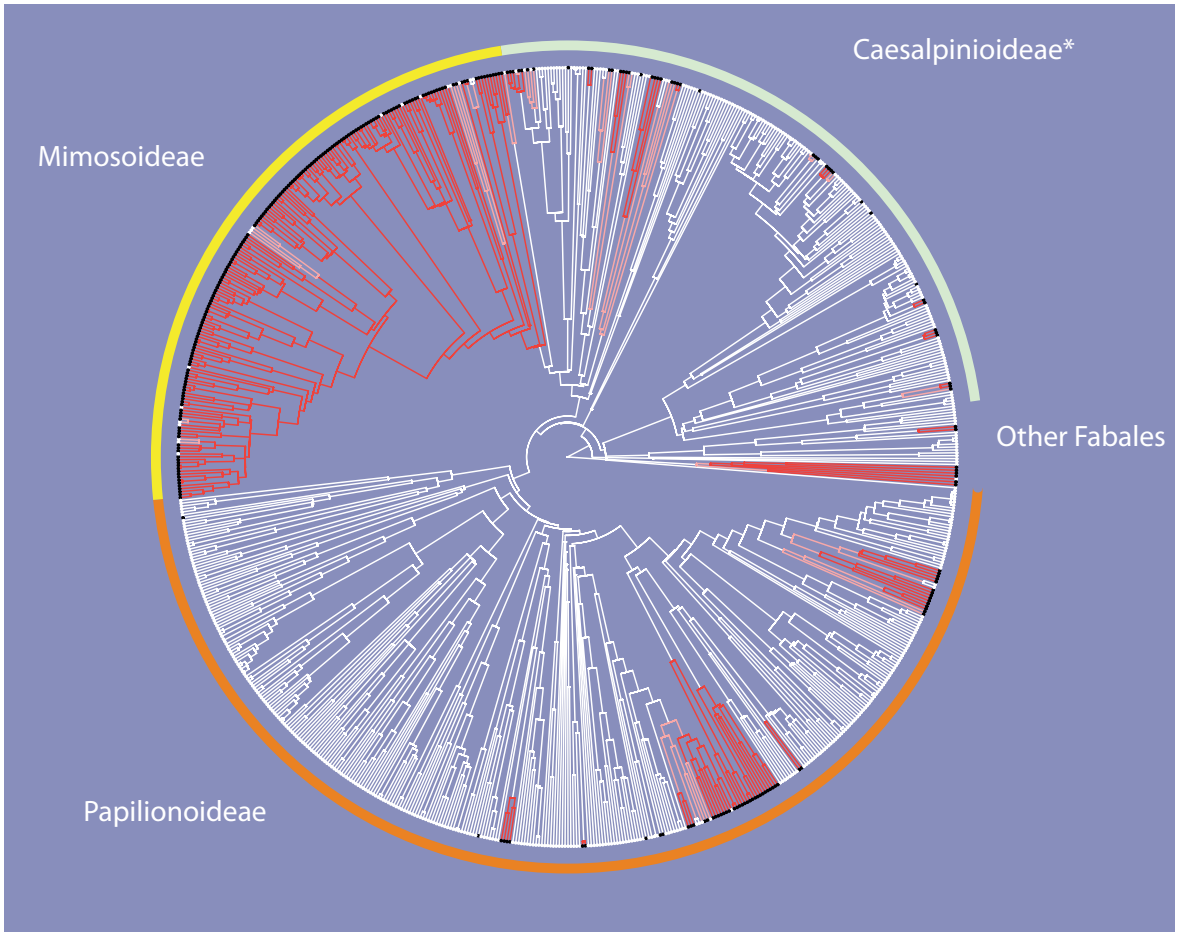

Figure 3.3: Marginal posterior probabilities for the liabilities, for EFN trait 1 of Marazzi *et al.* (2012) on the phylogeny inferred by Simon *et al.* (2009). Lineages with posterior probability > 0.7 colored red, lineages with posterior probability < 0.3 colored white, and remaining lineages colored pink.

conditional independence of separate lineages. Felsenstein's pruning algorithm for continuous characters is limited to Gaussian processes and breaks down if, for example, the transition probabilities are governed by Lévy processes (Landis *et al.*, 2013). In contrast, our approach works whenever we can numerically evaluation transition densities, and indeed only a few minor changes would transform our Algorithm 2 to one implementing on a far more complex evolutionary process.

3.7 Extension

3.7.1 Integration for multivariate traits

The numerical integration algorithm and the precursor model from Marazzi *et al.* (2012) assumes that each leaf has one binary character. Naturally, there are datasets where each species has multiple characters or a potentially polymorphic set of single-trait individuals. We can extend the numerical integration algorithm to multivariate character sets by first generalizing the multi-dimensional numerical integration formula given in Cheney and Kincaid (2012). Let $f(x_1, x_2, ..., x_m)$ be a function in m dimensions, let $y_{ij} = a_i + \frac{j(b_i - a_i)}{N}$ be the j-th precalculated evaluation point in dimension i for $j \in \{0, 1, ..., N\}$, and let w_{ij} be the weight in dimension i corresponding to y_{ij}. The numerical integration formula therefore can be written as

$$\int_{a_1}^{b_1} \dots \int_{a_m}^{b_m} f(x_1, ..., x_m)\, \mathrm{d}x_1 \dots \mathrm{d}x_m \approx \sum_{j_1=0}^{N} \dots \sum_{j_m=0}^{N} \left(\prod_{k=1}^{m} w_{kj_k} \right) f(y_{1j_1}, ..., y_{mj_m}) \tag{3.44}$$

The extended numerical integration method is shown in Algorithm 4.

Algorithm 4: Compute probability of a multivariate character.

Input:

N: Number of intervals in numerical integration.

$t_1, \ldots, t_{2n-2}$: branch lengths in tree.

θ_i: parameters at node i for $i \in \{1, 2, ..., 2n-1\}$

$z_1, \ldots, z_n$ observed character ($z_i \in \mathbb{R}^m$)

Output:

Probability L of observed character.

Construct the vector $\mathbf{x} = [0, 1, 2, \ldots, N]/N$.

Construct the vector $\mathbf{w} = [1, 4, 2, 4, 2, \ldots, 4, 2, 1]/3$ as in (3.10)

Compute the path length p_i from the root to each node i.

Initialize $F_i[k] \leftarrow 1$ for all nodes i and $k \in \{0, 1, \ldots, N\}^m$.

For all $i = 1, 2, \ldots, 2n-1$

For all $j = 1, 2, \ldots, m$

$$X_{ij} \leftarrow (b_{ij} - a_{ij})\mathbf{x} + a_{ij}$$

For all tip nodes $i = 1, 2, \ldots, n$

Let j be the index of the parent of node i

For $k \in \{0, 1, \ldots, N\}^m$

$$F_i[k] \leftarrow \left(\prod_{j=1}^{m} \frac{b_{ij} - a_{ij}}{N}\right) \sum_{\ell \in \{0,1,\ldots,N\}^m} \left(\prod_{j=1}^{m} \mathbf{w}_{\ell_j}\right) f_i\left(X_i[\ell]; X_j[k], \theta_i\right)$$

where $X_i[\ell] = \{X_{i1}[\ell_1], X_{i2}[\ell_2], \ldots, X_{im}[\ell_m]\}$

For all internal nodes $i = n+1, ..., 2n-2$, excluding the root

Let j be the index of the parent of node i

Let u, v be the indices of the children of node i

For $k = 0, 1, \ldots, N$

$$F_i[k] \leftarrow \left(\prod_{j=1}^{m} \frac{b_{ij} - a_{ij}}{N}\right) \sum_{\ell \in \{0,1,\ldots,N\}^m} \left(\prod_{j=1}^{m} \mathbf{w}_{\ell_j}\right) f_i\left(X_i[\ell]; X_j[k], \theta_i\right) F_u[\ell] F_v[\ell]$$

Let u, v be indices of the the children of the root.

$$L \leftarrow \left(\prod_{j=1}^{m} \frac{b_{2n-1,j} - a_{2n-1,j}}{N}\right) \sum_{\ell \in \{0,1,\ldots,N\}^m} \left(\prod_{j=1}^{m} \mathbf{w}_{\ell_j}\right) f_{2n-1}\left(X_{2n-1}[\ell]; \mu_r, \theta_{2n-1}\right) F_u[\ell] F_v[\ell]$$

Algorithm 4: Pseudo-code of the likelihood approximation algorithm for a multivariate character. The nodes are numbered in increasing order from tips to the root.

Chapter 4

Computing likelihoods of trees for diffusion models

In the previous chapter, we explored a dynamic programming algorithm which computes likelihoods of trees and relevant parameters given genetic data. The algorithm computes the likelihood from a multidimensional integral based on transition densities of quantitative characters by using numerical integration to compute each individual integral using a finite number of data points. The algorithm is efficient, simple to use, and has fast convergence (depending on the numerical integration rule used), but only works if the transition densities are explicitly known and can give highly inaccurate approximations if too few data points are used.

In this chapter, we consider a particular family of continuous trait models, where the 'trait' is the ancestral allele frequencies and the transition densities are determined implicitly via a diffusion. The fact that transition densities are not immediately available means that the algorithms in Chapter 3 cannot be applied immediately. We also take a different approach to numerical integrations, using basis functions rather than evaluation of the partial likelihoods at specific grid points. The result is higher efficiency, perhaps at the cost of simplicity.

We first give a review of diffusion based models.

4.1 Description of the model

The use of diffusion processes to model changes in allele frequencies over time dates back to Fisher (1922), where the change in frequency is approximated by a heat equation (Wahl, 2011). However, Sewall Wright independently came up with another differ-

ential equation for allele frequency (Wright, 1931). Fisher examined the manuscript for Wright (1931), declared Wright's equation to be correct, declared the equation in Fisher (1922) to be incorrect, and wrote to Wright in regards to the equations. Wright acknowledged Fisher's message in the published version of Wright (1931) and Fisher acknowledged Wright's findings on page 87 of Fisher (1930).

Let $f(x;p,t)$ be the probability density of allele frequency x at time t given the initial allele frequency p. Wright (1945) defined f by means of the Kolmogorov forward equation, a PDE (partial differential equation) of the form

$$\frac{\partial f(x;p,t)}{\partial t} = -\frac{\partial}{\partial x}\left(a(x)f(x;p,t)\right) + \frac{1}{2}\frac{\partial^2}{\partial x^2}\left(b(x)f(x;p,t)\right) \tag{4.1}$$

where $a(x)$ is the drift coefficient and $b(x)$ is the diffusion coefficient (see pages 137-138 of Ewens (2004)). The function $f(x;p,t)$ also satisfies the Kolmogorov backward equation

$$\frac{\partial f(x;p,t)}{\partial t} = a(p)\frac{\partial f(x;p,t)}{\partial p} + \frac{1}{2}b(p)\frac{\partial^2 f(x;p,t)}{\partial p^2} \tag{4.2}$$

(see page 138 of Ewens (2004)). We will be mostly working with the backward equation in this chapter. The continuous path processes defined by these PDEs are called diffusion processes, or simply, diffusions. Let $x(t)$ denote the allele frequency at time t and let T denote the set of time values in the process. The continuity of a diffusion process is defined by the concept of sample path continuity, namely

$$E\left(|x(t) - x(s)|^{\alpha}\right) \leq C|t-s|^{1+\beta}$$

for some $\alpha > 0, \beta > 0, C > 0$, and $h > 0$ and for all $s, t \in T$ such that $|t-s| \leq h$ (see page 39 of Kloeden and Platen (1992)).

In this chapter, we will focus on one standard diffusion model for allele frequency change, a model which includes the effects of drift, mutation, and selection on allele frequency. We begin with a review of the diffusion process, the bulk of the review based on pages 17, 98, 137-138, and 156-158 of Ewens (2004). The random process models changes over time in the proportion $x(t)$ of allele 1 individuals to the total population, with a state space $x(t) \in [0,1]$. There are four parameters h, s, u, and v; the parameters s and h determine the three genotype fitnesses

$$w_{11} = 1+s, \qquad w_{12} = 1+sh, \qquad w_{22} = 1. \tag{4.3}$$

Parameter u is the mutation rate from allele 1 to allele 2, and parameter v is the mutation rate from allele 2 to allele 1. The diffusion model assumes that s, u, and v

are all $O\left(N^{-1}\right)$ and utilizes scaled versions of these three parameters

$$\alpha = 2Ns, \qquad \beta_1 = 2Nu, \qquad \beta_2 = 2Nv. \tag{4.4}$$

We will assume that u and v are both non-zero, implying that the process has no fixed points and has a unique stationary distribution (see page 98 of Ewens (2004)). If we let δx and δt denote the change in x and t, respectively, then three of the moments of δx in a diffusion process are expressed as

$$E\left(\delta x\right) = \quad a(x)\delta t + o\left(\delta t\right) \tag{4.5}$$

$$\text{var}\left(\delta x\right) = \quad b(x)\delta t + o\left(\delta t\right) \tag{4.6}$$

$$E\left(\left|\delta x\right|^3\right) = \qquad o\left(\delta t\right). \tag{4.7}$$

The moments of the particular diffusion model we look at are

$$E\left(\delta x|x\right) \quad = \left(\alpha x(1-x)(x+h(1-2x)) - \beta_1 x + \beta_2(1-x)\right)(2N)^{-1} + o\left(N^{-1}\right) \tag{4.8}$$

$$\text{var}\left(\delta x|x\right) \quad = \left(x(1-x)\right)\delta t + o\left(N^{-1}\right) \tag{4.9}$$

$$E\left(\left|\delta x\right|^3\right) \quad = o\left(N^{-1}\right). \tag{4.10}$$

These moments are derived from the transition matrix of a discrete state space model with transition probabilities

$$p_{ij} = \binom{2N}{j}(\psi_i)^j(1-\psi_i)^{2N-j}. \tag{4.11}$$

Here,

$$\psi_i = \bar{w}^{-1}\left(\left\{w_{11}x^2 + w_{12}x(1-x)\right\}(1-u) + \left\{w_{12}x(1-x) + w_{22}(1-x)^2\right\}v\right) \tag{4.12}$$

and

$$\bar{w} = w_{11}x^2 + 2w_{12}x(1-x) + w_{22}(1-x)^2. \tag{4.13}$$

Moments (4.8) - (4.10) are consistent with moments (4.5) - (4.7) if we choose $\delta t = \left(2N\right)^{-1}$ and we choose the following drift and diffusion coefficients:

$$a(x) = \qquad \alpha x(1-x)(x+h(1-2x)) - \beta_1 x + \beta_2(1-x) \tag{4.14}$$

$$b(x) = \qquad x(1-x). \tag{4.15}$$

We will now describe the model we use for allele frequencies on a binary population tree with k leaves and $k-1$ internal nodes. For each node v, let x_v denote the random proportion of allele 1 in the population at node v and if v is not the root, we let p_v

denote the parental node of v. In addition, assume that each node is numbered from 1 to $r = 2k - 1$ such that the leaves are numbered from 1 to k, the root is numbered $2k - 1$, and the parents have larger numbers than their children.

Let $\pi(x_r)$ denote the stationary distribution at the root from which allele frequency x_r is drawn. The stationary (equilibrium) distribution satisfies the relation

$$\lim_{t\to\infty} f(x_r; p, t) = \pi(x_r) \tag{4.16}$$

irrespective of the initial value p (see page 90 of Ewens (2004)). The random sampling of allele frequency begins at the root, where a frequency is randomly chosen from $\pi(\cdot)$ and assigned to the root. Afterwards, at each node v descended from the root, the allele frequency x_v is randomly sampled from $f(\cdot; x_{p_v}, t_v)$ where t_v is the length of the branch between v and p_v. The frequencies at the non-leaf nodes are sampled in any order where each node is assigned an allele frequency after its parent node has been assigned a frequency. Let l denote a leaf and let n_l denote the total number of individuals in the sample at leaf l. After a leaf l has been assigned an allele frequency x_l, the number of individuals $m_l \leq n_l$ which have allele 1 is randomly chosen through binomial sampling from the distribution

$$P(n_l|m_l, x_l) = \binom{n_l}{m_l} x_l^{m_l}(1 - x_l)^{n_l - m_l}. \tag{4.17}$$

Let D denote the distribution of allele counts among the leaves and let θ denote the parameters of the model. The likelihood $P(D|\theta)$ is obtained from integrating over all allele frequencies at all nodes

$$P(D|\theta) = \int_0^1 \dots \int_0^1 \pi(x_r) \left(\prod_{i=1}^{2k-2} f(x_i; x_{p_i}, t_i) \right) \left(\prod_{l=1}^{k} P(n_l|m_l, x_l) \right) \mathrm{d}x_1 \dots \mathrm{d}x_{2k-1}. \tag{4.18}$$

There have been several methods for inferring trees and relevant parameters given the observed allele frequency data. Among these are methods based on the multi-population AFS (allele frequency spectrum) which is described in Gutenkunst *et al.* (2009) as "the joint distribution of allele frequencies across diallelic variants;" some sources (for example, Hernandez *et al.* (2007)) refer to the spectrum as the site frequency spectrum (or SFS).

Gutenkunst *et al.* (2009) discuss a method which models the AFS by using a diffusion approximation; however, their method uses a numerical strategy which scales poorly in the sample size and number of populations. Sawyer and Hartl (1992) and

Bustamante *et al.* (2001) discuss methods which infer the maximum likelihood estimates of parameters from the AFS based on the Poisson random field model, which assumes that the sites are independent of each other. Hernandez *et al.* (2007) investigate inference by using a context-dependent mutation model and a mathematical model for correcting the site frequency spectrum to account for ancestral misidentification. Caicedo *et al.* (2007) discuss inference of parameters from the SFS through the use of a bottleneck model, a selective sweeps model, and a combination of a bottleneck model with either a migration model or a selective sweeps model. Nielsen *et al.* (2009) discuss inference methods which use statistics related to the two dimensional (or two population) site frequency spectrum; these statistics are used to analyze which type of selection occurs and its effects. Chen (2012) applies coalescent theory to derive the joint AFS analytically, extend the joint AFS to multiple populations, and include effects by relevant models such as migration. Robinson *et al.* (2014) look at the effect of sample size on model selection and on parameter inference based on the AFS in single population models and in two-population models.

Other methods include methods where an allele frequency value is randomly sampled at the root from some distribution and the allele frequency evolves along each branch according to a transition density such as in Sirén *et al.* (2011; 2013). The model in Sirén *et al.* (2011) also includes binomial sampling of allele counts at each leaf given the allele frequencies sampled at each leaf. The model we use is similar to the model in Sirén *et al.* (2011), except that we use a stationary distribution at the root which satisfies (4.2) and transition densities that also satisfy (4.2).

Transition densities that satisfy (4.2), or any governing differential equation for that matter, are not always explicitly available. Even if a transition density cannot be obtained directly from a differential equation, there are methods for obtaining a numerical solution to the equation. One such method is the computation of a spectral representation of the transition density of a diffusion process; this is described in Section 15.13 of Karlin and Taylor (1981). A spectral representation is a weighted sum of functions of the form $\exp(-\lambda_i t)\,\phi_i(x)$, where λ_i is an eigenvalue of the differential equation of the diffusion process and $\phi_i(x)$ is the associated eigenfunction of the differential equation (see page 331 of Karlin and Taylor (1981)). Song and Steinrücken (2012) give examples of spectral representations of solutions to (4.2), both of which use eigenfunctions based on Gegenbauer polynomials: one representation is for the case of genetic selection and no mutation, and the other representation is for the case of recurrent mutation with or without general diploid selection.

Another way to numerically approximate transition densities from differential equations is to numerically solve the differential equations using Chebyshev polynomials, which are what we mostly work with in this chapter. One such solution method is to integrate the differential equation and find the Chebyshev expansion of the transition density that best fits the integrated system (see pages 100-111 of Fox and Parker (1968)). Another method is to find the Chebyshev expansion that best fits the original differential equation (see pages 111-114 of Fox and Parker (1968)). We mostly work with the integration based method in this chapter.

4.2 Overview of the algorithm

A common strategy for the computation of likelihoods from a multidimensional integral is Monte Carlo integration. Unfortunately, Monte Carlo integration has a convergence rate of $O\left(N^{-\frac{1}{2}}\right)$ where N is the number of function evaluations. As an alternative, we consider numerical integration; however, we do not integrate over all dimensions simultaneously, due to the curse of dimensionality. Instead, we compute the integral by using dynamic programming to integrate over one dimension at a time and we store data in each dimension to avoid any potential recalculation.

We already explored one such approach in Chapter 3 (published in Hiscott *et al.* (2016)), a numerical integration algorithm where we compute each integral via quadrature using data at a finite number of points. At each node v apart from the root, we compute the partial likelihood $\mathcal{F}_v(x)$ of the data either at v (if v is a leaf) or at leaves descended from v conditional on the value x at p_v. Let $f(x_v|x_{p_v}, \theta_v)$ denote the transition density from x_{p_v} to x_v given parameters θ_v. $\mathcal{F}_v(x)$ therefore satisfies the following three recurrences.

- If v is a leaf,
$$\mathcal{F}_v(x) = \int f(x_v|x, \theta_v) P_v\left(D_v|x_v\right) \, \mathrm{d}x_v. \tag{4.19}$$
- If v is a non-root node with children L and R,
$$\mathcal{F}_v(x) = \int f(x_v|x, \theta_v) \mathcal{F}_L(x_v) \mathcal{F}_R(x_v) \, \mathrm{d}x_v. \tag{4.20}$$
- If v is the root with children L and R,
$$\mathcal{F}_v(x) = \int \pi(x_v) \mathcal{F}_L(x_v) \mathcal{F}_R(x_v) \, \mathrm{d}x_v. \tag{4.21}$$

These are Equations (3.15)—(3.17) in Chapter 3.

For each node v, let x_v denote the continuous variable at v and let

$$X_v[k] = a_v + k\frac{b_v - a_v}{N}$$

denote the values of x_v used in the quadrature, where N is the number of function evaluations at each node of the tree and where a_v and b_v are finite and depend on the model. Let $F_v[k]$ denote the approximation of $\mathcal{F}_v\left(X_{p_v}[k]\right)$, where p_v is the parental node of v. The algorithm in Chapter 3 computes $F_v[k]$ so that it satisfies the following recurrences for quadrature weights w_{vi} at node v and for $i \in \{0, 1, ..., N\}$:

- If v is a leaf,

$$F_v[k] = \sum_{i=0}^{N} w_{vi} f\left(X_v[i] | X_{p_v}[k], \theta_v\right) P_v\left(D_v | X_v[i]\right). \tag{4.22}$$

- If v is a non-root node with children L and R,

$$F_v[k] = \sum_{i=0}^{N} w_{vi} f\left(X_v[i] | X_{p_v}[k], \theta_v\right) F_L[i] F_R[i]. \tag{4.23}$$

- If v is the root with children L and R,

$$F_v[k] = \sum_{i=0}^{N} w_{vi} \pi(X_v[i]) F_L[i] F_R[i]. \tag{4.24}$$

These are numerically integrated approximations of Equations (3.15)—(3.17) in Chapter 3. If we let n denote the number of leaves, the runtime is $O\left(nN^2\right)$; far more efficient than if we integrate over all dimensions simultaneously. The algorithm has an asymptotic error rate equal to the error rate of whichever numerical integration technique is used at each node. As an example, Simpson's rule is applied to each node in the cases discussed in Chapter 3, so the entire algorithm has an error rate of $O\left(N^{-4}\right)$, as a function of N. We can therefore obtain faster convergence than with Monte Carlo integration while avoiding any blowup in computation time due to multiple dimensions. In addition, the algorithm is reasonably simple to apply to any likelihood where the transition densities are known.

Despite its merits, however, the numerical integration method in Chapter 3 has limitations that hinder its applicability to some likelihood computations. First of all, the method assumes that the transition densities are explicitly available, and it is not

always possible to obtain explicit expressions of transition densities. Second of all, even with the transition densities known, the numerical integration can generate large errors due to the use of a finite number of integration points. As pointed out in Section 3.5.2 of Chapter 3, a transition density becomes highly peaked along short branches, and this leads to large integration errors when the Simpson's rule is used in one dimension, let alone multiple dimensions. In this chapter, we introduce a method which generalizes the method in Chapter 3, is more accurate, and to which numerical solutions to PDEs, including (4.2), can be applied directly.

For this method, we define an integral

$$\mathcal{G}_v(\phi) = \int \phi(y)\mathcal{F}_v(y)\omega^*(y)\,\mathrm{d}y \tag{4.25}$$

for an arbitrary function ϕ and a weight function ω^*. Note that this integral is a generalization of the approximations $\mathcal{F}_v(x)$ computed in Chapter 3, which we obtain if we substitute $\delta(y-x)$ in place of ϕ and let $\omega^*(y)=1$.

In practice, we select a set of basis functions $\phi_1, \phi_2, \ldots$ and define

$$G_v[k] = \int \phi_k(y)\mathcal{F}_v(y)\omega^*(y)\,\mathrm{d}y. \tag{4.26}$$

These integrals generalize the approximations $F_v[k]$ computed in Chapter 3, which we obtain if we use $\delta(y - X_{p_v}[k])$ in place of ϕ and let $\omega^*(y)=1$. The method we discuss in this chapter utilizes shifted Chebyshev polynomials $T_k^*(x)$, $k \in \{0,1,2,...\}$. Chebyshev polynomials are a family of orthogonal polynomials commonly used for function interpolation; an example of literature describing the Chebyshev polynomials is Mason and Handscomb (2003). For the remainder of this chapter, we define the integrals

$$G_v[k] := \mathcal{G}_v(T_k^*) \tag{4.27}$$

and define $\omega^*(x) = \frac{1}{\sqrt{x-x^2}}$, which is the weight function for shifted Chebyshev polynomials. We will derive the recurrences for $G_v[k]$ in the sections that follow.

4.2.1 Likelihood at a leaf

At each leaf v, let n_v denote the total number of individuals at v and let r_v denote the number of individuals with allele 1. We then evaluate

$$G_v[k] = \int_0^1 T_k^*(p)\mathcal{F}_v(p)\omega^*(p)\,\mathrm{d}p \tag{4.28}$$

$$= \int_0^1 T_k^*(p)\omega^*(p) \int_0^1 f(x;p,t_v)\binom{n_v}{r_v}x^{r_v}(1-x)^{n_v-r_v}\,\mathrm{d}x\,\mathrm{d}p, \tag{4.29}$$

where f is a solution to (4.2), t_v denotes the time between non-root node v and its parental node p_v. To find $G_v[k]$, we define

$$g_v(p,t) := \int_0^1 f(x;p,t)\binom{n_v}{r_v}x^{r_v}(1-x)^{n_v-r_v}\,\mathrm{d}x. \tag{4.30}$$

We then multiply both sides of (4.2) by $\binom{n_v}{r_v}x^{r_v}(1-x)^{n_v-r_v}$ and integrate with respect to x to obtain the PDE

$$\frac{\partial g_v(p,t)}{\partial t} = a(p)\frac{\partial g_v(p,t)}{\partial p} + \frac{1}{2}b(p)\frac{\partial^2 g_v(p,t)}{\partial p^2} \tag{4.31}$$

with initial condition

$$g_v(p,0) = \int_0^1 \delta(x-p)\binom{n_v}{r_v}x^{r_v}(1-x)^{n_v-r_v}\,\mathrm{d}x = \binom{n_v}{r_v}p^{r_v}(1-p)^{n_v-r_v} \tag{4.32}$$

which is based on the fact that $f(x;p,0) = \delta(x-p)$ (Kimura, 1955). Later in this chapter, we discuss how we solve the above PDE numerically using operators on weighted sums of Chebyshev polynomials. We obtain the values of

$$G_v[k] = \int_0^1 T_k^*(p)\omega^*(p)g_v(p,t_v)\,\mathrm{d}p \tag{4.33}$$

directly from the numerical solution to (4.31).

4.2.2 Likelihood along a branch

Let v be a node which is neither a leaf nor the root, and let L and R denote the child nodes of v. The integrals we evaluate for v are

$$G_v[k] = \int_0^1 T_k^*(p)\mathcal{F}_v(p)\omega^*(p)\,\mathrm{d}p \tag{4.34}$$

$$= \int_0^1 T_k^*(p)\omega^*(p)\int_0^1 f\left(x;p,t_v\right)\mathcal{F}_L(x)\mathcal{F}_R(x)\,\mathrm{d}x\,\mathrm{d}p. \tag{4.35}$$

Similar to the integrals at the leaves, we find $G_v[k]$ by finding

$$g_v(p,t) := \int_0^1 f(x;p,t)\mathcal{F}_L(x)\mathcal{F}_R(x)\,\mathrm{d}x. \tag{4.36}$$

This satisfies

$$\frac{\partial g_v(p,t)}{\partial t} = a(p)\frac{\partial g_v(p,t)}{\partial p} + \frac{1}{2}b(p)\frac{\partial^2 g_v(p,t)}{\partial p^2} \tag{4.37}$$

such that

$$g_v(p,0) = \mathcal{F}_L(p)\mathcal{F}_R(p). \tag{4.38}$$

We derive an expression for $g_v(p,0)$ in terms of $G_L[i]$ and $G_R[j]$ after a description of the relevant properties of shifted Chebyshev polynomials.

4.2.3 Likelihood at the root

Let r denote the root and let L and R denote the child nodes of r. At the root, we compute the likelihood of the entire tree given the entire dataset,

$$P(D|\theta) = \int_0^1 \pi\left(x\right) \mathcal{F}_L(x) \mathcal{F}_R(x)\, \mathrm{d}x, \tag{4.39}$$

where $\pi\left(x\right)$ is the stationary distribution, a distribution that satisfies the equation

$$-a(x)\pi\left(x\right) + \frac{1}{2}\frac{\mathrm{d}\left\{b(x)\pi\left(x\right)\right\}}{\mathrm{d}x} = 0, \tag{4.40}$$

see pages 145-146 of Ewens (2004). After we find $\pi(x)$, we find the numerical approximation of $P(D|\theta)$ through the use of different integration techniques; we describe this process later on in this chapter.

4.3 Chebyshev polynomials

4.3.1 Standard Chebyshev polynomials

Chebyshev polynomials are a family of polynomials defined on the interval $[-1, 1]$ and widely used in numerical analysis for function approximation and numerical integration (Mason and Handscomb, 2003). There are four kinds of Chebyshev polynomials (Mason and Handscomb, 2003); we focus on the first kind, the most widely used. These Chebyshev polynomials are defined recursively by

$$T_0(x) = 1; \tag{4.41}$$

$$T_1(x) = x; \tag{4.42}$$

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x),\ n \geq 2. \tag{4.43}$$

Equivalently, the Chebyshev polynomials are defined by the identity

$$T_n(x) = \cos\left(n \arccos(x)\right). \tag{4.44}$$

The Chebyshev polynomials form an orthogonal family. Let $\omega(x)$ denote the weight function

$$\omega(x) = \frac{1}{\sqrt{1-x^2}}. \tag{4.45}$$

Then, for all m and n, we have

$$\int_{-1}^{1} T_m(x)T_n(x)\omega(x)dx = \begin{cases} \pi & \text{if } m = n = 0; \\ \frac{\pi}{2} & \text{if } m = n > 0; \\ 0 & \text{otherwise.} \end{cases} \tag{4.46}$$

We will also make use of three identities which involve Chebyshev polynomials. The first of these identities is

$$xT_n(x) = \frac{1}{2}\left(T_{|n-1|}(x) + T_{n+1}(x)\right), \tag{4.47}$$

which follows directly from (4.43). The second identity is

$$T_m(x)T_n(x) = \frac{1}{2}\left(T_{|m-n|}(x) + T_{m+n}(x)\right), \tag{4.48}$$

which is a consequence of (4.44), as is the third identity

$$\int T_m(x)dx = \begin{cases} \frac{1}{2}\left(\frac{T_{m+1}(x)}{m+1} - \frac{T_{|m-1|}}{m-1}\right) + C_m & \text{if } m \neq 1; \\ \frac{1}{4}T_2(x) + C_1 & \text{if } m = 1, \end{cases} \tag{4.49}$$

where C_m is a constant.

4.3.2 Shifted Chebyshev polynomials

Since allele frequencies range between 0 and 1 and the Chebyshev polynomials discussed in the previous section are defined in the interval $[-1, 1]$, we apply a change of variable. Shifted Chebyshev polynomials of the first kind are defined by $T_n^*(x) = T_n(2x-1)$ for $x \in [0, 1]$; see Section 1.3.1 of Mason and Handscomb (2003) for more information. Shifted Chebyshev polynomials are also orthogonal, and satisfy

$$\int_0^1 T_m^*(x)T_n^*(x)\omega^*(x)dx = \begin{cases} \pi & \text{if } m = n = 0; \\ \frac{\pi}{2} & \text{if } m = n > 0; \\ 0 & \text{otherwise,} \end{cases} \tag{4.50}$$

where $\omega^*(x) = (x - x^2)^{-1/2}$.

Applying the change of variables to (4.43), the shifted Chebyshev polynomials satisfy the recurrence relation

$$T_n^*(x) = 2(2x-1)T_{n-1}^*(x) - T_{n-2}^*(x) \tag{4.51}$$

for $n \geq 2$, where $T_0^*(x) = 1$ and $T_1^*(x) = 2x - 1$.

From (4.43) and (4.51), we obtain the identities

$$xT_n^*(x) = \frac{1}{4}\left(T_{|n-1|}^*(x) + 2T_n^*(x) + T_{n+1}^*(x)\right) \tag{4.52}$$

and

$$(1-x)T_n^*(x) = \frac{1}{4}\left(2T_n^*(x) - T_{|n-1|}^*(x) - T_{n+1}^*(x)\right). \tag{4.53}$$

Applying the change of variables to (4.48), we have

$$T_m^*(x)T_n^*(x) = \frac{1}{2}\left(T_{|m-n|}^*(x) + T_{m+n}^*(x)\right) \tag{4.54}$$

From (4.49), we have the integral identity

$$\int T_m^*(x)dx = \begin{cases} \frac{1}{4}\left(\frac{T_{m+1}^*(x)}{m+1} - \frac{T_{|m-1|}^*(x)}{m-1}\right) + C_m^* & \text{if } m \neq 1; \\ \frac{1}{8}T_2^*(x) + C_1^* & \text{if } m = 1. \end{cases} \tag{4.55}$$

where C_m^* is a constant.

By Weierstrass's theorem, for any continuous function f on $[0,1]$ and for any $\epsilon > 0$, there exists a polynomial $p_n(x)$ of sufficiently large degree n such that $||f - p_n||_\infty < \epsilon$. This polynomial can therefore be expressed as a weighted sum of Chebyshev polynomials

$$p_n(x) = \sum_{i=0}^{n} b_i T_i^*(x) \tag{4.56}$$

where

$$b_n = \begin{cases} \frac{1}{\pi}\int_0^1 p_n(x)\omega^*(x)\,\mathrm{d}x & \text{if } n = 0; \\ \frac{2}{\pi}\int_0^1 p_n(x)T_n^*(x)\omega^*(x)\,\mathrm{d}x & \text{if } n > 0, \end{cases} \tag{4.57}$$

so that every continuous function on $[0,1]$ can be approximated arbitrarily closely using shifted Chebyshev polynomials. In addition, if f is infinitely differentiable (or, by extension, analytic) on $[0,1]$, then for any $\rho > 1$, there exists a constant $K(\rho)$ such that

$$\left|\left| f(x) - \sum_{i=0}^{N} c_i T_i^*(x) \right|\right|_\infty = K(\rho)\rho^{-N}, \tag{4.58}$$

where

$$c_n = \begin{cases} \frac{1}{\pi}\int_0^1 f(x)\omega^*(x)\,\mathrm{d}x & \text{if } n = 0; \\ \frac{2}{\pi}\int_0^1 f(x)T_n^*(x)\omega^*(x)\,\mathrm{d}x & \text{if } n > 0. \end{cases} \tag{4.59}$$

4.3.3 Further properties of the shifted Chebyshev polynomials

Here we establish several additional properties of shifted Chebyshev polynomials, all of which are important for our algorithm. For completeness, we provide proofs of these results; for similar results, see Baszenski and Tasche (1997).

Proposition 1. *Suppose that* $g(x) = \sum_{i=0}^{\infty} a_i T_i^*(x)$, $h(x) = \sum_{i=0}^{\infty} b_i T_i^*(x)$, *and* $f(x) = \sum_{i=0}^{\infty} c_i T_i^*(x)$, *where*

$$c_i = \begin{cases} a_0 b_0 + \frac{1}{2}\sum_{j=1}^{\infty} a_j b_j & \text{if } i = 0; \\ \frac{1}{2}\sum_{j=0}^{\infty}\left(a_j b_{|i-j|} + a_{|i-j|}b_j + a_j b_{i+j} + a_{i+j}b_j\right) & \text{if } i > 0. \end{cases} \tag{4.60}$$

Then $f(x) = g(x)h(x)$ *for all* $x \in [0, 1]$.

Proof

If we multiply $g(x)$ and $h(x)$ together, we can express the product using (4.54) as follows

$$\begin{aligned}
g(x)h(x) &= \left(\sum_{i=0}^{\infty} a_i T_i^*(x)\right)\left(\sum_{j=0}^{\infty} b_j T_j^*(x)\right) \\
&= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} a_i b_j T_i^*(x) T_j^*(x) \\
&= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} \frac{a_i b_j}{2}\left(T_{|i-j|}^*(x) + T_{i+j}^*(x)\right) \\
&= a_0 b_0 T_0^*(x) + \sum_{i=1}^{\infty} \frac{a_i b_i}{2}\left(T_0^*(x) + T_{2i}^*(x)\right) \\
&\quad + \sum_{i=0}^{\infty}\left(\sum_{j\neq i} \frac{a_i b_j}{2}\left(T_{|i-j|}^*(x) + T_{i+j}^*(x)\right)\right) \\
&= \left(a_0 b_0 + \frac{1}{2}\sum_{i=1}^{\infty} a_i b_i\right) T_0^*(x) \\
&\quad + \frac{1}{2}\sum_{i=1}^{\infty}\left(\sum_{j=0}^{\infty}\left(a_j b_{|i-j|} + a_{|i-j|} b_j + a_j b_{i+j} + a_{i+j} b_j\right)\right) T_i^*(x)
\end{aligned}$$

□

Proposition 2. *For all* i, j, *and* k,

$$\int_0^1 T_i^*(x) T_j^*(x) T_k^*(x) \omega^*(x)\,\mathrm{d}x = \begin{cases} \pi & \text{if } i = j = k = 0; \\ \frac{\pi}{4}\left(\delta_{i,|j-k|} + \delta_{i,j+k}\right) & \text{otherwise} \end{cases} \tag{4.61}$$

where

$$\delta_{a,b} = \begin{cases} 1 & \text{if } a = b; \\ 0 & \text{otherwise.} \end{cases}$$

Proof

From (4.54), we get

$$\begin{aligned}
\int_0^1 T_i^*(x) T_j^*(x) T_k^*(x) \omega^*(x)\,\mathrm{d}x &= \frac{1}{2}\int_0^1 T_i^*(x)\left(T_{|j-k|}^*(x) + T_{j+k}^*(x)\right)\omega^*(x)\,\mathrm{d}x \\
&= \frac{1}{2}\int_0^1 \left(T_i^*(x) T_{|j-k|}^*(x) + T_i^*(x) T_{j+k}^*(x)\right)\omega^*(x)\,\mathrm{d}x
\end{aligned}$$

From (4.46), we then get

$$\int_0^1 T_i^*(x)T_{|j-k|}^*(x)\omega^*(x)dx = \begin{cases} \pi & \text{if } i = |j-k| = 0; \\ \frac{\pi}{2}\delta_{i,|j-k|} & \text{otherwise.} \end{cases}$$

$$\int_0^1 T_i^*(x)T_{j+k}^*(x)\omega^*(x)dx = \begin{cases} \pi & \text{if } i = j+k = 0; \\ \frac{\pi}{2}\delta_{i,j+k} & \text{otherwise.} \end{cases}$$

Therefore,

$$\int_0^1 T_i^*(x)T_j^*(x)T_k^*(x)\omega^*(x)\,\mathrm{d}x = \begin{cases} \pi & \text{if } i = j = k = 0; \\ \frac{\pi}{4}\left(\delta_{i,|j-k|} + \delta_{i,j+k}\right) & \text{otherwise} \end{cases}$$

□

Since binomial sampling is used in the tree model from which we calculate likelihoods, we express the binomial density in terms of Chebyshev polynomials.

Proposition 3. *For $n \in \mathbb{N}$ and $n \geq m \in \mathbb{N}$,*

$$\binom{n}{m} x^m (1-x)^{n-m} = \sum_{i=0}^{n} \sigma_{m,n,i} T_i^*(x) \tag{4.62}$$

where

$$\sigma_{m,n,i} = \begin{cases} \frac{1}{n+1}\binom{n}{m} \sum_{j=1}^{n+1} x_j^m \left(1-x_j\right)^{n-m} & \text{if } i = 0; \\ \frac{2}{n+1}\binom{n}{m} \sum_{j=1}^{n+1} x_j^m \left(1-x_j\right)^{n-m} T_i^*\left(x_j\right) & \text{if } i > 0. \end{cases} \tag{4.63}$$

and where

$$x_j = \frac{1}{2}\left(1 + \cos\left(\frac{\left(j-\frac{1}{2}\right)\pi}{n+1}\right)\right) \tag{4.64}$$

Proof

Since x^m and $(1-x)^{n-m}$ are polynomials of degrees m and $n-m$, respectively, (4.62) is a polynomial of degree n. If we let $y = 2x - 1$, then we can express (4.62) as

$$\binom{n}{m}\left(\frac{y+1}{2}\right)^m \left(\frac{1-y}{2}\right)^{n-m} \tag{4.65}$$

Since (4.65) is still a polynomial of degree n and it is defined on $[-1, 1]$, it can be expressed exactly as a sum of Chebyshev polynomials of degree no larger than n

$$\sum_{i=0}^{n} \sigma_{m,n,i} T_i(y)$$

where

$$\sigma_{m,n,i} = \begin{cases} \frac{1}{n+1}\binom{n}{m}\sum_{j=1}^{n+1}\left(\frac{y_j+1}{2}\right)^m\left(\frac{1-y_j}{2}\right)^{n-m} & \text{if } i=0; \\ \frac{2}{n+1}\binom{n}{m}\sum_{j=1}^{n+1}\left(\frac{y_j+1}{2}\right)^m\left(\frac{1-y_j}{2}\right)^{n-m} T_i\left(y_j\right) & \text{if } i>0. \end{cases} \tag{4.66}$$

and where

$$y_j = \cos\left(\frac{\left(j-\frac{1}{2}\right)\pi}{n+1}\right) \tag{4.67}$$

based on pages 147 and 151 of Mason and Handscomb (2003).

If we express y in terms of x, we get

$$\sigma_{m,n,i} = \begin{cases} \frac{1}{n+1}\binom{n}{m}\sum_{j=1}^{n+1} x_j^m\left(1-x_j\right)^{n-m} & \text{if } i=0; \\ \frac{2}{n+1}\binom{n}{m}\sum_{j=1}^{n+1} x_j^m\left(1-x_j\right)^{n-m} T_i^*\left(x_j\right) & \text{if } i>0. \end{cases}$$

where

$$x_j = \frac{1}{2}\left(1+\cos\left(\frac{\left(j-\frac{1}{2}\right)\pi}{n+1}\right)\right)$$

□

Note that, in practice, it will be more efficient, and numerically stable, to use standard Chebyshev interpolation algorithms when determining the coefficients $\sigma_{m,n,i}$; see Chapter 8 of Trefethen (2000).

4.3.4 Solving transition densities using Chebyshev bases

At several steps in the algorithm, we wish to find the coefficients of the Chebyshev expansion of the integral

$$g(p,t) = \int_0^1 f(x;p,t)F(x)\,\mathrm{d}x$$

where $F(x)$ is some function for which we already have the coefficients for the Chebyshev expansion and $f(x;p,t)$ is the solution to the PDE

$$\frac{\partial f(x;p,t)}{\partial t} = \mathcal{L}f(x;p,t) \tag{4.68}$$

with the initial condition $f(x;p,0) = \delta(x-p)$ and where

$$\mathcal{L}f(x;p,t) = a(p)\frac{\partial f(x;p,t)}{\partial p} + \frac{1}{2}b(p)\frac{\partial^2 f(x;p,t)}{\partial p^2}.$$

Here, we write $a(p)$ and $b(p)$ as

$$a(p) = \sum_{i=0}^{n} a_i p^i \tag{4.69}$$

$$b(p) = \sum_{i=0}^{n} b_i p^i. \tag{4.70}$$

For $\mathbf{c} = \{c_i | i \in \{0, 1, 2, \ldots\}\}$ such that

$$\sum_{i=0}^{\infty} c_i T_i^*(x) = f(x; p, t)$$

and $\mathbf{d} = \{d_i | i \in \{0, 1, 2, \ldots\}\}$ such that

$$\sum_{i=0}^{\infty} d_i T_i^*(x) = \mathcal{L} f(x; p, t),$$

let $\mathcal{M}$ be an operator such that $\mathcal{M}\mathbf{c} = \mathbf{d}$ and let $(\mathcal{M}\mathbf{c})_i = d_i$.

Proposition 4. *If* $F(x) = \sum_{i=0}^{\infty} \alpha_i T_i^*(x)$ *and*

$$g(p, t) = \int_0^1 f(x; p, t) F(x) \, \mathrm{d}x,$$

then, for all $t \geq 0$,

$$g(p, t) = \sum_{i=0}^{\infty} c_i(t) T_i^*(p)$$

where $\mathbf{c}(t) = \{c_i(t) | i \in \{0, 1, 2, \ldots\}\}$ *is the solution to the ODE*

$$\frac{d}{dt}\mathbf{c}(t) = M\mathbf{c}(t) \tag{4.71}$$

where $c_i(0) = \alpha_i$ *for all* $i \in \{0, 1, 2, \ldots\}$.

Proof
If we multiply $\frac{\partial f(x;p,t)}{\partial t}$ by $F(x)$ and integrate over the interval $x \in [0, 1]$, we obtain

$$\int_0^1 F(x) \frac{\partial f(x; p, t)}{\partial t} \, \mathrm{d}x = \int_0^1 F(x) \mathcal{L} f(x; p, t) \, \mathrm{d}x \tag{4.72}$$

$$\frac{\partial}{\partial t} \left(\int_0^1 F(x) f(x; p, t) \, \mathrm{d}x \right) = \mathcal{L} \left(\int_0^1 F(x) f(x; p, t) \, \mathrm{d}x \right) \tag{4.73}$$

$$\frac{\partial}{\partial t} \left(g(p, t) \right) = \mathcal{L} \left(g(p, t) \right) \tag{4.74}$$

$$\frac{\partial}{\partial t} \left(\sum_{i=0}^{\infty} c_i(t) T_i^*(p) \right) = \mathcal{L} \left(\sum_{i=0}^{\infty} c_i(t) T_i^*(p) \right) \tag{4.75}$$

$$\sum_{i=0}^{\infty} \left(\frac{\partial}{\partial t} c_i(t) \right) T_i^*(p) = \sum_{i=0}^{\infty} \left(\mathcal{M}\mathbf{c}(t) \right)_i T_i^*(p). \tag{4.76}$$

Therefore, $\frac{\partial}{\partial t} c_i(t) = \left(\mathcal{M}\mathbf{c}(t) \right)_i$ and we obtain the differential equation

$$\frac{\partial}{\partial t}\mathbf{c}(t) = \mathcal{M}\mathbf{c}(t).$$

Similarly, if we multiply $f(x;p,0)$ by $F(x)$ and integrate over the interval $x \in [0,1]$, we obtain

$$\int_0^1 F(x)f(x;p,0)\,\mathrm{d}x = \int_0^1 F(x)\delta(x-p)\,\mathrm{d}x \tag{4.77}$$

$$g(p,0) = F(p) \tag{4.78}$$

$$\sum_{i=0}^{\infty} c_i(0)T_i^*(p) = \sum_{i=0}^{\infty} \alpha_i T_i^*(p), \tag{4.79}$$

which gives us the initial condition on $\mathbf{c}(t)$, which is $c_i(0) = \alpha_i$ for all i.

□

4.4 Approximation of transition densities

Due to the infinite sums in the Chebyshev expansions, the algorithm we have described cannot be applied in practice, so instead, we will use a finite number of Chebyshev polynomials. We partition the operator $\mathcal{M}$ as

$$\mathcal{M} = \begin{bmatrix} \mathcal{M}_N & \mathcal{U} \\ \mathcal{V} & \mathcal{W} \end{bmatrix}, \tag{4.80}$$

where $\mathcal{M}_N$ acts upon vectors of length $N+1$ (or coefficients of degree N Chebyshev expansions) and outputs vectors of length $N+1$. We then approximate the likelihood $P(D|\theta)$ in (4.18) by the following algorithm:

1. At each leaf $v \in \{1,\ldots,k\}$:

   (a) For $i \in \{0,\ldots,N\}$, compute coefficient $\tilde{c}_i(0)$ of initial condition vector $\tilde{\mathbf{c}}(0)$ as

   $$\tilde{c}_i(0) = \begin{cases} 0 & \text{if } i > N; \\ \sigma_{n_v,r_v,i} & \text{otherwise.} \end{cases}$$

   (b) Compute $\tilde{\mathbf{c}}(t_v)$ based on

   $$\frac{\partial}{\partial t}\tilde{\mathbf{c}}(t) = \mathcal{M}_N\tilde{\mathbf{c}}(t).$$

   (c) For $i \in \{0,\ldots,N\}$, set

   $$G_v[i] = \begin{cases} \pi\tilde{c}_0(t_v) & \text{if } i = 0; \\ \frac{\pi}{2}\tilde{c}_i(t_v) & \text{if } i > 0. \end{cases} \tag{4.81}$$

2. At each internal node $v \in \{k+1, \ldots, 2k-2\}$:

   (a) Let L and R be the two children of node v.

   (b) Set $l_0 = \frac{1}{\pi} G_L[0]$ and $r_0 = \frac{1}{\pi} G_R[0]$.

   (c) For $i \in \{1, \ldots, N\}$, set $l_i = \frac{2}{\pi} G_L[i]$ and $r_i = \frac{2}{\pi} G_R[i]$.

   (d) For $i \in \{0, \ldots, N\}$, compute coefficient $\tilde{c}_i(0)$ of initial condition vector $\tilde{\mathbf{c}}(0)$ as

$$\tilde{c}_i(0) = \begin{cases} \tilde{l}_0 \tilde{r}_0 + \frac{1}{2} \sum_{j=1}^{\infty} \tilde{l}_j \tilde{r}_j & \text{if } i = 0; \\ \frac{1}{2} \sum_{j=0}^{\infty} \left(\tilde{l}_j \tilde{r}_{|i-j|} + \tilde{l}_{|i-j|} \tilde{r}_j + \tilde{l}_j \tilde{r}_{i+j} + \tilde{l}_{i+j} \tilde{r}_j \right) & \text{otherwise.} \end{cases}$$

   (e) Compute $\tilde{\mathbf{c}}(t_v)$ based on

$$\frac{\partial}{\partial t} \tilde{\mathbf{c}}(t) = \mathcal{M}_N \tilde{\mathbf{c}}(t).$$

   (f) For $i \in \{0, \ldots, N\}$, set

$$G_v[i] = \begin{cases} \pi \tilde{c}_0(t_v) & \text{if } i = 0; \\ \frac{\pi}{2} \tilde{c}_i(t_v) & \text{if } i > 0. \end{cases} \tag{4.82}$$

3. Let L and R be the two children of the root.

4. Set $l_0 = \frac{1}{\pi} G_L[0]$ and $r_0 = \frac{1}{\pi} G_R[0]$.

5. For $i \in \{1, \ldots, N\}$, set $l_i = \frac{2}{\pi} G_L[i]$ and $r_i = \frac{2}{\pi} G_R[i]$.

6. For $i \in \{0, \ldots, N\}$, set

$$c_i = \begin{cases} \tilde{l}_0 \tilde{r}_0 + \frac{1}{2} \sum_{j=1}^{\infty} \tilde{l}_j \tilde{r}_j & \text{if } i = 0; \\ \frac{1}{2} \sum_{j=0}^{\infty} \left(\tilde{l}_j \tilde{r}_{|i-j|} + \tilde{l}_{|i-j|} \tilde{r}_j + \tilde{l}_j \tilde{r}_{i+j} + \tilde{l}_{i+j} \tilde{r}_j \right) & \text{otherwise.} \end{cases}$$

7. Compute approximate likelihood

$$\begin{aligned} \tilde{P}(D|\theta) &= \int_0^1 \pi(x) \left(\sum_{i=0}^{N} c_i T_i^*(x) \right) \mathrm{d}x \\ &= \sum_{i=0}^{N} c_i \left(\int_0^1 \pi(x) T_i^*(x) \, \mathrm{d}x \right) \end{aligned} \tag{4.83}$$

If either $\beta_1 < 1$ or $\beta_2 < 1$, then $\pi(x)$ is not bounded above (see page 175 of Ewens (2004)), so there would not exist a Chebyshev expansion of $\pi(x)$. However, as long as β_1 and β_2 are both greater than 0, $\pi(x)$ is still integrable and we can use applicable numerical integration techniques to compute each integral in (4.83).

4.4.1 Truncation error

To determine the order of the truncation error induced by using a finite number of basis functions, we first find the error by truncating $\mathbf{c}(t)$, where $\mathbf{c}(t)$ is a solution to

$$\frac{\partial}{\partial t}\mathbf{c}(t) = \mathcal{M}\mathbf{c}(t)$$

and that $\mathbf{c}(0) = \alpha$. We partition $\mathbf{c}(t)$ as

$$\mathbf{c}(t) = \begin{bmatrix} \tilde{\mathbf{c}}(t) \\ \epsilon(t) \end{bmatrix},$$

where $\tilde{\mathbf{c}}(t)$ consists of the $N+1$ terms of $\mathbf{c}(t)$. Since in our case, $\mathbf{c}(0)$ and $\mathbf{c}(t)$ are analytic, $||\epsilon(t)|| = K(\rho)\rho^{-N}$ for any $\rho > 1$, for some $K(\rho) > 0$, and for bounded t.

From our partition of $\mathbf{c}(0)$, we obtain the differential equation

$$\frac{\partial}{\partial t}\tilde{\mathbf{c}}(t) = \mathcal{M}_N\tilde{\mathbf{c}}(t) + \mathcal{U}\epsilon(t) \tag{4.84}$$

with initial condition $\tilde{\mathbf{c}}(0) = \tilde{\alpha}$. If we let vector $\hat{\mathbf{c}}(t)$ denote a solution to the equation

$$\frac{\partial}{\partial t}\hat{\mathbf{c}}(t) = \mathcal{M}_N\hat{\mathbf{c}}(t)$$

with initial condition $\hat{\mathbf{c}}(0) = \tilde{\mathbf{c}}(0)$.

If we let $\upsilon(t) = \tilde{\mathbf{c}}(t) - \hat{\mathbf{c}}(t)$, then we obtain the differential equation

$$\frac{\partial}{\partial t}\upsilon(t) = \mathcal{M}_N\tilde{\mathbf{c}}(t) + \mathcal{U}\epsilon(t) - \mathcal{M}_N\hat{\mathbf{c}}(t) = \mathcal{M}_N\upsilon(t) + \mathcal{U}\epsilon(t) \tag{4.85}$$

with initial condition $\upsilon(0) = 0$. We solve this equation by using the integrating factor method to obtain

$$\upsilon(t) = \exp(M_N t)\int_0^t \exp(-M_N\tau)\,\mathcal{U}\epsilon(\tau)\,\mathrm{d}\tau$$

where M_N is a matrix such that $\mathcal{M}_N\tilde{\mathbf{c}}(t) = M_N\tilde{\mathbf{c}}(t)$. Using the logarithmic norm μ, the matrix norms $||\exp(M_N t)||$ and $||\exp(-M_N t)||$ are bounded above by $\exp(\mu(M_N)\,t)$ and $\exp(\mu(-M_N)\,t)$, respectively. If we assume that $\mu(M_N) > 0$ and $\mu(-M_N) > 0$, then we can bound $||\upsilon(t)||$ above by

$$t\exp(\mu(M_N)\,t)\exp(\mu(-M_N)\,t)\,K\rho^{-N} = tK\left(\frac{\exp\left((\mu(M_N) + \mu(-M_N))\,\frac{t}{N}\right)}{\rho}\right)^N.$$

If we replace ρ with a function $r(N)$ such that $r(N) > 1$ for all N and such that

$$\frac{\exp\left((\mu(M_N) + \mu(-M_N))\,\frac{t}{N}\right)}{r(N)} < 1,$$

we obtain a bound on $||\upsilon(t)||$ that decreases exponentially, indicating spectral convergence. Even if either $\mu(M_N) \leq 0$ or $\mu(-M_N) \leq 0$, we can still find a bound which decreases exponentially.

4.5 Efficient solution of differential equations

In order to find the Chebyshev expansion of $g\left(x_{p_v}, t_v\right)$ for each non-root node v, we solve (4.2) through the use of operators on coefficient vectors.

Suppose that $f(x) = \sum_{i=0}^{\infty} c_i T_i^*(x)$ and $\mathbf{c} = (c_0, c_1, ...)$. Let $\mathcal{A}$ be the *integration operator* defined by

$$\int f(x)dx = \int \sum_{i=0}^{\infty} c_i T_i^*(x)dx = \sum_{i=0}^{\infty} (\mathcal{A}\mathbf{c})_i T_i^*(x).$$

Similarly, let $\mathcal{B}$ denote the operator which multiplies a function by x, so

$$xf(x) = \sum_{i=0}^{\infty} c_i x T_i^*(x) = \sum_{i=0}^{\infty} (\mathcal{B}\mathbf{c})_i T_i^*(x).$$

Both of the operators $\mathcal{A}$ and $\mathcal{B}$ have simple expressions and both have bandwidth one. Here, $\mathcal{A}_{ij} = (\mathcal{A}e_j)_i$ and and $\mathcal{B}_{ij} = (\mathcal{B}e_j)_i$, where e_j is the vector with 1 in position j and 0 elsewhere.

Proposition 5. *Operators $\mathcal{A}$ and $\mathcal{B}$ satisfy the relations*

$$\mathcal{A}_{ij} = \begin{cases} \frac{1}{2} & \text{if } i = 0 \text{ and } j = 1; \\ \frac{1}{8} & \text{if } i = 1 \text{ and } j = 2; \\ \frac{1}{4(i+1)} & \text{if } i > 1 \text{ and } j = i + 1; \\ -\frac{1}{4(i-1)} & \text{if } i > 1 \text{ and } j = i - 1; \\ 0 & \text{otherwise;} \end{cases} \tag{4.86}$$

and

$$\mathcal{B}_{ij} = \begin{cases} \frac{1}{2} & \text{if } i = j \text{ or if } i = 0 \text{ and } j = 1; \\ \frac{1}{4} & \text{if } i > 0 \ j = i \pm 1; \\ 0 & \text{otherwise.} \end{cases} \tag{4.87}$$

Proof

Since

$$\int T_i^*(x)\,\mathrm{d}x = \sum_{j=0}^{\infty} (\mathcal{A}e_j)_i T_j^*(x) = \sum_{j=0}^{\infty} \mathcal{A}_{ij} T_j^*(x),$$

we obtain the relation (4.86) from (4.52). Similarly, since

$$xT_i^*(x) = \sum_j (\mathcal{B}e_j)_i T_j^*(x) = \sum_j \mathcal{B}_{ij} T_j^*(x)$$

we obtain the relation (4.87) from (4.55). □

While the differentiation operator is dense with respect to the Chebyshev basis (see Section 2.4.5 of Mason and Handscomb (2003) for an expression), we can avoid having to apply it directly by applying a technique appearing in Fox and Parker (1968).

Theorem 2. *Let $\mathcal{M}$ be the operator defined by*

$$\mathcal{L}f = \mathcal{L}\left(\sum_i c_i T_i^*(p)\right) = \sum_i (\mathcal{M}\mathbf{c})_i T_i^*(p), \tag{4.88}$$

where

$$\mathcal{L}f = a(p)\frac{\partial f}{\partial p} + \frac{1}{2}b(p)\frac{\partial^2 f}{\partial p^2}$$

and $a(p), b(p)$ are polynomials of degree n. Let

$$\begin{aligned}\mathcal{C} = &\left(\frac{1}{2}\sum_{i=0}^{n} b_i \mathcal{B}^i\right) + \mathcal{A}\ \left(\sum_{i=0}^{n}\left(a_i - (i+1)b_{i+1}\right)\mathcal{B}^i\right) \\ &+ \mathcal{A}^2\ \left(\sum_{i=0}^{n-1}\left(\frac{1}{2}(i+1)(i+2)b_{i+2} - (i+1)a_{i+1}\right)\mathcal{B}^i\right)\end{aligned} \tag{4.89}$$

where a_i and b_i satisfy the relations

$$a(p) = \sum_{i=0}^{n} a_i p^i \tag{4.90}$$

$$b(p) = \sum_{i=0}^{n} b_i p^i. \tag{4.91}$$

Then,

$$\mathcal{A}^2\mathcal{M} = \mathcal{C}.$$

Proof

We can solve (4.88) by integrating it twice to obtain

$$\begin{aligned}\iint \mathcal{L}f \,\mathrm{d}p^2 &= \iint \left(\sum_i (\mathcal{M}\mathbf{c})_i T_i^*(p)\right) \mathrm{d}p^2 \\ &= \int \left(\sum_{i=0}^{\infty} \left(\mathcal{A}\mathcal{M}\mathbf{c}\right)_i T_i^*(p)\right) \mathrm{d}p \\ &= \sum_{i=0}^{\infty} \left(\mathcal{A}^2\mathcal{M}\mathbf{c}\right)_i T_i^*(p)\end{aligned}$$

and by applying a method described on pages 100-101 of Fox and Parker (1968) to obtain

$$\begin{aligned}
\iint \mathcal{L}f \,\mathrm{d}p^2 =& \iint \left(a(p)\frac{\partial f}{\partial p} + \frac{1}{2}b(p)\frac{\partial^2 f}{\partial p^2} \right) \mathrm{d}p^2 \\
=& b(p)f + \int (a(p) - b'(p))f \,\mathrm{d}p + \iint \left(\frac{1}{2}b''(p) - a'(p) \right) f \,\mathrm{d}p^2 \\
=& \frac{1}{2}b(p) \left(\sum_i c_i T_i^*(p) \right) + \int (a(p) - b'(p)) \left(\sum_i c_i T_i^*(p) \right) \mathrm{d}p \\
&+ \iint \left(\frac{1}{2}b''(p) - a'(p) \right) \left(\sum_i c_i T_i^*(p) \right) \mathrm{d}p^2 \\
=& \frac{1}{2} \sum_{j=0}^{n} \sum_i \left(\mathcal{B}^j b_j \mathbf{c} \right)_i T_i^*(p) + \sum_{j=0}^{n} \sum_{i=1}^{\infty} \left(\mathcal{A}\mathcal{B}^j \left(a_j - (j+1)b_{j+1} \right) \mathbf{c} \right)_i T_i^*(p) \\
&+ \sum_{j=0}^{n-1} \sum_{i=2}^{\infty} \left(\mathcal{A}^2 \mathcal{B}^j \left((j+1)(j+2)b_{j+2} - (j+1)a_{j+1} \right) \mathbf{c} \right)_i T_i^*(p).
\end{aligned}$$

Expressing both of the above relations in terms of the coefficients, we get

$$\begin{aligned}
\mathcal{A}^2\mathcal{M}\mathbf{c} =& \left(\frac{1}{2} \sum_{j=0}^{n} b_j \mathcal{B}^j \mathbf{c} \right) + \mathcal{A} \left(\sum_{j=0}^{n} \left(a_j - (j+1)b_{j+1} \right) \mathcal{B}^j \mathbf{c} \right) \\
&+ \mathcal{A}^2 \left(\sum_{j=0}^{n-1} \left(\frac{1}{2}(j+1)(j+2)b_{j+2} - (j+1)a_{j+1} \right) \mathcal{B}^j \mathbf{c} \right) \\
=& \left(\frac{1}{2} \sum_{j=0}^{n} b_j \mathcal{B}^j + \mathcal{A} \sum_{j=0}^{n} \left(a_j - (j+1)b_{j+1} \right) \mathcal{B}^j \right. \\
&\left. + \mathcal{A}^2 \sum_{j=0}^{n-1} \left(\frac{1}{2}(j+1)(j+2)b_{j+2} - (j+1)a_{j+1} \right) \mathcal{B}^j \right) \mathbf{c} \\
=& \mathcal{C}\mathbf{c}
\end{aligned}$$

□

We note that the operators $\mathcal{A}^2$ and $\mathcal{C}$ in Theorem 3 have bandwidths 2 and $n+1$, respectively; these bandwidths enable rapid approximation of transition densities.

In order to find $\mathcal{M}$, we first take into account the fact that the double integration could lead to extra constant and linear terms. We determine the values of i such that either $(\mathcal{C}e_i)_0 \neq 0$ or $(\mathcal{C}e_i)_1 \neq 0$, then directly find $\mathcal{M}e_i$ for these values of i. We then find the remaining $\mathcal{M}e_i$ by solving $\mathcal{A}^2\mathcal{M}\mathbf{v} = \mathcal{C}\mathbf{v}$ for all $\mathbf{v}$ such that $(\mathcal{C}\mathbf{v})_0 = 0$ and $(\mathcal{C}\mathbf{v})_1 = 0$.

For each internal node v, in order to find the approximation $\tilde{\mathbf{c}}_v$ which is an approximate solution to the initial value problem described by

$$\frac{\partial}{\partial t}\mathbf{c}_v(t) = \mathcal{M}\mathbf{c}_v$$
$$\mathbf{c}_v(0) = \mathbf{d}_v$$

for some coefficient vector $\mathbf{d}_v$, we use the fourth-order Runge-Kutta Method, or RK4, which has a global error rate of $O\left(\triangle t_v^4\right)$ (see pages 106 and 107 of Gockenbach (2002)). We truncate both $\mathbf{c}$ and $\frac{\partial}{\partial t}\mathbf{c}$ to $N+1$ terms. Let M denote the number of time steps, let $\triangle t_v = \frac{t_v}{M}$ denote the length of each time step, and let $\tilde{\mathbf{c}}^i$ denote the approximation of $\mathbf{c}^i$ such that

$$\mathbf{F}_v\left(x_{p_v}, \triangle t_v i\right) = \sum_{j=0}^{\infty} \mathbf{c}_j^i T_j^*(x)$$

Then, by the Runge-Kutta method,

$$\tilde{\mathbf{c}}^{i+1} = \tilde{\mathbf{c}}^i + \frac{\triangle t_v}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$

where

$$\begin{aligned}
k_1 =& \mathcal{M}\tilde{\mathbf{c}}^i \\
k_2 =& \mathcal{M}\left(\tilde{\mathbf{c}}^i + \frac{\triangle t_v}{2}k_1\right) \\
k_3 =& \mathcal{M}\left(\tilde{\mathbf{c}}^i + \frac{\triangle t_v}{2}k_2\right) \\
k_4 =& \mathcal{M}\left(\tilde{\mathbf{c}}^i + \triangle t_v k_3\right).
\end{aligned}$$

If M Runge-Kutta time steps are used at each node below the root of an m-leaf tree, then the total Runge-Kutta induced error is $O\left(m\triangle t_{max}^4\right)$, where t_{max} is the largest branch length on the tree.

4.6 Experimental assessment

For our experiments, we simulate allele frequency changing along the branches and, at each leaf, we draw the number of allele 1 individuals at the leaf from a binomial distribution based on the simulated allele frequency at the leaf. For the simulations, we first sample an allele frequency at the root by randomly drawing a frequency from the stationary distribution π at the root. We do this by sampling a uniform random number y, then finding allele frequency x such that $\int_0^x \pi(z)\,\mathrm{d}z = y$.

Once we have our initial frequency x, we simulate diffusion processes along the branches by first writing (4.2) as a stochastic differential equation

$$\mathrm{d}X_t = a\left(X_t\right)\,\mathrm{d}t + \sqrt{b\left(X_t\right)}\,\mathrm{d}B_t \tag{4.92}$$

where X_t is the allele frequency drawn at time t and B_t is a Wiener process (see Itô (2004)). We choose integer M to be the number of time steps per branch and set $\mathrm{d}t_v = \frac{T_v}{M}$, where t_v is the time along the branch between node v and its parent node p_v, and where T_v is the total time between v and p_v. Along each branch, we simulate the stochastic differential equation using the Euler-Maruyama scheme, which is described in Saito and Mitsui (1993).

Assume that the nodes are visited in an order such that the root is visited first and such that each remaining node is not visited before its parent node. We use the following algorithm to sample the allele counts:

1. At the root, draw an allele frequency x from stationary distribution $\pi(x)$.

2. At each remaining node v, simulate the change in allele frequency between p_v and v via the Euler-Maruyama scheme. Record the allele frequency at node v as x_v.

3. At each leaf l, draw m_l from a binomial distribution $B\left(n_l, x_l\right)$.

After simulating data, we run two computational experiments. For the first experiment, we compute the approximate likelihood from each distribution at multiple values of M (number of time steps per node) and a fixed value of N (number of Chebyshev polynomials per node). For the second experiment, we fix M and compute the approximate likelihood from each distribution for multiple values of N.

4.6.1 Protocol

We run experiments on two different trees: a balanced binary 4-leaf tree and a balanced binary 16-leaf tree. In each tree, all branches are the same length and we use a parameter γ for the branch length. For an example of a tree with one of the values of γ we use, see Figure 4.1. In each experiment, all leaves have the same number of individuals, for which we use a parameter n_l.

For our simulations and experiments, we use the following parameter values:

- scaled mutation rates $\beta_1 = \beta_2 = 0.001$

- fitness parameters:
  - $\alpha \in \{0, 0.1\}$
  - $h = \{0.5, 0.1, 0.9\}$ (Note: the value of h is redundant if $\alpha = 0$)
- $\gamma \in \{0.01, 0.02\}$
- $n_l = 40$.

In the experiments in which M is fixed, we fix M at 100 and we use

$$N \in \{2, 4, \ldots, 98, 100\}.$$

In the experiments in which N is fixed, we fix N at either 50 or 100 and use

$$M \in \{2, 4, \ldots, 98, 100\}.$$

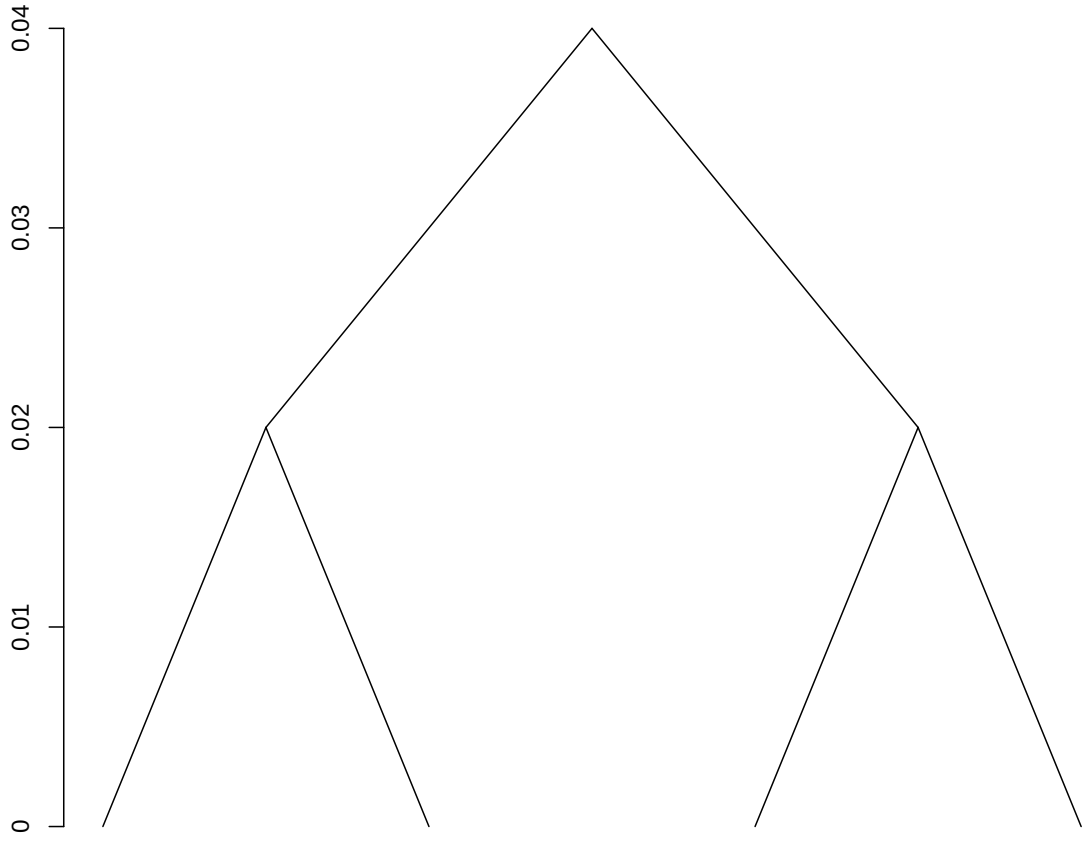

Figure 4.1: A balanced binary 4-leaf tree where $\gamma = 0.02$.

4.7 Results

The results are summarized in Figures 4.2 through 4.7 and in Tables 4.1 through 4.4. In Figure 4.2, we plot the logarithm of the absolute error of the approximate likelihood computed from the different simulated distributions (each represented by a different line) on the 4-leaf tree computed at different values of N with M is fixed at 100 and for two different values of γ. In Figure 4.3, we plot the logarithm of the absolute error computed from the distributions on the 16-leaf tree for the same values of σ and γ used for the computations for Figure 4.2; in both figures, the horizontal axis is N. In Figure 4.4, we compare the error dynamics for different sets of selection parameters on the 16-leaf tree. In Figure 4.5, we plot the logarithm of the error of the approximations of the likelihoods on the 4-leaf tree at different values of M where N is fixed at either 50 or 100 and $\gamma \in \{0.01, 0.02\}$; in these plots the horizontal axis is $\log(M)$. In Figure 4.6, we plot the errors of the approximations of the likelihoods on the 16-leaf under the same parameters as in Figure 4.5. In Figure 4.7, we compare the error dynamics on 16-leaf tree using the same selection parameters as in Figure 4.4 and fixing N at 50. If an approximation is either greater than 1 or no greater than 0, we exclude its error from the plots.

If the algorithm converges exponentially as N increases (assuming that M is fixed and sufficiently large or that integrals are evaluated exactly), then the logarithm of error should decrease linearly or faster. This is because, based in Section 4.4.1, the truncation error ϵ_N at N of the entire likelihood is $K(\rho)\rho^{-N}$ for $\rho > 1$ and some constant $K(\rho)$. Therefore,

$$\log \epsilon_N = \log(K(\rho)) - N log \rho.$$

For each distribution analyzed in the experiments with fixed M, since the actual likelihood is unavailable, we compute the error of an approximation as the absolute difference between the approximation itself and the approximation when $N = 100$. Our choice of N is due to the fact that, as Figures 4.5 through 4.7 show, if either M is too small or N is too large, the error could be considerably large.

The plots in Figures 4.2 through 4.4 show error plots that in general decrease faster than linearly. In all plots, there are two intervals of N in which there are different rates of decrease in error. It is unclear why there is a discontinuity in the rate of decrease, but this might be related to the fact that there are 40 individuals at each leaf or related to the error of the Runge-Kutta method being affected by the number of Chebyshev polynomials. In both intervals, the rate of decrease gradually increases as N increases.

These plots, therefore, imply that the approximation converges exponentially as N increases, as long as N is not too large. The maximum branch length, the number of leaves, and the selection parameters all appear to have no significant effect on the convergence rate.

If the approximate likelihood converges quartically as M increases while N is fixed, then the slope of the logarithm of the error against $\log(M)$ should be roughly -4 or no greater than -4. As before, the actual likelihood is unavailable, so we compute the error of an approximation at $M = a$ as the absolute difference between the approximation at $M = a$ and the approximation when $M = 100$. In each plot in Figures 4.5 through 4.7, we include a dashed line of slope -4 to compare the error plots to. As the error plots in Figures 4.5 through 4.7 in general have a slope that is roughly -4, the errors decrease quartically. Although Figure 4.7 shows that the selection parameters have no significant effect on the convergence rate, Figures 4.5 and 4.6 show that the number of leaves, the value of γ, and the value of N can effect the convergence rate. On either tree, the approximations appear to require larger M values to converge as either N or γ increases. The discontinuity in rate of decrease in error could be due to a build-up of error in the iterations of the Runge-Kutta method because of the truncation error of $\mathcal{M}$ and the Runge-Kutta induced error $O\left(m\left(\frac{\gamma}{M}\right)^4\right)$ where $m \in \{4, 16\}$. This could also explain why increasing the number of leaves also appears to increase the M value required for convergence.

Let $N(e)$ denote the minimum value of N such that for any distribution D in the sample, for likelihood approximation $P_{(N,M)}(D|\theta)$, and with M fixed at 100,

$$\left|P_{(N,100)}(D|\theta) - P_{(100,100)}(D|\theta)\right| < e.$$

In Table 4.1, we present values of $N(10^{-8})$ and $N(10^{-10})$ we obtained from the experiments where M is fixed at 100. While increasing γ somehow leads to a decrease in $N(10^{-8})$, it leads to an increase in $N(10^{-10})$. If the number of leaves increases, both $N(10^{-8})$ and $N(10^{-10})$ increase as well. The selection parameters have little, if any, effect on $N(10^{-8})$ or $N(10^{-10})$.

In Table 4.2, we present values of $M_N(e)$, where $M_N(e)$ is the minimum value of M such that for any distribution D in the sample and for fixed N,

$$\left|P_{(N,M)}(D|\theta) - P_{(N,100)}(D|\theta)\right| < e.$$

In particular, we present values of $M_N(10^{-8})$, $M_N(10^{-10})$, and $M_N(10^{-12})$ obtained from our experiments with fixed N. As Table 4.2 shows, increasing the number of

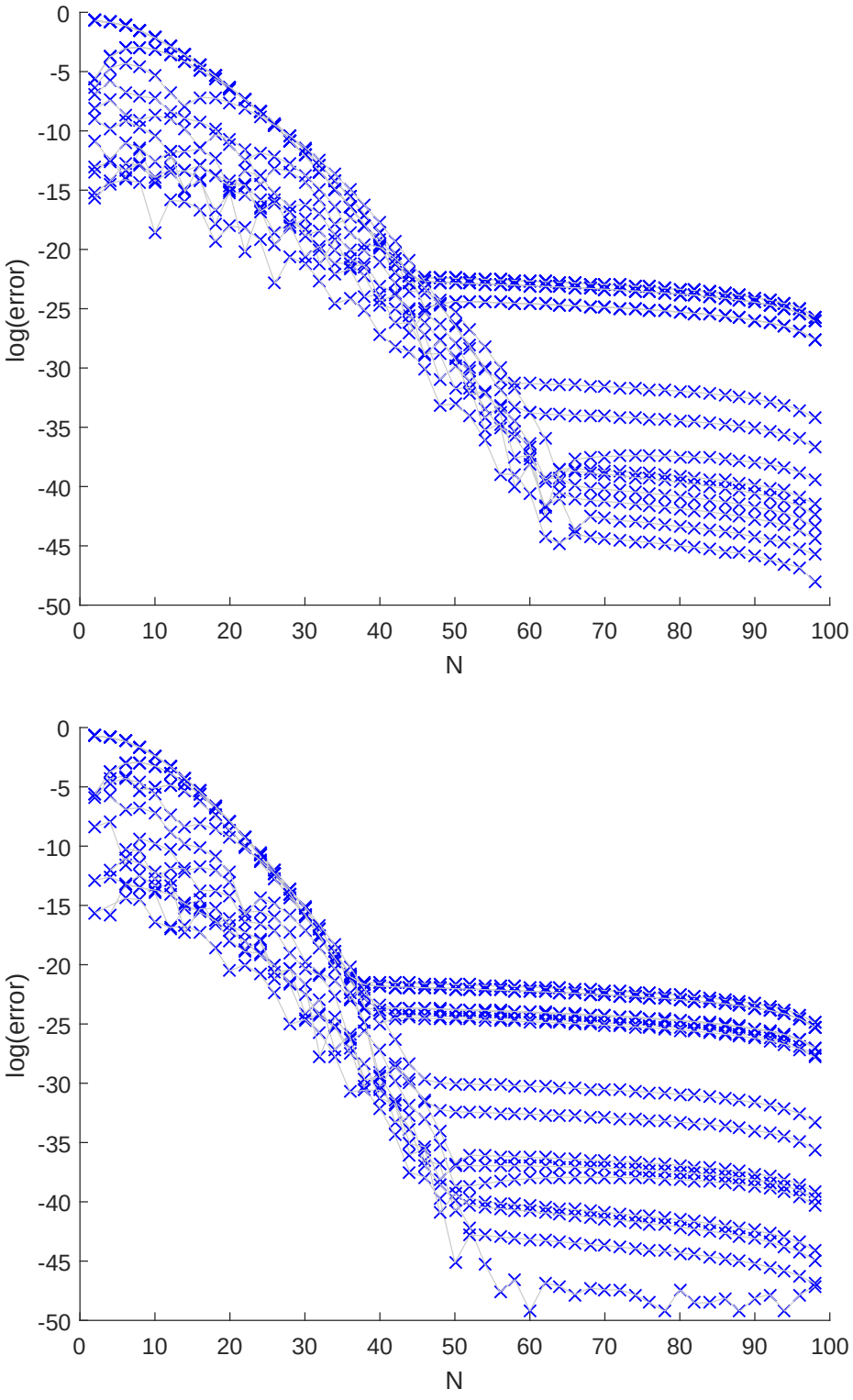

Figure 4.2: Convergence results for the 4-leaf tree when M is fixed at 100, $\beta_1 = \beta_2 = 0.001$, $\alpha = 0$, and $\gamma = 0.01$ (top) or 0.02 (bottom).

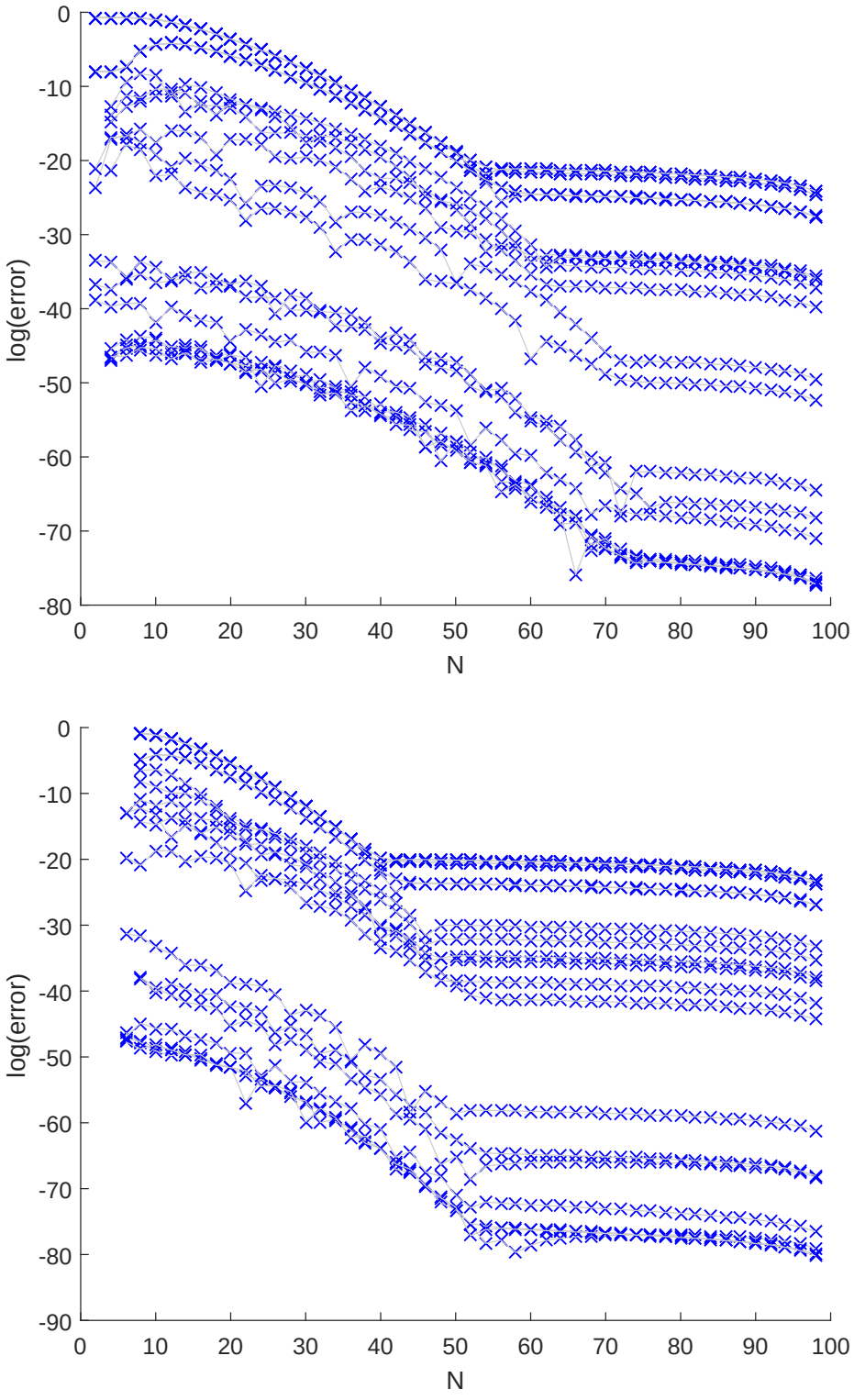

Figure 4.3: Convergence results for the 16-leaf tree when M is fixed at 100, $\beta_1 = \beta_2 = 0.001$, $\alpha = 0$, and $\gamma = 0.01$ (top) or 0.02 (bottom).

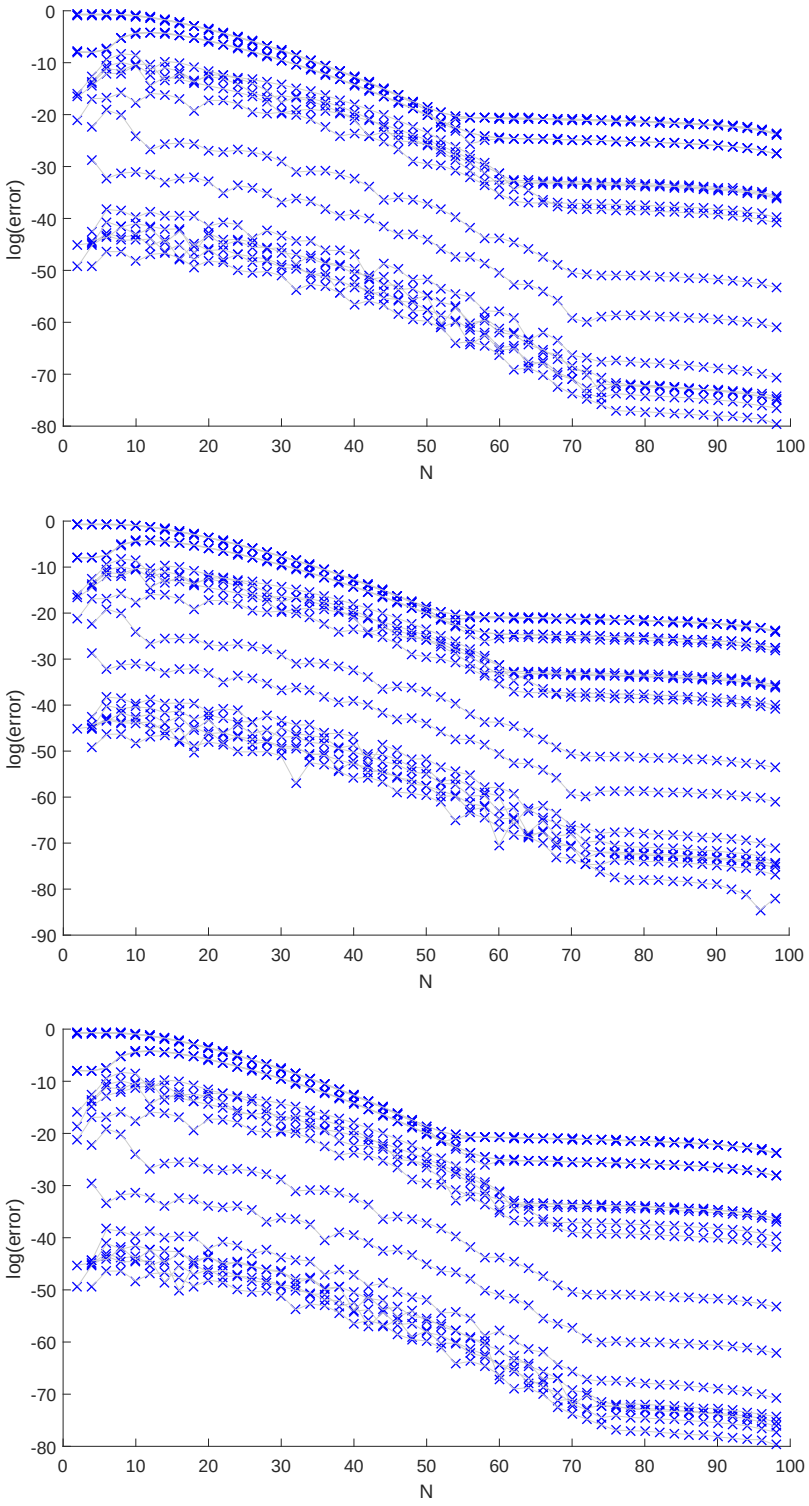

Figure 4.4: Convergence results for the 16-leaf tree when M is fixed at 100, $\beta_1 = \beta_2 = 0.001$, $\gamma = 0.01$, and $\alpha = 0.1$. The top plot consists of results from when $h = 0.5$, the middle plot consists of results from when $h = 0.1$, and the bottom plot consists of results from when $h = 0.9$.

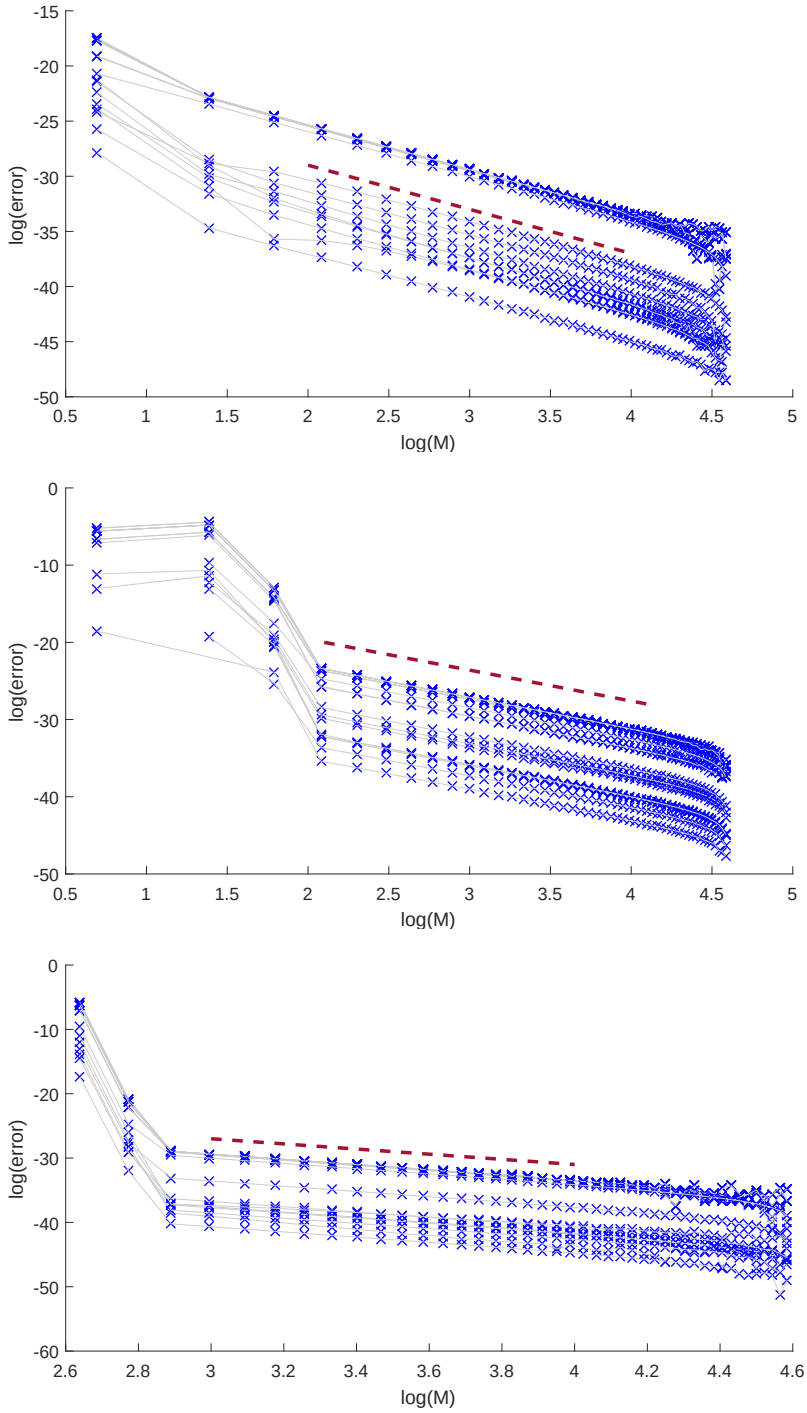

Figure 4.5: Convergence results for the 4-leaf tree when N is fixed at 50 (top and middle plots) or 100 (bottom plot), $\beta_1 = \beta_2 = 0.001$, $\alpha = 0$, and $\gamma = 0.01$ (top and bottom plots) or 0.02 (middle plot).

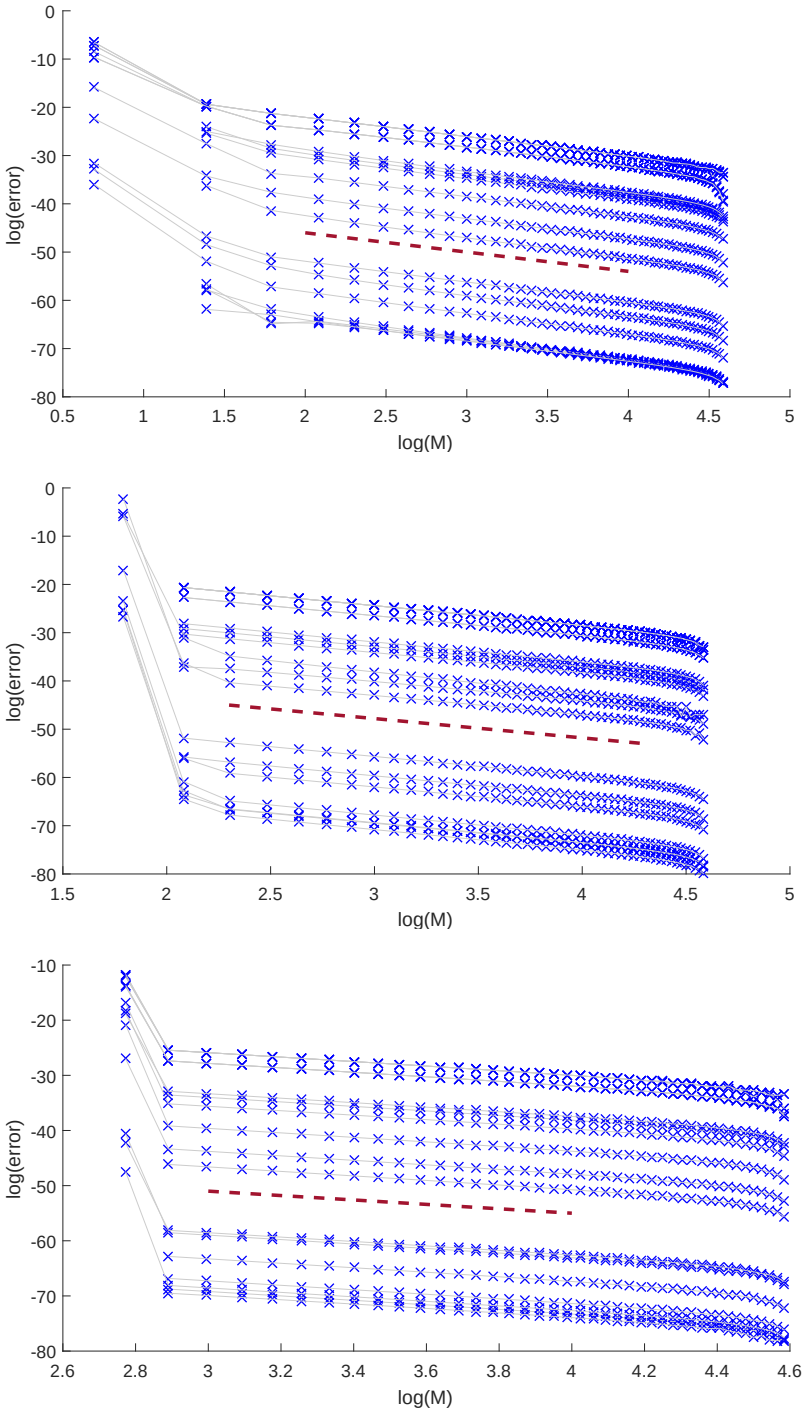

Figure 4.6: Convergence results for the 16-leaf tree when N is fixed at 50 (top and middle plots) or 100 (bottom plot), $\beta_1 = \beta_2 = 0.001$, $\alpha = 0$, and $\gamma = 0.01$ (top and bottom plots) or 0.02 (middle plot).

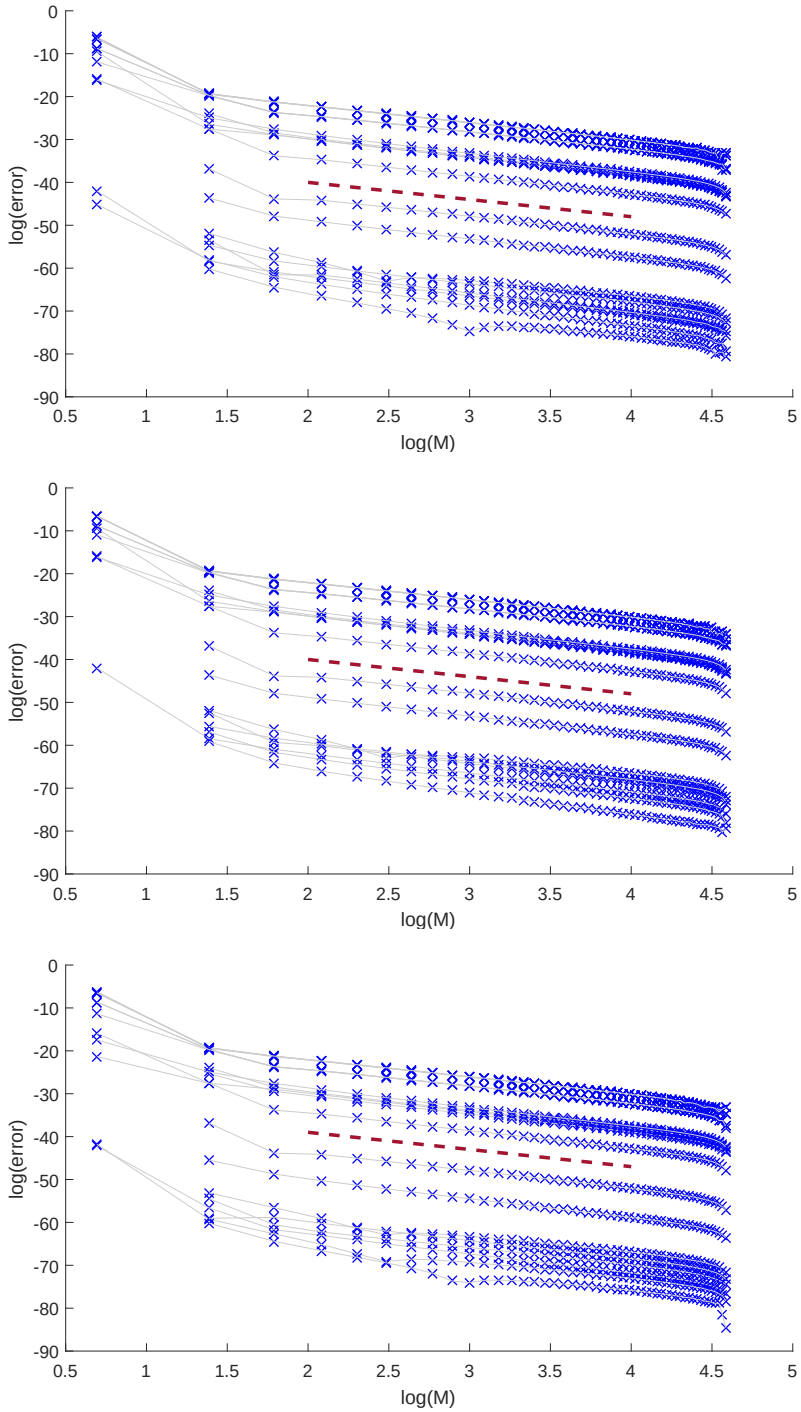

Figure 4.7: Convergence results for the 16-leaf tree when N is fixed at 50, $\beta_1 = \beta_2 = 0.001$, $\gamma = 0.01$, and $\alpha = 0.1$. The results in the plots come from when $h = 0.5$ (top plot), 0.1 (middle plot), or 0.9 (bottom plot).

Leaves	α	h	γ	$N(10^{-8})$	$N(10^{-10})$
4	0	n/a	0.01	42	74
4	0	n/a	0.02	36	82
16	0	n/a	0.01	50	94
16	0	n/a	0.02	38	98
16	0.1	0.5	0.01	50	98
16	0.1	0.1	0.01	50	96
16	0.1	0.9	0.01	50	96

Table 4.1: Values of $N(10^{-8})$ and $N(10^{-10})$ obtained from the experiments with M fixed at 100.

leaves, N, or γ increases, $M_N(10^{-8})$, $M_N(10^{-10})$, and $M_N(10^{-12})$ all increase. As with our values of $N(10^{-8})$ and $N(10^{-10})$, changing the section parameters appears to have little effect on $M_N(e)$.

Leaves	α	h	γ	N	$M_N(10^{-8})$	$M_N(10^{-10})$	$M_N(10^{-12})$
4	0	n/a	0.01	50	4	6	14
4	0	n/a	0.02	50	8	8	24
4	0	n/a	0.01	100	16	18	18
16	0	n/a	0.01	50	4	10	30
16	0	n/a	0.02	50	8	16	46
16	0	n/a	0.01	100	18	18	32
16	0.1	0.5	0.01	50	4	10	30
16	0.1	0.1	0.01	50	4	10	30
16	0.1	0.9	0.01	50	4	10	30

Table 4.2: Values of $M_N(10^{-8})$, $M_N(10^{-10})$, and $M_N(10^{-12})$ obtained from the experiments with N fixed at either 50 or 100.

Earlier in this section, we point out that we exclude any errors associated with approximations that are greater than 1 or no greater than 0. We denote R as the average number of excluded errors per distribution in an experiment. We present values of R from our experiments with fixed M in Table 4.3 and present values of R from our experiments with fixed N in Table 4.4. As the tables show, the selection parameters

have little effect on R and increasing γ, N (in the case of fixed N), or the number of leaves increases R. Figures 4.2 through 4.4 show that, in the experiments with fixed M, the excluded errors are mostly associated with small values of N. Similarly, excluded errors in the fixed N experiments are mostly associated with small values of M, as shown in Figures 4.5 through 4.7.

Leaves	α	h	γ	R
4	0	n/a	0.01	0
4	0	n/a	0.02	0.65
16	0	n/a	0.01	0.4
16	0	n/a	0.02	2.6
16	0.1	0.5	0.01	0.4
16	0.1	0.1	0.01	0.45
16	0.1	0.9	0.01	0.4

Table 4.3: Values of R obtained from the experiments with fixed M.

Leaves	α	h	γ	N	R
4	0	n/a	0.01	50	0
4	0	n/a	0.02	50	0.65
4	0	n/a	0.01	100	6.1
16	0	n/a	0.01	50	0.4
16	0	n/a	0.02	50	2.6
16	0	n/a	0.01	100	7.25
16	0.1	0.5	0.01	50	0.4
16	0.1	0.1	0.01	50	0.45
16	0.1	0.9	0.01	50	0.4

Table 4.4: Values of R obtained from the experiments with fixed N.

Based on our experiments, the value of M is more critical than the value of N. First of all, the convergence rate of the error induced by the Runge-Kutta integrations is dependent on both the maximum branch length of the tree and M. Second of all, the value of M affects how large N can be without the Runge-Kutta integrations causing significant errors.

4.8 Discussion

By incorporating dynamic programming, Chebyshev expansions of transition densities, and RK4, we achieve quartic convergence as M increases while N is fixed. If M is fixed and sufficiently large, then we achieve spectral convergence as N increases; however, how large N can be without causing significant mathematical errors is dependent on both M and maximum branch length. Neither convergence rate is significantly affected by the selection parameters or by the number of leaves on the tree. However, when M is fixed, the value of N required for the desired level of accuracy is dependent on maximum branch length and on the number of leaves. In addition, the value of M required for the desired level of accuracy is dependent on maximum branch length, on the number of leaves, and on N.

Despite the algorithm's convergence and the fact that the explicit forms of the transition densities are not necessary for the computations, the algorithm is not applicable to every diffusion process. As stated in Section 4.1, we assume that both mutation rates are greater than 0. However, there are diffusion processes in which there is no mutation whatsoever (see Sections 5.2 through 5.4 of Ewens (2004)) and processes where there is mutation in only one direction (see Section 5.5 of Ewens (2004)). To compute likelihoods based on these processes, one would have to consider a root distribution other than the stationary distribution.

Chapter 5

Future directions

With our work in Chapter 2, we were the first to introduce the method of delayed acceptance sampling to computational biology. Unfortunately, finding a working implementation of delayed acceptance turned out to be more difficult than we at first thought. If the approximation of the likelihood is poor, then the algorithm is more likely to reject samples that would be accepted by standard MCMC, resulting in poor mixing. Much of our work is dealing with the pernicious problem of acquisition error. However, even with the problem fixed in theory, mixing problems with the additional N variable made the approach ineffective. We have yet to fully understand how this may be best addressed. We began a search for alternative approximations, including those based on gene frequency models. This search, in turn, led us to explore efficient techniques for computing likelihoods under gene frequency models.

We plan to extend the numerical integration approach in Chapter 3 to a more general computational approach. The next step is to implement the approach for a range of models, thereby establishing it as a standard tool for computational biologists. The numerical integration approach is still not as automated as we would like; it can take some work to get a quadrature method working well on a model. Therefore, it would be good to find more robust algorithms; this, in part, led to the more general basis functions methods in Chapter 4.

Regarding the basis function approach, the next step is for us to greatly expand upon the experimental assessment of the approach. This includes using trees with varying branch lengths, using larger trees, and using empirical data as opposed to simulated data. In addition, we will look into alternative numerical solutions to the diffusion equations, such as those based on distribution theory. Our current intention, however, is to publish a robust and fully tested method as soon as we can.

Even though the work presented in this thesis was performed with analyses in evolutionary biology in mind, we believe that the work presented in this chapter could be extended to other biological contexts. The algorithms described in this thesis are applied to cases with binary trees, but it is possible that they could be extended to other diagrams, including ones not necessarily in biological contexts. In any case, the methods in this thesis have the potential to be used for further advancements in analytical software or algorithms.

References

Akaike, H. (1974). A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, *19*(6), 716–723.

Baszenski, G. and Tasche, M. (1997). Fast polynomial multiplication and convolutions related to the discrete cosine transform. *Linear Algebra and its Applications*, *252*(1-3), 1–25.

Bryant, D., Bouckaert, R., Felsenstein, J., Rosenberg, N. A., and RoyChoudhury, A. (2012). Inferring species trees directly from biallelic genetic markers: bypassing gene trees in a full coalescent analysis. *Molecular Biology and Evolution*, *29*(8), 1917–1932.

Bustamante, C. D., Wakeley, J., Sawyer, S., and Hartl, D. L. (2001). Directional Selection and the Site-Frequency Spectrum. *Genetics*, *159*(4), 1779–1788.

Caicedo, A. L., Williamson, S. H., Hernandez, R. D., Boyko, A., Fledel-Alon, A., York, T. L., Polato, N. R., Olsen, K. M., Nielsen, R., McCouch, S. R., Bustamante, C. D., and Purugganan, M. D. (2007). Genome-Wide Patterns of Nucleotide Polymorphism in Domesticated Rice. *PLoS Genet*, *3*(9), 1–12.

Cavalli-Sforza, L. L. and Edwards, A. W. (1967). Phylogenetic analysis. Models and estimation procedures. *American Journal of Human Genetics*, *19*(3 Pt 1), 233–257.

Chen, H. (2012). The joint allele frequency spectrum of multiple populations: A coalescent theory approach. *Theoretical Population Biology*, *81*(2), 179 – 195.

Cheney, E. and Kincaid, D. (2012). *Numerical Mathematics and Computing*. Boston, MA: Cengage Learning.

Christen, J. A. and Fox, C. (2005). Markov chain Monte Carlo Using an Approximation. *Journal of Computational and Graphical Statistics*, *14*(4), 795–810.

Dahlquist, G. and Björck, Å. (2008). Numerical Integration. In *Numerical Methods in Scientific Computing*, Volume 1, Chapter 5. Philadelphia, PA: SIAM.

Davis, P. J. and Rabinowitz, P. (1984). *Methods of numerical integration*. Orlando, FL: Academic Press.

Degnan, J. H. and Rosenberg, N. A. (2009). Gene tree discordance, phylogenetic inference and the multispecies coalescent. *Trends in Ecology & Evolution*, *24*(6), 332 – 340.

Drummond, A. J., Nicholls, G. K., Rodrigo, A. G., and Solomon, W. (2002). Estimating mutation parameters, population history and genealogy simultaneously from temporally spaced sequence data. *Genetics*, *161*(3), 1307–1320.

Ewens, W. (2004). *Mathematical Population Genetics 1: Theoretical Introduction*. Interdisciplinary Applied Mathematics. Springer New York.

Farris, J. S. (1972). Estimating Phylogenetic Trees from Distance Matrices. *The American Naturalist*, *106*(951), 645–668.

Felsenstein, J. (1968). *Statistical inference and the estimation of phylogenies*. Ph. D. thesis, Department of Zoology, University of Chicago, Chicago, IL.

Felsenstein, J. (1973). Maximum likelihood and minimum-steps methods for estimating evolutionary trees from data on discrete characters. *Systematic Zoology*, *22*(3), 240–249.

Felsenstein, J. (1981a). Evolutionary trees from DNA sequences: a maximum likelihood approach. *Journal of Molecular Evolution*, *17*(6), 368–376.

Felsenstein, J. (1981b). Evolutionary Trees From Gene Frequencies and Quantitative Characters: Finding Maximum Likelihood Estimates. *Evolution*, *35*(6), 1229–1242.

Felsenstein, J. (1988). Phylogenies and quantitative characters. *Annual Review of Ecology and Systematics*, *19*, 445–471.

Felsenstein, J. (2002). Quantitative characters, phylogenies, and morphometrics. In N. MacLeod (Ed.), *Morphology, shape and phylogeny*, 27–44. London: Taylor and Francis.

Felsenstein, J. (2004). *Inferring phylogenies*. Sunderland, Massachusetts: Sinauer Associates, Inc.

Felsenstein, J. (2005). Using the quantitative genetic threshold model for inferences between and within species. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *360*(1459), 1427–1434.

Felsenstein, J. (2009). Estimating effective population size from samples of sequences: inefficiency of pairwise and segregating sites as compared to phylogenetic estimates. *Genetical Research*, *59*(2), 139–147.

Felsenstein, J. (2012). A comparative method for both discrete and continuous characters using the threshold model. *The American Naturalist*, *179*(2), 145–156.

Felsenstein, J. and Churchill, G. A. (1996). A Hidden Markov Model approach to variation among sites in rate of evolution. *Molecular Biology and Evolution*, *13*(1), 93–104.

Fisher, R. (1930). *The Genetical Theory of Natural Selection*. Oxford: Oxford University Press.

Fisher, R. A. (1922). On the dominance ratio. *Proceedings of the Royal Society of Edinburgh*, *42*, 321–341.

FitzJohn, R. G. (2012). Diversitree: comparative phylogenetic analyses of diversification in R. *Methods in Ecology and Evolution*, *3*(6), 1084–1092.

Fox, C., Haario, H., and Christen, J. A. (2013). Inverse Problems. In P. Damien, P. Dellaportas, N. G. Polson, and D. A. Stephens (Eds.), *Bayesian Theory and Applications*, Chapter 31. Oxford: Oxford University Press.

Fox, L. and Parker, I. (1968). *Chebyshev polynomials in numerical analysis*. Oxford mathematical handbooks. Oxford: Oxford University Press.

Freckleton, R. P. (2012). Fast likelihood calculations for comparative analyses. *Methods in Ecology and Evolution*, *3*(5), 940–947.

Gelman, A. (2004). Parameterization and Bayesian Modeling. *Journal of the American Statistical Association*, *99*(466), 537–545.

Gilks, W. R., Richardson, S., and Spiegelhalter, D. J. (1996). Introducing markov chain monte carlo. In W. R. Gilks, S. Richardson, and D. J. Spiegelhalter (Eds.), *Markov Chain Monte Carlo in Practice*, Chapter 1. London: Chapman and Hall.

Gockenbach, M. (2002). *Partial Differential Equations: Analytical and Numerical Methods*. Number v. 1 in Partial Differential Equations: Analytical and Numerical Methods. Philadelphia: Society for Industrial and Applied Mathematics.

Gutenkunst, R. N., Hernandez, R. D., Williamson, S. H., and Bustamante, C. D. (2009). Inferring the Joint Demographic History of Multiple Populations from Multidimensional SNP Frequency Data. *PLoS Genet*, *5*(10), 1–11.

Hamilton, M. (2009). *Population Genetics*. UK: Wiley-Blackwell.

Hansen, T. F. (1997). Stabilizing selection and the comparative analysis of adaptation. *Evolution*, *51*(5), 1341–1351.

Harmon, L. J., Losos, J. B., Jonathan Davies, T., Gillespie, R. G., Gittleman, J. L., Bryan Jennings, W., Kozak, K. H., McPeek, M. A., Moreno-Roark, F., Near, T. J., *et al.* (2010). Early bursts of body size and shape evolution are rare in comparative data. *Evolution*, *64*(8), 2385–2396.

Heled, J. (2012). Sequence diversity under the multispecies coalescent with Yule process and constant population size. *Theoretical Population Biology*, *81*(2), 97 – 101.

Heled, J. and Drummond, A. J. (2010). Bayesian Inference of Species Trees from Multilocus Data. *Molecular Biology and Evolution*, *27*(3), 570–580.

Hernandez, R. D., Williamson, S. H., and Bustamante, C. D. (2007). Context Dependence, Ancestral Misidentification, and Spurious Signatures of Natural Selection. *Molecular Biology and Evolution*, *24*(8), 1792–1800.

Hiscott, G., Fox, C., Parry, M., and Bryant, D. (2016). Efficient Recycled Algorithms for Quantitative Trait Models on Phylogenies. *Genome Biology and Evolution*, *8*(5), 1338–1350.

Ho, L. and Ané, C. (2014). A linear-time algorithm for Gaussian and non-Gaussian trait evolution models. *Systematic Biology*, *63*(3), 397.

Itô, K. (2004). *Diffusion Processes*. John Wiley & Sons, Inc.

Karlin, S. and Taylor, H. (1981). *A Second Course in Stochastic Processes*. Elsevier Science.

Khaitovich, P., Enard, W., Lachmann, M., and Pääbo, S. (2006). Evolution of primate gene expression. *Nature Reviews Genetics*, *7*(9), 693–702.

Khaitovich, P., Pääbo, S., and Weiss, G. (2005). Toward a neutral evolutionary model of gene expression. *Genetics*, *170*(2), 929–939.

Kimura, M. (1955). Solution of a Process of Random Genetic Drift with a Continuous Model. *Proceedings of the National Academy of Sciences of the United States of America*, *41*(3), 144–150.

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

Kingman, J. F. C. (1982b). On the Genealogy of Large Populations. *Journal of Applied Probability*, *19*, 27–43.

Kloeden, P. and Platen, E. (1992). *Numerical Solution of Stochastic Differential Equations*. Applications of Mathematics: Stochastic Modelling and Applied Probability. Springer-Verlag Berlin Heidelberg.

Kuhner, M. K., Beerli, P., Yamato, J., and Felsenstein, J. (2000). Usefulness of single nucleotide polymorphism data for estimating population parameters. *Genetics*, *156*(1), 439–447.

Kutsukake, N. and Innan, H. (2013). Simulation-based Likelihood Approach for Evolutionary Models of Phenotypic Traits on Phylogeny. *Evolution*, *67*(2), 355–367.

Lande, R. (1976). Natural selection and random genetic drift in phenotypic evolution. *Evolution*, 314–334.

Landis, M. J., Schraiber, J. G., and Liang, M. (2013). Phylogenetic analysis using Lévy processes: finding jumps in the evolution of continuous traits. *Systematic Biology*, *62*(2), 193–204.

Larget, B. and Simon, D. (1998). Faster likelihood calculations on trees. Technical report.

Lemey, P., Rambaut, A., Welch, J. J., and Suchard, M. A. (2010). Phylogeography takes a relaxed random walk in continuous space and time. *Molecular Biology and Evolution*, *27*(8), 1877–1885.

Lepage, T., Lawi, S., Tupper, P., and Bryant, D. (2006). Continuous and tractable models for the variation of evolutionary rates. *Mathematical Biosciences*, *199*(2), 216–233.

Liu, J. (2013). *Monte Carlo Strategies in Scientific Computing*. Springer Series in Statistics. Springer New York.

Liu, J. S. and Chen, R. (1998). Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, *93*(443), 1032–1044.

Maddison, W. P. (1997). Gene Trees in Species Trees. *Systematic Biology*, *46*(3), 523–536.

Marazzi, B., Ané, C., Simon, M. F., Delgado-Salinas, A., Luckow, M., and Sanderson, M. J. (2012). Locating evolutionary precursors on a phylogenetic tree. *Evolution*, *66*(12), 3918–3930.

Mason, J. C. and Handscomb, D. C. (2003). Chebyshev polynomials. *Chapman & Hall*.

Meudt, H. M., Lockhart, P. J., and Bryant, D. (2009). Species delimitation and phylogeny of a New Zealand plant species radiation. *BMC Evolutionary Biology*, *9*(1), 111.

Nei, M. (1972). Genetic Distance between Populations. *The American Naturalist*, *106*(949), 283–292.

Nei, M. (1987). Genetic distance and molecular phylogeny. *Population Genetics and Fishery Management. Seattle. USA*, 193–223.

Nei, M. and Roychoudhury, A. (1974). Sampling variances of heterozygosity and genetic distance. *Genetics*, *76*(2), 379–390.

Nielsen, R., Hubisz, M. J., Hellmann, I., Torgerson, D., Andrés, A. M., Albrechtsen, A., Gutenkunst, R., Adams, M. D., Cargill, M., Boyko, A., *et al.* (2009). Darwinian and demographic forces affecting human protein coding genes. *Genome Research*, *19*(5), 838–849.

Nielsen, R., Mountain, J. L., Huelsenbeck, J. P., and Slatkin, M. (1998). Maximum-Likelihood Estimation of Population Divergence Times and Population Phylogeny in Models without Mutation. *Evolution*, *52*(3), 669–677.

Notohara, M. (1990). The coalescent and the genealogical process in geographically structured population. *Journal of Mathematical Biology*, *29*(1), 59–75.

O'Meara, B. C. (2012). Evolutionary inferences from phylogenies: a review of methods. *Annual Review of Ecology, Evolution, and Systematics*, *43*, 267–285.

Pamilo, P. and Nei, M. (1988). Relationships between gene trees and species trees. *Molecular Biology and Evolution*, *5*(5), 568–583.

Pond, S. L. K. and Muse, S. V. (2004). Column sorting: Rapid calculation of the phylogenetic likelihood function. *Systematic Biology*, *53*(5), 685–692.

Rannala, B. and Yang, Z. (2003). Bayes estimation of species divergence times and ancestral population sizes using DNA sequences from multiple loci. *Genetics*, *164*(4), 1645–1656.

Reynolds, J., Weir, B. S., and Cockerham, C. C. (1983). Estimation of the coancestry coefficient: Basis for a short-term genetic distance. *Genetics*, *105*(3), 767–779.

Robinson, J. D., Coffman, A. J., Hickerson, M. J., and Gutenkunst, R. N. (2014). Sampling strategies for frequency spectrum-based population genomic inference. *BMC Evolutionary Biology*, *14*(1), 254.

Rodriguez, F., Oliver, J., Marin, A., and Medina, J. R. (1990). The general stochastic model of nucleotide substitution. *Journal of theoretical biology*, *142*(4), 485–501.

Ronquist, F. (2004). Bayesian inference of character evolution. *Trends in Ecology & Evolution*, *19*(9), 475–481.

Rosenberg, N. A. and Nordborg, M. (2002). Genealogical trees, coalescent theory and the analysis of genetic polymorphisms. *Nature Reviews Genetics*, *3*(5), 380–390.

RoyChoudhury, A. and Thompson, E. A. (2012). Ascertainment correction for a population tree via a pruning algorithm for likelihood computation. *Theoretical Population Biology*, *82*(1), 59–65.

Saito, Y. and Mitsui, T. (1993). Simulation of stochastic differential equations. *Annals of the Institute of Statistical Mathematics*, *45*(3), 419–432.

Saitou, N. and Nei, M. (1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, *4*(4), 406–425.

Sawyer, S. A. and Hartl, D. L. (1992). Population genetics of polymorphism and divergence. *Genetics*, *132*(4), 1161–1176.

Simon, M. F., Grether, R., de Queiroz, L. P., Skema, C., Pennington, R. T., and Hughes, C. E. (2009). Recent assembly of the Cerrado, a neotropical plant diversity hotspot, by in situ evolution of adaptations to fire. *Proceedings of the National Academy of Sciences*, *106*(48), 20359–20364.

Sirén, J., Hanage, W. P., and Corander, J. (2013). Inference on Population Histories by Approximating Infinite Alleles Diffusion. *Molecular Biology and Evolution*, *30*(2), 457–468.

Sirén, J., Marttinen, P., and Corander, J. (2011). Reconstructing population histories from single nucleotide polymorphism data. *Molecular Biology and Evolution*, *28*(1), 673–683.

Sokal, R. R. and Michener, C. D. (1958). A statistical method for evaluating systematic relationships. *Univ Kans Sci Bull*, *38*, 1409–1438.

Song, Y. S. and Steinrücken, M. (2012). A Simple Method for Finding Explicit Analytic Transition Densities of Diffusion Processes with General Diploid Selection. *Genetics*, *190*(3), 1117–1129.

Stamatakis, A. (2006). RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics*, *22*(21), 2688–2690.

Stevens, P. F. (1991). Character states, morphological variation, and phylogenetic analysis: a review. *Systematic Botany*, *16*(3), 553–583.

Swofford, D. L. (2002). PAUP*: Phylogenetic analysis using parsimony (* and other methods), Version 4.0b.

Trefethen, L. (2000). *Spectral Methods in MATLAB*. Software, Environments, and Tools. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104).

Voelckel, C., Gruenheit, N., Biggs, P., Deusch, O., and Lockhart, P. (2012). Chips and tags suggest plant-environment interactions differ for two alpine Pachycladon species. *BMC Genomics*, *13*(1), 322.

Wahl, L. M. (2011). Fixation when N and s Vary: Classic Approaches Give Elegant New Results. *Genetics*, *188*(4), 783–785.

Wilson, I. J. and Balding, D. J. (1998). Genealogical Inference From Microsatellite Data. *Genetics*, *150*(1), 499–510.

Wright, S. (1931). Evolution in Mendelian populations. *Genetics*, *16*(2), 97–159.

Wright, S. (1934). An analysis of variability in number of digits in an inbred strain of guinea pigs. *Genetics*, *19*(6), 506.

Wright, S. (1945). The differential equation of the distribution of gene frequencies. *Proceedings of the National Academy of Sciences*, *31*(12), 382–389.

Wright, S. (1951). The genetical structure of populations. *Annals of eugenics*, *15*(4), 323–354.

Yang, Z. (2006). *Computational molecular evolution*, Volume 21. Oxford: Oxford University Press.