Department of Mathematics and Statistics

A Sequential Monte Carlo Algorithm with Transformations for Bayesian Model Exploration: Applications in Population Genetics

by

Richard James Culliford

Thesis submitted for the degree of
Doctor of Philosophy

Applied Statistics

Department of Mathematics and Statistics

April 2019

Declaration of Authorship

I confirm that this is my own work and the use of all material from other sources has been properly and fully acknowledged.

SIGNATURE:

DATE:

Abstract

Given a statistical model that attempts to explain the data, calculating the Bayes' posterior distribution of the models parameters is desirable. The marginal likelihood of the model is also of interest, which is used for model comparison. However, for most applications, only estimates of these two measurements can be obtained with a class of methods that give consistent estimates being Monte Carlo algorithms.

This thesis attempts to improve both the process in inferring a high-dimensional posterior distribution and the corresponding model marginal likelihood, on the condition that we can define an ordered set of statistical models in which deterministic transformations between each adjacent model can be applied. We propose an adaption of the sequential Monte Carlo algorithm, which we term the "transformation Sequential Monte Carlo" algorithm. The key feature of this algorithm is by defining a series of target distributions, that make use of said mentioned model transformations, we aim to infer high dimensional models by using easier to estimate posteriors from lower dimensional models with a model transformation applied.

Our proposed algorithm has advantages over many established MC methods. One notable advantage is that we can tailor the algorithm if we wish to update a posterior distribution by including additional observations, but these observations also correspond to a new parameter set that needs to be inferred. Alternatively it is useful where the parameter space can become too large to explore using basic MC methods, for example if there exists an exponential or factorial relationship with observation size and the number of discrete values, but using a lower dimensional model and incorporating it into the model exploration assists with convergence.

We test these strengths of tSMC under three applications, which include two population genetics applications being ancestral reconstruction under the coalescent and the other being the Structure algorithm.

Acknowledgements

I give thanks to the Modernising Medical Microbiology research group (http://modmedmicro.nsms.ox.ac.uk/) for providing half of the funding to make this P.h.D project possible and inviting me to their seminars. I hope to still stay in touch and contribute to any future projects. Also I send my thanks to the University of Reading for their half of the total funding.

I would like to thank Dr. John P. Huelsenbeck for the prompt response after I requested population allele data used in chapter 5. I also thank Dr. Daniel Lawson for discussing his adaption of the Structure model, as well as for showing great support for collaboration with the University of Reading Mathematics and Statistics department. I am very grateful for discussions and exchange of ideas with Dr. Felipe Medina-Aguayo and Dr. Mark Bell and I hope to collaborate and support your research in the future. I also appreciate for help received by the P.h.D staff of Peta-Ann King and Kristine Albridge when I had queries regarding my thesis.

I thank Dr. Daniel Wilson for his contribution to our research and the recently published research paper based on our findings in this investigation. I also feel extremely lucky to have Dr. Richard G. Everitt as my supervisor for four years. I find it hard to believe that time has gone that quickly. Finally I would like to thank Celine and my mum for all the love and support (and tolerance) over the four years of my P.h.D. project.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation and Framework

This thesis focuses on the Bayesian inference problem where we wish to consider a statistical model that describes the data, y, with the ith observation of the dataset defined by $y_i \in Y$ where Y is a sample space that contains the complete set of values that y_i can possibly have. Bayes' inference involves inferring the posterior distribution for the model parameters $\theta \in \Theta$, see section 1.2 of this chapter for an expanded introduction, and for expedition purposes in this chapter we assume that Θ represents a continuous multi-dimensional parameter space. We define the model likelihood of the data as $f(y|\theta)$, the prior distribution as $p(\theta)$ and the posterior distribution as $\pi(\theta|y)$ which is defined by

$$\pi(\theta|y) = \frac{f(y|\theta)p(\theta)}{\int_\Theta f(y|\theta)p(\theta)d\theta}. \tag{1.1}$$

A common obstacle to obtain the exact form of (1.1) is solving the integral in (1.2) representing the distribution of the data marginalised over the parameters, termed as the marginal likelihood (ML),

$$Z(y) = \int_\Theta f(y|\theta)p(\theta)d\theta. \tag{1.2}$$

Over the last few decades Monte Carlo (MC) algorithms, see section 1.3, have been proposed that allow us to either estimate (1.2) or to bypass this formula to receive an estimate of $\pi(\theta|y)$. As described in chapter 2, we may desire this marginal likelihood as a measure to compare different models and see what provides the best model fit. The algorithm we present in this thesis attempts to estimate the marginal likelihood, but it was also primarily aimed for a particular situation in which we wish to use Monte Carlo (as described below).

Suppose that for a particular model we are interested in the idea of sequentially adding data observations over time and then updating the posterior distribution. Under Bayes' theorem when introducing a new set of observations $y_2 \subset Y$, having already found or estimated the posterior of $\pi(\theta \mid y_1)$, then given (1.1) we can find the posterior distribution for the combined data $y = \{y_1, y_2\}$ proportional to

$$\pi(\theta \mid y) \quad \propto \quad \pi(\theta \mid y_1) f(y_2 \mid \theta, y_1), \tag{1.3}$$

where (1.3) is known up to a normalising constant. If we knew the exact form of the posterior distribution then it is trivial to calculate it with the new data, however since this may not be achievable then again we are dependent on Monte Carlo methods. The most standard Monte Carlo methods, such as Markov chain Monte Carlo (MCMC) does not allow us to update a Monte Carlo estimate of $\pi(\theta|y_1)$ with y_2, we specifically explain why in section 1.4, and we would be forced to have a completely new run of this algorithm.

However through methods such as particle filters or sequential Monte Carlo (SMC), to be further discussed in sections 1.5 and 1.6 of this chapter, it is possible to obtain an estimate of (1.3) by updating $\pi(\theta|y_1)$ with y_2. This type of inference is useful on the condition that the posterior of $\pi(\theta|y)$ is more similar to $\pi(\theta|y_1)$, in comparison to the prior distribution of $p(\theta)$.

Furthermore we consider the scenario where adding one observation would in-

crease the number of parameters by at least one parameter. If the observations and parameters are distributed by a state space model then we can apply particle filters, described in section 1.6.1, however we are interested in applications which cannot be expressed this way. We would have to consider a proposal that proposes extra parameters additional to those already present, see chapter 2 for a review on some of these methods.

In this thesis we will describe an algorithm that will ideally start with a low-dimensional model and move to a higher dimensional model, while also providing a solution to devising accurate proposals by inferring a sequence of posterior distributions that eventually targets the true target distribution.

We believe the algorithm is best applied when there is a direct relationship between the size of the observations and the number of parameters in a model. For example where gradual sets of observations are submitted over some real-world time period, where separate Monte Carlo runs would be needed for the different sets of data because of the stated relationship between the model and observations. An application of how our proposed algorithm may be used this way is shown in chapter 4.

Furthermore it is useful in high dimensional models when an initial proposal is difficult to devise, but easier to construct under a low-dimensional model. By starting from a simpler model we aim to gradually build up to a high-dimensional model providing that the posteriors of two high-dimensional models are very similar. Chapter 3 and 5 exploit this sole condition alone and do not take into account an incrementally increasing observation size. For the remainder in chapter 1 we give a basic introduction to the Bayesian inference problem and how Monte Carlo methods attempt inference of the posterior distribution. Much of what is discussed in this chapter is considered general statistical knowledge with exceptions including the niche class of particle filters, SMC samplers and associated properties which we do reference. We recommend the following sources; Andrieu *et al.* (2003); Bernardo and Smith

(1994); Chib and Greenberg (1995); Doucet and Johansen (2011); Liu (2004); Mackay (2003); Robert and Casella (2004); Roberts and Rosenthal (2004); Särkkä (2013); Tierney (1994). The remaining chapters give the following contributions.

Chapter 2 gives a brief introduction to model comparison techniques, introduces Reversible jump Markov chain Monte Carlo (RJMCMC) and explains the key components that we apply to our algorithm. This chapter introduces our proposed algorithm of "transformation sequential Monte Carlo" (tSMC), including the strengths and weaknesses of the algorithm, the extensions to tSMC to improve posterior inference, how results should be interpreted and how it has advantages over other standard model-transitions algorithms.

In chapter 3 we apply tSMC in inferring the posterior distribution of a series of univariate Gaussian mixture models. By using this application we investigate the tSMC algorithm extensions, as mentioned in chapter 2, on the general algorithm before deciding whether the said adjustments are appropriate to be used in chapters 4 and 5.

In Chapter 4 we present a tSMC adaption for genealogy reconstruction under coalescent theory and describe the model assumptions made in our applied example. We discuss how it can compete/coexist with other maximum likelihood or Monte Carlo based methods.

Finally in chapter 5 we describe the Structure application in relation to Dirichlet Process mixture models. We propose how these class of algorithms can be adapted into tSMC and how gradually increasing the number of populations can be achieved through tSMC (in comparison to increasing the number of parameters).

1.2 Bayesian Statistics

Bayesian theory has origins in Bayes and Price (1743), which primarily was an analysis and discussion on the probability of an event occurring given the data. Given

some data $y \in Y$, we define a statistical model for the data. In comparison to frequentist statistics we claim uncertainty on the true values of each of the model parameters, θ, by design. For purposes of exposition we let θ be a continuous random vector of $\theta \in \Theta \subset \mathbb{R}^d$ where d is the dimensional size of θ. The posterior distribution, $\pi(\theta|y)$, is the distribution of the parameters after data has been introduced. We also define a prior distribution, $p(\theta)$, which is based on the prior information of θ that is commonly based from model assumptions given the research field. When knowledge of a parameter relating to some physical system is established, we can assume an informative prior which accurately represents our prior knowledge of θ. They might have strong cut-off points that give minuscule probabilities for improbable values, and thus they concentrate the potential posterior distribution on a smaller range of values. Alternatively if we have little information on θ we may choose to assign a weakly informative prior, characterised by long distributional tails and weaker peaks of probability density (a more flat density). An example of a non-informative prior are "Jeffreys priors" (Jeffreys, 1946) which are invariant to any transformation of the parameter set, which means that should a transformation be applied to a parameter then the new prior can be constructed by using the "change-of-variables" formula on the untransformed prior distribution (if not invariant you would have to invent a new prior based on model assumptions). Although the weaknesses with Jeffreys priors are depending on the application the prior has a chance of being an improper prior, which is a prior that is not normalised and whose probability distribution does not sum to one which may also lead to improper posterior distributions. Furthermore Jeffreys priors are harder to use and solve in high-dimensional models. We also note that the priors have hyper-parameters where we could assume additional uncertainty by setting hyper-priors on a subset of these hyper-parameters, which are usually applied to account for additional group differences depending on the data and/or model. These type of models can be termed as "Hierarchical Bayesian models".

As briefly mentioned in section 1.1 the posterior distribution in equation (1.1) is

derived from the likelihood and prior. It requires equation (1.2) to be evaluated but this integral can be intractable to solve, notably for high-dimensional problems. It is possible to avoid such a calculation of the marginal likelihood using conjugate priors, where the prior-likelihood combination gives a posterior that has the same distribution as the prior. However it is usually an option only for the simplest of prior-likelihood relationships and not for the applications shown in chapters 3-5, which consider high-dimensional parameter space.

A class of methods that could approximate this integral are "numerical methods", which consider splitting the complete parameter space of a marginal parameter into a large number of N intervals segments and estimating the integral by combining estimates of the integral in these smaller intervals. However these methods scale very poorly in high dimensions. Overall when assigning N interval segments per dimension d for each integrand the computational cost is proportional to $O(N^d)$.

Otherwise we could consider Laplace's approximation (which we make use of in chapter 4) to estimate the normalisation constant. Given that we wish to estimate $\int_\Theta \pi(\theta)d\theta$, we first Taylor-expand around the log of $\pi(\theta)$ defined by

$$\widetilde{\log(\pi(\theta))} = \log(\pi(\tilde{\theta})) - 0.5(\theta-\tilde{\theta})^{\mathrm{T}}H(\theta-\tilde{\theta}), \tag{1.4}$$

where $\tilde{\theta}$ is the maximum a posteriori (MAP) of θ, being the value that maximises the posterior distribution, and H is the Hessian matrix given by

$$H_{ij} = -\frac{\partial^2}{\partial\theta_i\partial\theta_j} log(\pi(\theta))\bigg|_{\theta=\tilde{\theta}}. \tag{1.5}$$

By taking the exponential of the Taylor expansion in (1.4) we note that

$$\begin{aligned}\widetilde{\pi(\theta)} &= \pi(\tilde{\theta})\exp(-0.5(\theta-\tilde{\theta})^{\mathrm{T}}H(\theta-\tilde{\theta})) \\ &\propto \exp(-0.5(\theta-\tilde{\theta})^{\mathrm{T}}H(\theta-\tilde{\theta})),\end{aligned} \tag{1.6}$$

which is a density of Normal$(\theta|\mu=\tilde{\theta},\tau=H)$, where μ and τ is the mean and precision

respectively of a Gaussian distribution, so estimates of $\pi(\theta)$ can be sampled from this distribution. The approximation of (1.2) is given by the normalisation constant of said Gaussian distribution defined in (1.7),

$$\widetilde{\int_{\Theta} \pi(\theta) d\theta} \quad = \quad \pi(\tilde{\theta}) \sqrt{\frac{(2\pi)^d}{\det|H|}}, \tag{1.7}$$

where "det" defines the determinant of a matrix. Depending on the application it is usually recommended that $\pi(\theta)$ is expressed in the form of $\exp(\eta(\theta))$, where $\eta : \Theta \to \mathbb{R}^d$ being a function on the parameter space of θ, such that we can obtain a Gaussian approximation through Laplace's approximation. However it may not always be possible for certain unnormalised posteriors to be defined through a Taylor expansion, for example if discrete parameters are present then it is not possible to apply Laplace's approximation. The Laplace approximation to the normalisation constant is not invariant should a nonlinear transformation be applied to θ, i.e the method is basis-dependent and we would have different estimates for the marginal likelihood. Finally the approximation is only appropriate when the posterior is justifiably Gaussian distributed and is not multi-modal.

Our focus then shifts onto Monte Carlo techniques to obtain an approximation for $\pi(\theta|y)$, which either lack the weaknesses or at least is not as hard to use in high dimensions in comparison to numerical methods or Laplace approximations.

1.3 Monte Carlo Methods

In this section we give a basic introduction to the theory and convergence properties of Monte Carlo (MC) algorithms. Although MC methods can be used for different applications, for example they can be used for numerical optimisation, we focus on the class of algorithms regarding integral estimation.

The first Monte Carlo methods were developed within the 1940s, with the very first paper on the subject published by Metropolis and Ulam (1949). The first form

of the Markov chain algorithm, the Metropolis algorithm, was developed by Metropolis *et al.* (1953) which focused on particle physics applications but only considered symmetric proposals on some function. They also emphasised certain properties that allow for a convergence to a stationary distribution (explained in section 1.4). The Metropolis-Hastings algorithm by Hastings (1970) was a generalised form of the Metropolis algorithm which also allowed for non-symmetric proposals and described the target distribution as an invariant distribution of the Markov chain. Afterwards Gibbs sampling was introduced as a special case of the Metropolis Hastings algorithm in Geman and Geman (1984), based on earlier research by Josiah Gibbs in the early 1900s. Despite the introduction of these stated Markov chain Monte Carlo methods in the 1970s-1980s, widespread use was restrained by poor computer processing power. However after several research papers, notably starting with both applied and suggested applications in Gelfand and Smith (1990) such as the Exchangeable Poisson model, an increase of computational power from new systems and the introduction of BUGS software (Bayesian Inference using Gibbs Sampling, Lunn *et al.* (2009)) were the advantages of using MCMC fully displayed. The complete history of MCMC is far more complex than stated in this thesis, where there existed additional developments similar to the stated research that are less popular, and we would recommend more advanced discussions by Hitchcock (2003); Robert and Casella (2011); Tanner and Wong (2010).

MC methods provide a solution to estimate integrals of the form $\int_\Theta \eta(\theta)\pi(\theta)d\theta$ where $\eta : \Theta \to \mathbb{R}^d$ is some function on the parameter space of θ, and $\pi : \Theta \to \mathbb{R}^d$ is a probability density of θ. Notably the integral form of the marginal likelihood, stated in (1.2), can be expressed this way where $\eta(\theta) = f(y|\theta)$ and $\pi(\theta)$ is the prior distribution $p(\theta)$. Alternatively where $\eta(\theta) = \theta$ and $\pi(\theta)$ is the posterior distribution of $\pi(\theta|y)$, this integral is the posterior expectation.

A standard MC approximation involves drawing N independent samples of $\theta = \{\theta^1, ..., \theta^N\}$ which are sampled from $\pi(\theta)$. Directly sampling from the target distribu-

tion is termed "Perfect Monte Carlo Sampling", and thus we use the approximation in (1.9) to receive an unbiased estimate of the integral,

$$\mathrm{E}_\pi[\eta(\theta)] = \int_\Theta \eta(\theta)\pi(\theta)d\theta, \tag{1.8}$$

$$\hat{\mathrm{E}}_\pi[\eta(\theta)] = \frac{1}{N}\sum_{i=1}^{N}\eta(\theta^i). \tag{1.9}$$

Consider the Dirac measure δ on the sample, in which $\delta_{\theta^{(i)}}(\theta)$ is equal to one when $\theta^{(i)} = \theta$ and zero everywhere else. An empirical estimate of the target distribution, when $\eta(\theta) = \theta$, can be calculated by taking a weighted sum of Dirac measures, in the basic MC algorithm we assume they all have equal probability, defined by

$$\hat{\pi}(\theta_{1:N}) = \frac{1}{N}\sum_{i=1}^{N}\delta_{\theta^{(i)}_{1:N}}(\theta_{1:N}). \tag{1.10}$$

However the clear weakness with this most basic MC algorithm is that it requires that sampling from $\pi(\theta)$ is feasible. In the cases where it is not feasible we focus on alternative MC methods which will be explained within the rest of this chapter.

Before these alternative MC solutions are discussed we should note several convergence properties regarding (1.9) and (1.10). By the "law of large numbers"

$$\frac{1}{N}\sum_{i=1}^{N}\eta(\theta^i) \rightarrow \int_\Theta \eta(\theta)\pi(\theta)d\theta, \tag{1.11}$$

as $N \rightarrow \infty$ and again each θ^i was sampled from $\pi(\theta)$. We note that (1.11) follows the strong law of law numbers and converges almost surely to its target (and thus it converges in probability). Overall (1.11) states that the estimator is "consistent", meaning that as N increases then the estimator of (1.9) will eventually converge to what it is aiming to estimate. Furthermore the central limit theorem states that

$$\hat{\mathrm{E}}_\pi[\eta(\theta)] - \mathrm{E}_\pi[\eta(\theta)] \rightarrow \text{Normal}\left(\mu = 0, \tau = \left(\frac{\sigma^2}{N}\right)^{-1}\right), \tag{1.12}$$

where μ and τ is the mean and precision respectively of a Gaussian/normal distribu-

tion, as $N \to \infty$ and $\sigma^2 = \text{Var}_\pi[\eta(\theta)] < \infty$. Equation (1.12) states that the Monte Carlo estimate $\hat{\text{E}}_\pi[\eta(\theta)]$ converges in distribution to a Gaussian distribution with mean $\text{E}_\pi[\eta(\theta)]$ and variance σ^2/N. Overall using (1.11) as an estimate to the integral gives an unbiased estimator, with $\text{E}\left[\hat{\text{E}}_\pi[\eta(\theta)]\right] - \text{E}_\pi[\eta(\theta)] = 0$, where $\text{E}\left[\hat{\text{E}}_\pi[\eta(\theta)]\right]$ is the expectation of the estimate $\hat{\text{E}}_\pi[\eta(\theta)]$ given a large number of similar Monte Carlo algorithm runs that infer the same target distribution.

We would also consider the non-negative mean squared error (MSE), being the square of the errors, where we define it in the form

$$\text{MSE}_\pi[\hat{\text{E}}[\eta(\theta)]] = \left(\text{E}\left[\hat{\text{E}}_\pi[\eta(\theta)]\right] - \text{E}_\pi[\eta(\theta)]\right)^2 + \text{Var}\left[\hat{\text{E}}_\pi[\eta(\theta)]\right], \tag{1.13}$$

where $\text{Var}\left[\hat{\text{E}}_\pi[\eta(\theta)]\right]$ is the variance, also known as the "Monte Carlo variance", of the estimate $\hat{\text{E}}_\pi[\eta(\theta)]$. It may be used to determine how efficient the Monte Carlo estimator of $\text{E}_\pi[\eta(\theta)]$ is, where we aim to minimise the MSE by ensuring that the Monte Carlo variance is as low as possible. The "bias" is termed as $\left(\text{E}\left[\hat{\text{E}}_\pi[\eta(\theta)]\right] - \text{E}_\pi[\eta(\theta)]\right)$ where if we cannot obtain an unbiased estimator, as explained earlier, we at least desire for this value to be as minuscule as possible.

The class of Monte Carlo methods allow us to sample from $\pi(\theta)$ if this target distribution has a non-standard distribution. One of the simplest examples of such an algorithm is "rejection sampling". The basic premise is given $\pi(\theta)$, which might not be known up to a normalisation constant, instead of sampling from it we consider an easy to sample distribution $g(\theta)$. We also set $R \in \mathbb{R}^+$ such that $\pi(\theta) < R \times g(\theta)$. At each state we sample from $g(\theta)$ and $u \sim \text{Unif}(0, 1)$, and we accept θ^i as part of the Monte Carlo sample if $u < \pi(\theta^i)/(R \times g(\theta^i))$. This process is continued until an appropriately sized Monte Carlo sample is collected. However it is not a practical method for high dimensional problems as trying to find the best possible $R \times g(\theta)$ that follows the overall gradient or shape of $\pi(\theta)$, while still only marginally above the target distribution for all θ, is not an easy or potentially possible task. For example if

a large R is required for $\pi(\theta) < R \times g(\theta)$ for a high dimensional θ then the probability of an accepted proposal is approximately $1/R$ which has the potential to be a very small probability depending on the size of R (Andrieu *et al.*, 2003).

Alternatively a sample may be obtained through a different Monte Carlo method that considers the Markov chain family of algorithms, the core details are explained in the next section. We would also extend to other popular forms of Monte Carlo methods including Importance Sampling and sequential Monte Carlo in sections 1.5 and 1.6 respectively.

1.4 Markov Chains and Markov Chain Monte Carlo

Consider a stochastic process, or random process, which is a series of random variables $(\theta^0, ..., \theta^N)$ indexed by some time scale set $(0, ..., N)$. The process is used to model the changes of a system of variables in time. We also briefly note that we define θ^0 to represent some initial value for the stochastic process, and we use the notation more often when using other algorithms that only consider some time index (we change the notation again when we start to use 'particles' in the later sections). The only time index we consider is a discrete time process $(0, ..., N) \subset \mathbb{Z}^+$, where each θ^i stays in their state for exactly one unit of time. If we can go by the assumption that the conditional distribution of θ^i given all of its past states of $\theta^0, ..., \theta^{i-1}$ is the same as the conditional distribution dependent on θ^{i-1} only, and not on any past or future states, then the joint probability of the random variables can be given by

$$\begin{aligned} Pr(\theta^0, ..., \theta^N) &= Pr(\theta^0, ..., \theta^{N-1}) \times Pr(\theta^N | \theta^0, ..., \theta^{N-1}) \\ &= Pr(\theta^0) \prod_{i=1}^{N} Pr(\theta^i | \theta^0, ..., \theta^{i-1}) \\ &= Pr(\theta^0) \prod_{i=1}^{N} Pr(\theta^i | \theta^{i-1}), \end{aligned} \tag{1.14}$$

this is also known as the "memory-less" property. We consider this class of stochastic processes known as Markov processes and focus on discrete time Markov processes, or Markov chains as they are usually called, which apply the type of kernels shown in (1.15). We propose each θ^i sequentially using the transition kernel, based on (1.14),

$$K(\theta^{i-1}, \theta^i) \quad = \quad Pr(\theta^i|\theta^{i-1}). \tag{1.15}$$

Specifically these are time homogeneous Markov chains as all conditional probabilities are independent of the time index such that

$$Pr(\theta^{i+j}|\theta^{i+j-1} = R) \quad \equiv \quad Pr(\theta^i|\theta^{i-1} = R), \tag{1.16}$$

for $j \in \mathbb{N}$ and $R \in \mathbb{R}^d$. Furthermore we state that a Markov chain has a stationary distribution if there exists a distribution π (also termed as an invariant probability distribution) such that,

$$\pi(\theta^i) \quad = \quad \int_\Theta \pi(\theta^{i-1}) K(\theta^{i-1}, \theta^i) d\theta^{i-1}. \tag{1.17}$$

Furthermore π is the limiting distribution of a Markov chain if no matter what state we start in the chain, the current distribution will eventually converge to π as the number of applied kernels go to infinity,

$$\pi \quad = \quad \lim_{N\to\infty} Pr(\theta^N = R|\theta^0). \tag{1.18}$$

A Markov chain Monte Carlo (MCMC) algorithm is where the Markov chain is constructed in such a way that some desired target distribution, that we wish to infer, is the chains limiting distribution. Therefore given that all marginal points are eventually sampled from the same target distribution we aim to obtain a dependent MC sample from a Markov chain. Three conditions need to be satisfied in order to ensure this, and we consider them as general guidelines when constructing Markov kernels

used in this investigation. The first condition is that π is the stationary distribution of the chain.

The second condition regards if each state is "irreducible". At a current state, and has to hold true for all states, it is possible to reach any other state in the parameter space through a finite number of Markov transitions, e.g in the discrete time case $\exists i' > i$ such that $Pr(\theta^{i'} = R'|\theta^i = R) > 0$. For example it is possible to move from say R to R' within $i' - i$ kernel moves.

Finally the Markov chain must not get stuck in a cycle of revisiting the same states of the chain in multiples of m iterations, i.e the chain must be "aperiodic". For example given a series of subsets $(\Theta^0, ..., \Theta^m) \subset \Theta, m \in \mathbb{N}^+$ with Θ representing the parameter space for all θ. Then periodicity exists if at a certain state $Pr(\theta^i \in \Theta^j|\theta^{i-1}) = 1$ for all $\theta^{i-1} \in \Theta^{j-1}$ and $i, j \in \mathbb{N}$, and furthermore $Pr(\theta^i \in \Theta^0|\theta^{i-1} \in \Theta^m) = 1$.

We consider the differences between the estimated distribution generated from a Markov chain and the true target distribution through the total variation distance,

$$||K^n(\theta, \Theta) - \pi(\theta)||_{TV} \quad = \quad \sup_\Theta |K^n(\theta, \Theta^{'}) - \pi(\theta)|. \tag{1.19}$$

A limiting distribution in the Markov chain implies convergence in total variation distance, i.e

$$\lim_{n\to\infty} ||K^n(\theta, \Theta) - \pi(\theta)||_{TV} \quad = \quad 0. \tag{1.20}$$

Under the weaker conditions of irreducibility and π being a stationary distribution, a Strong Law of Large Numbers holds. Under some additional conditions a central limit theorem holds, with the σ^2 term (seen in (1.12)) when using Markov chains (Jones, 2004) defined by

$$\sigma^2 \quad = \quad \mathrm{Var}[E[\eta(\theta^0)]] + 2\sum_{i=1}^{\infty} \mathrm{Cov}(E[\eta(\theta^0)], E[\eta(\theta^i)]). \tag{1.21}$$

One way of ensuring stationarity with respect to π is to chose a Markov chain kernel

such that it fulfills the sufficient condition termed the detailed balance formula,

$$\pi(\theta^i)K(\theta^i, \theta^{i-1}) \quad = \quad \pi(\theta^{i-1})K(\theta^{i-1}, \theta^i). \tag{1.22}$$

A chain that satisfies detailed balance with respect to the target distribution π implies that π is a stationary distribution of the chain. It is an easy condition to check, and therefore is used in the construction of most MCMC algorithms. Overall MCMC has a minimum computational cost of $O(N)$.

However if we refer back to equation (1.3), the reason we cannot solely use MCMC to update a posterior distribution with more observations is because the stationary distribution must be exactly the same within each state of the chain but increasing or decreasing the number of observations will change the stationary distribution. Therefore we can't, for example, define each state of the chain to differ by the number of observations. Therefore we consider importance sampling solutions, as seen in sections 1.5 and 1.6. Before discussing these methods, we give a brief introduction to two of the most common MCMC algorithms and their corresponding kernel moves.

1.4.1 Gibbs Sampler and Metropolis-Hastings Algorithm

Suppose we define θ as d-dimensional where each marginal parameter is defined by θ_j, and θ_{-j} representing the joint set of parameters that does not include θ_j. Gibbs sampling can be performed if we can obtain full conditionals on each parameter i.e $f(\theta_j|\theta_{-j})$. The Gibbs sampler algorithm is displayed in algorithm 1, where $q(\cdot)$ is a simple to sample from distribution, and uses the kernel

$$K(\theta^{i-1}, \theta^i) \quad = \quad \prod_{j=1}^{d} \pi(\theta^i_j|\theta^i_{-j}). \tag{1.23}$$

Otherwise an algorithm that is applied more widely than Gibbs sampling is the Metropolis Hastings algorithm, shown in algorithm 2.

Algorithm 1 Standard Gibbs Sampling Algorithm

Set variable ordering of $\theta_1, \theta_2, ..., \theta_d$
Set chain length N
$\theta^0 \sim q(\cdot)$
for $i = 1 : N$ **do**
  for $j = 1 : d$ **do**
    $\theta_j^i \sim \pi(\theta_j^i | \theta_{-j}^i)$
  end for
end for

Algorithm 2 Standard Metropolis Hastings Algorithm

Set chain length N
$\theta^0 \sim q(\cdot)$
for $i = 1 : N$ **do**
  $\tilde{\theta}^i \sim q(\cdot\,|\theta^{i-1})$
  $\alpha(\theta^{i-1}, \tilde{\theta}^i) = \min\left\{1, \dfrac{\pi(\tilde{\theta}^i) q(\theta^{i-1}|\tilde{\theta}^i)}{\pi(\theta^{i-1}) q(\tilde{\theta}^i|\theta^{i-1})}\right\}$
  $u \sim$ Unif(0,1)
  $\theta^i = \tilde{\theta}^i$ if $u < \alpha(\theta^{i-1}, \theta^i)$, otherwise set $\theta^i = \theta^{i-1}$
end for

The transition kernel of Metropolis Hastings is given by

$$\begin{aligned} K(\theta^{i-1}, \theta^i) &= q(\theta^i|\theta^{i-1})\alpha(\theta^{i-1}, \theta^i) \\ &+ (1 - \int_\Theta q(\theta^i|\theta^{i-1})\alpha(\theta^{i-1}, \theta^i) d\theta^i)\delta_{\theta^{i-1}}(\theta^i), \end{aligned} \tag{1.24}$$

where $q(\theta^i|\theta^{i-1})$ is a simple to sample from distribution that uses parameters from a previous iteration and $\alpha(\theta^{i-1}, \theta^i)$ is the function of the acceptance probability that the value proposed for θ^i will be the next iteration given the previous iteration of θ^{i-1}. For further proofs regarding how this kernel fulfills the criterion of the detailed balance equation or how the acceptance probability function shown in algorithm 2 is designed to satisfy detailed balance, see Chib and Greenberg (1995); Roberts and Rosenthal (2004). The success of the convergence depends on the proposal $q(\cdot|\theta^{i-1})$, and a balance of the proportion of accepted proposals has to be considered. If the proposal has a large variance then many of the moves will be rejected which leads to

high correlation of the sampled chain. Otherwise if the variance is too small then the chain will explore the distribution slowly and may not visit multiple modes if these are present in the target distribution. A basic example of such proposal distributions includes the independent sampler, $q(\cdot|\theta^{i-1}) \sim q(\cdot)$, which has no dependency on the previously iterated state. Another example is the random walk sampler where the proposal is centered on θ^{i-1}, e.g Normal($\mu = \theta^{i-1}, \tau$) being a Gaussian distribution, or Unif($\theta^{i-1} - 1, \theta^{i-1} + 1$) being a continuous uniform distribution. Due to the form of algorithm 2 it is possible to sample from π while only knowing its distribution up to a normalisation constant, which makes the Metropolis Hastings algorithm a very viable solution to applications that apply Bayes' theorem.

1.5 Importance Sampling

Importance sampling (IS) is another Monte Carlo method which is given special attention as sequential Monte Carlo techniques, described in the next section, build on top of importance sampling. Similarly to rejection sampling, instead of simulating directly from $\pi(\theta)$ we consider a simple to simulate proposal distribution $g(\theta)$ which is similar to the target distribution. Here we use the term "particles" to explain the complete set of proposals for the target distribution, see Annealed Importance sampling which is explained later within this section as to how changes to the initially generated particles can be proposed. Furthermore the ith particle is defined as θ^i in comparison to section (1.4) where we used it to represent the ith state within a Markov chain. A rearrangement of (1.8) and (1.9) gives us

$$\begin{aligned} \mathrm{E}_\pi[\eta(\theta)] &= \int_\Theta \eta(\theta)\frac{\pi(\theta)}{g(\theta)}g(\theta)d\theta \\ &= \mathrm{E}_g\left[\eta(\theta)\frac{\pi(\theta)}{g(\theta)}\right] \qquad (1.25) \\ \hat{\mathrm{E}}_g\left[\eta(\theta)\frac{\pi(\theta)}{g(\theta)}\right] &= \frac{1}{N}\sum_{i=1}^{N}\eta(\theta^i)\tilde{w}(\theta^i), \qquad (1.26) \end{aligned}$$

providing that the normalisation constant for $\pi(\theta)$ is known (see (1.28) when it is unknown). We define each $\tilde{w}(\theta^i) = \pi(\theta^i)/g(\theta^i)$, $\tilde{w}_i > 0$, as an unnormalised importance weight with the normalised IS weights defined by $\tilde{w}(\theta^i)/N$. This proposal distribution should be as close to the shape of $\pi(\theta)$, as the variance of the importance sampling estimator is proportional to $1 + \mathrm{Var}_g(\tilde{w}(\theta^i))$. It is highly advisable to set $g(\theta)$ to be more heavy-tailed than $\pi(\theta)$ to prevent the risk of having estimators with infinite variance (Robert and Casella, 2004). Where the target distribution is only known up to a normalisation constant, such as in Bayes' problems, we would need consider an alternative formulation. Given that the normalisation constant is defined by $\int_\Theta \pi(\theta)d\theta$ for an unnormalised $\pi(\theta)$ (as the integral would be equivalent to 1 if it was normalised) then we rewrite the approximation as,

$$\begin{aligned}
\mathrm{E}_\pi[\eta(\theta)] &= \int_\Theta \eta(\theta) \frac{\left(\frac{\pi(\theta)}{g(\theta)} g(\theta)\right)}{\left(\int_\Theta \frac{\pi(\theta)}{g(\theta)} g(\theta) d\theta\right)} d\theta \\
&= \frac{E_g\left[\eta(\theta)\frac{\pi(\theta)}{g(\theta)}\right]}{E_g\left[\frac{\pi(\theta)}{g(\theta)}\right]} \qquad (1.27) \\
\hat{\mathrm{E}}_\pi[\eta(\theta)] &= \frac{\frac{1}{N}\sum_{i=1}^{N} \eta(\theta^i)\tilde{w}(\theta^i)}{\frac{1}{N}\sum_{i=1}^{N} \tilde{w}(\theta^i)} \\
&= \sum_{i=1}^{N} \eta(\theta^i) w(\theta^i), \qquad (1.28)
\end{aligned}$$

with a normalised importance weight of $w(\theta^i) = \tilde{w}(\theta^i)/\sum_{i=1}^{N}\tilde{w}(\theta^i)$. Thus importance sampling is a viable solution for posterior distribution estimation. We note that although (1.28) is asymptotically unbiased, it is biased for a finite sample in comparison to (1.26) which is an unbiased estimator regardless of the size of N. A weighted

empirical estimate of the target distribution can be obtained via,

$$\hat{\pi}(\theta^{1:N}) \quad = \quad \frac{1}{N}\sum_{i=1}^{N} w(\theta^i)\delta_{\theta^i}(\theta^{1:N}). \tag{1.29}$$

Otherwise IS continues to follow the standard MC convergence properties as described in section 1.3. However a challenge with using importance sampling is trying to devise a proposal distribution for high dimensional distribution, as with more parameters to infer means a smaller probability of the parameters simultaneously being in areas of high probability.

One adaption of importance sampling methods that works better in high dimensions is "annealed Importance Sampling" (AIS) (Neal, 2001). The appeal of AIS is that we try and close the distance between the initial particles $\theta_0 \sim \eta(\cdot)$, which is easy to sample, and the target $\theta_T \sim \pi_T(\cdot)$. Although AIS could be explained from Neal (2001), we explain the algorithm as described by Tokdar and Kass (2010) as this variant is far more similar to sequential Monte Carlo samplers (shown in section 1.6.3).

Note that when describing a particle in AIS we use θ_t^i to represent a parameter at the ith particle within time t of a time index, and sometimes we may use θ_{jt}^i to represent the jth dimension of said particle if θ_t^i is multidimensional. We use this notation for this algorithm and when using any SMC algorithm (see the next section) throughout the rest of the thesis.

Given that the initial set of particles has been generated via $\theta_0 \sim \eta(\cdot)$, we move the particles θ_0 to θ_1 by applying some kernel function $K_1(\theta_0, \theta_1)$ which targets an intermediate distribution ρ_1. This is repeated by applying individual kernels $K_j(\theta_{j-1}, \theta_j)$, where these kernels could be MCMC updates, which target $\rho_j(\cdot)$ to receive each θ_j before finally targeting θ_T. We could define these intermediate distributions via an annealed geometric scheme of $\rho_t = (\eta(\theta))^{1-\varphi_t}(\pi_T(\theta))^{\varphi_t}$ for some $(\varphi_0 = 0, \varphi_1, ..., \varphi_T = 1)$, such that each $\rho_{t-1} \approx \rho_t$. Each unnormalised importance

weight is then given by

$$\tilde{w}^i = \frac{\rho_1(\theta_0^i)}{\rho_0(\theta_0^i)}\frac{\rho_2(\theta_1^i)}{\rho_1(\theta_1^i)}...\frac{\rho_T(\theta_{(T-1)}^i)}{\rho_{T-1}(\theta_{(T-1)}^i)}. \tag{1.30}$$

However a weakness with AIS is that it suffers from particle degeneracy, an attribute that is explained in section 1.6.2.

1.6 Sequential Monte Carlo

The standard sequential Monte Carlo (SMC) sampler algorithm has flexibilities that give it an advantage when applied with Bayesian inference in comparison to both importance sampling and MCMC based algorithms, which we also remark in section 1.7. Before discussing sequential Monte Carlo samplers, which is used as the general framework for our proposed methods in this investigation, we first discuss Bayesian filtering as an introduction to the concept of particle filters. The importance of resampling is also explained. Finally sequential Monte Carlo samplers are introduced.

1.6.1 Sequential Bayesian Filtering for the State Space Model

We give a brief explanation of the state-space model, also known as hidden Markov models (HMM), of

$$\begin{aligned} \theta_0 &\sim p_0(\cdot) \\ \theta_t &\sim p(\cdot|\theta_{t-1}) \\ y_t &\sim f(\cdot|\theta_t), \end{aligned} \tag{1.31}$$

where each θ_t and y_t is indexed to some discrete time $t \in (0, ...T)$, specifically the time index for the second equation is $(1, ...T)$ and for the third equation is $(0, ..., T)$, and we assume that their corresponding densities are homogeneous (i.e independent

from the time index). Here $p_0(\cdot)$ is our initial distribution to generate θ_0, $p(\cdot|\theta_{t-1})$ is a Markov transition kernel, and both $f(\cdot|\theta_0)$ and $f(\cdot|\theta_t)$ are the distributions for observations y_0 and y_t respectively. While we know the values of the observations $y_{0:T}$, we do not know the true values of the hidden variables $\theta_{0:T}$. Therefore we aim to estimate the distributions of these hidden variables via

$$\pi(\theta_{0:T}|y_{0:T}) \quad = \quad \frac{f(y_{0:T}|\theta_{0:T})p(\theta_{0:T})}{Z(y_{0:T})}. \tag{1.32}$$

As stated in section 1.2, we have the standard problem of the normalisation constant $Z(y_{0:T})$ having an intractable integral. Naturally the joint distribution of $\pi(\theta_{0:T}|y_{0:T})$ may be simulated through MCMC. We could simulate the joint and dependent parameter set $\theta_{0:T}$ in one proposed MH move, but devising a single high-dimensional proposal would be difficult as simultaneously proposing each θ_t to their respective probability modes becomes harder with increasing T. Alternatively we could perform MCMC updates on blocks of the state space of length $L \in \{1, ..., T\}$ and targeting

$$\pi(\theta_{t:(t+L)}|y_{0:T}, \theta_{0:(t-1)}, \theta_{(t+L+1):T}) \quad \propto \quad \prod_{i=t}^{t+L+1} p_i(\theta_i|\theta_{i-1}) \prod_{i=t}^{t+L} f(y_i|\theta_i), \tag{1.33}$$

providing that L is sufficiently small enough such that a good proposal is made. However this is not likely to be a good strategy if the posterior dependence between states is strong, and setting L to be small can increase the computational cost if the observation size is large (Andrieu *et al.*, 2010). It is possible to simulate exactly from $\pi(\theta_{0:T}|y_{0:T})$ when the models are finite state space HMM (Doucet and Johansen, 2011; Frühwirth-Schnatter, 1994). Alternatively it is also possible to simulate exactly if p_t and f_t are Gaussian distributed and linear, for example this is the foundation for the Kalman Filter algorithm (Doucet and Johansen, 2011; Kalman, 1960).

However a good strategy for the general case, that does not have the drawbacks of the stated MCMC solutions, is to obtain a sequential set of the estimated densities using the particle filter (Gordon *et al.*, 1993). The term "filtering" regards to

how we estimate the current state of a chain using the past history of the chain, where we estimate $\pi(\theta_{0:t}|y_{0:t})$ by updating our estimate of $\pi(\theta_{0:(t-1)}|y_{0:(t-1)})$. Otherwise "smoothing" is its corresponding counterpart, where we target $\pi(\theta_{0:T}|y_{0:T})$ or the marginal distributions of $\pi(\theta_t|y_{0:T})$ given that we have access to the observation states of $y_{0:T}$. In general, particle filters are exclusively used for state space models and nonlinear filtering applications (Doucet and Johansen, 2011; Särkkä, 2013).

In the particle filtering algorithm we firstly approximate the posterior distribution of $\pi_0(\theta_0|y_0)$ using importance sampling, with $q_0(\theta_0|y_0)$ as our importance sampler using a defined particle size N. This yields an unnormalised importance weight of

$$\tilde{w}(\theta_0) \quad = \quad \frac{f(y_0|\theta_0)p_0(\theta_0)}{q_0(\theta_0|y_0)}, \tag{1.34}$$

and the weights are used as shown in section (1.5) to obtain the estimated posterior distribution and other summary statistics for $\pi(\theta_0|y_0)$.

At the next time index at $t = 1$ we would consider the joint target distribution of $\pi(\theta_{0:1}|y_{0:1})$, where $\pi(\theta_{0:1}|y_{0:1}) \propto [\pi(\theta_0|y_0)]\,[f(y_1|\theta_1)p(\theta_1|\theta_0)]$ and the importance sampler proposal of $q(\theta_{0:1}|y_{0:1}) \propto q_0(\theta_0|y_0)q_1(\theta_1|y_{0:1}, \theta_0)$ is applied to formulate the next unnormalised importance weight of,

$$\tilde{w}(\theta_1) \quad = \quad \frac{f(y_1|\theta_1)p(\theta_1|\theta_0)}{q_1(\theta_1|y_{0:1}, \theta_0)} \frac{f(y_0|\theta_0)p_0(\theta_0)}{q_0(\theta_0|y_0)}. \tag{1.35}$$

From (1.35) we are using a proposal of $q_{t-1}(\theta_{0:(t-1)}|y_{0:(t-1)}, \theta_{0:(t-2)})$ to help construct a proposal for $\pi(\theta_{0:t}|y_{0:t})$, although there is usually no need to use the most elements of the set $\{y_{0:(t-1)}, \theta_{0:(t-2)}\}$ to construct a proposal $q_t(\theta_t|y_{0:t}, \theta_{0:t-1})$ except for niche applications (Doucet and Johansen, 2011). Therefore by using multiple importance sampling updates we have a standard particle filter given in algorithm 3.

The optimal proposal (Doucet and Johansen, 2011) for the particle filter, which is usually not available, to minimise the variance of the importance weights is given

Algorithm 3 Standard Particle Filter without resampling (see section 1.6.2 for discussion on resampling)

Set Particle Size N
Set Markov Chain Length T
for $i = 1 : N$ **do**
  $\theta_0^i \sim q_0(\cdot|y_0)$
  $\tilde{w}(\theta_0^i) = \dfrac{f(y_0|\theta_0)p_0(\theta_0)}{q_0(\theta_0|y_0)}$
end for
for $i = 1 : N$ **do**
  $w(\theta_0^i) = \tilde{w}(\theta_0^i)/\sum_{i=1}^{N}\tilde{w}(\theta_0^i)$
end for
for $t = 1 : T$ **do**
  for $i = 1 : N$ **do**
    $\theta_t^i \sim q_t(\cdot|y_{0:t}, \theta_{0:(t-1)}^i)$
    $\tilde{w}(\theta_t^i) = w(\theta_{t-1}^i)\dfrac{f(y_t|\theta_t^i)p(\theta_t^i|\theta_{t-1}^i)}{q_t(\theta_t^i|y_{0:t}, \theta_{0:(t-1)}^i)}$
  end for
  for $i = 1 : N$ **do**
    $w(\theta_t^i) = \tilde{w}(\theta_t^i)/\sum_{i=1}^{N}\tilde{w}(\theta_t^i)$
  end for
end for

by

$$\begin{aligned} q_t(\theta_t|y_{0:t}, \theta_{0:(t-1)}) &= p(\theta_t|y_t, \theta_{t-1}) \\ &= \frac{f(y_t|\theta_t)p(\theta_t|\theta_{t-1})}{Z(y_t|\theta_{t-1})}. \end{aligned} \tag{1.36}$$

However we may desire to use the methodology used for filtering in alternative situations. In section 1.6.3 we use methodology based on particle filters to simulate from the posterior on a set of static parameters of θ_T. Before we explain this in section 1.6.3, we first remark on resampling in section 1.6.2 and why we should almost always use it for any particle filtering algorithm.

1.6.2 Resampling

After multiple sequential weight updates in SMC methods, weight degeneracy will occur because of increasing variance of the importance weights due to the growing distance between the importance sampling proposal and the target distribution (Doucet *et al.*, 2000; Gordon *et al.*, 1993; Kong *et al.*, 1994). As the time index t continues to infinity it is guaranteed that one particle will contain all the weight (Doucet *et al.*, 2001). Thus degeneracy leads to an estimated distribution that is not representative of the target distribution.

Therefore it is recommended to include a resampling step for both particle filters and SMC algorithms. If the particles are showing excessive degeneracy then the particles, and their corresponding ancestry and weights, are resampled to obtain a new set of particles of $\tilde{\theta}_k^{1:N}$. Resampling methods involve removing particles with low weights and replacing them with replicates of existing particles whose corresponding weights are large, thus the term "resampling" meaning that an estimate of the posterior density is being sampled (Doucet *et al.*, 2001). Resampling itself also increases the variance of the posterior estimates. Thus we avoid resampling at every step and only perform it when enough weight degeneracy has occurred.

An illustration of resampling within the particle filter is shown in figure 1.1, represented in the four steps. In step 1 we sample from a distribution $q_0(\theta_0|y_0)$ that estimates the posterior distribution of $\pi(\theta_0|y_0)$. In step 2 the particles are weighted, using (1.34), with some particles clearly having more weight then others. In step 3 a resampling algorithm is initiated to duplicate particles, with the variation of the resampled particles varying depending on the resampling algorithm itself but most schemes always favour high weighting particles, and afterward the particles are set to have equal weights. Finally in step 4 a proposal $q_1(\theta_1|\theta_0, y_{0:1})$, which uses the resampled particles as part of the proposal, is made that estimates the marginal posterior distribution of θ_1 and this is followed by reweighting the particles, given in (1.35), that gives particle weights corresponding to $\pi(\theta_{0:1}|y_{0:1})$.

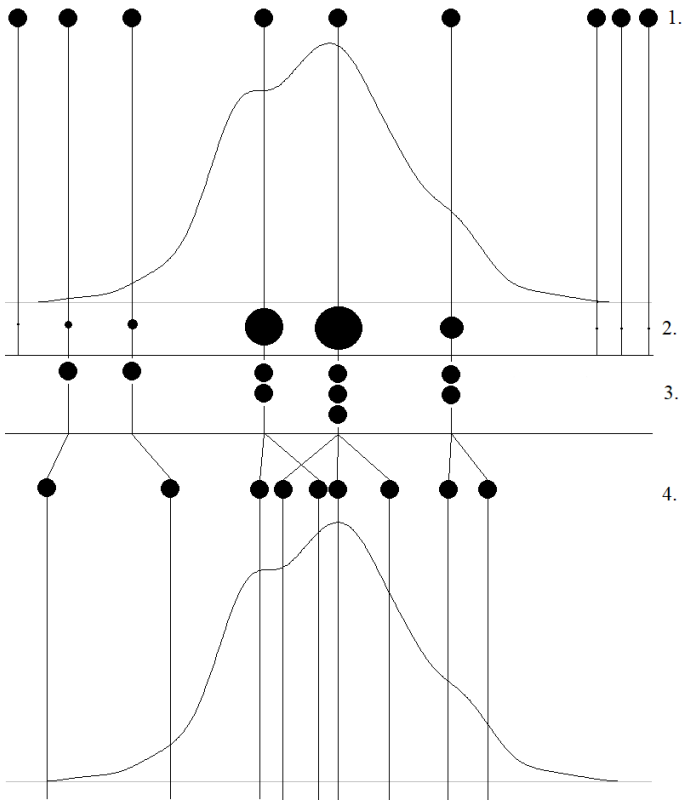


Figure 1.1: An illustration of resampling within the particle filter, where 9 particles are considered and we are assuming that we are estimating a one dimensional parameter at each indexed time. In this example, what can be seen is that some very low weighted particles are replaced.

One common way to measure degeneracy is by calculating the effective sample size (ESS), which takes values between 1 and N, defined as,

$$\frac{1}{\sum_{i=1}^{N}\left(\tilde{w}^i(\theta^i)\right)^2}. \tag{1.37}$$

A predefined threshold is set, and one which will be used when applying such algorithms in the investigation, in which we set it to half of the particle size (Doucet and Johansen, 2011). If the ESS is below this threshold then resampling will be performed.

There are many types of resampling schemes, however "stratified resampling", see algorithm 4, will be applied throughout the investigation. The concept of stratified

Algorithm 4 Stratified Resampling Algorithm. The computational cost is $O(N)$ for stratified resampling

Set original particles $\theta^1, ..., \theta^N$
Set normalised weights $w^1, ..., w^N$
for $i = 1 : N$ **do**
  $u^i \sim \text{Unif}(0, 1)$
  $\tilde{u}^i = \frac{(i-1)+u^i}{N}$
  Find the minimum k such that $\tilde{u}_t^i \leq w_t^1 + ... + w_t^k$
  $\tilde{\theta}^i = \theta^k$
end for

resampling is that different partitions of the cumulative weights are explored, meaning that particle with high weights are still likely to be selected but the probability is not as high if resampled particles were to be selected if instead $\tilde{u}^i \sim \text{Unif}(0, 1)$. Consider the following example, given six particles and $cw^{1:N} = (0.069, 0.247, 0.293, 0.519, 0.901, 1)$ where $cw^{1:N}$ represents each cumulative normalised weights. Then suppose that the particles selected are $\tilde{i} = (1, 2, 4, 5, 5, 6)$ if given $\tilde{u} = (0.05, 0.213, 0.357, 0.4, 0.79, 0.946)$ as we desire the minimum k such that $\tilde{u}^i \leq w^1 + ... + w^k$. Thus we have a set of resampled particles of $\tilde{\theta} = (\theta^1, \theta^2, \theta^4, \theta^5, \theta^5, \theta^6)$.

The most common resampling algorithms being the multinomial, systematic and stratified sampling algorithms give unbiased estimates to the target distribution. Sys-

tematic resampling is considered a potentially good resampling scheme as well, however as it relies on producing samples dependently it means that there is no established theory on its resampling variance reduction in comparison to other schemes. Meanwhile the stratified resampling algorithm has theoretical evidence that it gives a smaller variance than certain resampling algorithm such as multinomial resampling (Douc and Cappé, 2005; Hol *et al.*, 2006).

1.6.3 Sequential Monte Carlo for Static Bayesian Inference

We now consider a sequence of static target distributions of $(\pi_1, ..., \pi_T)$ where $\theta_t \sim \pi_t$, $\theta_t \in \Theta$ and $t \in (0, ..., T)$ is a sequence of natural numbers with T indexing the final target distribution. Otherwise we define π_0 to represent some joint proposal distribution for π_1. Unlike particle filters we do not have each individual observation y_t distributed by a specific distribution dependent on θ_t, i.e $y_t \sim f(\cdot|\theta_t)$. Through sequential importance sampling (SIS) methods we desire to transition a set of particles generated from distribution π_0 to a target distribution of π_T, by moving a set of particles $\{\theta_0^i\}_{i=1}^N$ drawn from π_0 to regions of high probability density in π_1 by using π_0 combined with some kernel as our importance sampler. The process continues up to π_T, which uses π_{T-1} and a moving kernel as the importance sampler, which is similar to particle filtering.

One advantage of using SIS based methods is that we may choose to estimate the posterior distribution of the joint parameters given that we include an additional subset of observational values that are added over time. For example an estimation of $\pi_t \equiv \pi(\theta|y_{1:t})$ can be made using the previously inferred $\pi_{t-1} \equiv \pi(\theta|y_{1:(t-1)})$. This is useful if the computational complexity increases with sample size, providing that π_t and π_{t-1} are similar. It also prevents running an additional MCMC with the complete set of observations (Chopin, 2002).

Alternatively we may desire to gradually approach the target distribution of π_T through some tempering effect via annealing, an example being an annealed impor-

tance sampling scheme of $\pi_i \propto \pi_T^{\varphi_t}(\pi_0)^{1-\varphi_t}$ for $0 = \varphi_0 < \varphi_1 < ..... < \varphi_T = 1$, where the sample path eventually converges to π_T. Raising the target distribution to some power of $\varphi_t < 1$ allows for the acceptance of less likely parameter subsets under the posterior distribution. Therefore like AIS we can sequentially move towards the target distribution in which any sampled values are weighted down, before sampling from the target distribution, if they are unrepresentative of the true distribution. Although we clarify that SIS is more "general" than AIS, such as sequentially including observations over time and resampling (Del Moral *et al.*, 2006; Neal, 2001).

We propose to move the particles at each target generated from the original importance distribution using a Markov kernel K_t. This Markov kernel could be a simple independent move that is not dependent on the previous values of the particles or alternatively it could be a random walk move, for example we could propose a Gaussian random walk with the mean for each ith particle being θ_t^i and the variance being an appropriate value (Del Moral *et al.*, 2006; Doucet and Johansen, 2011).

However if are we interested in static models, estimating the posterior of θ_t and not $\theta_{0:t}$, we need to marginalise the importance distribution. Instead of sampling from $q_t(\theta_{0:t})$ we have a proposal distribution from $q_t(\theta_t)$ given by

$$q_t(\theta_t^i) \quad = \quad \int ..... \int_{\Theta_0 \times ... \times \Theta_{t-1}} q_0(\theta_0^i) \prod_{j=1}^{t} K_j(\theta_j^i | \theta_{j-1}^i) d\theta_0 d\theta_1, ..., d\theta_{t-1}, \qquad (1.38)$$

where $q_t(\theta_{0:t}) = q(\theta_{0:(t-1)}) K_t(\theta_t | \theta_{t-1})$ for example. Naturally it is usually impossible to solve the integral in (1.38) for an arbitrary kernel. So first we consider the formula for the unnormalised weights under the standard sequential importance sampling algorithm which samples sequentially from a target $\pi_t(\theta_{0:t})$, in which we emphasise that this can take the form of a normalised or unnormalised distribution, for increasing t (Doucet and Johansen, 2011; Liu, 2004), with the general form being

$$\tilde{w}_t^i \quad = \quad \frac{\pi_t(\theta_{0:t}^i)}{q_t(\theta_{0:t}^i)}$$

$$
\begin{aligned}
&= \frac{\pi_t(\theta^i_{0:t})}{q_0(\theta^i_0)\prod_{j=1}^{t} K_j(\theta^i_j|\theta^i_{j-1})} \\
&= \frac{\pi_{t-1}(\theta^i_{0:(t-1)})\pi_t(\theta^i_{0:t})}{\pi_{t-1}(\theta^i_{0:(t-1)})q_0(\theta^i_0)K_t(\theta^i_t|\theta^i_{t-1})\prod_{j=1}^{t-1} K_j(\theta^i_j|\theta^i_{j-1})} \\
&= \frac{\pi_{t-1}(\theta^i_{0:(t-1)})}{q_{t-1}(\theta^i_{0:(t-1)})}\frac{\pi_t(\theta^i_{0:t})}{\pi_{t-1}(\theta^i_{0:(t-1)})K_t(\theta^i_t|\theta^i_{t-1})} \\
&= \tilde{w}^i_{t-1}\frac{\pi_t(\theta^i_{0:t})}{\pi_{t-1}(\theta^i_{0:(t-1)})K_t(\theta^i_t|\theta^i_{t-1})}, \qquad (1.39)
\end{aligned}
$$

as $w^i_{t-1} = \pi_{t-1}(\theta^i_{0:(t-1)})/q_{t-1}(\theta^i_{0:(t-1)})$. The concept of sequential Monte Carlo samples is how it defines the joint target distribution to be a product of multiple artificial backward Markov kernels of $L_{t-1}(\theta_{t-1}|\theta_t)$ such that we formulise the most recent marginal distribution, $\pi_t(\theta_t)$, in the chain as

$$
\pi_t(\theta^i_{0:t}) = \pi_t(\theta^i_t)\prod_{j=1}^{t} L_{j-1}(\theta^i_{j-1}|\theta^i_j). \qquad (1.40)
$$

Therefore the marginal distribution of the joint distribution is derived by construction as this involves simply integrating out a set of backward kernels (Del Moral *et al.*, 2006). We reconsider the unnormalised importance weight update given by (1.39) and manipulate the formula by substituting (1.40) into (1.39) to obtain

$$
\begin{aligned}
\tilde{w}^i_t &= \tilde{w}^i_{t-1}\frac{\pi_t(\theta^i_{0:t})}{\pi_{t-1}(\theta^i_{0:(t-1)})K_t(\theta^i_t|\theta^i_{t-1})} \\
&= \tilde{w}^i_{t-1}\frac{\pi_t(\theta^i_t)\prod_{j=1}^{t} L_{j-1}(\theta^i_{j-1}|\theta^i_j)}{\pi_{t-1}(\theta^i_{t-1}))K_t(\theta^i_t|\theta^i_{t-1})\prod_{j=1}^{t-1} L_{j-1}(\theta^i_{j-1}|\theta^i_j)} \\
&= \tilde{w}^i_{t-1}\frac{\pi_t(\theta^i_t)L_{t-1}(\theta^i_{t-1}|\theta^i_t)}{\pi_{t-1}(\theta^i_{t-1})K_t(\theta^i_t|\theta^i_{t-1})}. \qquad (1.41)
\end{aligned}
$$

We then define the unnormalised importance weight via,

$$\tilde{w}_t^i = w_{t-1}^i \frac{\pi_t(\theta_t^i) L_{t-1}(\theta_{t-1}^i|\theta_t^i)}{\pi_{t-1}(\theta_{t-1}^i) K_t(\theta_t^i|\theta_{t-1}^i)}. \tag{1.42}$$

Therefore it is possible to bypass the marginalisation problem as described earlier, and obtain weights that are only dependent on the previous time point (Del Moral *et al.*, 2006). Overall a standard SMC sampler algorithm with resampling is shown in algorithm 5, providing that we are using half the particle size as an ESS threshold for resampling with a computational cost of $O(NT)$.

It is still important to choose the most optimal form of both $L_{t-1}(\theta_{t-1}|\theta_t)$ and $K_t(\theta_t|\theta_{t-1})$ that provides the minimum variance of the weights at each state. The most optimum set of kernels is when we consider an importance sampler of $q_t(\theta_t)$, as stated previously this has to be marginalised from $q_t(\theta_{0:t}) = q_0(\theta)\prod_{j=1}^{t} K_j(\theta_j|\theta_{j-1})$, and have the following relationship with the forward and backward kernels,

$$L_{t-1}(\theta_{t-1}|\theta_t) = \frac{q_{t-1}(\theta_{t-1}) K_t(\theta_t|\theta_{t-1})}{q_t(\theta_t)}, \tag{1.43}$$

in which the substitution of (1.43) into (1.41) simply gives us the standard importance weight update as described in section 1.5 (Del Moral *et al.*, 2006). Again as such a marginalisation is usually not possible we consider sub-optimal choices. In our adaptions we choose MCMC moves, where the backward kernel is given by the reversal of the forward kernel,

$$L_{t-1}(\theta_{t-1}|\theta_t) = \frac{\pi_t(\theta_{t-1}) K_t(\theta_t|\theta_{t-1})}{\pi_t(\theta_t)}, \tag{1.44}$$

where the form of both π_{t-1} and π_t can be known up to a normalisation constant. Using this kernel simplifies the reweighting step to

$$\tilde{w}_t^i = w_{t-1}^i \frac{\pi_t(\theta_{t-1}^i)}{\pi_{t-1}(\theta_{t-1}^i)}. \tag{1.45}$$

Algorithm 5 SMC Sampler algorithm with resampling

Set Particle Size N
Set Number of Target Distributions T
for $i = 1 : N$ **do**
  $\theta_0^i \sim q_0(\cdot)$
  $\tilde{w}_0(\theta_0^i) = \pi_0(\theta_0^i)/q_0(\theta_0^i)$
end for
for $i = 1 : N$ **do**
  $w_0(\theta_0^i) = \tilde{w}_0(\theta_0^i)/\sum_{i=1}^{N} \tilde{w}_0(\theta_0^i)$
end for
if $1/\sum_{i=1}^{N} \left(w_0^i\right)^2 < N/2$ **then**
  Resample Particles under Stratified Resampling algoirthm
end if
for $t = 1 : T$ **do**
  for $i = 1 : N$ **do**
    $\theta_t^i \sim K_t(\cdot|\theta_{t-1}^i, \cdot)$
    $\tilde{w}_t^i = w_{t-1}^i \frac{\pi_t(\theta_t^i) L_{t-1}(\theta_{t-1}^i|\theta_t^i)}{\pi_{t-1}(\theta_{t-1}^i) K_t(\theta_t^i|\theta_{t-1}^i)}$
  end for
  for $i = 1 : N$ **do**
    $w_t^i = \tilde{w}_t^i / \sum_{j=1}^{N} \tilde{w}_t^j$
  end for
  if $1/\sum_{i=1}^{N} \left(w_t^i\right)^2 < N/2$ **then**
    Resample Particles under Stratified Resampling algorithm
  end if
end for

Under this sup-optimal kernel it is possible to adjust algorithm 5 such that we perform all the steps in the ordering of "Reweighting-Resampling-MCMCKernel", in which from (1.44) the MCMC kernel targets the distribution $\pi_t(\cdot)$, in comparison to "MCMCKernel-Reweighting-Resampling".

Regardless of including a resampling scheme in the SMC algorithm, particle degeneration can be quickly hastened due to other factors. One potential factor is caused by a large difference between distributions at each Markov state. Application dependent factors can also lead to higher variances, for example if the type of kernels cause small acceptance rates but with large jumps to different areas of probability. If particle degeneration occurs too quickly then resampling will also occur at a rapid rate, and if this restricts exploration of the parameter space of θ then only small non-overlapping subsets of high probability density will be visited. Regardless resampling should almost always be included, as in many situations it is impossible for a proposal distribution to greatly match a target distribution and thus large weight degeneracy is inevitable. The rate of particle degeneracy, shown through the ESS, as the algorithm progresses from start to finish should be analysed and we consider this as a diagnostic to access the quality of either the MCMC kernel and potentially the resampling scheme. However analysing the ESS is only appropriate on the condition that the posterior distribution has shown good convergence. An example would be to avoid having the density of a parameter focused on a single value because the kernel failed to explore the posterior.

On a quick note there also exists particle MCMC (PMCMC) (Andrieu *et al.*, 2010), and the basic concept of this algorithm is that it applies an SMC algorithm within an MCMC sampler. Each state considers an acceptance probability between the previous state of the chain and a weighted sampled particle from the SMC component of the algorithm.

1.7 Discussion

We have described the minimum statistical concepts that are necessary to understand how we develop a solution to our problem stated in section 1.1. Although we do not go into detail on model comparison techniques, these are instead discussed in chapter 2.

The most basic Monte Carlo simulation described in section 1.3, independent Monte Carlo sampling, requires that it is possible to simulate from the target distribution. An example is a posterior distribution derived from a conjugate prior. This does not make them appropriate for most Bayes' application problem where it is not possible to directly sample a distribution up to a normalisation constant. Importances sampling performs more effectively if the dimensional size is small and thus an importance distribution is easier to propose. Prior knowledge of where the target distribution has significant probability mass is required to construct an effective importance sampling, but this might not be available and thus MCMC methods may be a strong option. We would apply MCMC methods as a method to explore the full possible parameter space of each parameter to identify subsets of high probability density, either through single parameter moves or by transitioning the full parameter set to a new state within one move. Even when applying MCMC in high-dimensions, the chain can converge providing that the Markov chain is long enough and local exploration of the parameters is sufficient. Nevertheless devising an efficient proposal is difficult when multimodality exists within the target distribution, and how to devise MCMC to explore large probability valleys in the joint posterior can also be difficult to devise.

If multimodality is present then using sequential Monte Carlo sampler techniques has shown to be effective in practical applications, as we start from a long-tailed proposal which should at least sample all potential probability modes before eventually converging to a much narrower posterior (Paulin *et al.*, 2019). As stated previously,

SMC is also useful if we expect the complete data to arrive in subsets over time and prevents multiple MCMC based runs to be initiated for new data. We note that each described scheme has ever increasing computational complexity, and depending on which method to use depends on background knowledge on how complex the posterior distribution is expected to be.

While SMC samplers can be used in cases where observations are added one by one, the most general form of the algorithm does not consider inferring additional parameters for each observation added to the posterior. Although particle filters do consider new parameters to be inferred with each observation included, the applications we consider lack the same relationship between the parameters and previous states shown in (1.31). In chapter 2 we offer a proposed solution for our research questions in this thesis, which applies most of the stated methods in this chapter and across model methods which are again described in the next chapter.

Chapter 2

Model Comparison and Transformation Sequential Monte Carlo

The previous chapter described relevant background information for the investigation, consisting of some fundamentals of Bayesian statistics and Monte Carlo methods. Chapter 2 introduces our solution for the inferential problem described at the beginning of chapter 1. We call the method "transformation sequential Monte Carlo" (tSMC).

Within section 2.1 we first review methods for Bayesian model comparison. These algorithms either provide an estimate of the model posterior distribution of $\pi(m|y)$ or otherwise a Bayes factor (BF) (Jeffreys, 1998) used to compare two models. We give an introduction to reversible jump Markov chain Monte Carlo (RJMCMC) (Green, 1995) which allows for a jump from one statistical model to one with a different parameter space. This is through trans-dimensional proposals, or alternatively termed as "across model" moves, that make use of deterministic transformations. We explain an extension of the algorithm by Karagiannis and Andrieu (2013) which proposes an adaption that addresses one of the weaknesses of RJMCMC.

Section 2.2 describes the general form of tSMC. We state how this adaption of

a sequential Monte Carlo algorithm estimates a sequence of posterior distributions, of differing parameter space, by transitioning through each of these nested models via deterministic transformations similar to those in RJMCMC. The strengths and weaknesses of the tSMC approach, such as how well we expect tSMC to perform in high dimensional models, are also explained.

Finally section 2.3 describes the extensions to improve the efficiency of the basic tSMC algorithm.

2.1 Overview of Approaches to Bayesian Model Comparison

Suppose that we have a set of models $\{m_k\}_{k=1}^{K} \subset M$ with $\theta_{m_k} \in \Theta_{m_k}$ being the corresponding parameters for model m_k. We also consider a union of spaces $\cup_{m_k \in M}\{m_k\} \times \Theta_{m_k}$ where the sample space of $\Theta_{m_k} \subset \mathbb{R}^{d_{m_k}}$ may consist of differing dimensions $d_{m_k} \in \mathbb{N}$. Furthermore, depending on the algorithm we use to infer the posterior of each θ_{m_k}, we may define a model m_0 which usually represents proposals for model parameters of m_1.

The posterior distribution $\pi(m_k|y)$ is obtained by

$$\pi(m_k|y) = \frac{p(m_j)\int_{\Theta_{m_k}} f(y|\theta_{m_k}, m_k)p(\theta_{m_k}|m_k)d\theta_{m_k}}{\sum_{j=1}^{K}\Big(p(m_j)\int_{\Theta_{m_j}} f(y|\theta_{m_j}, m_j)p(\theta_{m_j}|m_j)d\theta_{m_j}\Big)}, \tag{2.1}$$

where $f(y|m_k, \theta_{m_k})$, $p(m_k)$ and $p(\theta_{m_k}|m_k)$ represents the likelihood given the data and the prior distributions for m_k and $\theta_{m_k}|m_k$ respectively. Naturally the highest $\pi(m_k|y)$ states the best model fit given the observational data. Alternatively to compare two models we may consider using the Bayes factor to see the evidence for favoring one

model m_i over the other m_j, which is given by

$$\begin{aligned} \text{BF} &= \frac{\pi(m_i|y)p(m_i)}{\pi(m_j|y)p(m_j)} \\ &= \frac{Z(y|m_i)}{Z(y|m_j)}, \end{aligned} \tag{2.2}$$

in which $Z(y|m_j)$ is the marginal likelihood given that the parameters from model m_j have been marginalised. The Bayes factors becomes the posterior odds when assuming a discrete uniform prior over the models. If BF < 1 then the weight of evidence in the Bayes factor is in favor of model m_j rather than model m_i and vice versa when BF > 1, but to what degree this can be considered as strong or weak evidence is up to personal interpretation. Some caution should be taken when using the marginal likelihood $Z(y|m_k)$ as a primary source of model comparison however, as it is sensitive to the joint prior distribution of the model parameters. For example given two models, which have parameters $\{\theta_1\}$ and $\{\theta_1, \theta_2\}$, then if the individual priors are uninformative this is likely to cause the smallest model to be favoured. Furthermore, as typical of the Bayes' formula, such a calculation of (2.1) is usually intractable unless conjugate priors are used. Algorithms to estimate this posterior or the BF are considered, but as discussed later each algorithm has their limitations regarding the accuracy and computational cost in estimating the true ML. We could also consider alternatives to the ML to compare models, for example we could calculate the Hyvärinen scores of the models (Hyvärinen, 2005), which can be estimated through SMC (Shao *et al.*, 2018), although this can only be applied to certain types of models.

In section 2.1.1 we state how $\pi(m_k|y)$, $\pi(y|m_k)$ or Bayes factors can be estimated from Monte Carlo output or by initiating a completely new algorithm. We also pay particular attention to nested models, which we later consider when defining our algorithm. In section 2.1.2 we give special attention to reversible jump MCMC algorithms in that gives an estimate of $\pi(m_k|y)$ by exploring the joint posterior of $\pi(m_k, \theta|y)$. We also state a few other algorithms that are perform a similar role, in terms of model

comparison, in section 2.1.4. Within most of this section we hide the conditionality on m_k when describing most of these algorithms, for simplicity, and only bring it up when relevant.

2.1.1 Direct Marginal likelihood Estimation

2.1.1.1 Standard Importance Sampling, AIS and SMC Methods

First we consider importance sampling based methods to estimate a marginal likelihood in which marginalising the model parameters $\theta \in \Theta$ is not possible. In chapter 1 we explained how the posterior can be estimated through importance sampling. Indeed, by using $g(\theta)$ as an importance sampling proposal we can also obtain a Monte Carlo estimate of the marginal likelihood given by

$$\begin{aligned} Z(y) &= \int_{\Theta} f\left(y \mid \theta\right) p\left(\theta\right) d\theta \\ &= \int_{\Theta} \frac{f\left(y \mid \theta\right) p\left(\theta\right)}{g\left(\theta\right)} g\left(\theta\right) d\theta \\ &= \mathrm{E}_g\left[\frac{f(y|\theta)p(\theta)}{g(\theta)}\right] \qquad (2.3) \\ \hat{Z}(y) &= \frac{1}{N}\sum_{i=1}^{N} \tilde{w}(\theta^i), \qquad (2.4) \end{aligned}$$

where we let $\tilde{w}(\theta^i) = f\left(y \mid \theta^i\right) p(\theta^i)/g(\theta^i)$ and $\theta^i \sim g(\cdot)$ for $i \in \{1, ..., N\}$. However, similar to the problem of estimating the posterior distribution, a low Monte Carlo variance of (2.4) will only occur if $g(\theta)$ is similar to the target distribution. This becomes harder to design in high-dimensional or complex models (Agapiou *et al.*, 2017; Gelman and Meng, 1998). For example we could use the prior distribution as an importance sampler in which the marginal likelihood may by estimated through

$$\begin{aligned} Z(y) &= \mathrm{E}_p\left[\frac{f(y|\theta)p(\theta)}{p(\theta)}\right] \qquad (2.5) \\ \hat{Z}(y) &= \frac{1}{N}\sum_{i=1}^{N} f(y|\theta^i), \qquad (2.6) \end{aligned}$$

however since the prior is usually much wider than the likelihood, unless again for a suitably high N, then it will lead to the underestimation of the ML (Newton and Raftery, 1994; Raftery *et al.*, 2007).

We refer back to chapter 1 where we discussed AIS which has proven to decrease the variance of the Monte Carlo estimate of the posterior or marginal likelihood estimate in comparison to a standard IS (Neal, 2001). Instead of one importance sampler we gradually transition through a series of T intermediate distributions of $\rho_t = (\eta(\theta_t))^{1-\varphi_t}(\pi_T(\theta_t))^{\varphi_t}$ for $(0 = \varphi_0 < \varphi_1 < ..... < \varphi_T = 1)$, where the set $(\theta_0..., \theta_t, ..., \theta_T)$ is generated through a series of kernels, such as a series of MH or Gibbs sampler kernels, in order to gradually move from usually a more wider proposal to narrower posterior space. Defining each of the unnormalised weights by $\tilde{w}^i = (\rho_1(\theta_0^i)/\rho_0(\theta_0^i)) \times ..... \times \left(\rho_T(\theta_{(T-1)}^i)/\rho_{T-1}(\theta_{(T-1)}^i)\right)$, the marginal likelihood is estimated by substituting the AIS weights into (2.4). Furthermore we state an algorithm that applies AIS very similarly in section 2.1.1.3.

SMC samplers, as described in chapter 1, follow from AIS where notably it considers resampling steps at certain states to prevent degeneracy in the particle set and due to this the formulation of the estimated marginal likelihood is a little different. In SMC we therefore consider

$$\begin{aligned} \frac{\widehat{Z_T(y)}}{Z_0(y)} &= \prod_{t=1}^{T} \frac{\widehat{Z_t(y)}}{Z_{t-1}(y)} \\ &= \prod_{t=1}^{T} \sum_{i=1}^{N} w_{t-1}(\theta_{t-1}^i)\tilde{w}_t(\theta_{(t-1):t}^i), \end{aligned} \tag{2.7}$$

where if MCMC kernels were used to generate each θ_t^i then the incremental weights are defined by $\tilde{w}_t(\theta_{(t-1):t}^i) = \pi_t(\theta_{t-1}^i)/\pi_{t-1}(\theta_{t-1}^i)$. Furthermore $w_{t-1}(\theta_{t-1}^i)$ are the normalised weights after assessing whether a resampling algorithm should be applied or not. Again Neal (2001) shows that if we were to use a series of annealed target distributions of $\pi_t(\theta|y, \varphi_t) \propto (\pi_T(\theta|y)^{\varphi_t}(p(\theta))^{1-\varphi_t}$, then (2.7) reduces down to an estimate of $\hat{Z}_T(y)$ as a normalised prior distribution has a normalising constant of 1.

2.1.1.2 Importance Sampling for Bayes Factors

We consider how importance sampling can be used to estimate the Bayes factors between two models that are nested. For example consider two posterior distributions that correspond to a low dimensional model m_1 with parameters $\{\theta_1\} \in \Theta_1$ which is subsetted to the other model m_2 with parameters $\{\theta_1, \theta_2\}$ (and $\theta_2 \in \Theta_2$). We can obtain a Bayes factor between the two models, providing that they are normalised, by considering the following relationship,

$$\begin{aligned}
\frac{Z(y|m_2)}{Z(y|m_1)} &= \frac{Z(y|m_2)}{Z(y|m_1)} \int_{\Theta_1 \times \Theta_2} \pi(\theta_1, \theta_2|y, m_2) d\theta_1 d\theta_2 \\
&= \frac{1}{Z(y|m_1)} \int_{\Theta_1 \times \Theta_2} f(y|\theta_1, \theta_2) p(\theta_1, \theta_2) d\theta_1 d\theta_2 \\
&= \frac{1}{Z(y|m_1)} \int_{\Theta_1 \times \Theta_2} \frac{f(y|\theta_1, \theta_2) p(\theta_1, \theta_2)}{q(\theta_2|\theta_1)\pi(\theta_1|y, m_1)} q(\theta_2|\theta_1)\pi(\theta_1|y, m_1) d\theta_1 d\theta_2 \\
&= \int_{\Theta_1 \times \Theta_2} \frac{f(y|\theta_1, \theta_2) p(\theta_1, \theta_2)}{q(\theta_2|\theta_1) f(y|\theta_1) p(\theta_1)} q(\theta_2|\theta_1)\pi(\theta_1|y, m_1) d\theta_1 d\theta_2. \qquad (2.8)
\end{aligned}$$

Thus by considering standard Monte Carlo theorem we have the estimate of the ratio given by

$$\widehat{\frac{Z(y|m_2)}{Z(y|m_1)}} = \frac{1}{N} \sum_{i=1}^{N} \tilde{w}^i, \qquad (2.9)$$

where $\tilde{w}^i = f(y|\theta_1^i, \theta_2^i) p(\theta_1^i, \theta_2^i) / q(\theta_2^i|\theta_1^i) f(y|\theta_1^i) p(\theta_1^i)$ and using the importance proposals $\theta_1^i \sim \hat{\pi}(\cdot|y, m_1)$, which is based on a Monte Carlo estimate of the posterior distribution $\pi(\cdot|y, m_1)$, and $\theta_2^i \sim q(\cdot|\theta_1^i)$. The main appeal of using (2.8) and (2.9) is that creating an importance proposal to estimate a low dimensional posterior of $\pi(\theta_1|y)$ is easier than designing a proposal for a higher dimensional posterior $\pi(\theta_1, \theta_2|y)$, and thus we could use a proposal for the non-nested parameters θ_2 that might be conditional on θ_1 (given $\hat{\pi}(\theta_1|y)$). Naturally the proposal $q(\theta_2|\theta_1)\hat{\pi}_1(\theta_1|y, m_1)$ still needs to be a close fitting match to $\pi_2(\theta_1, \theta_2|y, m_2)$ to avoid high variance estimates of (2.9). Furthermore (2.8) can be expanded to include AIS with target distributions of the

form,

$$\rho_t = (q(\theta_2|\theta_1)f(y|\theta_1)p(\theta_1))^{1-\varphi_t}(f(y|\theta_1,\theta_2)p(\theta_1,\theta_2))^{\varphi_t}. \quad (2.10)$$

An SMC approach with resampling is also possible.

2.1.1.3 Other Importance Sampling Approaches

While MCMC bypasses the calculation of the marginal likelihood to obtain a Monte Carlo estimate of the posterior distribution, the most common methods being Gibbs and MH samplers do not automatically give an estimate of the marginal likelihood by design. Nevertheless it is still possible to obtain this estimate by post-processing MCMC output, and one of the simplest methods that allows this is the harmonic mean of the likelihood. The harmonic mean estimator (HME) is formulated by considering that

$$\begin{aligned}\frac{1}{Z(y)} &= \frac{1}{Z(y)}\int_\Theta p(\theta)d\theta \\ &= \int_\Theta \frac{f(y|\theta)p(\theta)}{f(y|\theta)Z(y)}d\theta \\ &= \int_\Theta \frac{1}{f(y|\theta)}\pi(\theta|y)d\theta \\ &= \mathrm{E}_\pi\left[\frac{1}{f(y|\theta)}\right], \end{aligned} \quad (2.11)$$

providing that the prior is proper as an improper prior may lead to an infinite marginal likelihood (Baele *et al.*, 2012; Friel and Wyse, 2012). We consider an IS approach and use $\hat{\pi}(\theta|y)$, derived from MCMC, as our proposed importance sampler to obtain a sample of $\theta = \{\theta^1, \theta^2, ..., \theta^N\}$. Then a marginal likelihood estimate can be obtained by,

$$\hat{Z}(y) = \left(\frac{1}{N}\sum_{i=1}^{N}(f(y|\theta^i))^{-1}\right)^{-1}. \quad (2.12)$$

While being one of the more simplest methods that use MCMC output, N usually needs to be impossibly large for the estimate to be accurate. This is since we would be applying a narrow posterior as a proposal for a wide prior target distribution, and this leads to an underestimation of $(Z(y))^{-1}$ as we are not accounting for density in most of the probability space. Naturally, considering the reciprocal, the HME provides an overestimation of $Z(y)$. In addition the true marginal likelihood value is sensitive to the prior on θ, however the harmonic mean itself isn't (Friel and Wyse, 2012; Xie *et al.*, 2011).

We also take note of the "stepping-stone" algorithm, which is an approach introduced by Xie *et al.* (2011), and is very similar to AIS with regards to the algorithms use of annealed intermediate distributions although it does generate each θ_{t-1} differently. The methods strictly considers a series of posterior densities of $\pi_t(\theta|y,\varphi_{t-1}) \propto (\pi(\theta|y))^{\varphi_t}(p(\theta))^{1-\varphi_t} = (f(y|\theta))^{\varphi_t}p(\theta)$ and so the algorithm estimates $Z(y)$. A path from the prior to the posterior is considered, with multiple reweighting steps, and gives a marginal likelihood estimate of

$$\begin{aligned}\hat{Z}(y) &= \prod_{t=1}^{T}\widehat{\frac{Z_t(y)}{Z_{t-1}(y)}} \\ &= \prod_{t=1}^{T}\left(\frac{1}{N}\sum_{i=1}^{N}\frac{(f(y|\theta_{t-1}^i))^{\varphi_t}}{(f(y|\theta_{t-1}^i))^{\varphi_{t-1}}}\right) \\ &= \frac{1}{N}\prod_{t=1}^{T}\left(\sum_{i=1}^{N}(f(y|\theta_{t-1}^i))^{\varphi_t-\varphi_{t-1}}\right), \end{aligned} \tag{2.13}$$

which is identical to AIS when using the prior an importance proposal. Unlike AIS, at each state we sample N iterations of θ_{t-1} from a MCMC algorithm which targets $\pi_{t-1}(\theta|y,\varphi_{t-1})$.

We also describe "path-sampling", also termed as thermodynamic integration, in the application of Bayes formula via the algorithms proposed by Gelman and Meng (1998); Lartillot and Philippe (2006). This set of algorithms attempt to estimate $\log(Z(y|m_2)) - \log(Z(y|m_1))$ between two different models m_1 and m_2. We could

again consider an annealed scheme $(f(y|\theta)p(\theta))^{\varphi}(p(\theta))^{1-\varphi}$ like AIS which simplifies the log ratio to $\log(Z(y|m_2)) = Z(y)$ given a normalised prior, although other annealed schemes are possible should distribution m_1 not be of the form of a prior. We now show how under this scheme the integration over $[0, 1]$ is formulated by considering the first moment of

$$\begin{aligned} Z(y|\varphi) &= \int_{\Theta} f(y|\theta, \varphi)p(\theta|\varphi)d\theta \\ &= \int_{\Theta} (f(y|\theta))^{\varphi}p(\theta)d\theta, \end{aligned} \tag{2.14}$$

so $Z(y|m_1) = Z(y|\varphi = 0) = 1$ and $Z(y|m_2) = Z(y|\varphi = 1) = Z(y) = \int_{\Theta} f(y|\theta)p(\theta)d\theta$. This is gives us the stated ratio,

$$\begin{aligned} \log\left(\frac{Z(y|m_2)}{Z(y|m_1)}\right) &= \log(Z(y)) \\ &= \int_0^1 \frac{\partial\left(\log(Z(y|\varphi))\right)}{\partial\varphi}d\varphi \\ &= \int_0^1 \frac{1}{Z(y|\varphi)}\frac{\partial Z(y|\varphi)}{\partial\varphi}d\varphi \\ &= \int_0^1 \frac{1}{Z(y|\varphi)}\frac{\partial[\int_{\Theta} f(y|\theta, \varphi)p(\theta|\varphi)d\theta]}{\partial\varphi}d\varphi \\ &= \int_0^1 \left(\int_{\Theta} \frac{\tilde{\pi}(y, \theta|\varphi)}{Z(y|\varphi)}\frac{1}{\tilde{\pi}(y, \theta|\varphi)}\frac{\partial f(y|\theta, \varphi)p(\theta|\varphi)}{\partial\varphi}d\theta\right)d\varphi \\ &= \int_0^1 \left(\int_{\Theta} \pi(\theta|y, \varphi)\frac{\partial\log(f(y|\theta, \varphi)p(\theta|\varphi))}{\partial\varphi}d\theta\right)d\varphi \\ &= \int_0^1 \mathrm{E}_{\pi_{\varphi}}\left[\frac{\partial\log\left(f(y, \theta|\varphi)p(\theta|\varphi)\right)}{\partial\varphi}\right]d\varphi, \end{aligned} \tag{2.15}$$

where $\tilde{\pi}(y, \theta|\varphi) = f(y, \theta|\varphi)p(\theta|\varphi)$ is the unnormalised posterior distribution (conditional on some tuning value φ, $0 < \varphi < 1$), $Z(y|\varphi)$ is the marginal likelihood corresponding to the said unnormalised posterior and $\mathrm{E}_{\pi_{\varphi}}$ is the expectation with respect to $\pi(\theta|y, \varphi)$ (Lartillot and Philippe, 2006). The adaption by Gelman and Meng (1998) sets a prior on φ, for example Unif$(0, 1)$, and considered estimating the log

Bayes factor in (2.15) via

$$\log(Z(y)) = \mathrm{E}_{\tilde{\pi}_\varphi}\left[\frac{\partial \log(f(y|\theta,\varphi)p(\theta|\varphi))}{\partial\varphi}\frac{1}{p(\varphi)}\right], \tag{2.16}$$

where $\mathrm{E}_{\tilde{\pi}_\varphi}$ is the expectation with respect to $p(\varphi)\times\pi(\theta|y,\varphi)$. Therefore an estimate of the Bayes factor is then given by,

$$\widehat{\log(Z(y))} = \frac{1}{N}\sum_{j=1}^{N}\frac{1}{p(\varphi^j)}\frac{\partial \log(f(y|\theta^j,\varphi^j)p(\theta^j|\varphi^j))}{\partial\varphi^j}, \tag{2.17}$$

where we could draw N number of samples of φ and θ from $p(\varphi)\times\pi(\theta|y,\varphi)$. Overall it is not appropriate to plan on drawing from all possible models that are conditional on the complete continuous variable set of φ especially if we depend on MH based algorithms to estimate the form of $\pi(\theta|y,\varphi)$. Furthermore we do require that the set of φ^j is spread out fairly evenly across $[0,1]$, otherwise we fail to cover the full probability space sufficiently.

In Lartillot and Philippe (2006) they instead consider a fixed sequence $(\varphi_0 = 0,\varphi_1,...,\varphi_T = 1)\subset[0,1]$ in which they consider a model m_0, which contains prior assumptions of model m_T representing the posterior $\pi(\theta|y)$, in comparison to a set of sampled φ. They perform individual Markov chain Monte Carlo runs targeting each of the posteriors to collect N samples from each target. We then estimate each

$$U_t = \mathrm{E}_{\pi_\varphi}\left[\frac{\partial \log\left(f(y,\theta|\varphi_t)p(\theta|\varphi_t)\right)}{\partial\varphi_t}\right] \tag{2.18}$$

$$\tilde{U}_t = \frac{1}{N}\sum_{i=1}^{N}\frac{\partial \log\left(f(y,\theta^i|\varphi_t)p(\theta|\varphi_t)\right)}{\partial\varphi_t}. \tag{2.19}$$

Finally, as we have an integral between 0 and 1, Simpson's triangulation gives an estimate of the log ratio via,

$$\widehat{\log(Z(y))} = \frac{1}{T}\left(\frac{1}{2}\tilde{U}_0 + \sum_{t=1}^{T-1}\tilde{U}_t + \frac{1}{2}\tilde{U}_T\right). \tag{2.20}$$

While this form of path-sampling by Lartillot and Philippe (2006) is more flexible,

the use of Simpson's triangulation makes (2.20) suffer from discretisation bias as it is approximating a continuous integral by using a finite sequence of points. This bias decreases with increasing T and each $\{\varphi_{t-1}, \varphi_t\}$ being appropriately spaced.

There has been some applied evidence that annealed importance sampling does perform better than at least the harmonic mean estimator (Friel and Wyse, 2012). What Baele *et al.* (2012) concluded is that the stepping stone algorithm, despite the higher computational cost, offered improvements in the error bound of ML estimates over the harmonic mean estimator and path-sampling algorithm (and furthermore doesn't suffer from discretisation bias). Otherwise the path sampling algorithm performs better than the harmonic mean estimator, which includes other versions such as the stabilised HME, in terms of how their algorithm gave the correct ML in multiple applications (Baele *et al.*, 2012; Friel and Pettitt, 2008).

While the estimation of the marginal likelihood given in (2.13) has similarities with the AIS/SMC calculation of the marginal likelihood, the stepping-stone algorithm requires that each θ_{t-1} is estimated through MCMC. In comparison, a standard SMC algorithm given by Del Moral *et al.* (2006) already estimates and samples θ_{t-1} by design and thus is computationally faster. Therefore an AIS or SMC approach would be considered a superior option to some of these popular choices which apply MCMC output.

2.1.2 Across Model Transitions and Reversible Jump MCMC (RJMCMC) Algorithms

We may desire to estimate each $\pi(m_k|y)$ by exploring the posterior on model space. We could use a Metropolis-Hastings algorithm that allows for a proposal to jump to a different model. Referring back to the acceptance probability stated in chapter 1 for the MH algorithm, the acceptance ratio in this case would be

$$\alpha(m^{i-1}, \tilde{m}^i) = \min\left\{1, \frac{Z(y|\tilde{m}^i)p(\tilde{m}^i)q(m^{i-1}|\tilde{m}^i)}{Z(y|m^{i-1})p(m^{i-1})q(\tilde{m}^i|m^{i-1})}\right\}, \tag{2.21}$$

where $\tilde{m}^i \sim q(\cdot|m^{i-1})$. Clearly we cannot implement the above because we cannot define the true value of $Z(y|m)$, and if we did know these values then then there is no need to perform such an MCMC in the first place since model comparison can just be done using the marginal likelihoods. Instead we consider pseudo marginal Metropolis Hastings (PMMH) methods (see for example Andrieu and Roberts (2009)) in which if some distribution cannot be evaluated then an unbiased estimator may be used in its place in a standard Metropolis Hastings algorithm. Here we use a slightly different pseudo-marginal approach to the one in Andrieu and Roberts (2009), where instead we consider an unbiased estimator of the acceptance probability as described in Karagiannis and Andrieu (2013). One potential estimator is an approximation based on importance sampling. If we wish to use an importance sampling estimate for the ratio $Z(y|\tilde{m}^i)/Z(y|m^{i-1})$ where $\tilde{m}^i$ is a higher dimensional model then model m^{i-1}, we can use the nested models based importance sampling estimate in (2.8). If we define the two models to be $\pi(\theta_1|y, m^{i-1})$ and $\pi(\theta_1, \theta_2|y, \tilde{m}^i)$ then by substituting an importance sampling estimate of $Z(y|\tilde{m}^i)/Z(y|m^{i-1})$, based on (2.8) with only one particle in said importance sampler, into (2.21) we receive

$$\begin{aligned}\alpha(m^{i-1}, \tilde{m}^i) &= \min\left\{1, \frac{Z(y|\tilde{m}^i)p(\tilde{m}^i)q(m^{i-1}|\tilde{m}^i)}{Z(y|m^{i-1})p(m^{i-1})q(\tilde{m}^i|m^{i-1})}\right\} \\ &\approx \min\left\{1, \frac{f(y|\theta_1, \theta_2, \tilde{m}^i)p(\theta_1, \theta_2|\tilde{m}^i)p(\tilde{m}^i)}{q(\theta_2|\theta_1, m^{i-1})f(y|\theta_1, m^{i-1})p(\theta_1|m^{i-1})p(m^{i-1})}\right. \\ &\quad \left.\times\frac{q(m^{i-1}|\tilde{m}^i)}{q(\tilde{m}^i|m^{i-1})}\right\}. \end{aligned} \tag{2.22}$$

This is essentially the groundwork for Reversible jump Markov chain Monte Carlo ((RJMCMC), see for example Green (1995); Hastie and Green (2012); Richardson and Green (1997)), an across-model Markov chain algorithm that allows us to perform trans-dimensional moves from the parameter space on some model $m_k \in M$ to the new space within a different model $m_{k'}$. The algorithm introduced in Green (1995) explains that the acceptance probability of (2.22) results in an MCMC algorithm that

targets the posterior distribution on model space. However using multiple importance points in (2.22) does not correctly target the posterior distribution unless the number of particles is infinite (Alquier *et al.*, 2016) but, as established in Green (1995), the acceptance probability in (2.22) which uses a single importance point does in fact result in the correct posterior distribution.

We now describe the general RJMCMC algorithm and how it extends the approach in (2.22). For simplicity we assume that each model m_k has a direct transformation to model $m_{k'}$ within some index sequence of models. Furthermore we work on the disjoint union of spaces of $\cup_{m_k \in M}\{m_k\} \times \Theta_{m_k}$, and $\theta_{m_k} \in \Theta_{m_k} \subset \mathbb{R}^{d_{m_k}}$ (with $d_{m_k} \in \mathbb{N}$) are the parameters for each corresponding m_k. Given that the dimensions of $\mathbb{R}^{d_{m_k}}$ and $\mathbb{R}^{d_{m_{k'}}}$ differ then to jump between m_k and $m_{k'}$ we may require drawing from a vector of random variables. We consider $u_{m_k} \in U_{m_k \to m_{k'}} \subset \mathbb{R}^{r_{m_k}}$, with density $\psi_{m_k \to m_{k'}}(\cdot)$, which is required to transition to $m_{k'}$ and $u_{m_{k'}} \in U_{m_{k'} \to m_k} \subset \mathbb{R}^{r_{m_{k'}}}$, with density $\psi_{m_{k'} \to m_k}(\cdot)$, used to transition back to model m_k. We assume that the normalisation constants of $\psi_{m_k \to m_{k'}}(u_{m_k})$ and $\psi_{m_{k'} \to m_k}(u_{m_{k'}})$ are known and these densities can always be evaluated. These two auxiliary variable sets are needed to give full posterior exploration of both m_k and $m_{k'}$ and must be set such that it meets the dimension matching criterion of $d_{m_k} + r_{m_k} = d_{m_{k'}} + r_{m_{k'}}$.

Furthermore we apply a deterministic function, $h : \mathbb{R}^{d_{m_k}} \times \mathbb{R}^{r_{m_k}} \to \mathbb{R}^{d_{m_{k'}}} \times \mathbb{R}^{r_{m_{k'}}}$, on $\{\theta_{m_k}, u_{m_k}\}$ to give the transformed sample of $\{\theta_{m_{k'}}, u_{m_{k'}}\}$. When the previously mentioned dimension matching criterion is met then h is bijective and its inverse is differentiable (a diffeomorphism), a required condition to use RJMCMC since we are making a transformation on the parameter space. In our illustrations we define the Jacobian of the transformation as

$$\begin{aligned} J_{m_{k'} \to m_k} &= J^{-1}_{m_k \to m_{k'}} \\ &= \left(\left| \frac{\partial h(\theta_{m_k}, u_{m_k})}{\partial (\theta_{m_k}, u_{m_k})} \right| \right)^{-1}. \end{aligned} \tag{2.23}$$

There may be the case where the new parameters for $m_{k'}$ are generated directly from $u_{m_{k'}} \in U_{m_{k'} \to m_k} \subset \mathbb{R}^{r_{m_{k'}}}$ with density $\psi_{m_{k'} \to m_k}(\cdot)$ such that the identity transformation $\theta_{m_{k'}} = \{\theta_{m_k}, u_{m_{k'}}\}$ is applied with Jacobian equal to 1 as in equation (2.22). The general case of RJMCMC we have presented also corresponds to using an importance sampling estimators for the acceptance probability, with the addition of a transformation to produce the proposal distribution. The general form of of RJMCMC is shown in algorithm 6. We could perform a kernel move K_{m_k} that applies, for example, say a standard MCMC proposal on the parameters of the current model m_k either if the proposed model to jump to is the current model (which is what is considered in algorithm 6) or apply a kernel after an across model move has been made.

An estimate of a marginal density of $\pi(m_k|Y)$ is obtained via the proportion that the Markov chain was within model m_j, same as a standard Monte Carlo estimate. What we presented in (2.22) is a special case of RJMCMC where we strictly apply the identity transformation and we use nested models. While the general RJMCMC also uses a single point IS estimator the models do not need to be nested and a deterministic transformation is applied. However the efficiency of RJMCMC is dependent on the choice of u_{m_k} and $u_{m_{k'}}$ and the associated transformations to transition between models m_k and $m_{k'}$. If poor choices are made then the transformation will give a high variance IS estimate of the acceptance probability will be made, resulting in an inefficient MCMC and may fail to explore the complete model space. This is particularly an issue if each model is high dimensional since this increases the variance of the estimate of the acceptance probability. A potential option is to use some adaptive form of across model transformations (for a general review see Brooks *et al.* (2003); Hastie (2005); Hastie and Green (2012); Sisson (2005)), however we do not consider such modifications.

Algorithm 6 Reversible Jump MCMC Algorithm.

Set N chain length
$m_k^{(0)} \sim q(\cdot)$
$\theta_{m_k}^{(0)} \sim K(\cdot)$
for $i = 1 : N$ **do**
  $\tilde{m}_{k'}^{(i)} | m_k^{(i-1)} \sim q(m_k^{(i-1)}, \cdot)$
  if $\tilde{m}_{k'}^{(i)} = m_k^{(i-1)}$ **then**
    $m_{k'}^{(i)} = \tilde{m}_{k'}^{(i)}$
    $\theta_{m_k}^{i} \sim K_{m_k}(\cdot | \theta_{m_k}^{(i-1)})$
  else if $\tilde{m}_{k'}^{i} \neq m_k^{(i-1)}$ **then**
    $u_{m_k}^{i} \sim \psi_{m_k \to m_{k'}}(\cdot)$
    $\{\tilde{\theta}_{m_{k'}}^{i}, \tilde{u}_{m_{k'}}^{i}\} = h(\theta_{m_k}^{(i-1)}, u_{m_k}^{i})$
    $\alpha_{m_k \to m_{k'}}(\theta_{m_k}^{(i-1)}, \tilde{\theta}_{m_{k'}}^{(i)}) = \min \left\{ 1, \frac{\pi(\tilde{m}_{k'}^{i}, \tilde{\theta}_{m_{k'}}^{i} | y) \psi_{m_{k'} \to m_k}(\tilde{u}_{m_{k'}}^{i})}{\pi(m_k^{(i-1)}, \theta_{m_k}^{(i-1)} | y) \psi_{m_k \to m_{k'}}(u_{m_k}^{i}) J_{m_{k'} \to m_k}} \right.$
    $\left. \times \frac{q(m_k^{(i-1)} | \tilde{m}_{k'}^{i})}{q(\tilde{m}_{k'}^{i} | m_k^{(i-1)})} \right\}$
    $u \sim \text{Unif}(0,1)$
    if $u \leq \alpha_{m_k \to m_{k'}}(\theta_{m_k}^{(i-1)}, \tilde{\theta}_{m_{k'}}^{(i)})$ **then**
      $m_k^{i} = \tilde{m}_{k'}^{i}$
      $\theta_{m_k}^{i} = \tilde{\theta}_{m_{k'}}^{i}$
    else
      $m_k^{i} = m_k^{(i-1)}$
      $\theta_{m_k}^{i} = \theta_{m_k}^{(i-1)}$
    end if
  end if
end for

2.1.3 Annealed Importance Sampling Reverse Jump MCMC

A solution to the problem stated at the end of the previous section is given by a modified RJMCMC algorithm by Karagiannis and Andrieu (2013), termed annealed importance sampling reversible jump MCMC (AIS-RJMCMC), where the basic premise of the algorithm is that instead of using an importance sampling update shown in (2.22) we instead consider an AIS based unbiased estimator. Similar to AIS or SMC we have a time parameter $t \in (0, ..., T)$ where $T \in \mathbb{N}$, and in this scenario we consider the total number of intermediate distributions to transition from model m_k to model $m_{k'}$. Only one particle is used within the AIS as we would be substituting the particle into a MH acceptance probability, similar to what is shown in (2.22).

We clarify that from here we start to start to consider a parameter $\theta^i_{m_k t}$ that belongs to model m_k, is part of the ith particle and we are at time t of some process. We may choose to extend this to include the jth dimension of a parameter by defining $\theta_{m_k jt}$, see chapter 3 where we use this notation. This is especially important as we use this notation in our proposed solution in section 2.2. Although in the case of AIS-RJMCMC we only have the one particle being $\theta_{m_k t}$.

A series of forward annealing densities of $\rho_t(\theta_{m_{k'}t}, u_{m_{k'}t}; m_k \to m_{k'})$ is defined, and this may also be expressed in the form of backward annealing densities but for simplicity our investigation considers these densities under the forward case. Like AIS the idea is then to transition from an initial posterior distribution representing the current model m_k of

$$\rho_0(\theta_{m_{k'}}, u_{m_k}; m_k \to m_{k'}) \quad \propto \quad \pi(m_k, \theta_{m_k 0}|y)\psi_{m_k \to m_{k'}}(u_{m_k 0})J_{m_{k'} \to m_k}, \tag{2.24}$$

to a target distribution representing model $m_{k'}$ of

$$\rho_T(\theta_{m_{k'}} u_{m_{k'}}; m_k \to m_{k'}) \quad \propto \quad \pi(m_{k'}, \theta_{m_{k'}T}|y)\psi_{m_{k'} \to m_k}(u_{m_{k'}T}). \tag{2.25}$$

What is noticeable is how (2.24) and (2.25) represent the numerator and denominator of an RJMCMC acceptance probability respectively, with the auxiliary variables and the Jacobian that results from transforming the parameter space also defined in algorithm 6 and section 2.1.2.

AIS-RJMCMC adapts algorithm 6 where after $\{\theta_{m_{k'}}, u_{m_{k'}}\}$ is generated, a series of kernels that target a set of annealed intermediate distributions is applied in order to obtain $\{\theta_{m_{k'}T}, u_{m_{k'}T}\}$ representing of $m_{k'}$. Thus starting with $\{\theta_{m_{k'}0}, u_{m_{k'}0}\}$, we generate a path that sequentially moves through the set of $(\rho_1(\theta_{m_{k'}1}, u_{m_{k'}1}), ..., \rho_T)$. Each $\{\theta_{m_{k'}t}, u_{m_{k'}t}\}$ is generated from each of the corresponding transition kernels of $K_t(\theta_{m_{k'}(t-1)}, u_{m_{k'}(t-1)}, m_k \to m_{k'})$, for example each K_t might be a series of MCMC moves. If the kernel was designed to have have the reversibility and symmetry and conditions (for example, Metropolis Hastings kernels), then the acceptance probability in algorithm 6 is then defined by

$$\alpha^{(0:T)}_{m_k \to m_{k'}} \equiv \frac{q(m_k|m_{k'}, \cdot)}{q(m_{k'}|m_k, \cdot)} \prod_{t=1}^{T} \frac{\rho_t(\theta_{m_{k'}(t-1)}, u_{m_{k'}(t-1)}; m_k \to m_{k'})}{\rho_{t-1}(\theta_{m_{k'}(t-1)}, u_{m_{k'}(t-1)}; m_k \to m_{k'})}, \tag{2.26}$$

and we consider (2.26) in particular for our proposed algorithm. A recommended scheme for the intermediate distributions is to use geometric averages, similar to what is suggested for AIS, with an annealed sequence of $\varphi_t = (t/T)^R$ for $R \in \mathbb{N}$. Each annealed intermediate distribution is defined by

$$\begin{aligned} \rho_t(\theta_{m_{k'}t}, u_{m_{k'}t}; m_k \to m_{k'}) \propto\ & (\pi(m_k, \theta_{m_kt}|y)\psi_{m_k \to m_{k'}}(u_{m_kt})J_{m_{k'} \to m_k})^{1-\varphi_t} \\ & \times(\pi(m_{k'}, \theta_{m_{k'}t}|y)\psi_{m_{k'} \to m_k}(u_{m_{k'}t}))^{\varphi_t}, \end{aligned} \tag{2.27}$$

where $\varphi_0 = 0$, $\varphi_T = 1$ and each φ_t may be evenly spaced or may follow some function of t (such as geometric spacing). As stated with AIS, a geometric scheme allows to transition from (2.24) to (2.25) in a smooth manner by asserting more initial power on the posterior parameter space of model m_k which reduces the weighted impact of an inefficient transformation proposal and allows for the proposal to explore the

parameter space through the Markov transition moves.

2.1.4 Other Past Approaches for ML Estimation or Exploration of Joint Model Space

We give a brief explanation of other approaches which consider trans-dimensional space. Less general and more application specific approaches that are similar to tSMC are displayed in chapters 3-5 respectively.

In Jasra *et al.* (2008) they define a standard SMC algorithm which applies MH or Gibbs kernels to the parameters of some model, similar to what we introduced in chapter 1, however the kernel also includes a RJMCMC-like proposal to jump to a new model. They performed this base adaption within their "interacting sequential Monte Carlo samplers" algorithm which considers using parallel samplers, or simultaneous SMC samplers, for some defined number of states in the Markov chain. The most notable feature is that each sampler can be constrained to explore a specific subset of models, so for example one sampler could explore the joint parameter space of the three highest defined dimensional models while another sampler could explore the three lowest dimensional models. This allows for a more effective exploration of the space of the models. Once a certain number of states have been completed for all samplers it is subjected to one final kernel/reweighting/resampling step. Note the applied kernel at this state is set to be identical for all parallel runs, such that all particles share the same parameter space. After performing another identical kernel they then sample particles from all the runs and use a single SMC sampler for the remainder of the algorithm. However the issue regarding if a RJMCMC move can successfully transition between high-dimensional models remains unchanged in this algorithm, with the most safest scenario in their algorithm being to dedicate a SMC sampler to each specific model.

Zhou *et al.* (2016) presented SMC-1 which is very similar to the non-parallel SMC sampler by Jasra *et al.* (2008) which we just discussed. It is a SMC algorithm,

as described in chapter 1, that uses an annealed series of intermediate distribution where again one of the kernels takes the form of a RJMCMC proposal. They also gave an alternative algorithm being SMC-3 where while it is a SMC algorithm with annealed distributions, these intermediate distributions differ from the usual geometric annealed/bridged scheme as seen with AIS. In this algorithm, instead of jumping to a random model, they consider exploring a sequential set of models of $(m_0, m_1, ..., m_K)$. Assuming that we start with parameters sampled from $\pi(\cdot|m_{k-1})$, each particle considers the basic unnormalised posterior model, conditional on the model type, multiplied by a prior on the models. This prior takes two values of m_{k-1} and m_k such that as $t \to T$ then the prior will gradually favor m_k, i.e $Pr(m_k|t) = \eta(t/T)$ for some increasing bijection $\eta : [0, 1] \to [0, 1]$, and thus all particles will be a representative sample of m_k when $t = T$. The kernels naturally incorporate RJMCMC proposals, where if a particle is currently in one model then a RJMCMC move is proposed to transition to its pairwise model and vice versa.

Persing *et al.* (2015) used a variation of the particle MCMC to transition between models, where as a reminder PMCMC runs a SMC algorithm within a MCMC algorithm where each state considers an acceptance probability between the previous state of the chain and a weighted sampled particle from the SMC component of the algorithm (Andrieu *et al.*, 2010). The main feature in their PMCMC algorithm is how it starts by first sampling a model from some distribution to jump to, however all the parameters are sampled from their respective conditional distributions (their prior distributions are also another option). They then apply a standard SMC algorithm that applies geometric based intermediate distributions to explore the parameter space of the sampled model. In comparison to the other RJMCMC described methods, they do not transform a current set of parameters in order to transition between models, but instead from some easy to sample proposal distribution.

2.2 Transformation SMC, Algorithm Properties and Justifications

In this section we now introduce transformation SMC and how it is used for Bayes' model comparison, and other scenarios, in section 2.2.1. We go in depth of the advantages and disadvantages of this approach in sections 2.2.2 and 2.2.3.

2.2.1 The tSMC Algorithm

Unlike the AIS-RJMCMC algorithm discussed in section 2.1.3 which considers an importance sampler of nested models as an estimator to a MCMC acceptance probability, we instead use a pure SMC sampler which uses ideas from RJMCMC. As an adaption of the SMC sampler it is possible to obtain not only the weighted points of a posterior distribution for a set of models, but also estimate the corresponding marginal likelihood of the model. If we were using MCMC then we would need to run additional algorithms such as the stepping stone algorithm. We wish to estimate the posterior distributions of models $(m_0, m_1, ..., m_K) \subset M$, where there is some natural ordering of the model space. We desire to infer up to the highest dimensional model of m_K. The difference between each m_k would usually be the number of parameters, but most importantly there must exist deterministic transformations between each adjacent model. While we explain the algorithm in terms of Bayesian model comparison, it is possible to apply the algorithm in other scenarios, such as data point tempering (see chapter 4), by defining a different sequence of models that differ by the size of the observations that they are modeling. Alternatively, in chapter 5 we use tSMC in a scenario where the total number of parameter has no linear relationship with the observations, but an increase in the observation size will increase the discrete parameter space for a subset of parameters (in which we explain our justifications for our approach in said chapter). The algorithm will work best given that the difference between each m_{k-1} and m_k is small and each model is nested within a successive

model, such that a subset of nested parameters in model m_k has marginal posteriors that are slightly different from the same parameters in m_{k-1}.

Transformation SMC applies a similar sequence of target distribution from Karagiannis and Andrieu (2013) where we wish to estimate the posterior $\pi(\theta_{m_k}|y, m_k)$ by using the set of particles from $\pi(\theta_{m_{k-1}}|y, m_{k-1})$ as part of an importance sampler. As model m_k differs by a few parameters we need to transform the parameter set $\theta_{m_{k-1}}$ so that it is in the new parameter space. This could be achieved by either generating the missing parameters from some distribution $\psi_{m_{k-1}\to m_k}(\cdot)$ and applying an identity transformation. Alternatively we apply some transformation to the existing parameters, and define $\psi_{m_{k-1}\to m_k}(\cdot)$ and $\psi_{m_k\to m_{k-1}}(\cdot)$ to ensure the dimension matching criterion is met.

For the rest of the chapter, and thesis, we refer to the set $\{\theta_{m_k}, u_{m_k}\}$ as a result of using a transformation on $\{\theta_{m_{k-1}}, u_{m_{k-1}}\}$. Overall we define the importance proposal for the parameters of model m_k by

$$\pi(\theta_{m_{k-1}}, m_{k-1})\psi_{m_{k-1}\to m_k}(u_{m_{k-1}})J_{m_k\to m_{k-1}}, \tag{2.28}$$

where $\pi(\theta_{m_{k-1}}, m_{k-1})$ is the unnormalised posterior of model m_{k-1}, $u_{m_{k-1}}$ are the auxiliary variables related to the transformation to m_k and $J_{m_k\to m_{k-1}}$ is the Jacobian of the inverse transformation. The sampler is used to aim to infer a target distribution

$$\pi(\theta_{m_k}, m_k)\psi_{m_k\to m_{k-1}}(u_{m_k}), \tag{2.29}$$

where we still define the auxiliary variables u_{m_k} for the inverse of the transformation to be part of the target distribution. We choose to gradually converge to m_k by using a series of annealed intermediate distributions similar to (2.27) being

$$\begin{aligned}\rho_t &= \left(\pi(\theta_{m_{k-1}t}, m_{k-1})\psi_{m_{k-1}\to m_k}(u_{m_{k-1}t})J_{m_k\to m_{k-1}}\right)^{1-\varphi_t} \\ &\quad\times\left(\pi(\theta_{m_kt}, m_k)\psi_{m_{k-1}\to m_k}(u_{m_kt})\right)^{\varphi_t},\end{aligned} \tag{2.30}$$

such that

$$\rho_0 = \pi(\theta_{m_{k-1}0}, m_{k-1})\psi_{m_{k-1}\to m_k}(u_{m_{k-1}0})J_{m_k\to m_{k-1}} \tag{2.31}$$

$$\rho_T = \pi(\theta_{m_kT}, m_k)\psi_{m_k\to m_{k-1}}(u_{m_kT}). \tag{2.32}$$

We consider the particle set of $\{\theta_{m_kT}, u_{m_kT}\}$ to represent the MC estimate of the posterior distribution for the parameters of model m_k.

The most basic form of the tSMC algorithm considers the current schedule time t, with $\varphi_t = \{\varphi_0, \varphi_1, ..., \varphi_T\}$ where $\varphi_0 = 0$ and $\varphi_T = 1$, between m_{k-1} and m_k with corresponding normalised weights of $w_{m_kt} \propto \rho_t/\rho_{t-1}$. Thus our algorithm takes the form of an annealed SMC algorithm, see chapter 1. Naturally when making a new transition from model m_k, after the previous transition is completed once $\varphi_T = 1$, we reset $t = 0$. Furthermore the only type of kernel we apply in tSMC is a MCMC kernel, which simplifies the unnormalised weight calculation to

$$\begin{aligned}
\tilde{w}_{m_jt} &= w_{m_k(t-1)}\frac{\rho_t(\theta_{m_k(t-1)}, u_{m_k(t-1)}; m_{k-1}\to m_k)}{\rho_{t-1}(\theta_{m_k(t-1)}, u_{m_k(t-1)}; m_{k-1}\to m_k)} \\
&= w_{m_k(t-1)}\frac{(\pi(m_{k-1}, \theta_{m_{k-1}(t-1)})\psi_{m_{k-1}\to m_k}(u_{m_{k-1}(t-1)})J_{m_k\to m_{k-1}})^{1-\varphi_t}}{(\pi(m_{k-1}, \theta_{m_{k-1}(t-1)})\psi_{m_{k-1}\to m_k}(u_{m_{k-1}(t-1)})J_{m_k\to m_{k-1}})^{1-\varphi_{t-1}}} \\
&\quad\times\frac{(\pi(m_k, \theta_{m_k(t-1)})\psi_{m_k\to m_{k-1}}(u_{m_k(t-1)}))^{\varphi_t}}{(\pi(m_k, \theta_{m_k(t-1)})\psi_{m_k\to m_{k-1}}(u_{m_k(t-1)}))^{\varphi_{t-1}}}.
\end{aligned} \tag{2.33}$$

Just like an annealed scheme the first few intermediate distributions will initially favor the joint parameter space of the posterior of m_{k-1} along with any auxiliary variables and the Jacobian of the initiated transformation in (2.31) before gradually favoring the posterior of model m_k defined by (2.32). We also apply resampling and we initiate it should the ESS drop too low.

The initial model m_0 can be set to be the prior distribution for the parameters in m_1, in which we then can obtain a series of marginal likelihood estimates of $(Z(y|m_1), ..., Z(y|m_k))$ as explained in previously sections. Otherwise the algorithm

instead estimates a set of Bayes factor comparing a sequence of models instead.

Exploration of the parameter space is made by applying at least one MCMC kernel, after a resampling step, of $K_t(\cdot|\theta_{m_k(t-1)}, u_{m_k(t-1)})$ which target ρ_t as its stationary distribution. We specifically use MCMC kernels on the parameter set $\{\theta_{m_k t}, u_{m_k t}\}$, and then apply the reverse transformation to receive $\{\theta_{m_{k-1} t}, u_{m_{k-1} t}\}$. We accept a kernel proposal with acceptance probability of

$$\begin{aligned}\alpha(\theta^i_{m_k(t-1)}, \tilde{\theta}^i_{m_k t}) &= \min\Bigg\{1, \frac{(\pi(m_{k-1}, \tilde{\theta}^i_{m_{k-1} t})\psi_{m_{k-1}\to m_k}(\tilde{u}^i_{m_{k-1} t}))^{1-\varphi_t}}{(\pi(m_{k-1}, \theta^i_{m_{k-1}(t-1)})\psi_{m_{k-1}\to m_k}(u^i_{m_k(t-1)}))^{1-\varphi_t}} \\ &\quad \times \frac{(\tilde{J}_{m_k\to m_{k-1}})^{1-\varphi_t}(\pi(m_k, \tilde{\theta}^i_{m_k t})\psi_{m_k\to m_{k-1}}(\tilde{u}^i_{m_k t}))^{\varphi_t}}{(J_{m_k\to m_{k-1}})^{1-\varphi_t}(\pi(m_k, \theta^i_{m_k(t-1)})\psi_{m_k\to m_{k-1}}(u^i_{m_k(t-1)}))^{\varphi_t}} \\ &\quad \times \frac{q(\theta^i_{m_k(t-1)}|\tilde{\theta}^i_{m_k t})}{q(\tilde{\theta}^i_{m_k t}|\theta^i_{m_k(t-1)})}\Bigg\}, \end{aligned} \tag{2.34}$$

where $q(\theta^i_{m_k(t-1)}|\tilde{\theta}^i_{m_k t})/q(\tilde{\theta}^i_{m_k t}|\theta^i_{m_k(t-1)})$ is our proposal ratio given that we are using one MH update.

The basic tSMC algorithm is shown in algorithm 7, note that we go by the assumptions that we are not starting from an existing run of the algorithm.

2.2.2 Justification

We now show that tSMC is a special case of a standard (fixed-dimensional) SMC sampler through defining a sequence of target distributions that results in the weight update as described in the previous section.

For simplicity we begin with the case where we have a sequence of nested models and use the identity transformation. We do strongly note that we do not need this nested model requirement for tSMC to be used, for example each model may be very different to each other but good transformations that connect the sequences of models may exist, however it is done for illustration purposes. We assume that the model set $(m_0, m_1, ..., m_k, ..., m_K)$ is successively nested within each other, each m_k (excluding m_0) has parameter $\theta_{1:k}$ and we define the parameters θ_k to represent the parameters of

Algorithm 7 The Standard tSMC algorithm.

Set model $(m_0, ..., m_K)$, Set Particle Size N, Set Resampling Threshold
for $i = 1 : N$ **do**
$w_{m_0 0} = \frac{1}{N}$
$\theta^i_{m_0 0} \sim K_{m_0}(\cdot)$
end for
for $k = 1 : K$ **do**
for $i = 1 : N$ **do**
$u^i_{m_{(k-1)}0} \sim \psi_{m_{k-1} \to m_k}(\cdot)$
$\{\theta^i_{m_k 0}, u^i_{m_k 0}\} = h(\theta^i_{m_{(k-1)}0}, u^i_{m_{(k-1)}0})$
end for
for $t = 1 : T$ **do**
for $i = 1 : N$ **do**
$$\tilde{w}^i_{m_k t} = w^i_{m_k(t-1)} \frac{\rho_t(\theta^i_{m_k(t-1)}, u^i_{m_k(t-1)}; m_{k-1} \to m_k)}{\rho_{t-1}(\theta^i_{m_k(t-1)}, u^i_{m_k(t-1)}; m_{k-1} \to m_k)}$$
end for
for $i = 1 : N$ **do**
$$w^i_{m_k t} = \frac{\tilde{w}^i_{m_k t}}{\sum_{j=1}^{N} \tilde{w}^i_{m_k t}}$$
end for
if $\left(\sum_{i=1}^{N} (w^i_{m_k t})^2\right)^{-1} <$ Threshold **then**
Resample via Stratified Resampling algorithm
end if
for $i = 1 : N$ **do**
$\{\tilde{\theta}^i_{m_k t}, \tilde{u}^i_{m_k t}\} \sim K_t(\cdot | \theta^i_{m_k(t-1)}, u^i_{m_k(t-1)})$
$\{\tilde{\theta}^i_{m_{k-1} t}, \tilde{u}^i_{m_{k-1} t}\} = h^{-1}(\tilde{\theta}^i_{m_k t}, \tilde{u}^i_{m_k t})$
$\alpha_{t-1 \to t}(\{\theta^i_{m_k(t-1)}, u^i_{m_k(t-1)}\}, \{\tilde{\theta}^i_{m_k t}, \tilde{u}^i_{m_k t}\}) =$
$$\min\left\{1, \frac{\rho_t(\tilde{\theta}^i_{m_k t}, \tilde{u}^i_{m_k t}; m_{k-1} \to m_k)}{\rho_t(\theta^i_{m_k(t-1)}, u^i_{m_k(t-1)}; m_{k-1} \to m_k)}\right\}$$
$u \sim$ Unif(0,1)
if $u \leq \alpha_{t-1 \to t}(\theta^i_{m_k(t-1)}, \theta^i_{m_k t})$ **then**
$\{\theta^i_{m_k t}, u^i_{m_k t}\} = \{\tilde{\theta}^i_{m_k t}, \tilde{u}^i_{m_k t}\}$
$\{\theta^i_{m_{k-1} t}, u^i_{m_{k-1} t}\} = h^{-1}(\tilde{\theta}^i_{m_k t}, \tilde{u}^i_{m_k})$
else
$\{\theta^i_{m_k t}, u^i_{m_k t}\} = \{\theta^i_{m_k(t-1)}, u^i_{m_k(t-1)}\}$
$\{\theta^i_{m_{k-1} t}, u^i_{m_{k-1} t}\} = h^{-1}(\theta^i_{m_k(t-1)}, u^i_{m_k(t-1)})$
end if
end for
end for
end for

m_k that are exclusive to any models that precede it. We are interested in estimating a sequence of posterior distributions defined by

$$\pi(\theta_{1:k}|y) \quad \propto \quad p(\theta_{1:k}|m_k)f(y|\theta_{1:k}, m_k), \tag{2.35}$$

for all $k \in (1, ..., K)$, and each normalising constant of $Z(y|m_k)$. In our tSMC algorithm we actually define the true target distribution of each m_k to be

$$\pi(\theta_{1:K}|y, m_k) \quad \propto \quad f(y|\theta_{1:k}, m_k)p_1(\theta_1), ..., p_k(\theta_k), ..., p_K(\theta_K), \tag{2.36}$$

in which we assume for all of our applications that each distribution is defined on the random vector $\theta_{1:K}$ rather than $\theta_{1:k}$. Note that (2.36) has as its marginal on $\theta_{1:k}$ the posterior $\pi(\theta_{1:k}|y, m_k)$. Furthermore, the normalising constant of (2.36) is equal to $Z(y|m_k)$. Thus, ignoring any intermediate distributions dictated by a tempering scheme as described earlier, then the normalised weights are proportional to

$$\begin{aligned} w_{m_k} \quad &\propto \quad \frac{\pi(\theta_{1:K}|y, m_k)}{\pi(\theta_{1:K}|y, m_{k-1})} \\ &= \quad \frac{f(y|\theta_{1:K}, m_k)p_1(\theta_1), ..., p_k(\theta_k), p_{k+1}(\theta_{k+1}), ..., p_K(\theta_K)}{f(y|\theta_{1:K}, m_{k-1})p_1(\theta_1), ..., p_{k-1}(\theta_{k-1}), p_k(\theta_k), ..., p_K(\theta_K)} \\ &= \quad \frac{f(y|\theta_{1:k}, m_k)}{f(y|\theta_{1:(k-1)}, m_{k-1})}. \end{aligned} \tag{2.37}$$

In one transformation proposal in chapter 3, we suggest this identity transformation for at least a subset of parameters in which the normalised weights almost take the form of (2.37). What can also be seen is that when MCMC kernels on the space $\theta_{1:K}$ are applied, we only need to perform them on $\theta_{1:k}$ as the parameters $\theta_{(k+1):K}$ are not present in (2.37).

However what we do consider is that the parameters of $\theta_{1:(k-1)}$ that target m_{k-1} may not be an appropriate fit for model m_k when including the new subset θ_k, and in many scenarios there may exist some constraint between variables (for example $\sum_{i=1}^{k} \theta_i = 1$). Therefore our final justification for the tSMC approach is that while the

priors may not be suitable enough for proposing all of $\theta_{1:k}$ in high dimensions, we believe that applying a transformation on $\theta_{1:k-1}$ is a superior option given that the current estimates are appropriate for models m_{k-1} and m_k have marginal parameters that are similar. In chapter 3 we show this comparison of trying to generate each most of θ_k from prior distributions, against applying a transformation on a subset of $\theta_{1:(k-1)}$. Nevertheless should the transformation on parameters not be appropriate for the untransformed subset of parameters inferred from model m_{k-1} still require some changes in the estimates of their posterior probabilities when transitioning to m_k, then this can be done by applying MCMC kernels on $\theta_{1:k-1}$ and θ_k.

The general version of tSMC that is described in section 2.2.1 results from a similar argument to that used in the derivation of (2.37), where the target is chosen to estimate

$$\begin{aligned} \pi(\theta_{1:K}|y, m_k) &\propto f(y|\theta_{1:k}, m_k)p_1(\theta_1), ..., p_k(\theta_k) \\ &\quad \times p_{k+1}(u_{m_{k+1}})..., p_K(u_{m_K}), \end{aligned} \tag{2.38}$$

and when moving from model m_{k-1} to m_k a deterministic transformation is applied to the subvector $(\theta_{1:(k-1)}, u_{m_k})$.

Since we have shown tSMC to be a standard fixed-dimensional SMC sampler, in which we now go back to our standard notation of defining the model parameters of m_k as θ_{m_k}, we obtain the following Monte Carlo approximation,

$$\hat{\pi}(\theta_{m_k}|y, m_k) = \sum_{i=1}^{N} w_{m_k}^{(i)} \delta_{\theta_{m_k}^{(i)}}(\theta_{m_k}^{1:N}), \tag{2.39}$$

and we would have the following central limit theorem for some function η for the case where no resampling is performed Chopin (2004); Del Moral *et al.* (2006),

$$\left(\mathrm{E}_{\hat{\pi}(\theta_{m_k}|y,m_k)}[\eta(\theta_{m_k})] - \mathrm{E}_{\pi(\theta_{m_k}|y,m_k)}[\eta(\theta_{m_k})]\right) \rightarrow \text{Normal}\left(0, \frac{N}{\sigma_{IS}^2(\eta(\theta_{m_k}))}\right). \tag{2.40}$$

In the scenario with multinomial resampling is performed at every iteration we have,

$$\left(\mathrm{E}_{\hat{\pi}(\theta_{m_k}|y,m_k)}[\eta(\theta_{m_k})] - \mathrm{E}_{\pi(\theta_{m_k}|y,m_k)}[\eta(\theta_{m_k})]\right) \quad \rightarrow \quad \text{Normal}\left(0, \frac{N}{\sigma^2_{SMC}(\eta(\theta_{m_k}))}\right). \tag{2.41}$$

We note that $\sigma^2_{IS}(\eta(\theta_k))$ and $\sigma^2_{SMC}(\eta(\theta_k)$ follow from Del Moral *et al.* (2006) where under strong mixing assumptions and using a sup-optimal kernel, such as an MCMC, the variance of $\sigma^2_{SMC}(\eta(\theta_k)$ is uniformally bounded as the number of SMC iterations goes to infinity. However $\sigma^2_{IS}(\eta(\theta_k)$ tends to infinity as the number of SMC iterations goes to infinity. Finally Gerber *et al.* (2017) showed similar results when stratified resampling is applied.

Just like the standard SMC sampler algorithm, see section 2.1.1, we can take the product of each Bayes factor pairing to receive an estimate of the marginal likelihood for each model m_k given that m_0 are simple proposal distributions (for example they could be prior distributions) for the parameters of model m_1. The normalisation constant of the final model, by starting from a model m_0 that is normalised such as a model only containing prior distributions, given by tSMC is defined by

$$\begin{aligned}
\hat{Z}(y|m_K) &= \frac{\widehat{Z(y|m_1)}}{1}\frac{\widehat{Z(y|m_2)}}{Z(y|m_1}\ldots\frac{\widehat{Z(y|m_K)}}{Z(y|m_{K-1})} \\
&= \prod_{k=1}^{K}\prod_{t=1}^{T}\sum_{i=1}^{N} w_{m_k(t-1)}(\theta^i_{m_k(t-1)})\frac{\rho_t(\theta^i_{m_k(t-1)}, u^i_{m_k(t-1)}; m_{k-1} \rightarrow m_k)}{\rho_{t-1}(\theta^i_{m_k(t-1)}, u^i_{m_k(t-1)}; m_{k-1} \rightarrow m_k)} \\
&= \prod_{k=1}^{K}\prod_{t=1}^{T}\sum_{i=1}^{N} w_{m_k(t-1)}(\theta^i_{m_k(t-1)})\tilde{w}_{m_kt}(\theta^i_{m_k((t-1):t)}),
\end{aligned} \tag{2.42}$$

where $w_{m_k(t-1)}$ are normalised weights, after assessing whether a resampling algorithm should be applied or not, such that $w_{m_10}(\theta^i_{m_10}) = N^{-1}$ and each $w_{m_k0} = w_{m_{k-1}T}$. Otherwise $\tilde{w}_{m_kt}(\theta^i_{m_k((t-1):t)})$ are the incremental weights within each model transition. If there is either no resampling throughout the algorithm, or alternatively resampling at each state the estimate is unbiased i.e $\mathrm{E}[\hat{Z}(y|m_k)] = Z(y|m_k)$.

We would also be concerned about how the Monte Carlo error is affected for both the estimated posterior distribution and ML when a transition to a new model involves

an increase in the model parameter dimensions. Beskos *et al.* (2014a) and Beskos *et al.* (2014b) state that for a SMC sampler algorithm which uses annealed intermediate distributions, controlling the MC error in a standard importance sampling algorithm would require the number of particles to increasing exponentially in dimensional size d. Alternatively it can be controlled by using $O(d)$ intermediate distributions, leading to a computational cost that is quadratic in d. As we use one model to bridge to the next successive model, we only plan on introducing a few more dimensions at each iteration and therefore the number of intermediate distributions needed to control of the error may be less than $O(d)$.

2.2.3 Discussion of the Standard tSMC Adaption

2.2.3.1 Advantages of tSMC

As indicated when explaining the basic tSMC algorithm in sections 2.2.1 and 2.2.2, the core strengths of our proposed algorithm over other across model comparison algorithms are discussed within this subsection.

The efficiency from RJMCMC and AIS-RJMCMC can be poor due to a single particle importance sampling estimator, even if applying AIS does offer improvements. However our algorithm, instead of using multiple iterations of a high variance estimator, is going to estimate the Bayes factor using a single SMC which improves the variance of said Bayes factor by allowing for more particles instead of a single particle. Furthermore applying an annealed SMC scheme, instead of a single importance sampler to transition between different models, means any additional changes to the parameters that are needed after the transformation can be made via MCMC kernel moves.

We may use a model m_{k-1} to infer properties of a model of m_k which differ by a few parameters. The densities for most of the parameters would change very little, and any changes to their posteriors that are needed are proposed via MCMC kernel moves, for an increasing dimensional size.

A poor transformation can be compensated by resampling at certain states and through additional MCMC kernels. We show later that these methods can be adaptively made.

As this approach is an adaption of particle filters then performing computational parallelisation of most of the important processes, such as the reweighting and MCMC kernel steps, is possible (Johansen, 2009).

Finally if our initial model m_0 is the prior then we can receive an estimate of the marginal likelihood for each m_k.

2.2.3.2 Limitations and Improvements

The key limitation of tSMC that is shared across all applications is Gibbs samplers, or at least when using target distributions under a geometric bridging scheme of (2.30), can not be used as MCMC kernels for each variable θ_{m_kjt} if they have a continuous parameter space. This is due to how the geometric bridging intermediate distributions raise two different posteriors to a power and thus it is impossible to construct the conditional distributions of each θ_{m_kjt}. It might be possible under certain models, although we have not identified them. We now go into further depth of several improvements to the basic tSMC algorithm in algorithm 7 which will enhance the advantages of tSMC over other algorithms.

An issue in AIS-RJCMC regards how to design an appropriate choice for $\varphi = (\varphi_0, \varphi_1, ..., \varphi_t, ..., \varphi_T)$ given that there is no background information on the properties of the data. If a large particle size is believed to be required to achieve sufficient convergence or the computational cost to perform MCMC moves or evaluate each ρ_t is very large, then it is desirable to minimise the number of reweighting steps. However when designing φ it is necessary to set some of the early discrepancies between φ_{t-1} and φ_t to be very small to give the particles some chance to explore the parameter space and prevent a large variance in the particle weights when reweighting the particles soon after the initial model transition. Nevertheless devising a good schedule for

φ is tricky as it needs to be consistently appropriate for all m_{k-1} to m_k model transitions, and it is unlikely that there is a φ that is appropriate for all model transitions. For example there could be an issue of T being too large, whether this occurs when transitioning between low or high dimensional models is likely to be application dependent, leading to unnecessary reweighting and kernel move steps despite this being a safe option to prevent a large variance in the posterior estimation. We discuss how we can propose adaptively in section 2.3.1.

In a MCMC algorithm the exploration of the parameter space will be poor if the kernels to transition to a new state are not chosen properly. What might be a good proposal for a certain type of distribution may not be appropriate for lower or higher dimensional models. Similarly the kernels may need adjusting as they may have either too low or high acceptance rates as target intermediate distributions with φ_t tend towards 1. As discussed previously we may want to use an adaptive scheme to dictate the number of intermediate distributions when there is uncertainity of how many of these distributions we require. However if there is very little movement in the particles then this would show as a high ESS, and therefore give a poor picture of convergence to an extended space. Therefore we take into account adaptive kernels when possible. In section 2.3.2, we discuss adaptive approaches for both problems to prevent the need for reruns and prior experimentation when applying our algorithm.

Furthermore there is also a question regarding which model transition, given multiple choices for $\{u_{m_{k-1}}, u_{m_k}\}$ and $h()$, is more appropriate to transition between regions of high probability density. To prevent multiple runs of the tSMC algorithm or some other investigation to figure out what the best possible transformation moves could be, we instead propose an adaption of the tSMC algorithm that assigns a subset of particles to different transformations in which the worst transformations are removed via reweighting and resampling. We explain the approach in depth in section 2.3.3.

2.3 tSMC Extensions and Diagnostics

2.3.1 Adaptive φ_t

As stated in section 2.2.3 some form of adaptive annealing is considered to avoid the dilemma of how many annealed intermediate distributions to set up. We still examine non-adaptive approaches too, and how each scheme differs with regards to Monte Carlo error. However some caution is required when using adaptive schemes (this also includes using adaptive MCMC kernels, see section 2.3.2) as they do induce some small bias to both parameter and marginal likelihood estimates. However the bias is still expected to be negligible in the estimates (Prangle *et al.*, 2018).

Instead of defining the annealing schedule via some preset pattern, either being a geometric or evenly spaced sequence etc, each discrepancy between φ_t and φ_{t-1} could be adjusted via a sequence of targets based on some measurement of particle degeneracy such as the effective sample size.

A set of targets to control the particle degeneracy is defined via a sequence of effective sample sizes $(RN, R^2N, ..., R^kN)$ with $0 < R < 1$. At the beginning of the algorithm when $t = 1$ and $\varphi_{t-1} = 0$, we search for φ_1 such that the ESS is equal to RN. Afterward we estimate φ_2, given φ_1, to have the ESS equal to the next target in the sequence of R^2N. The particles are resampled when the ESS is below the predefined threshold, being some value greater or equal to R^kN, which then reverts the ESS sequence back to RN. Naturally $\varphi_t = 1$ is accepted on the condition that it gives an ESS greater then its assigned R^jN. To identify each φ_t we use a bisection method.

Jasra *et al.* (2011) also apply an exact scheme of ESS targets to dictate the annealing schedule. It is also similar to another scheme by Del Moral *et al.* (2012) where they used an adaptive scheme to select the tolerance levels of an Approximate Bayesian Computation target distribution and force a steady decline of the ESS as they infer successive annealed target distributions, tempered by this tolerance level

parameter, within their SMC algorithm.

The issue of using the ESS, as defined in chapter 1 in which is more commonly used in SMC algorithms, is that it is actually a measurement of weight degeneracy between the current distribution and the joint distributions that came after a resampling step. While useful to decide whether to perform resampling on particles, it is not appropriate when we actually desire a measurement that dictates the discrepancy between the target distribution and the distribution solely before it. Therefore we consider a new measurement of the effective sample size, termed the conditional effective sample size (CESS) by Zhou *et al.* (2016), defined as

$$\begin{aligned} \text{CESS} &= N\sum_{i=1}^{N} w^i \left(\frac{\tilde{w}_t^i}{N\sum_{j=1}^{N} \tilde{w}_t^j w_{t-1}^j} \right) \\ &= \frac{N(\sum_{i=1}^{N} w_{t-1}^i \tilde{w}_t{}^i)^2}{\sum_{i=1}^{N} w_{t-1}^i \left(\tilde{w}_t{}^i\right)^2}. \end{aligned} \tag{2.43}$$

It is equal to the ESS if the particles are resampled after every reweighting step. Therefore by setting CESS to some constant CESS $\in (1, N]$, we find φ_t such that it is equal to said constant.

Zhou *et al.* (2016) claims that in their applications using the CESS schedule provided less variation in the marginal likelihoods estimates, for an SMC algorithm which used annealed intermediate distributions with a prior as an importance sampler, while mostly giving gradually increasing discrepancies between each φ_{t-1} and φ_t. In comparison using the ESS leads to uneven discrepancies, notably a huge discrepancy after resampling the particles. Overall what is desirable is for each of the successive discrepancies of $\varphi_t - \varphi_{t-1}$ to be increasing or steady as $\varphi_t \to 1$ by having roughly even distances between each successive distribution.

However there is one disadvantage from doing adaptive annealing in comparison

to a fixed annealing, whether such a φ_t can be practically defined by some finite precision if the value is negligible. For example we found that there were cases where φ_1 could not be found if the quality of the transformation is poor as it leads to many initial zero-weighted particles, with such an example shown in chapter 3. Jasra *et al.* (2011) also suffered the same problem in their proposed across model SMC algorithm. Therefore to compensate for such a move where the quality of the transformation is unknown, especially in cases where we generate $\{\theta_{m_{k'}}, u_{m_{k'}}\}$ directly from a uninformative prior distribution, we may set φ_1 to some very small value and either allow for a resampling step naturally or force a resampling move.

2.3.2 Adaptive MCMC Proposals

We aim to have appropriate kernels that can explore the space of the new target distribution of $\{\theta_{m_k}, u_{m_k}\}$ given that they were proposed using $\{\theta_{m_{k-1}}, u_{m_{k-1}}\}$.

Jasra *et al.* (2011) considered an MCMC kernel move for each independent parameter being a mean-centered Gaussian random walk at time t, with the scaling being the variance of the parameters at $t-1$. They also applied an additional procedure where if the acceptance rates of the MCMC moves for a particular parameter become too large, by breaching a predefined value, then the tuning variance is multiplied by some positive factor β. Respectively if the rates become too small then the tuning variance is reduced. The assumption made is that the empirical variance from time $t-1$ will provide an appropriate tuning to the parameters at time t, and furthermore this continued to perform well under adaptive annealing using the ESS as a threshold. Under this scheme, the vast majority of acceptance rates for each parameter eventually did converge to some low-variance range of values at each t.

Theoretically, given that the marginal posterior distribution of a parameter is close to Gaussian distributed, then when using a Gaussian random walk proposal for one parameter what can give an optimum acceptance rate of around 0.44 is using a tuning variance of $(2.38\sigma)^2$ where σ is the standard deviation of the target. If

several parameters are updated at the same time then it is $(2.38\Sigma)^2/d^2$ where d is the dimensional of the proposal distribution and Σ is the empirical covariance matrix. Furthermore the optimum acceptance rate is 0.234, and it is recommended to at least get acceptance rates to be between 0.15 and 0.4 to obtain at least 80% efficiency of the maximum asymptotic efficiency (as $d \to \infty$) of a MH algorithm (Gelman *et al.*, 1996; Roberts and Rosenthal, 2001, 2009). While it is not exactly stated how to achieve that same percentage of efficiency for a one component proposal, which again requires an optimum rate of 0.44 and that the target distribution is normally distributed, we could consider an acceptance rate between 0.2 and 0.6 covers most of the efficiency given graphical results of Gelman *et al.* (1996). However there is no certainty that a given marginal posterior for a parameter could be justified as being Gaussian distributed.

Thus when considering single parameter random walks we consider applying a relaxed scheme which uses the function of the empirical variance of the parameter at time $t-1$ as the tuning variance. This emperical variance is then multiplied by a fixed constant c depending if the acceptance rates are too high or too low. Under this scheme we will still try and obtain rates between 0.2 and 0.6, on the basis that Gaussian conditions could be fulfilled despite not having a complete picture of the true form of the posterior, and set c to either 2 or 0.5 when a rate breaks its respective boundaries.

Individual adaptive MCMC proposals will be given within chapters 3-5, each catered on the parameter set for each application.

2.3.3 Groundwork for Multiple Transformations

We desire to apply multiple transformations within our tSMC algorithm, and potentially adapt the type of transformations made. This would prevent multiple rounds of testing each type of transformation. It is an ongoing research topic in model transition MCMC algorithms to identify a series of adaptive model transition moves that can roughly consistently jump to regions of high probability density within

the alternative model or bound the proportion of accepted moves to a predefined range (Brooks *et al.*, 2003; Hastie, 2005; Hastie and Green, 2012; Sisson, 2005). However these type of adaptive transformations propose new across model moves based on the properties of a single Markov chain. Whether such adaptions can be similarly applied in a SMC type algorithm is another potential research question, however what first must be analysed is whether the best possible model transformation moves can be identified whilst transitioning between two models.

One possible implementation involves assigning individual particles to one of the across-model transitions through some distribution. The weights of the particles are pooled together during reweighting and resampling procedures, and will change to a different move during the resampling. We investigate what the particle history would be and how this would affect the estimated probabilities densities for each marginal parameter. Ideally by the end of a transition from model m_{k-1} to m_k the best move dominates the particle set early on, and will allow for the potential option of avoiding multiple tSMC runs for each move by analysing the algorithmic history and recommending what the best possible move would be for future iterations. This is based on similar MCMC schemes which apply multiple kernels and identify the most appropriate moves based on acceptance rates and overall quality of each proposal. We define $l_{(m_{k-1}\to m_k)\cdot}$ as a discrete "label" variable whose states (each ith state being $l_{(m_{k-1}\to m_k)i}$) represent the different possible transformations. How each of these labels are defined is dependent on the application, but overall they strictly dictate the form of the auxiliary distributions and the associated transformation.

We let $\psi_{m_{k-1}\to m_k}(\cdot)$ represent the proposal distribution for different transformations and $\psi_{m_k\to m_{k-1}}(\cdot)$ be a proposed auxiliary distribution to dictate what the reverse

transformation is. Then each intermediate distribution is then given by

$$\begin{aligned}\rho_t &= (\pi(m_{k-1}, \theta_{m_{k-1}t}|y)\psi_{m_{k-1}\to m_k}(u_{m_{k-1}t}|l_{(m_{k-1}\to m_k)i}))^{1-\varphi_t} \\ &\times (J_{m_k\to m_{k-1}}\psi_{l,m_{k-1}\to m_k}(l_{(m_{k-1}\to m_k)i}))^{1-\varphi_t} \\ &\times (\pi(m_k, \theta_{m_kt}|y)\psi_{m_k\to m_{k-1}}(u_{m_kt}|l_{(m_{k-1}\to m_k)i}))^{\varphi_t} \\ &\times (\psi_{l,m_k\to m_{k-1}}(l_{(m_{k-1}\to m_k)i}))^{\varphi_t}, \end{aligned} \tag{2.44}$$

in which we apply the ith transformation and corresponding reversed transformation. Usually we will let each possible type of transformation have equal probability of being selected.

2.3.4 Alternative Annealed Intermediate Distribution

An issue with the geometric annealing distributions, described in (2.30), is that if the quality of the transformation to a new parameter space is very inaccurate it may cause larges variations in the particles weights. In the most extreme case, proposals to a new model space might be so poor that almost all particles have probability zero at any φ_t which will break the tSMC algorithm, as well as other algorithms such as the discussed AIS-RJMCMC, as we have no positive weights for resampling. What we consider instead of a geometric annealing scheme is to use arithmetic annealing intermediate distributions, as suggested by Karagiannis and Andrieu (2013), of

$$\begin{aligned}\rho_t &= (1-\varphi_t)(\pi(m_{k-1}, \theta_{m_{k-1}t})\psi_{m_{k-1}\to m_k}(u_{m_{k-1}t})J_{m_k\to m_{k-1}}) \\ &+\varphi_t(\pi(m_k, \theta_{m_kt})\psi_{m_k\to m_{k-1}}(u_{m_kt})). \end{aligned} \tag{2.45}$$

Even if the transformation solely gave zero probability proposals, ρ_t would not equate to zero and potentially can be recovered by MCMC moves. There will exist stronger initial push to have parameter estimates for model m_k, generated from an across model move, into areas of high posterior probability mass at a faster rate as model

m_k will have more initial probability density mass at each ρ_t. This is due to φ_t being used as a multiplicative factor rather than an exponent and thus φ_t will have less impact on the magnitude of the joint densities for both (2.31) and (2.32). What property this intermediate distribution has instead is a close to equal representation of both models, in comparison to a gradual prioritisation of model m_k. However we must note that the true form of the arithmetic annealing intermediate distribution, when unnormalised densities are not applied, is actually given by

$$\begin{aligned}\rho_t &= \left(\frac{(1-\varphi_t)(\pi(m_{k-1},\theta_{m_{k-1}t})\psi_{m_{k-1}\to m_k}(u_{m_{k-1}t})J_{m_k\to m_{k-1}})}{\pi(m_{k-1}|y)}\right.\\ &\quad\left.\times\frac{(1-\varphi_t)\pi(m_{k-1}|y)}{(1-\varphi_t)\pi(m_{k-1}|y)+\varphi_t\pi(m_k|y)}\right)\\ &\quad+\left(\frac{\varphi_t(\pi(m_k,\theta_{m_k,t})\psi_{m_k\to m_{k-1}}(u_{m_kt}))}{\pi(m_k|y)}\right.\\ &\quad\left.\times\frac{\varphi_t\pi(m_k|y)}{(1-\varphi_t)\pi(m_{k-1}|y)+\varphi_t\pi(m_k|y)}\right),\end{aligned} \tag{2.46}$$

and therefore we may have large changes in the incremental weights if there is a great difference between the two marginal likelihoods of $\pi(m_{k-1}|y)$ and $\pi(m_k|y)$, especially when $\varphi_t \to 0$ or $\varphi_t \to 1$. Since we cannot define the true form of each of the marginal posteriors, then it is difficult to define a tempering scheme to compensate for unknown and sudden discrepancies in the weights. However we could apply an adaptive scheme, like in section 2.3.1, which defines the tempering scheme adaptively. This is an advantage over AIS-RJMCMC which only applies one particle and cannot apply a similar adaptive scheme.

2.3.5 Evaluating the Performance of the tSMC algorithm

In chapters 3-5 we apply tSMC to three different models. All of this is coded within R software (R Core Team, 2019). The software provides less complex operations than other programming languages and ideally it is preferred to test basic statistical concepts or ideas. However it is not appropriate to compare practical speed

with an existing programme in c++/Java. As R is an interpreted computer language which reads a series of lines step-by-step in comparison to a compiled language then it will usually be slower.

We estimate the Monte Carlo error from results given by our tSMC algorithm, with a focus on the marginal posterior densities of the model parameters and the marginal likelihood of the data. Otherwise we calculate the Bayes Factors instead of the marginal likelihood depending on how we define the models. We will still run equivalent intermediate distributions runs (or a fixed number of annealed distributions) for all ML tests, as we go by the assumption that a likelihood calculation would make up the bulk of the time to complete a script of a tSMC run. Therefore when appropriate we make comparisons with other MC algorithms, also redeveloped in R, with a measurement of MC error per likelihood calculation. While there exist proposed methods that attempt to determine Monte Carlo variance within one or fewer sweeps, see for example Lee and Whiteley (2018), we stick with the standard approach where we estimate the Monte Carlo variance from a number of runs.

Remaining attributes of tSMC are primarily tested within chapter 3. For example this includes evaluating the quality of the transformation proposals between each model transition by analysing the particle degeneracy given by the ESS and the overall history of the annealed intermediate distributions. These are graphical interpretations, and there is no established measurement that assesses the overall rate of particle degeneracy or the rate of increasing $\varphi_t - \varphi_{t-1}$ over multiple reweighting steps. We also use the number of intermediate distributions given by the adaptive scheme as an indication between the initial and final distributions in which better model transitions would have a small number of intermediate targets. Although as discussed in section 2.2.3 the particle degeneracy as given by the CESS or ESS should not solely be used as a good indicator for particle diversity and posterior convergence, and it is important we access the overall quality of the kernel moves made on the particles is analysed and the marginal posterior distributions. Adaptive MCMC proposals, or

standard proposals, and their associated Mean-Squared Jump distances and acceptance rates over time will also be analysed. The preferable outcome being consistent rates over time despite difficulties in achieving such rates for all marginal parameters using SMC or RJMCMC (Jasra *et al.*, 2011). Otherwise application specific diagnostics are stated in each chapter.

2.4 Discussion

In this chapter we have introduced transformation SMC, a SMC adaption to gradually infer distributional properties of a sequence of models under both fixed and increasing sample size. We have explained where tSMC can be the most useful, and where tSMC is definitely not appropriate such as when models are solely dependent on Gibbs sampling to give the best possible exploration of the model.

In chapter 3 we aim to apply our model in the application of univariate mixture models, and due to the simplicity of the application we primarily use it to test various properties of tSMC and advanced adaptions. Chapter 4 examines an application in genealogy tree reconstruction, an application that can be very difficult to as it involves a discrete parameter space that increases factorially with increasing sample size. Finally in chapter 5 we explore another type of mixture model, it is an example of exploring the same model but here we gradually augment the number of potential clusters that a set of allocations variables can take.

Chapter 3

Analysing tSMC with Applications in Mixture Models

This chapter focuses on estimating posterior densities and marginal likelihoods of a series of univariate Gaussian mixture models given some data $y \in Y \subset \mathbb{R}$. We primarily use this application to analyse the properties of our proposed tSMC algorithm, and its variants, as described in chapter 2. This chapter is split into the following subsections.

Section 3.1 describes the univariate Gaussian Mixture model. We also briefly explain the label-switching problem, and the solution we apply for it.

A review of past approaches to the mixture distribution problem are explained in section 3.2. We also state what contribution tSMC makes when inferring mixture models while also describing its limitations.

Section 3.3 considers the most appropriate priors, MCMC kernels and parameter transformations.

Section 3.4 gives a recap of the type of tests to run in tSMC and what type of comparison diagnostics we will consider for finite mixture models. We discuss our results in section 3.5.

Finally in section 3.6 we discuss strengths and weaknesses of the approach to finite mixture models, and how our adaptions could be expanded to include other

mixture models.

3.1 Mixture Models

3.1.1 The Finite Mixture Model

A mixture model is considered if exploratory analysis or otherwise suggests that the density of the data $\{y_i\}_{i=1}^n$ with $y_i \in Y$ is not unimodal or otherwise cannot be explained through fitting a single distribution. Each model m_k contains a total of k identically distributed "component" distributions. Many research questions assume that the total number of components k is unknown, and under Bayesian model assumptions this may treated as a random variable. We define the possible components as $\{a_1, ..., a_k\}$. The contribution that each component gives to the joint probability density of model m_k is dependent on their corresponding weight of $\omega_{m_k a_j}$, also termed as a mixture proportion, where $\omega_{m_k \cdot} = \{\omega_{m_k a_1}, ..., \omega_{m_k a_k}\}$ and $\sum_{j=1}^{k} \omega_{m_k a_j} = 1$.

A mixture model may also be parameterised using a set of allocation variables, $z_{1:n} = \{z_1, ..., z_n\}$, which correspond to each observation in the sample. The purpose of such variables is to cluster points into the components which could representative of some real world population, and the allocations of points to clusters are treated as missing data that must be inferred as part of the posterior. For example we could set a prior $p(z|\omega_{m_k}, m_k)$ such that $Pr(z_i = a_j) = \omega_{m_k a_j}$, with an additional prior on ω_{m_k} such as a Dirichlet distribution (we refer back to this in chapter 5), and each observation distributed by $y_i \sim f_{z_i}(\cdot|\theta_{m_k z_i}, \omega_{m_k z_i}, m_k)$ where $\theta_{m_k z_i}$ corresponds to the component indexed by z_i in model m_k. Sampling from the posterior with this type of parameterisation is sometimes dubbed as simulating "with completion", with more precise phrasing depending on how the allocation variables are inferred (Cappé *et al.*, 2003).

However we choose to ignore the allocation variables by integrating out the latent variables by taking the product of $f(y_i|\theta_{m_k z_i}, \omega_{m_k z_i}, m_k)$ and $p(z_i|w_{m_k z_j}, m_k)$, and

then simply integrate over all discrete allocation variables which gives us (3.2) as the distribution for one observation defined by

$$y_i \sim \sum_{j=1}^{k} f(y_i|\theta_{m_k a_j}, \omega_{m_k a_j}, m_k) p(z_i = a_j | w_{m_k a_j}, m_k), \tag{3.1}$$

such that

$$\sum_{j=1}^{k} f(y_i|\theta_{m_k a_j}, \omega_{m_k a_j}, m_k) p(z_i = a_j | w_{m_k a_j}, m_k) \equiv \sum_{j=1}^{k} \omega_{m_k a_j} f_{a_j}(y_i|\theta_{m_k a_j}, m_k). \tag{3.2}$$

Specific reasons why we choose to ignore the allocation labels, and otherwise how they can be implemented, is given in section 3.6. Furthermore we consider allocation of the observations in chapter 5 and give more in depth analysis of the methods that are used to allocate each label. From the selection of potential mixture distributions problems, the focus will be on Bayesian model comparison for a univariate mixture of Gaussian distribution. This is given by

$$y_i \sim \sum_{j=1}^{k} \omega_{m_k a_j} \text{Normal}(y_i; \mu_{m_k a_j}, \tau_{m_k a_j}), \tag{3.3}$$

where each $\mu_{m_k a_j}$ and $\tau_{m_k a_j}$ are the means and precisions respectively of a Gaussian distribution for each a_j component of $f_{a_j}(y_i|\theta_{m_k a_j}, m_k) = \text{Normal}(y_i; \mu_{m_k a_j}, \tau_{m_k a_j})$. The choice of Gaussian mixture models was considered as there already exists a range of results for comparison and a series of established Reversible Jump MCMC based transformation moves when transitioning from differing mixture models.

3.1.2 Label Switching

A common problem when inferring mixture models is given a set of components with a parameter set of $\theta_{m_k \cdot} = \{\theta_{m_k a_1}, ..., \theta_{m_k a_k}\}$, where each $\theta_{m_k a_j}$ share a common joint prior, is that the components are not identifiable by design. For illustration consider that there are a total of $k!$ permutations of the possible component orderings. Therefore given two types of permutations of p and p', such that

$(\omega_{m_k p}, \theta_{m_k p}) \equiv (\omega_{m_k p_1}, \omega_{m_k p_2}, ..., \omega_{m_k p_k}, \theta_{m_k p_1}, ..., \theta_{m_k p_k})$ for example, then regardless of the permutations we would have two symmetric likelihoods given by

$$\begin{aligned} f(y|\omega_{m_k p}, \theta_{m_k p}, m_k) &= \prod_{i=1}^{n} \left(\sum_{j=1}^{k} \omega_{m_k p_j} f_{a_j}(y_i|\theta_{m_k p_j}, m_k) \right) \\ &= \prod_{i=1}^{n} \left(\sum_{j=1}^{k} \omega_{m_k p'_j} f_{a_j}(y_i|\theta_{m_k p'_j}, m_k) \right) \\ &= f(y|\omega_{m_k p'}, \theta_{m_k p'}, m_k). \end{aligned} \tag{3.4}$$

Richardson and Green (1997) concluded that using a prior ordering on the means by setting $\mu_1 < ... < \mu_k$ proved to be effective in reducing the multimodality (due to this non-identifiability) of each marginal posterior parameter in univariate Gaussian models. This is in comparison to a sequential ordering of the precisions which failed to remove noticeable multimodality. However this prior does not necessarily mean that multimodality in the posterior distributions will be removed, at least with models that have more components than we might expect to find in the data, which we call "oversaturated" models (Stephens, 2000b). We adopt this solution of ordering the means for this chapter, but there are also other ways of tackling the problem. For example the Kullback-Leibler (KL) relabeling strategy to apply with MCMC involves considering a loss/cost function regarding all possible permutations and potentially latent allocation variables, and their corresponding classification probabilities of $\omega_{m_k a_j} f_{a_j}(y_i|\theta_{m_k a_j}, m_k) / \sum_{j=1}^{k} \omega_{m_k a_j} f_{a_j}(y_i|\theta_{m_k a_j}, m_k)$. There is also the strategy of performing a Monte Carlo algorithm without any constraints or orderings and then perform post-analysis to assign the component labels (Stephens, 1999). Otherwise it is advised to apply some post process simulation with different label assignments on each component to analyse the properties of the posterior (Richardson and Green, 1997). For further reading on recent advanced strategies to label switching with respect to Bayesian mixture models we would recommend Cron and West (2011); Jasra *et al.* (2005); Papastamoulis and Iliopoulos (2010); Rodríguez and Walker (2014).

3.2 Discussion of Past Approaches and tSMC Adaption

We primarily focus on past research in the application of model selection for univariate/multivariate mixture models, and efforts which allow for the chance of increasing/decreasing dimensional size within one run of the algorithm. We refer back to chapters 1-2 for the strengths and weaknesses of model comparison when using MCMC and SMC methods in which numerous adaptions have been proposed. The problem of using MCMC and SMC methods is that separate runs of the same algorithm are required if model comparison of a series of models, with the difference being the number of parameters to infer, was of interest. Furthermore the most reliable methods to estimate the ML from MCMC output tend to have higher computational cost. RJMCMC methods allow for the transition to different models and one of the algorithm's earliest uses was in the application univariate/multivariate Gaussian mixture models (see Brooks *et al.* (2003); Jasra *et al.* (2008, 2005); Papastamoulis and Iliopoulos (2009); Richardson and Green (1997); Zhang *et al.* (2004)). Again though it may struggle to transition to different parameter spaces if the across model transitions are inappropriate, and there exists a risk that the algorithm will not explore certain mixture models within a set length of the Markov chain as it becomes harder to devise better models transitions with particularly if the number of parameters for each component is large.

An alternative class of algorithms, termed the continuous time Markov chains Monte Carlo (CT-MCMC) by Cappé *et al.* (2003), also perform jumps between models. The basic concept introduced by Stephens (2000a) is that each state of a Markov chain involves initiating a continuous time Markov birth-death process. In this process a new component of a mixture distribution has a chance of being created which is dictated by some birth rate (for an explanation on the concept of a component birth or component death, see section 3.3.2), in which all weights are adjusted to incorpo-

rate the new component. For each component that is "born" or is already generated before the process starts, their independent death rates are also calculated. The rate is designed such that a bad proposal to the target distribution will lead to a higher rate. Therefore the number of components remains unchanged for some exponentially distributed time period and after this holding time has passed either a birth or death occurs. For example a component death occurs with probability given by its independent death rate divided by the sum of the birth and death rates, and afterwards all death rates are recalculated followed by another proposed holding time. Once the Markov process stops at some predefined total time then other kernels may be applied, such as MCMC moves on the model parameters etc, before moving onto the next state of the Markov chain. One adaption by Cappé *et al.* (2003), who commented on the lack of sufficient convergence when using birth/death moves alone, instead considered splitting a component and its corresponding parameters into two instead of birthing them and merging components instead of deleting them. CT-MCMC is slower and does not give any notable improvements to the estimate of the model posterior distribution than RJMCMC. Furthermore it still suffers from the same flaws of proposing new components from an existing high-dimensional model and any form of exploring the parameter space, which isn't a model transformation move, is only done at a certain state which comes after components have been proposed and deleted (Cappé *et al.*, 2003).

We conclude with Variational Bayes (VB) algorithms, a non Monte Carlo based series of methods with origins in Attias (1999), that gives variational posterior approximations. The basic concept of these algorithms for the application in model selection of mixture models is that it first involves constructing some tractable function, also termed as a variational distribution, that is an approximation to the posterior distribution. Afterwards, starting initially with a high dimensional model where the model posterior is likely to have small probability density, the algorithm attempts to maximise the marginal likelihood of the posterior through an iterative algorithm

that moves the allocation variables (if included in the model) and the component parameter estimates, and permanently (usually) deletes components based on minimising the Kull-back divergence between the tractable function and the posterior distribution (Ormerod and Wand, 2010). Note that we don't consider pure VB to find the best possible model for a set of data as the algorithm tends to underestimate the uncertainty of the ML, and furthermore it lacks the same convergence and variance consistency guarantees as Monte Carlo methods which were stated in chapter 1 (Grosse *et al.*, 2015). An interesting approach by McGrory *et al.* (2016) applies a sequential Monte Carlo algorithm and gradually adds data in batches over time. They again initiate on the proposed highest component size and use variational Bayes based kernel moves. The downside of this SMC approach however is that depending on the ordering of these sequential batches of data a component might be deleted too soon, although VB proposals were made in Wu *et al.* (2012) that allowed for the generation of a new components in a typical VB algorithm.

3.2.1 The tSMC Approach

As stated in chapter 2, we believe that exploiting the similarities in posteriors between neighboring models can assist with developing importance proposals for a high dimensional mixture model. In the case of univariate Gaussian distributions with a large number of components we might expect that the posterior distribution over the majority of the parameters will not change significantly when a new component is included, except with some adjustments to component precisions, and thus given that it is easier to estimate posterior distributions for low-dimensional models then it will assist with convergence at higher dimensions. Furthermore applying a series of intermediate distributions with MCMC kernels to adjust for poorer proposals, and move them into areas of high posterior probability density, is also beneficial. Unlike RJMCMC which depends on single transformation, tSMC makes N proposals (i.e N particles) which are weighted such that proposals which are in high probability

regions of the posterior will have larger weights while other particles will be deleted after resampling or gradually moved via kernel moves. Although CT-MCMC is similar with death rates affected by the quality of a proposal, it does not apply kernels to a mediocre proposal until the Markov time process is completed. Furthermore we receive an estimate of the ML by design without additional post-analysis.

3.3 Adaption of tSMC to the Univariate Mixture Model

Here we aim to estimate the marginal likelihood and posterior distribution of

$$\pi(\theta_{m_k\cdot}, \omega_{m_k\cdot}, m_k|y) \quad \propto \quad f(y|\theta_{m_k\cdot}, \omega_{m_k\cdot}, m_k)p(\theta_{m_k\cdot}, \omega_{m_k\cdot}, m_k), \tag{3.5}$$

for a set of models $(m_1, ..., m_K)$, with model m_0 containing only normalised proposals for model m_1. This application is an example where we have a sequence of successive nested models, each one differing by the inclusion of one Gaussian component. For the remainder for this section we first describe the complete form of the posterior distribution, followed by several types of transformation that we apply and finally describe the MCMC kernels we apply to the component weights and Gaussian parameters.

3.3.1 The Posterior Distribution

The unnormalised form of the posterior distribution is

$$\begin{aligned}\pi(\theta_{m_k\cdot}, \omega_{m_k\cdot}, m_k|y) \quad \propto \quad & f(y|\omega_{m_ka_1}, ..., \omega_{m_ka_k}, \mu_{m_ka_1}, ..., \mu_{m_ka_k}, \tau_{m_ka_1}, ..., \tau_{m_ka_k}, m_k) \\ & \times p(\omega_{m_ka_1}, ..., \omega_{m_ka_k})p(\mu_{m_ka_1}, ..., \mu_{m_ka_k}) \\ & p(\tau_{m_ka_1}|b)...p(\tau_{m_ka_k}|b)p(b)p(m_k),\end{aligned} \tag{3.6}$$

where $\theta_{m_ka_j} = \{\mu_{m_ka_j}, \tau_{m_ka_j}\}$ and b is explained later within this section. The likelihood is again defined in (3.3), being

$$\prod_{i=1}^{n}\sum_{j=1}^{k}\omega_{m_k a_j}\text{Normal}(y_i;\mu_{m_k a_j},\tau_{m_k a_j}). \tag{3.7}$$

We define the prior of each Gaussian mean by

$$\mu_{m_k a_j} \quad \sim \quad \text{Normal}(\mu=\bar{y},\tau=(y_{\max}-y_{\min})^{-2}). \tag{3.8}$$

where again μ and τ refers to the mean and precision respectively of a Gaussian distribution. Otherwise $\bar{y}$ is the observational mean, with $y_{\max}$ and $y_{\min}$ being the maximum and minimum values respectively of the observations. This prior ensures that it is likely that that each of the true distributions of the means are within the range of the prior, although a drawback is that it does not encourage closer fitting $\mu_{m_k a_j}$. Otherwise we consider the ordering of the Gaussian means of $\mu_{m_k a_1} < \mu_{m_k a_2} < ... < \mu_{m_k a_k}$, where there are $k!$ possible orderings. It also gives our joint prior of the Gaussian means defined by

$$p(\mu_{m_k a_1},...,\mu_{m_k a_k}) = \begin{cases} k!\prod_{j=1}^{k}\text{Normal}(\mu_{m_k a_j} \mid \bar{y},(y_{\max}-y_{\min})^{-2}) & \text{if } \mu_{m_k a_1} < ... < \mu_{m_k a_k} \\ 0 & \text{Otherwise} \end{cases}. \tag{3.9}$$

This acts as our basic solution to the label switching problem (Richardson and Green, 1997). For each precision we use a gamma distributed prior of

$$\tau_{m_k a_j} \quad \sim \quad \text{Ga}\left(\alpha=2,\beta=b\right), \tag{3.10}$$

where α and β are the shape and rate parameters respectively. Note that we also insert an additional prior on said rate parameter defined by

$$b \quad \sim \quad \text{Ga}(\alpha=0.2,\beta=10/(y_{\max}-y_{\min})^2). \tag{3.11}$$

We use this prior as we assume that we are uninformed about the true spread of the parameter space for each precision, and furthermore Richardson and Green (1997)

has shown that the marginal posterior distribution for the number of components is insensitive to the assumptions made in (3.10) and (3.11) while in comparison using different fixed β in (3.10) will give different posteriors.

For the priors that have mentioned so far, we have incorporated the data itself into the priors and is very close to what was used in Richardson and Green (1997). We recognise that it is not good practice, as it is a double use of data which we are not supposed to know in the first place as prior distributions are based on prior assumptions before receiving the data. Furthermore depending on how the hyperparameters are defined these priors are at risk of being appropriate for some datasets but not for others, to the point where it would of been simpler to set up a vague prior that would be appropriate for any predicted data variation (Berger, 2006). However it is still used in practice, such as the use of g-priors for regression coefficients when applying Bayes theorem to multiple regression models (Liang *et al.*, 2008), and again in this chapter we are more interested in analysing the properties of the tSMC algorithm then a true understanding of the data itself.

For the component weights we again assume no prior information and set an uninformative Dirichlet distribution of

$$\omega_{m_k \cdot} \sim \text{Dirichlet}(\alpha_1 = 1, ..., \alpha_k = 1), \tag{3.12}$$

where each α_i are the concentration parameters for a Dirichlet distribution.

Finally we set the prior on the number of components to be a discrete uniform distribution, thus assuming that we have no prior information on the required number of Gaussian distributions to represent the data of

$$p(m_k) = K^{-1}, \tag{3.13}$$

for $k \in \{1, ..., K\}$ where K is the pre-defined maximum number of joint distributions that is realistically expected from the data. Another credible prior for the number

of components would be a Poisson distribution, including its truncated variant, of Poi(λ) where we set λ to the most likely number of joint Gaussian distributions such that smaller probabilities are given the further the state deviates from λ (see Jasra (2006); Nobile and Fearnside (2007); Phillips and Smith (1995); Richardson and Green (1997)).

3.3.2 Model Transition Moves

We now describe how we plan on jumping between two parameter spaces. As stated from chapter 2 we simplify the notation and consider the set $\{\theta_{m_k \cdot t}, u_{m_k \cdot t}\}$ to refer to the results by using a transformation on the set $\{\theta_{m_{k-1} \cdot t}, u_{m_{k-1} \cdot t}\}$ and we are also reintroducing the time index t to the notation, representing the current intermediate distribution assuming a total of T intermediate distributions, as described in chapters 1 and 2.

3.3.2.1 Birth Move

One method to generate a new component for a higher dimensional model with at least one parameter is to generate the component parameters via auxiliary variables and then apply an identity transformation such that all new parameters are the equivalent to the auxiliary variables. To jump to a model with k components we consider the auxiliary variables $u_{m_{k-1}} = \{u_1, u_2, u_3\}$ such that

$$\begin{aligned} \hat{\mu} \equiv u_1 &\sim \text{Normal}(\mu = \bar{y}, \tau = (y_{\max} - y_{\min})^{-2}) \\ \hat{\tau} \equiv u_2 &\sim \text{Ga}\,(\alpha = 2, \beta = b) \\ \hat{\omega} \equiv u_3 &\sim \text{Beta}(1, k). \end{aligned} \tag{3.14}$$

The new mean and precision variables for the new components are proposed from their respective priors, as can be seen in (3.14). We reorder the set $(\mu_{m_{k-1}a_1}, ..., \mu_{m_{k-1}a_{k-1}}, \hat{\mu})$ to match the ordering of means as stated in (3.9) to get the set $(\mu_{m_k a_1}, ..., \mu_{m_k a_k})$

and this ordering applies for the other parameters too. A corresponding weight for Normal$(\mu_{m_k a_k}, \tau_{m_k a_k})$ is generated from u_3 with the other weights adjusted via $\omega_{m_k a_{(1:k-1)}} = \omega_{m_k a_k}(1 - \omega_{m_k a_{(1:k-1)}})$. From the transformation of the set of weights we receive a Jacobian of $J_{m_k \to m_{k-1}} = (1 - \hat{\omega})^{-(k-1)}$. This type of move is termed as a birth move, again based from Richardson and Green (1997), with the opposite of a birth move that removes a component termed as a death move. To add the new component we consider $a_{m_{k-1} \to m_k,\cdot}$ as the set of labels that can represent the newly created Gaussian component, with $a_{m_{k-1} \to m_k, i}$ as the specific ith label within this set. This is given with probability

$$\begin{aligned} a_{m_{k-1} \to m_k, i} &\sim \psi_{a, m_k \to m_{k-1}} \\ &= (k)^{-1}, \end{aligned} \tag{3.15}$$

and we do not propose a change to this auxiliary variable throughout the algorithm. This label is used to determine which component is to be deleted when we consider an inverse transformation to the model space of m_{k-1}. We also note that $\psi_{a, m_{k-1} \to m_k}(\cdot) = 1$ in this scenario. If we were to consider an importance weight of the tSMC algorithm, when transitioning from $k-1$ component model to one with k components where $\varphi_0 = 0$ and $\varphi_{T=1} = 1$, then this is defined by

$$\begin{aligned} \frac{\rho_T(\theta_{m_k \cdot 0}, \omega_{m_k \cdot 0}, u_{m_k}; m_{k-1} \to m_k)}{\rho_0(\theta_{m_k \cdot 0}, \omega_{m_k \cdot 0}, u_{m_k}; m_{k-1} \to m_k)} &= \frac{f(y|\theta_{m_k \cdot 0}, \omega_{m_k \cdot 0}, m_k) p(\theta_{m_k \cdot 0}) p(\omega_{m_k \cdot 0})}{f(y|\theta_{m_{k-1} \cdot 0}, \omega_{m_{k-1} \cdot 0}, m_{k-1}) p(\theta_{m_{k-1} \cdot 0}) p(\omega_{m_{k-1} \cdot 0})} \\ &\quad \times \frac{p(m_k)}{p(m_{k-1}) \psi_{m_{k-1} \to m_k}(u_3 | a_{m_{k-1} \to m_k, i}) J_{m_k \to m_{k-1}}} \\ &\quad \times \frac{\psi_{a, m_k \to m_{k-1}}(a_{m_{k-1} \to m_k, i})}{\psi_{a, m_{k-1} \to m_k}(a_{m_{k-1} \to m_k, i})} \end{aligned}$$

$$
\begin{aligned}
&= \frac{\left(\sum_{j=1}^{k}\omega_{m_k a_j 0} f_{a_j}(y|\theta_{m_k a_j 0}, m_k)\right) p(\omega_{m_k \cdot 0})}{\left(\sum_{j=1}^{k-1}\omega_{m_{k-1} a_j 0} f_{a_j}(y|\theta_{m_{k-1} a_j 0}, m_k)\right) p(\omega_{m_{k-1}\cdot 0})} \\
&\quad \times \frac{1}{\psi_{m_{k-1}\to m_k}(\widehat{\omega})((1-\widehat{\omega})^{-(k-1)})}\left(\frac{p(\mu_{m_k\cdot 0})p(\tau_{m_k\cdot 0})\psi_{a,m_k\to m_{k-1}}(a_{m_{k-1}\to m_k,i})}{p(\mu_{m_{k-1}\cdot 0})p(\tau_{m_k\cdot 0})}\right) \\
&= \frac{\left(\sum_{j=1}^{k}\omega_{m_k a_j 0} f_{a_j}(y|\theta_{m_k a_j 0}, m_k)\right) p(\omega_{m_k \cdot 0})}{\left(\sum_{j=1}^{k-1}\omega_{m_{k-1} a_j 0} f_{a_j}(y|\theta_{m_{k-1} a_j 0}, m_k)\right) p(\omega_{m_{k-1}\cdot 0})} \\
&\quad \times \frac{1}{\psi_{m_{k-1}\to m_k}(\widehat{\omega})((1-\widehat{\omega})^{-(k-1)})}, \qquad (3.16)
\end{aligned}
$$

as $\{\tau_{m_{k-1}a_1}, ..., \tau_{m_{k-1}a_{k-1}}, \hat{\tau}\} \equiv \{\tau_{m_k a_1}, ..., \tau_{m_k a_k}\}$ means that the precision priors cancel out in (3.16) and a similar logic follows with the prior of the means and $\psi_{a,m_k\to m_{k-1}}(\cdot) = (k)^{-1}$ canceling each other out. What we also do when explaining these transformation is suppress the dependence on all model parameters and the Jacobian on $a_{m_{k-1}\to m_k,i}$, where this was done to make the notation more simplistic regarding how the parameters can change for differing values of $a_{m_{k-1}\to m_k,i}$ (a conditionality we take note of in section 3.3.2.3).

3.3.2.2 Split Move

What was also considered was a split move which splits the weight, mean and precision of $\{\omega_s, \mu_s, \tau_s\}$ from a randomly selected component into two weights, means and variances. This involves the generation of auxiliary variables $u_{m_{k-1}} = \{u_1, u_2, u_3\}$ distributed by

$$
\begin{aligned}
u_1 &\sim \text{Beta}(2,2) \\
u_2 &\sim \text{Beta}(1,1) \\
u_3 &\sim \text{Beta}(2,2). \qquad (3.17)
\end{aligned}
$$

For the chosen Gaussian component we split its corresponding weight ω_{s}, mean μ_{s} and precision τ_{s} to represent the two new components, based on the Gaussian split move by Richardson and Green (1997) where

$$\begin{aligned}
\hat{\omega}_1 &= u_3\omega_{\mathrm{s}} \\
\hat{\omega}_2 &= (1-u_3)\omega_{\mathrm{s}} \\
\hat{\mu}_1 &= \mu_{\mathrm{s}} - u_1\sqrt{(\tau_{\mathrm{s}})^{-1}\frac{\hat{\omega}_2}{\hat{\omega}_1}} \\
\hat{\mu}_2 &= \mu_{\mathrm{s}} + u_1\sqrt{(\tau_{\mathrm{s}})^{-1}\frac{\hat{\omega}_1}{\hat{\omega}_2}} \\
\hat{\tau}_1 &= \left(u_2(1-u_1^2)(\tau_{\mathrm{s}})^{-1}\frac{\omega_{\mathrm{s}}}{\hat{\omega}_1}\right)^{-1} \\
\hat{\tau}_2 &= \left((1-u_2)(1-u_1^2)(\tau_{\mathrm{s}})^{-1}\frac{\omega_{\mathrm{s}}}{\hat{\omega}_2}\right)^{-1}.
\end{aligned} \tag{3.18}$$

The reverse move is termed a merge move, that takes the form of the following set of functions defined by

$$\begin{aligned}
\omega_{\mathrm{s}} &= \hat{\omega}_1 + \hat{\omega}_2 \\
\mu_{\mathrm{s}} &= \frac{\hat{\omega}_1\hat{\mu}_1 + \hat{\omega}_2\hat{\mu}_2}{\omega_{\mathrm{s}}} \\
\tau_{\mathrm{s}} &= \left(\hat{\omega}_1\frac{\hat{\mu}_1^2 + (\hat{\tau}_1)^{-1}}{\omega_{\mathrm{s}}} + \hat{\omega}_2\frac{\hat{\mu}_2^2 + (\hat{\tau}_2)^{-1}}{\omega_{\mathrm{s}}} - (\mu_{\mathrm{s}})^2\right)^{-1} \\
u_1 &= \frac{\hat{\mu}_2 - \hat{\mu}_1}{\sqrt{(\hat{\tau}_1)^{-1}\frac{\hat{\omega}_2}{\hat{\omega}_1}} + \sqrt{(\hat{\tau}_2)^{-1}\frac{\hat{\omega}_1}{\hat{\omega}_2}}} \\
u_2 &= 0.5 + \frac{\hat{\omega}_1\,(\hat{\tau}_1)^{-1} - \hat{\omega}_2\,(\hat{\tau}_2)^{-1}}{2\omega_{\mathrm{s}}(1-u_1^2)\tau_{\mathrm{s}}} \\
u_3 &= \frac{\hat{\omega}_1}{\omega_{\mathrm{s}}}.
\end{aligned} \tag{3.19}$$

The Jacobian $J_{m_k\to m_{k-1}}$ for this type of transformation move can be expressed, based on a transformed variant of the general form of $J_{m_{k-1}\to m_k}$ for multivariate Gaussian

models under 3.19 (Zhang *et al.*, 2004), by

$$\left((\omega_{\mathrm{s}})^4 \left((\tau_{\mathrm{s}})^{-0.5} / \hat{\omega}_1 \hat{\omega}_2\right)^{3/2} (1 - u_1^2) \left(\hat{\tau}_1 \hat{\tau}_2 / \tau_{\mathrm{s}}\right)^2\right)^{-1} \tag{3.20}$$

where $(\hat{\tau}_1 \hat{\tau}_2 / \tau_{\mathrm{s}})^2$ simply accounts for the determinant of the relationship between the precision and the variance of the data when transforming the general Jacobian in Zhang *et al.* (2004). Furthermore our reverse move only considers merging components that are directly adjacent to each other where there are $k - 1$ adjacent move pairs. We also add the further assumption, based in Richardson and Green (1997) and Karagiannis and Andrieu (2013), that for the fusion of (3.19) to hold true we require that $\mu_{m_k a_1}$ is directly followed by $\mu_{m_k a_2}$, with no other $\mu_{m_k a_j}$ in between, in order to prevent the label switching problem. It is possible to ignore this rule and instead consider all possible pairings, see for example Cappé *et al.* (2003), however it means removing the indicator variables of $\mu_{m_k a_1} < ... < \mu_{m_k a_k}$ and this could lead to a single component attempting to represent a subset of Gaussian components. Overall we consider the probability of a component to be split, and again we do not make any changes to these variables over time between a model transition, to be given by the auxiliary variable of

$$\begin{aligned} a_{m_{k-1} \to m_k, i} &\sim \psi_{a, m_{k-1} \to m_k}(\cdot) \\ &= (k-1)^{-1}, \end{aligned} \tag{3.21}$$

and this label is also used to identify which pairwise component pairing is to be merged where

$$\psi_{a, m_k \to m_{k-1}}(a_{m_{k-1} \to m_k, i}) = (k-1)^{-1}, \tag{3.22}$$

in order to perform the inverse calculation. When using this transformation proposal an importance weight between two models, given $\varphi_0 = 0$ and $\varphi_{T=1} = 1$, is given by

$$\begin{aligned}\frac{\rho_T(\theta_{m_k\cdot 0},\omega_{m_k\cdot 0},u_{m_k};m_{k-1}\to m_k)}{\rho_0(\theta_{m_k\cdot 0},\omega_{m_k\cdot 0},u_{m_k};m_{k-1}\to m_k)} &= \frac{f(y|\theta_{m_k\cdot 0},\omega_{m_k\cdot 0},m_k)p(\theta_{m_k\cdot 0})p(\omega_{m_k\cdot 0})}{f(y|\theta_{m_{k-1}\cdot 0},\omega_{m_{k-1}\cdot 0},m_{k-1})p(\theta_{m_{k-1}\cdot 0})p(\omega_{m_{k-1}\cdot 0})} \\ &\quad\times\frac{p(m_k)}{p(m_{k-1})\psi_{m_{k-1}\to m_k}(u_{m_{k-1}}|a_{m_{k-1}\to m_k,i})} \\ &\quad\times\frac{\psi_{a,m_k\to m_{k-1}}(a_{m_{k-1}\to m_k})}{\psi_{a,m_{k-1}\to m_k}(a_{m_{k-1}\to m_k})J_{m_k\to m_{k-1}}} \\ &= \frac{\left(\sum_{j=1}^{k}\omega_{m_k a_j 0}f_{a_j}(y|\theta_{m_k a_j 0},m_k)\right)p(\theta_{m_k\cdot 0})}{\left(\sum_{j=1}^{k-1}\omega_{m_{k-1}a_j 0}f_{a_j}(y|\theta_{m_{k-1}a_j 0},m_k)\right)p(\theta_{m_{k-1}\cdot 0})} \\ &\quad\times\frac{p(\omega_{m_k\cdot 0})}{p(\omega_{m_{k-1}\cdot 0})\psi_{m_{k-1}\to m_k}(u_{m_{k-1}})\left(\tau_{\mathrm{s}}/\hat{\tau}_1\hat{\tau}_2\right)^2} \\ &\quad\times\frac{1}{\left((\omega_{\mathrm{s}})^4\left((\tau_{\mathrm{s}})^{-0.5}/\hat{\omega}_1\hat{\omega}_2\right)^{3/2}(1-u_1^2)\right)^{-1}}, \end{aligned} \tag{3.23}$$

and again we suppress the dependence of the Jacobian and all other model parameters on $a_{m_{k-1}\to m_k,i}$.

3.3.2.3 Deconditioning the Model Proposals

There is a flaw regarding how both transformations, stated in sections 3.3.2.1 and 3.3.2.2, act as a proposal to an extended parameter space at each intermediate distribution. As the tSMC progresses to each intermediate distribution, the particles can become degenerate over the $a_{m_{k-1}\to m_k}$ variables since some of the transformations yield better proposals than others. Thus, despite choosing say a uniform distribution over the component label $a_{m_{k-1}\to m_k}$, our proposals for the labels will eventually fail to cover all of its possible states so only estimate a fraction of the marginal likelihood is estimated over time which would cause a notable variance in the Monte Carlo estimates. However the posterior distribution over the parameters is the same for each label, so the degeneracy in the labels does not effect the main posterior distribution that we want to infer. The birth move also has the same analogous problem.

This issue is also important where we wish to consider more than one type of transformation for moving between models. This involves applying an additional auxiliary variable dictating whether we use a birth or a split move. However such labels will also become degenerate. Therefore we choose to consider variance reduction methods.

We consider integrating out a set of variables that are conditional on the target distribution. While the purpose of such variables is to make sampling easier, it can also increase the variance of the posterior and the marginal likelihood. This is sometimes termed as deconditioning the model (Douc *et al.*, 2007; Liu *et al.*, 1994). In our case we choose to integrate out the auxiliary labels $a_{m_{k-1}\to m_k}$ stating how we initiate a birth/death on a component or split/merge one-two components respectively (similar to how the mixture model "with completion" was defined in section 3.1. For the birth move we generate a new component, and then consider summing over the discrete choices of which component is to be removed and thus we are deconditioning over the auxiliary variable. Therefore an importance weight of a transition between two models, with $\varphi_0 = 0$ and $\varphi_{T=1} = 1$, is defined in (3.24)

$$\begin{aligned}
\frac{\rho_T(\theta_{m_k\cdot 0}, \omega_{m_k\cdot 0}, u_{m_k}; m_{k-1} \to m_k)}{\rho_0(\theta_{m_k\cdot 0}, \omega_{m_k\cdot 0}, u_{m_k}; m_{k-1} \to m_k)} &= \frac{\left(\sum_{j=1}^{k} \omega_{m_k a_j 0} f_{a_j}(y|\theta_{m_k a_j 0}, m_k)\right)}{\rho_0(\theta_{m_k\cdot 0}, \omega_{m_k\cdot 0}, u_{m_k}; m_{k-1} \to m_k)} \\
&\quad \times p(\theta_{m_k\cdot 0}) p(\omega_{m_k\cdot 0}) p(m_k) \qquad (3.24) \\
\rho_0(\theta_{m_k\cdot 0}, \omega_{m_k\cdot 0}, u_{m_k}; m_{k-1} \to m_k) &= \sum_{a_{m_{k-1}\to m_k,i}} \left(\left(\sum_{j=1}^{k} \omega_{m_{k-1} a_j 0} f_{a_j}(y|\theta_{m_{k-1} a_j 0}, m_k) \right) \right. \\
&\quad \times p(\omega_{m_{k-1}\cdot 0}) p(\theta_{m_{k-1}\cdot 0}) p(m_{(k-1)}) \\
&\quad \psi_{m_{k-1}\to m_k}(\hat{\omega}|a_{m_{k-1}\to m_k,i}) \\
&\quad \left. \times \psi_{a,m_{k-1}\to m_k}(a_{m_{k-1}\to m_k,i}) J_{m_k\to m_{k-1}} \right), \qquad (3.25)
\end{aligned}$$

and we suppress the conditionality on the model parameters and the Jacobian. In particular $\theta_{m_{k-1}\cdot}$, $\omega_{m_{k-1}\cdot}$ and $J_{m_k\to m_{k-1}}$ will vary for differing $a_{m_{k-1}\to m_k,i}$. For example

if we consider a three component model in m_k, if the label is $a_{m_{k-1}\to m_k,1}$ then $\theta_{m_{k-1}\cdot}$ will represent the second and third Gaussian components of $\theta_{m_k\cdot}$, with $\omega_{m_{k-1}\cdot}$ similarly adjusted based on the inverse transformation on $\{\theta_{m_k\cdot}, \omega_{m_k\cdot}\}$. Furthermore we emphasise that $\psi_{a,m_k\to m_{k-1}}(a_{m_{k-1}\to m_k,i})$, integrates to one within (3.24), and this is similarly applied when we considering integrating out the auxiliary labels for the split move.

The deconditioned version of the split move, although we integrate over which pairwise components were thought to have originated from a split instead, is expressed by an importance weight of

$$\begin{aligned}
\frac{\rho_0(\theta_{m_k\cdot 0}, \omega_{m_k\cdot 0}, u_{m_k}; m_{k-1}\to m_k)}{\rho_0(\theta_{m_k\cdot 0}, \omega_{m_k\cdot 0}, u_{m_k}; m_{k-1}\to m_k)} &= \frac{\left(\sum_{j=1}^{k}\omega_{m_k a_j 0} f_{a_j}(y|\theta_{m_k a_j 0}, m_k)\right)}{\rho_0(\theta_{m_k\cdot 0}, \omega_{m_k\cdot 0}, u_{m_k}; m_{k-1}\to m_k)} \\
&\quad \times p(\theta_{m_k\cdot 0})p(\omega_{m_k\cdot 0})p(m_k) \qquad (3.26) \\
\rho_0(\theta_{m_k\cdot 0}, \omega_{m_k\cdot 0}, u_{m_k}; m_{k-1}\to m_k) &= \left(\sum_{a_{m_{k-1}\to m_k,i}}\left(\sum_{j=1}^{k}\omega_{m_{k-1}a_j 0} f_{a_j}(y|\theta_{m_{k-1}a_j 0}, m_k)\right)\right. \\
&\quad \times p(\theta_{m_{k-1}\cdot 0})p(\omega_{m_{k-1}\cdot 0})p(m_{k-1}) \\
&\quad \times \psi_{m_{k-1}\to m_k}(u_{m_{k-1}0}|a_{m_{k-1}\to m_k,i}) \\
&\quad \left. J_{m_k\to m_{k-1}}\right)\frac{1}{k-1}. \qquad (3.27)
\end{aligned}$$

Note that since we don't integrate over $\psi_{a,m_k\to m_{k-1}}(\cdot)$, which is equal to $(k-1)^{-1}$, this remains in the denominator term of (3.27)

Finally we consider deconditioning the tSMC adaption when we assign a subset of the particles to either the birth or split move defined by the label $l_{m_{k-1}\to m_k,i}$, as discussed in chapter 2. Not only do we remove the dependence over the auxiliary label variables, but also whether a birth or a split move was applied to a particle. We assume that each transformation has equal probability of being assigned to one of the transformations, i.e $Pr(l_{m_{k-1}\to m_k,1}) = Pr(l_{m_{k-1}\to m_k,2}) = 0.5$. Furthermore we sample

any auxiliary variables dependent on the ith type of transformation, for example

$$u_{m_{k-1}} \sim \psi_{m_{k-1}\to m_k, l_{m_{k-1}\to m_k,i}}(\cdot). \tag{3.28}$$

Therefore the importance weight between two models is given by

$$\begin{aligned}
\frac{\rho_T(\theta_{m_k\cdot 0}, \omega_{m_k\cdot 0}, u_{m_k}; m_{k-1}\to m_k)}{\rho_0(\theta_{m_k\cdot 0}, \omega_{m_k\cdot 0}, u_{m_k}; m_{k-1}\to m_k)} &= \frac{\left(\sum_{j=1}^{k}\omega_{m_k a_j 0} f_{a_j}(y|\theta_{m_k a_j 0}, m_k)\right)}{\rho_0(\theta_{m_k\cdot 0}, \omega_{m_k\cdot 0}, u_{m_k}; m_{k-1}\to m_k)} \\
&\quad \times p(\theta_{m_k\cdot 0})p(\omega_{m_k\cdot 0})p(m_k) \qquad (3.29) \\
\rho_0(\theta_{m_k\cdot 0}, \omega_{m_k\cdot 0}, u_{m_k}; m_{k-1}\to m_k) &= \left(\sum_{a_{m_{k-1}\to m_k,i}|l_{m_{k-1}\to m_k,1}}\left(\right.\right. \\
&\quad \left(\sum_{j=1}^{k}\omega_{m_{k-1} a_j 0} f_{a_j}(y|\theta_{m_{k-1} a_j 0}, m_{k-1})\right) \\
&\quad \times p(\theta_{m_{k-1}\cdot 0})p(\omega_{m_{k-1}\cdot 0})\psi_{m_{k-1}\to m_k, l_{m_{k-1}\to m_k,1}}(u_{m_{k-1}}) \\
&\quad \left.\left.\times J_{m_k\to m_{k-1}}\right)\right)\frac{1}{2} \\
&\quad + \left(\sum_{a_{m_{k-1}\to m_k,i}|l_{m_{k-1}\to m_k,2}}\left(\right.\right. \\
&\quad \left(\sum_{j=1}^{k}\omega_{m_{k-1} a_j 0} f_{a_j}(y|\theta_{m_{k-1} a_j 0}, m_{k-1})\right) \\
&\quad \times p(\theta_{m_{k-1}\cdot 0})p(\omega_{m_{k-1}\cdot 0})\psi_{m_{k-1}\to m_k, l_{m_{k-1}\to m_k,2}}(u_{m_{k-1}}) \\
&\quad \left.\left.\times J_{m_k\to m_{k-1}}\right)\right)\frac{1}{2(k-1)}, \qquad (3.30)
\end{aligned}$$

where again we suppress the conditionality of all the model parameters, auxiliary variables and the Jacobians on both $a_{m_{k-1}\to m_k,\cdot}$ and $l_{m_{k-1}\to m_k,\cdot}$. The denominator of (3.30) can essentially be thought of as the sum of both the deconditioned birth move importance proposal and deconditioned split move importance sampler.

A downside to performing any form of the deconditioned adaptions is that the computational cost will at least increase linearly depending on how many model transition proposals are applied and the complexity of each proposal itself.

3.3.3 MCMC Kernel Proposals

The within-models moves are inspired from Karagiannis and Andrieu (2013). We first perform Metropolis-Hastings kernels on the vector of component weights. This is then followed by performing single component moves on each mean of ascending order, and afterwards moves are made on each precision which is again proposed based on the ascending order of the component means. Finally the shared b hyperparameter term within each precision, as seen in (3.10) and (3.11), has a single MH proposal applied.

$$\begin{aligned}
\mu^{'}_{m_k a_j t} &\sim \text{Normal}\left(\mu = \mu_{m_k a_j t}, \tau = \left(\upsilon_{\mu_{m_k a_j} t}\right)^{-1}\right) && (3.31)\\
\log(\tau^{'}_{m_k a_j t}) &\sim \text{Normal}\left(\mu = \log(\tau_{m_k a_j t}), \tau = \left(\upsilon_{\tau_{m_k a_j} t}\right)^{-1}\right) && (3.32)\\
\log(b^{'}_{m_k t}) &\sim \text{Normal}\left(\mu = \log(b_{m_k t}), \tau = \left(\upsilon_{b_{m_k} t}\right)^{-1}\right). && (3.33)
\end{aligned}$$

We could choose to set each of the tuning variances $\upsilon_{\mu_{m_k a_j} t}$, $\upsilon_{\tau_{m_k a_j} t}$ and $\upsilon_{b_{m_k} t}$ via fixed values. For example for each component mean this could involve the difference of the range $\upsilon_{\mu_{m_k a_j} t} = y_{\max} - y_{\min}$ or alternatively we use the variance of the complete sample such that $\upsilon_{\mu_{m_k a_j} t} = \text{Var}(y)$ although depending on the sample has the potential to be a larger tuning variance than the range of the data. Large variances may have a great impact on the acceptance rates if proposals breach the component ordering prior, and since by our assumption that a good tuning variance for low dimensional model might not be appropriate for high-dimensional inference is why adaptive tuning schemes are considered.

At each state in the schedule the unadjusted tuning variances are updated before each MCMC step via

$$\begin{aligned}
\upsilon^{'}_{\mu_{m_k a_j} t} &= \text{Wt.Var}(\mu_{m_k a_j t}, w_{m_k t}) && (3.34)\\
\upsilon^{'}_{\tau_{m_k a_j} t} &= \text{Wt.Var}(\log(\tau_{\mu_{m_k a_j} t}), w_{m_k t}) && (3.35)\\
\upsilon^{'}_{b_{m_k} t} &= \text{Wt.Var}(\log(b_{m_k t}), w_{m_k t}), && (3.36)
\end{aligned}$$

where, for example, Wt.Var$(\mu_{m_k a_j t}, w_{m_k t})$ is the weighted variance of the particle estimates of $\mu_{m_k a_j t}$ given the normalised particle weights of $w_{m_k t}$. As stated in chapter 2, we will attempt bound the acceptance rates between 0.2 and 0.6. Considering the final tuning variances for the means, if the acceptance rates are greater than the upper bound we choose to set the tuning variance to this particular parameter to $\upsilon_{\mu_{m_k a_j t}} = \upsilon'_{\mu_{m_k a_j t}} \times c_{m_k t}$, where $c_{m_k t} = 2 \times c_{m_k (t-1)}$ and $c_{m_k 0} = 1$. If they are less than the lower bound then we set them to $\upsilon_{\mu_{m_k a_j t}} = \upsilon'_{\mu_{m_k a_j t}} \times c_{m_k t}$ where $c_{m_k t} = 0.5 \times c_{m_k (t-1)}$. Otherwise we simply define tuning variance as $\upsilon_{\mu_{m_k a_j t}} = \upsilon'_{\mu_{m_k a_j t}} \times 1$ and $c_{m_k t} = c_{m_k (t-1)}$. While these multiplicative factors stack together and carry over to the next MCMC step, they are reset when we make a new across model move to a new model.

A proposal to the component weights is made in logit space. Firstly one weight is removed, where we choose to remove the last weight of $\omega_{m_k a_k t}$, as there exists only $k-1$ degrees of freedom given that the weight vector has to sum to one. Afterwards a random walk is made on each of the logit transformed weights of

$$\log\left(\frac{\omega'_{m_k a_j t}}{1 - \sum_{l=1}^{k-1} \omega'_{m_k a_l t}},\right) = \text{Normal}\left(\log\left(\frac{\omega_{m_k a_j t}}{1 - \sum_{l=1}^{k-1} \omega_{m_k a_l t}}\right), (\upsilon_\omega)^{-1}\right), \quad (3.37)$$

and $\omega'_{m_k a_k t} = 1 - \sum_{l=1}^{k-1} \omega_{m_k a_l t}$. After prior testing we do not apply adaptive tuning to the weights, as the same scheme used for the means and precisions proved not to be effective. Therefore a fixed tuning variance υ_ω is considered instead, whose value is dependent on the data.

3.4 Tests and Adaptions for Univariate Mixture Models

As discussed previously in chapter 2 we will consider:

- If satisfactory convergence of the posterior has been achieved, in comparison to established techniques.
- The Monte Carlo variance of the marginal likelihood formulated of tSMC.
- In the case of having subsets of particle using either birth or split moves at certain states, we investigate the competition between the two across model moves.
- The ESS decay and its relationship to the discrepancies between intermediates distributions.

We test our tSMC adaption on two datasets, displayed in figures 3.1 and 3.2. One is the enzyme dataset containing 254 individuals introduced in Bechtel *et al.* (1993), and the second dataset is the adjusted galaxy dataset from Roeder (1990) containing 82 observations.

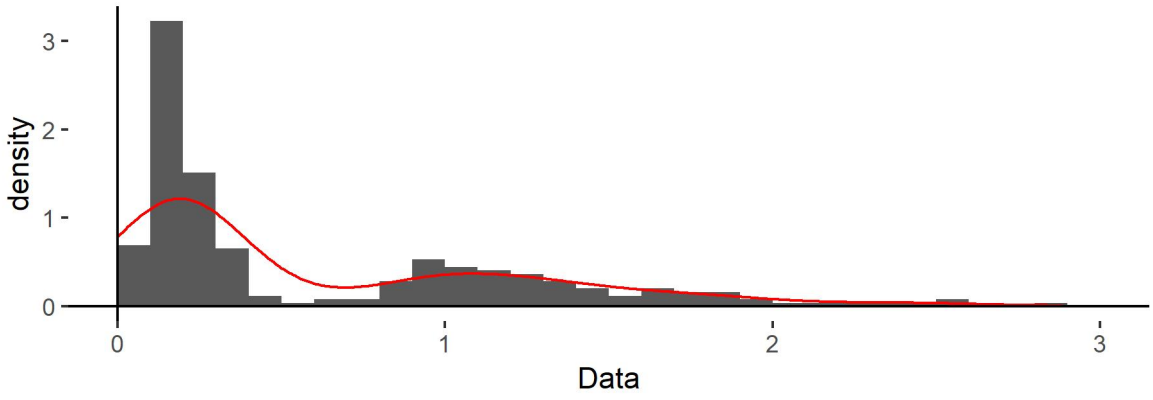


Figure 3.1: Kernel density plot for the enzyme data.

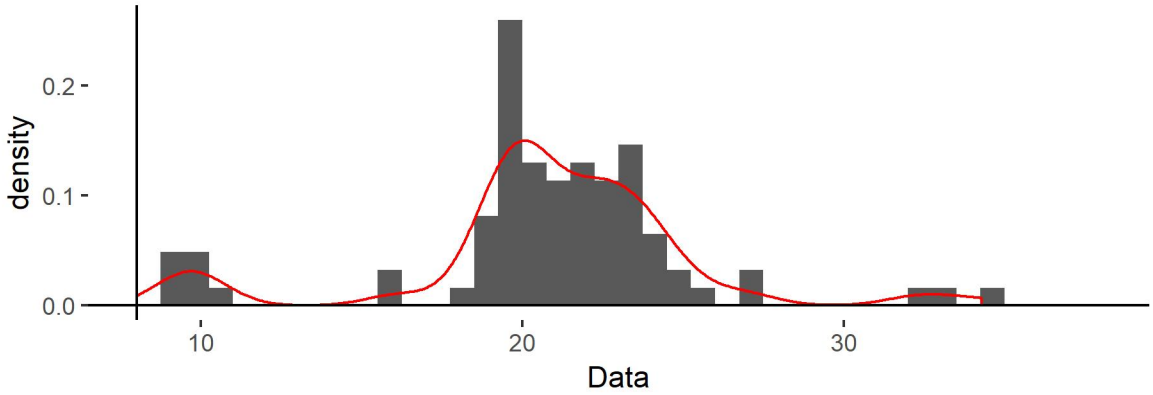


Figure 3.2: Kernel density plot forthe galaxy data.

For results used for displaying purposes we run tSMC for 10000 particles, under the priors and kernel moves mentioned in section 3.3. Otherwise for tests that involve the calculation of the marginal likelihood we run this up to eight univariate Gaussian components, at 250 particles with two sets of 50 runs in total. One set considers adaptively defining the number of intermediate distributions (adaptive annealing) and other uses a fixed number of intermediate distributions (fixed annealing).

Under the fixed annealing scheme we apply a series of intermediate distributions dictated by $\varphi_t = (t/T)^5$ with the total length of the set φ being 100.

In regards to choosing the CESS target to dictate adaptive annealing scheme, we choose to set it to $0.95P$. Zhou *et al.* (2016) set the CESS decay to $0.999P$ when applying this measurement to a standard annealed SMC algorithm (see Del Moral *et al.* (2006)), where for each model the associated priors were used as an importance proposal, and this lead to a total of 180-200 intermediate steps. However we have chosen to set it lower and aim to have at most 100 intermediate distributions per transition when using adaptive annealing, in which we analyse if adaptive annealing can still provide good convergence even though more intermediate distributions guarantees convergence. We also briefly illustrate the problems of using the ESS to adaptively define the intermediate distributions.

We compare the results of tSMC using am annealed SMC sampler algorithm that

uses the prior as our importance sampler, although due to the variable computational cost of adaptive annealing a true comparison can only be done with fixed annealing. For example given that our fixed annealing scheme applies 100 intermediate distributions per model transition then our estimates of the marginal likelihood for an eight component univariate Gaussian distribution from the tSMC algorithm is compared with annealed SMC algorithm runs that have a total of $100 \times 8 = 800$ intermediate distributions with discrepancies dictated by $\varphi_t = (t/T)^5$ and having the same particle size. When determining our best possible obtainable estimated marginal likelihood and posterior distribution, we run extremely long runs of the said SMC algorithm with 5000 particles and under 1000 intermediate steps.

Furthermore we compare our results with a long run of the RJMCMC adaption by Richardson and Green (1997). Their algorithm is exactly given in the "Miscellaneous Functions" CRAN package by Feng (2018), in which we can use the priors as described in section 3.3.1. However we should note that the algorithm considers the "with completion" adaption which attempts to infer the allocation variables, although there should be no difference in the Bayes factor or marginal likelihood estimates compared to the "without completion" model (providing that all prior distributions have the same hyperparameters) since the latter just have the allocation variables integrated out. The algorithm is run for $6,000,000$ iterations with a burn in of $1,000,000$ iterations.

We will consider using Monte Carlo error per number of likelihood calculations should the final results be very similar to each other. Regarding the comparison between tSMC and RJMCMC results we will compare the posterior odds (or Bayes factor) between each adjacent model. As we are using a uniform prior for the appropriateness of each model, given by (3.13), then the Bayes factors between each adjacent model is the equivalent of the posterior odds.

The main objective of this chapter is to analyse the properties of the tSMC algorithm by considering the following:

- Firstly we illustrate a few properties of the algorithm, especially regarding the extensions to the basic tSMC algrotihms such as the differences between using an ESS and CESS threshold to control the discrepancies between intermediate distributions when defining them adaptively.

- The second round of tests involves analysing the performance of both the birth and split move, under a fixed annealing and adaptive annealing target distribution. We repeat the two scenarios above, except we apply their respective deconditioned adaptions variants as stated in section 3.3.2.3.

- We investigate if tSMC can apply multiple transformations, in this case randomly assigning a particle one of the two moves, to set the ground work for adaptions that may make use of multiple proposals in in the future.

- We analyse the performance of tSMC when applying arithmetic series of target distributions. The tests will be based on the three previous points.

3.5 Results

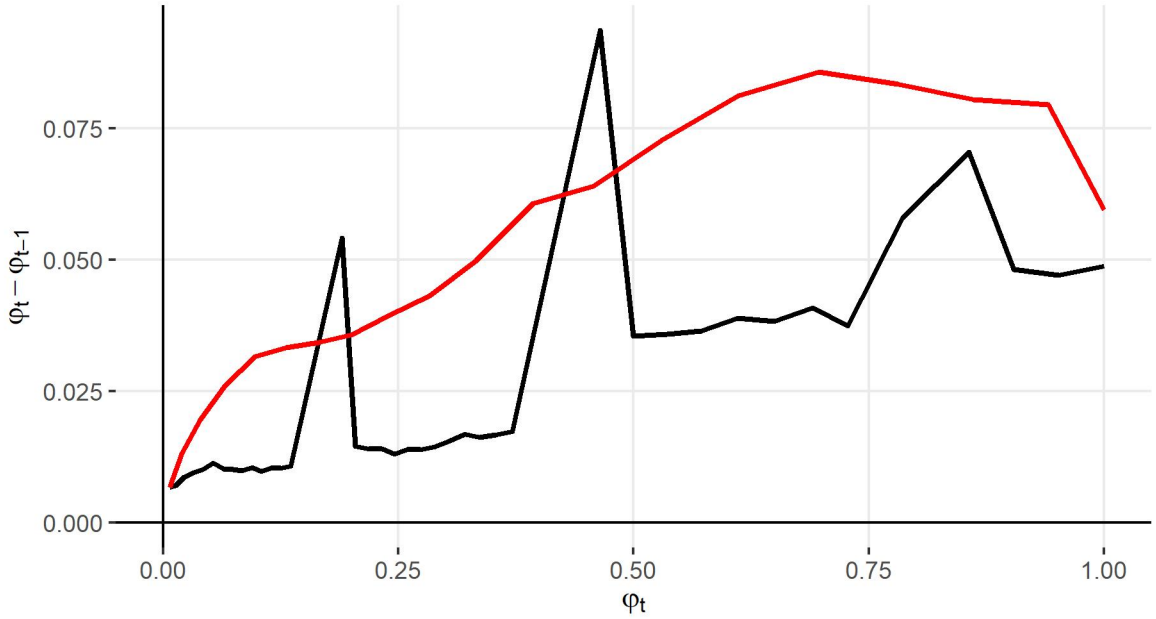
We considered some pre-testing to analyse whether the tSMC algorithm was at least functioning correctly and converging to a posterior that resembles an estimate generated by a standard SMC algorithm. We also explain other alterations to our adaptions due to discoveries during trial runs of each main tSMC adaption. Given an initial transformation for some parameter set $\{\theta_{m_{k-1}\cdot}, \omega_{m_{k-1}\cdot}, u_{m_{k-1}}\}$ to be transitioned to $\{\theta_{m_k\cdot}, \omega_{m_k\cdot}, u_{m_k}\}$, both of the applied transformations produced a large number of negligible weights which naturally lead to a low effective sample size. Furthermore the large variance between weights could also be due to the distance between the two intermediate distributions of $\rho_0(\cdot; m_{k-1} \to m_k)$ and $\rho_1(\cdot; m_{k-1} \to m_k)$, but this aspect would vary depending on model assumptions. Under fixed annealing we have a predefined value for φ_1, which gives an ESS estimate that usually leads to a resampling

step. However under the adaptive selection of intermediate distributions, many trial runs could not identify φ_1 that would have a low enough particle variance defined through the CESS. To compensate for this issue we have automatically set $\varphi_1 = 10^{-8}$ and perform adaptive annealing of each φ_t for the remaining transition states. This was a method also used by Jasra *et al.* (2011).

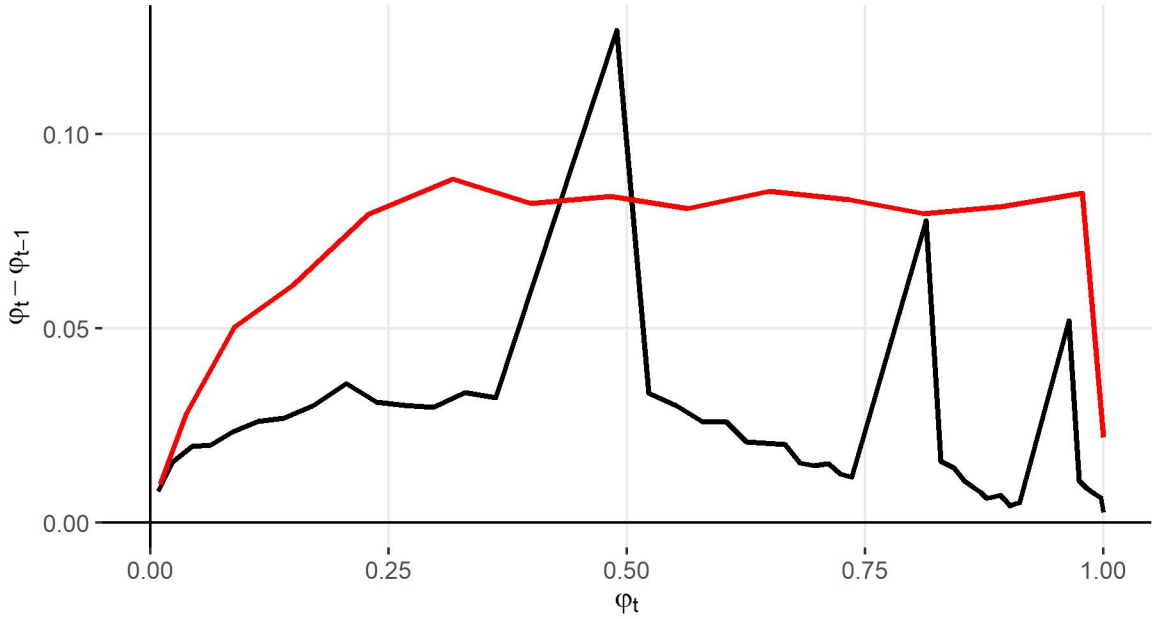
On a minor note, adaptively tuning the MCMC kernel was found not to work well when applied to a vector of component weights, and therefore we considered a fixed constant for the tuning variance of the component weights. The reason for this decision is because using a Gaussian walk with a weighted variance on the logit transformed variables often gave proposals that had little or large mass on a subset of weights, and were often rejected as a result. Under the enzyme dataset a tuning standard deviation of 0.4 for the logit weights proved to give acceptable weights between 0.2 to 0.6, which is roughly where we aim when it come to giving proposals to a multivariate object, and for the galaxy dataset we set this value to 0.8.

When using the Effective Sample Size as a measure to dictate the discrepancy between each $\{\varphi_{t-1}, \varphi_t\}$, see chapter 2 and the issues of using this as a measure of distance between two intermediate distributions, we consider a scheme of $\text{ESS}_i = 0.95\text{ESS}_{i-1}$ where $\text{ESS}_0 = N$. We compared this to the adaption that uses a fixed CESS target of $0.95N$. We analysed the history of the annealing discrepancies of each subsequent intermediate distribution when transitioning to an eight component model when using the split move.

As seen in figures 3.3, and 3.4 we have a very similiar pattern to what was shown in Zhou *et al.* (2016). The ESS scheme gave notably large discrepancies after a re-sampling step, but afterwards there would be a sequence of decreasing and eventually stable discrepancies until the next resampling step. In comparison using the CESS mostly gave gradually increasing discrepancies, although depending on the transition it could decrease but at a gradual rate, or otherwise the discrepancies between intermediate distribution remained stable after each reweighting step. Most importantly

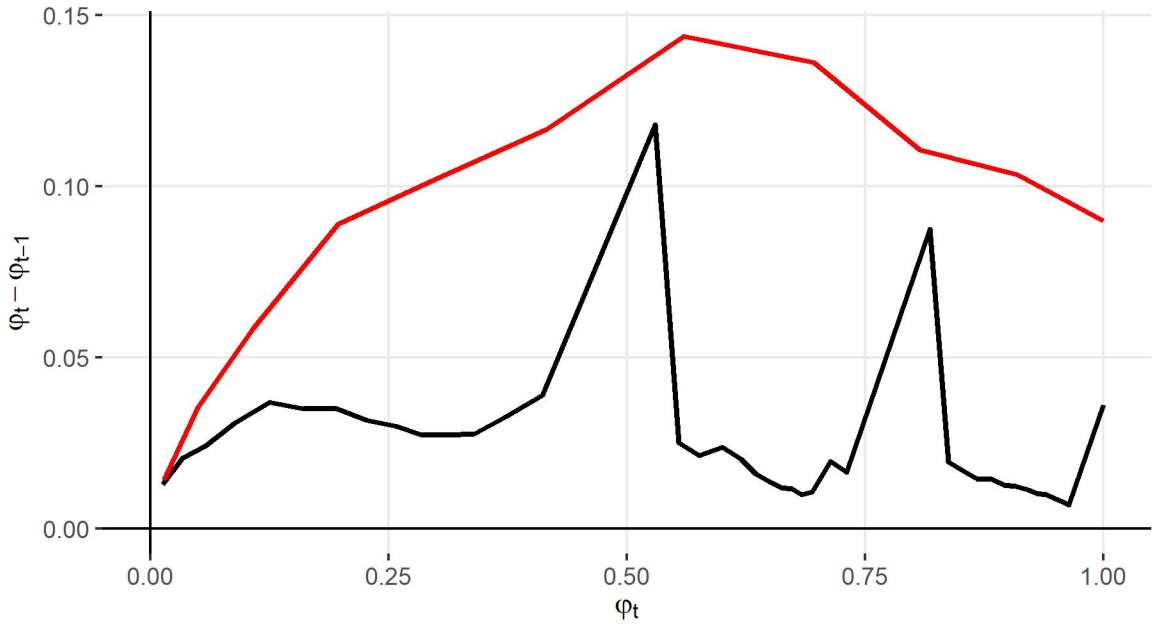


(a) Discrepancies of $\varphi_t - \varphi_{t-1}$ when transitioning from one to two Gaussian components.

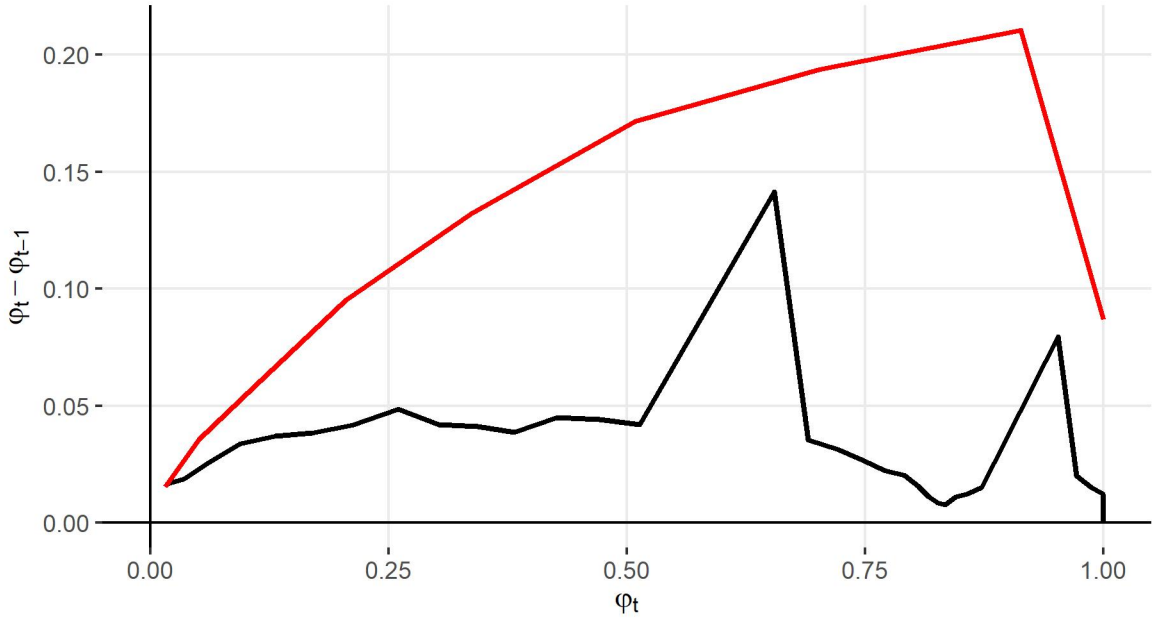


(b) Discrepancies of $\varphi_t - \varphi_{t-1}$ when transitioning from three to four Gaussian components.

Figure 3.3: Discrepancies, $\varphi_t - \varphi_{t-1}$, between intermediate distributions over time for low dimensional transitions. The black line represents the ESS dictated discrepancies, with the red line representing CESS dictated discrepancies.



(a) Discrepancies of $\varphi_t - \varphi_{t-1}$ when transitioning from five to six Gaussian components.



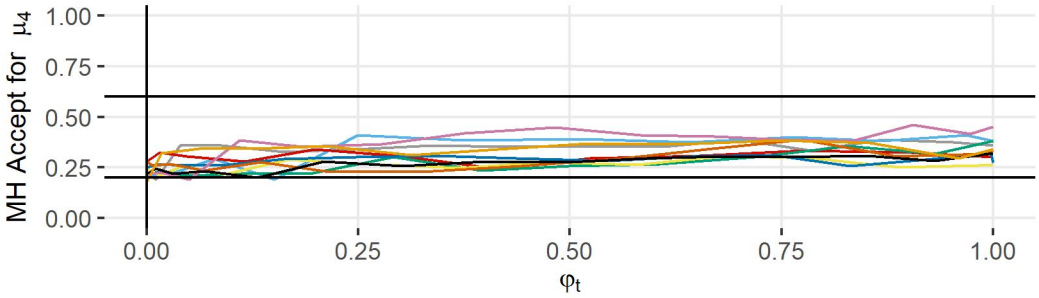
(b) Discrepancies of $\varphi_t - \varphi_{t-1}$ when transitioning from seven to eight Gaussian components.

Figure 3.4: Discrepancies, $\varphi_t - \varphi_{t-1}$, between intermediate distributions over time for high dimensional transitions. The black line represents the ESS dictated discrepancies, with the red line representing CESS dictated discrepancies.

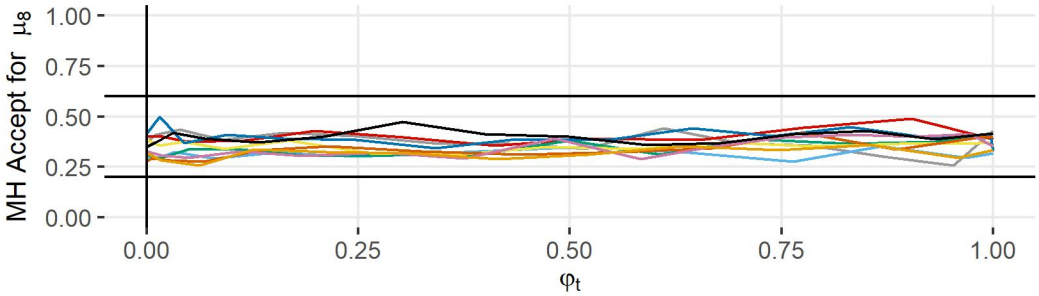
was that it was not affected by resampling. In either scheme we believe there wasn't any noticeable deviance of the posterior parameter estimation. Thus we believe using the CESS to dictate the discrepancies between each intermediate distribution to be the preferred option. The bisection method was shown to be fast enough for purpose to find each φ_t that corresponding to the CESS target, and no further effort was made to find faster root finding methods. Finally we do not make anymore comparisons between the CESS and ESS in the remaining chapters.

We considered how stable the acceptance probabilities were within the intermediate distributions, and if the various weighted variance tuning proposals for each parameter gave good initial acceptance probabilities (from the first intermediate distribution when transitioning from the two models). Figure 3.5 shows the acceptance rates for some of the model transitions, with the majority showing a similar pattern. What can be seen for most of these parameters is that they reach some steady convergence of acceptance probabilities and there are no major drops in the acceptance rates. This is slightly similar to what was shown in Jasra *et al.* (2011), who used the same adaptive algorithm but set the acceptance rates to be between 0.15 and 0.7, although the application showed this steady state to be far less variable on some parameters. From these tests we choose to apply our adaptive tuning scheme for all continuous parameters in this chapter and beyond.

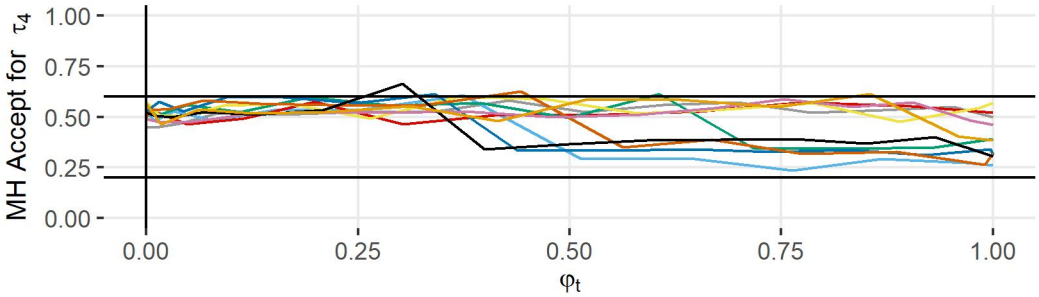
We display the estimated joint posterior densities in figures 3.6 to 3.9. These were specifically generated under 10000 particle under adaptively annealed intermediate distributions. While we only show the posterior distributions for the deconditioned birth move and deconditioned split move in these figures, all different adaptions of the tSMC algorithm gave approximately the same results as SMC and gave a good representation of the data itself. Although as we explain in the rest of this subsection, each of the tested transformations converged at different rates and gave different estimates of the marginal likelihood. Regarding the results from the galaxy dataset, under any of the schemes that infer a two Gaussian component univariate mixture model their



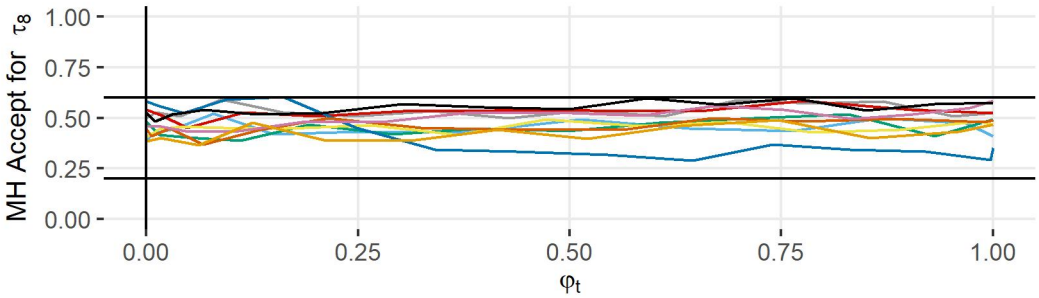
(a) Acceptance Probabilities for the fourth ordered mean.



(b) Acceptance probabilities for the eighth ordered mean.

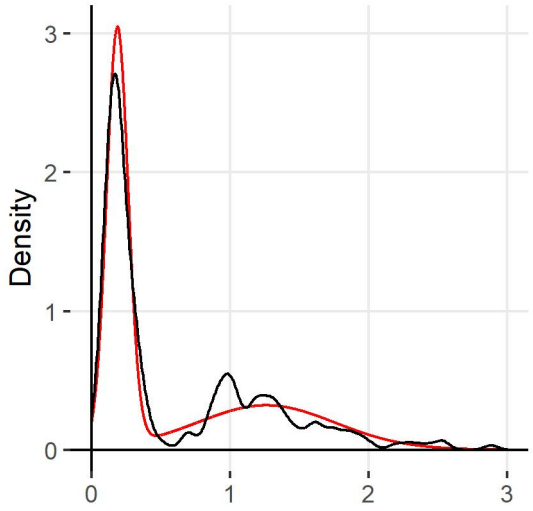


(c) Acceptance probabilities for the fourth ordered precision.

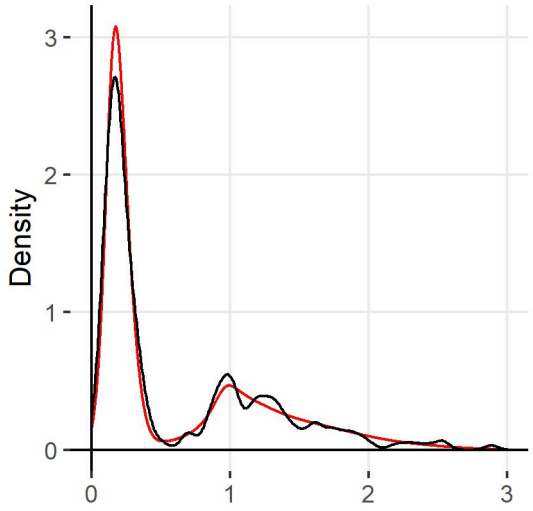


(d) Acceptance probabilities for the eighth ordered precision.

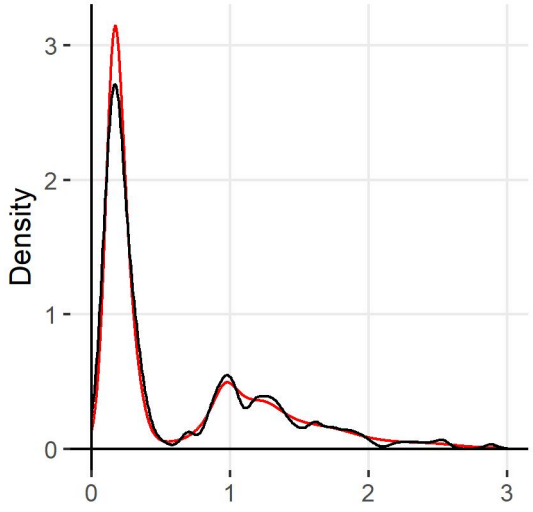
Figure 3.5: Acceptance probability plots for MH moves for several parameters when transitioning to an eight component univariate Gaussian distribution, this is shown over 10 runs of the tSMC algorithm.



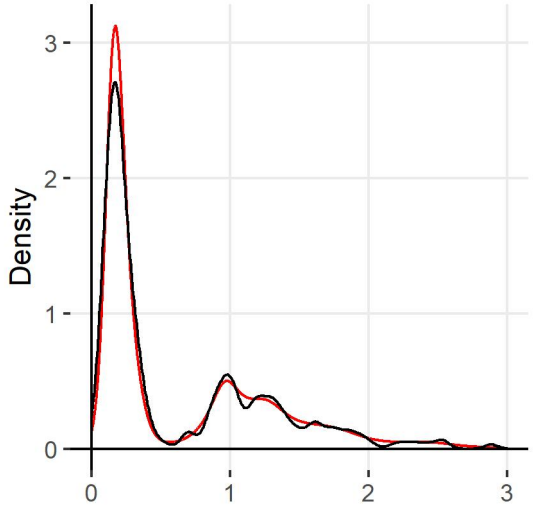
(a) Estimated posterior for two Gaussian components.



(b) Estimated posterior for four Gaussian components.

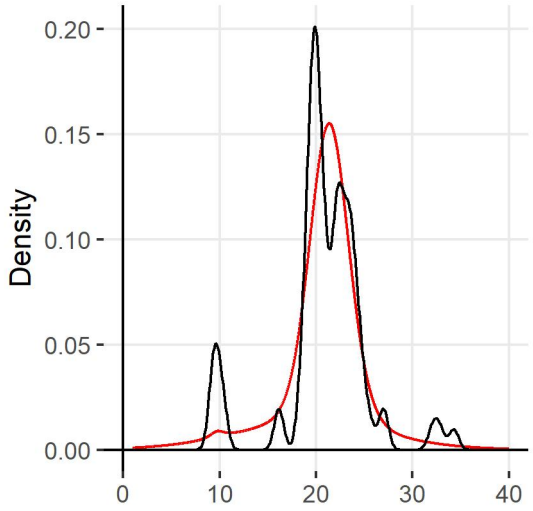


(c) Estimated posterior for six Gaussian components.

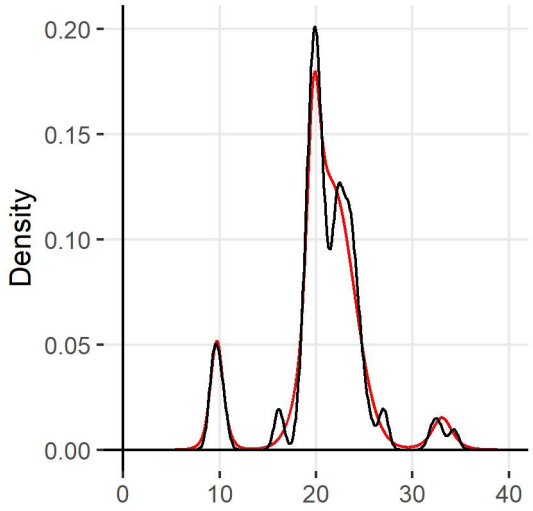


(d) Estimated posterior for eight Gaussian components.

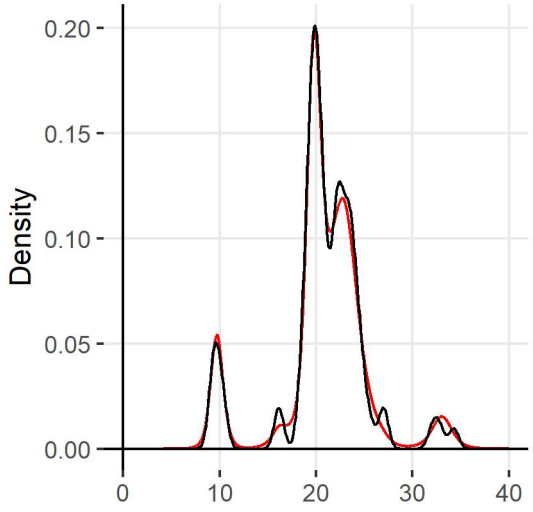
Figure 3.6: Estimated posterior density plots under the deconditioned birth transformation for the enzyme dataset. The red line represents the tSMC estimate and the black line represents the kernel density estimates of the data.



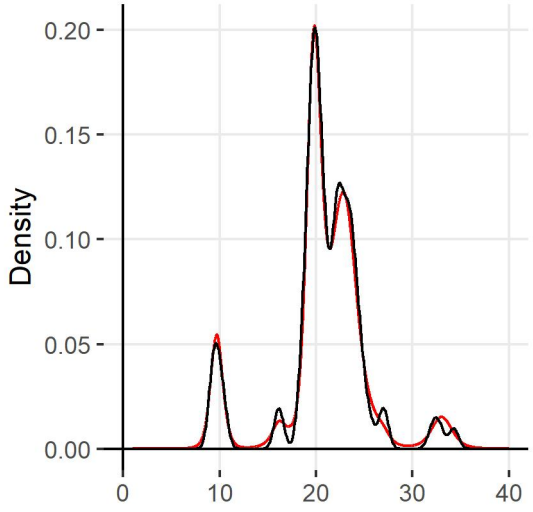
(a) Estimated posterior for two Gaussian components.



(b) Estimated posterior for four Gaussian components.

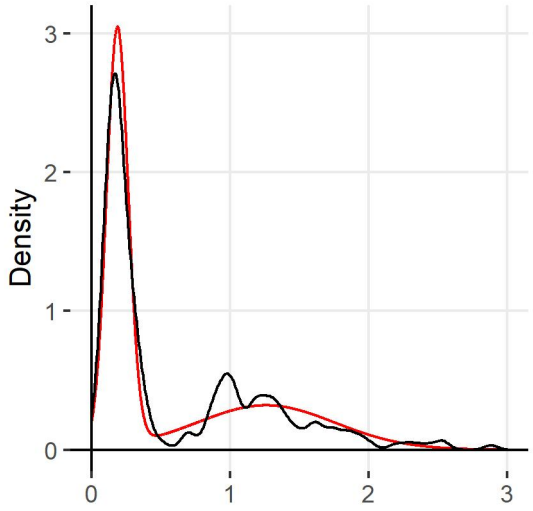


(c) Estimated posterior for six Gaussian components.

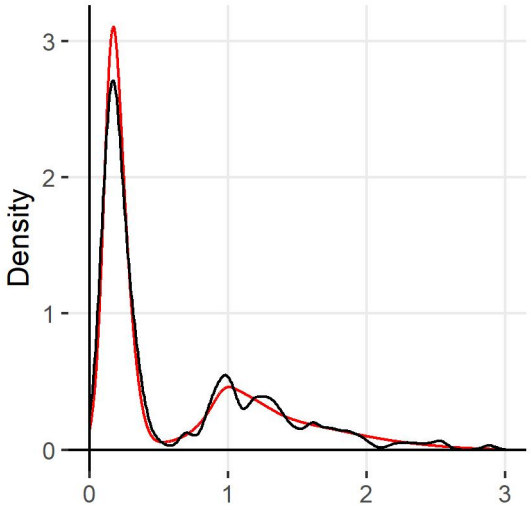


(d) Estimated posterior for eight Gaussian components.

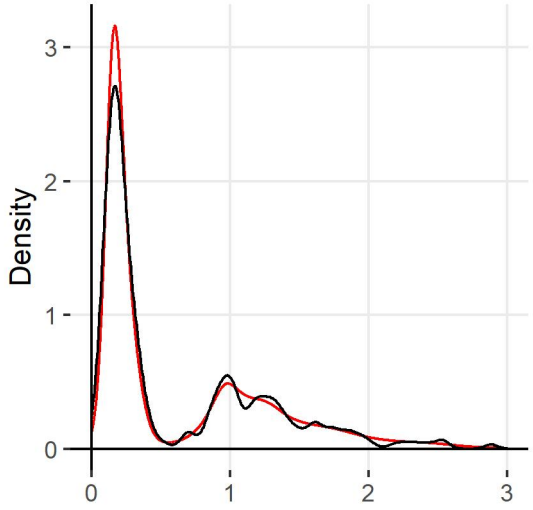
Figure 3.7: Estimated posterior density plots under the deconditioned birth transformation for the galaxy dataset. The red line represents the tSMC estimate and the black line represents the kernel density estimates of the data.



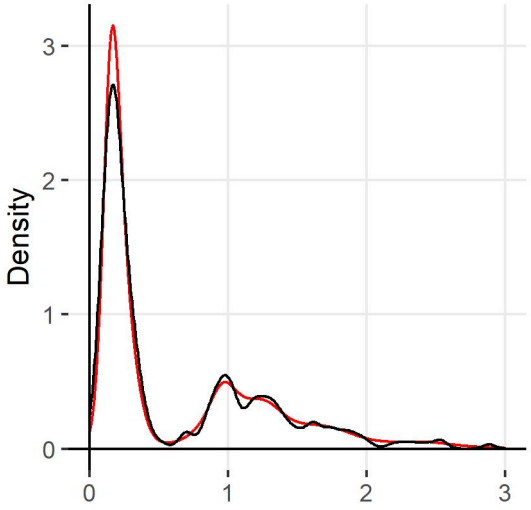
(a) Estimated posterior for two Gaussian components.



(b) Estimated posterior for four Gaussian components.

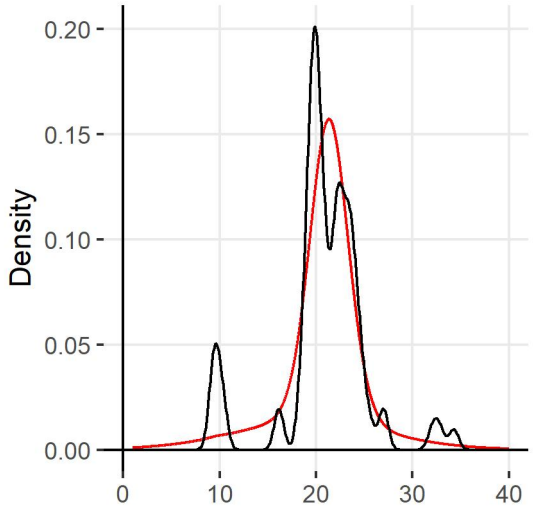


(c) Estimated posterior for six Gaussian components.

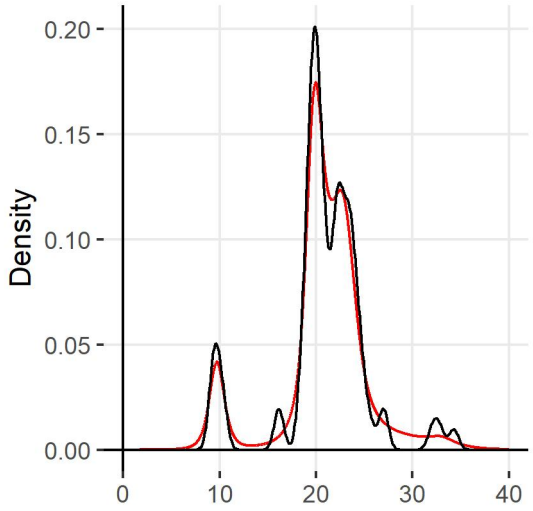


(d) Estimated posterior for eight Gaussian components.

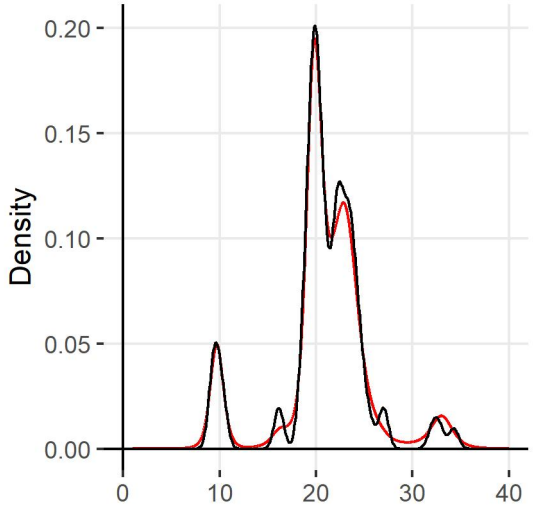
Figure 3.8: Estimated posterior density plots under the deconditioned split transformation for the enzyme dataset. The red line represents the tSMC estimate and the black line represents the kernel density estimates of the data.



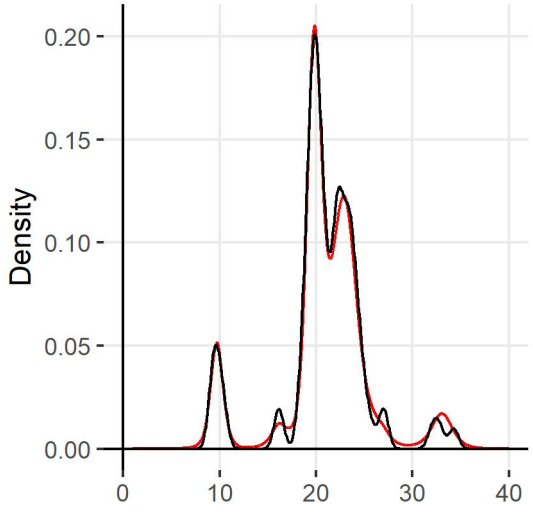
(a) Estimated posterior with two Gaussian components.



(b) Estimated posterior with four Gaussian components.



(c) Estimated posterior with six Gaussian components.



(d) Estimated posteriorwith eight Gaussian components.

Figure 3.9: Estimated posterior density plots under the deconditioned split transformation for the galaxy dataset. The red line represents the tSMC estimate and the black line represents the kernel density estimates of the data.

respective component Gaussian means are very close to each giving the impression of one Gaussian component. While we do not display the individual posterior densities for each parameter, there was no posterior density in minuscule subsets of the parameter (i.e no unusual peaks in the marginal density plots). Applying a fixed annealing schedule gave similar answers.

In figures 3.10 and 3.11 we display the distributions of the number of intermediate distributions, under each of the 50 adaptive annealing runs, it took to fully transition from a model only containing prior assumption for a single Gaussian model to each of the two to eight Gaussian component models. The performance of both the birth move and the deconditioned birth move performed poorly in comparison to their split transformation counterparts and more notably this large difference occurred even during the transition to a two Gaussian component model where we go further in depth of what exactly is happening within this transition with figures 3.12 to 3.15. In either case the deconditioned moves performed better in comparison to their conditioned counterparts. The performance of each move depended on the dataset, where in the enzyme data the deconditioned split move required the the least intermediate distributions on average to transition to an univariate Gaussian distributions with eight components, but in the galaxy data both the the deconditioned split move and the deconditioned birth/split move had roughly equal distribution of required intermediate distributions.

Plots for the ESS over φ_t, are shown from figures 3.12, to 3.16. We mainly plot results from the deconditioned adaptions, whose general pattern of particle degeneracy is mostly replicated by their respective conditional counterparts. For all the birth adaptions, shown in figure 3.12, there was an increasing rate of degeneracy as $\varphi_t \to 1$ when transitioning from 1-4 Gaussian components when applied with the enzyme dataset. As for the split transformation proposals, such a decay is not present with an example shown in figure 3.14. Figure 3.13 displays the ESS for the deconditioned birth move under the galaxy data, where it did not show the same rapid particle

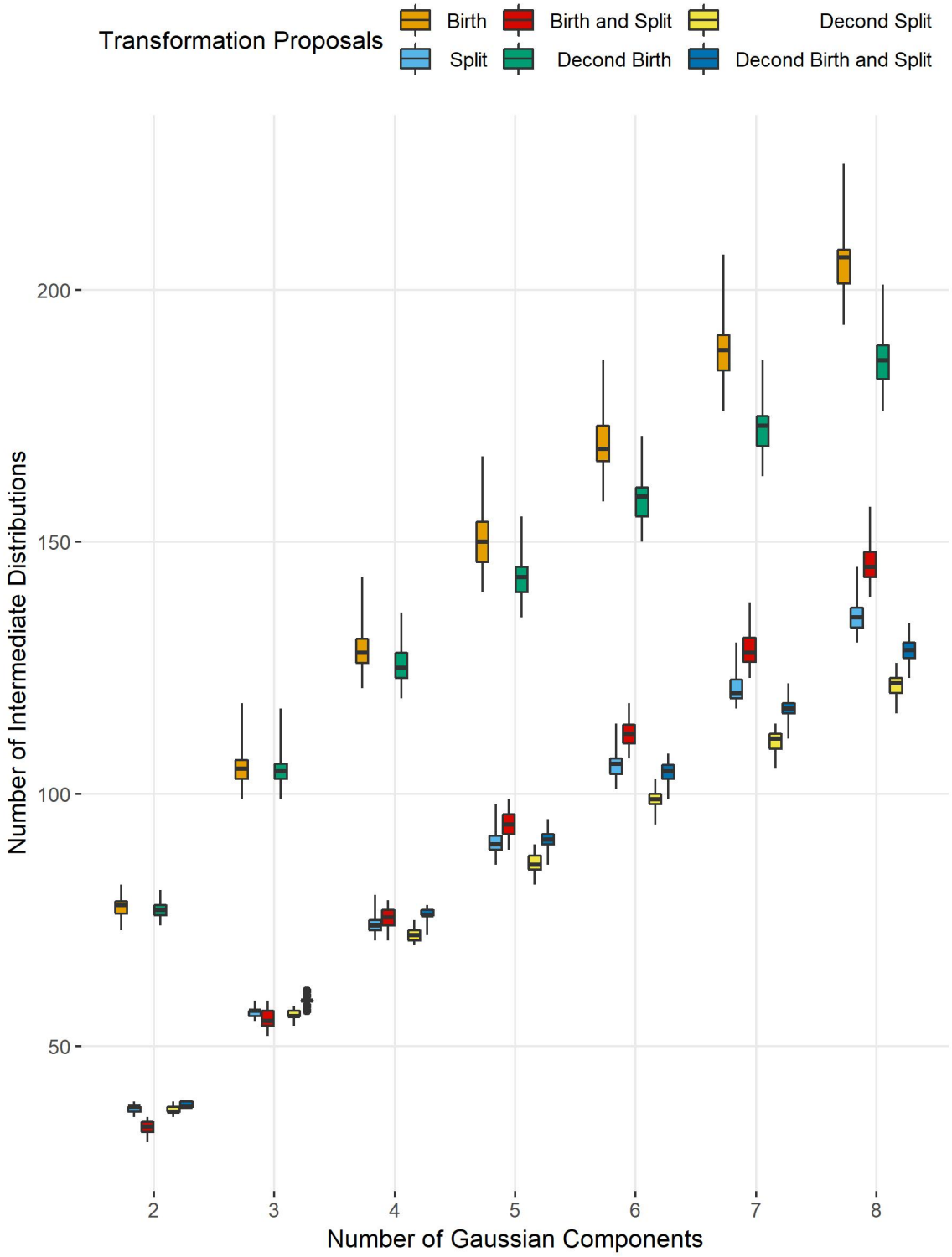


Figure 3.10: Cumulative number of intermediate distributions, from one to eight Gaussian component mixture, for the enzyme data.

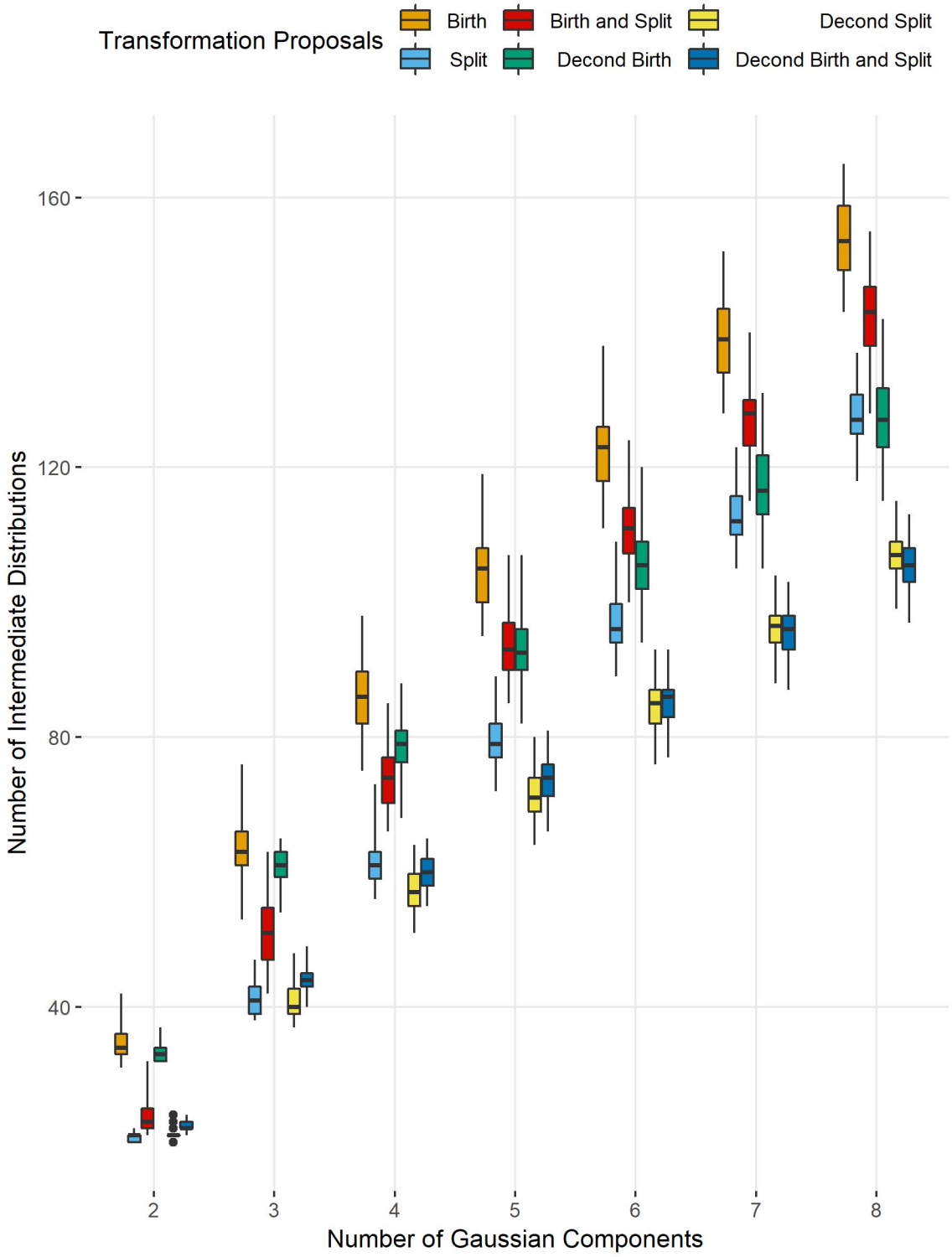
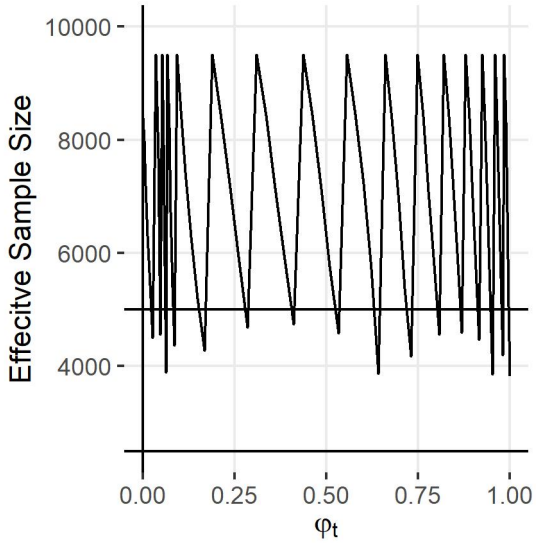
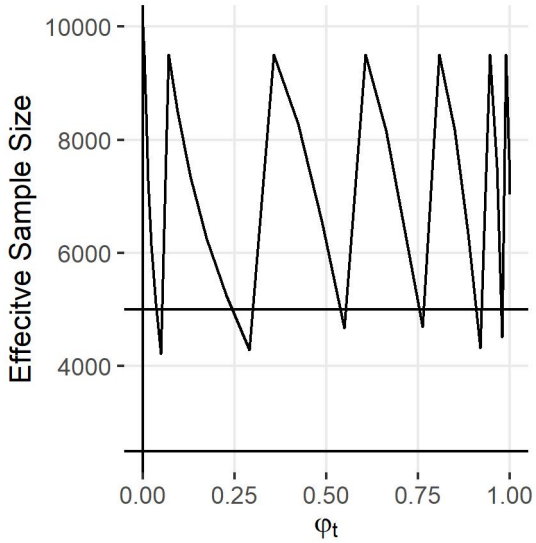


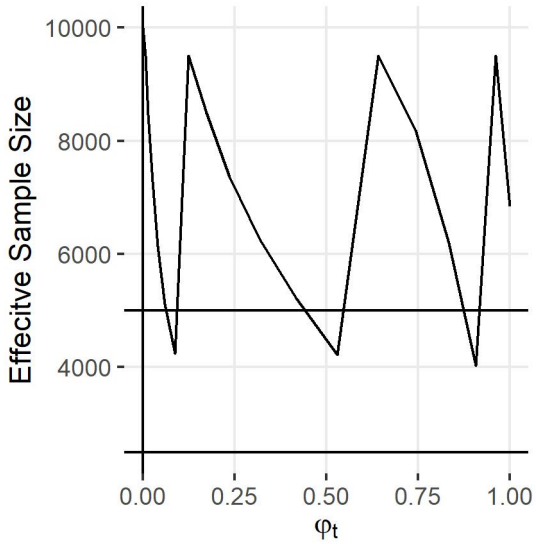
Figure 3.11: Cumulative number of intermediate distributions, from one to eight Gaussian component mixture, for the galaxy data.



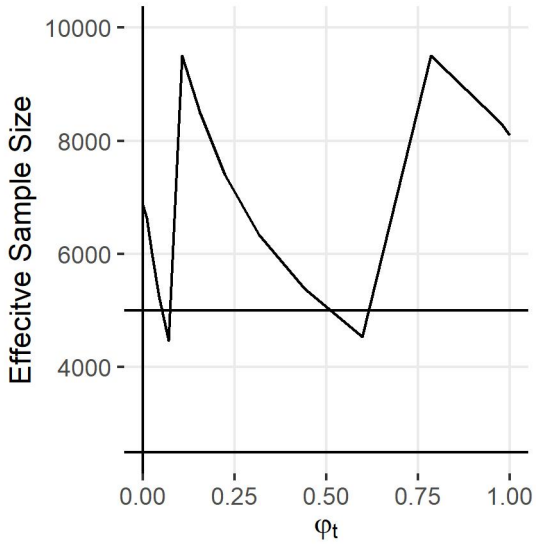
(a) ESS when transitioning from one to two Gaussian components.



(b) ESS when transitioning from two to three Gaussian components.

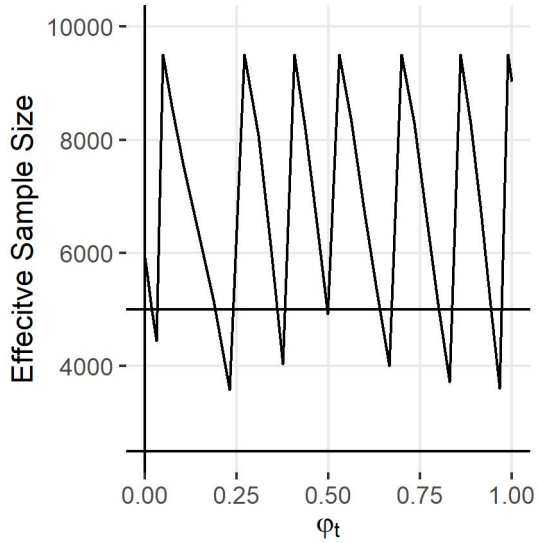


(c) ESS when transitioning from four to five Gaussian components.

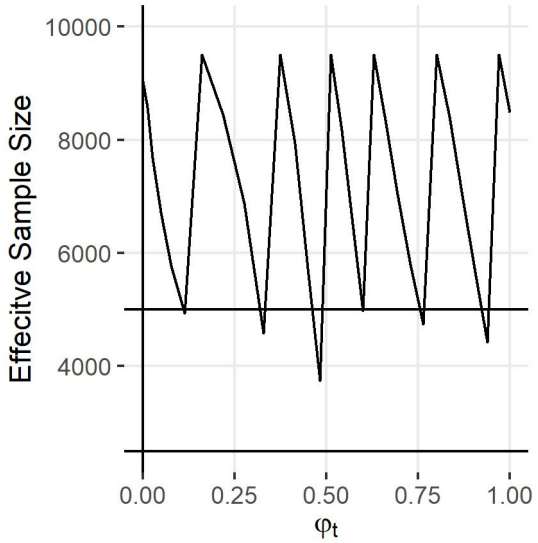


(d) ESS when transitioning from seven to eight Gaussian components.

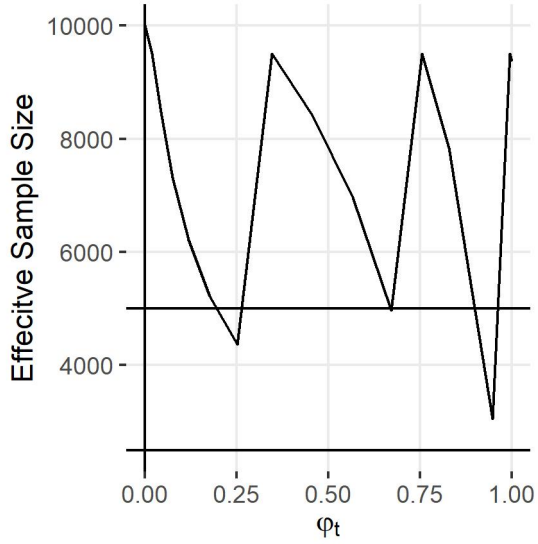
Figure 3.12: Effective sample size plots when applying the deconditioned birth move for the enzyme data. The straight line at ESS = 5000 represents the threshold for resampling.



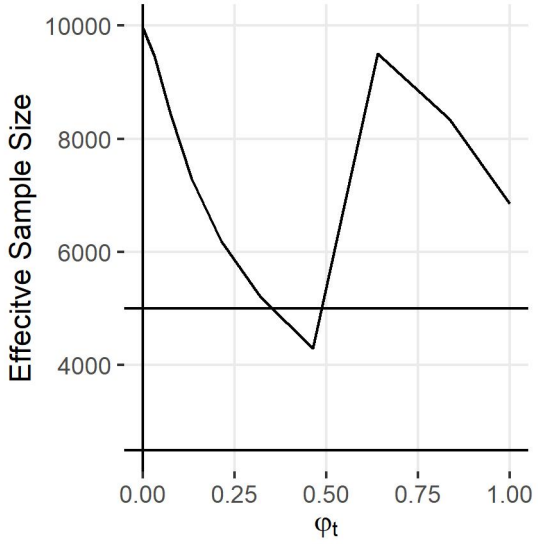
(a) ESS when transitioning from one to two Gaussian components.



(b) ESS when transitioning from two to three Gaussian components.

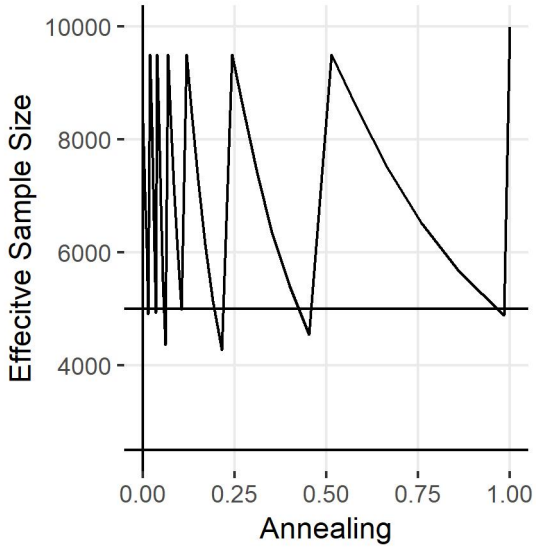


(c) ESS when transitioning from four to five Gaussian components.

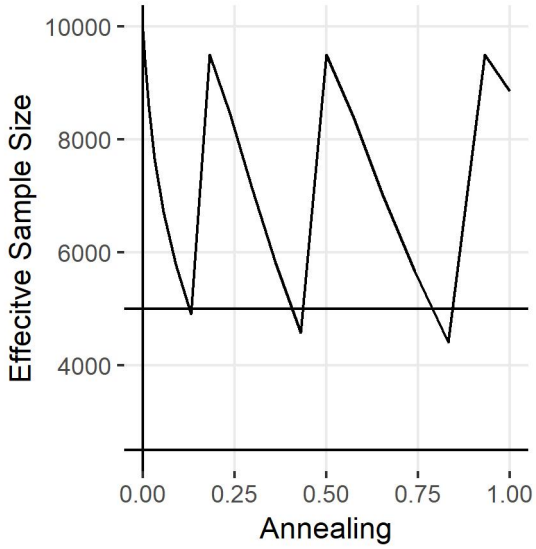


(d) ESS when transitioning from seven to eight Gaussian components.

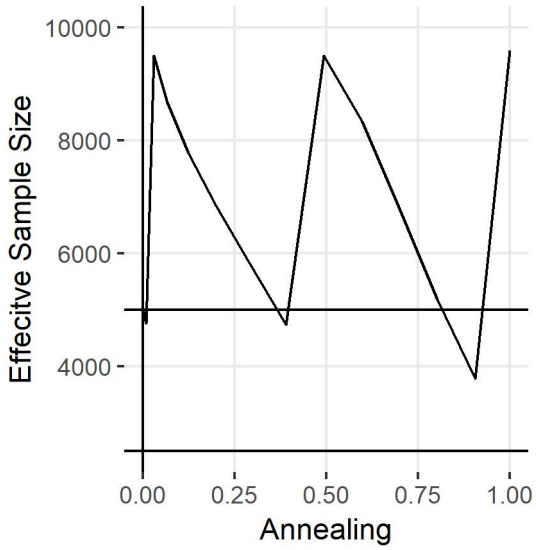
Figure 3.13: Effective sample size plots when applying the deconditioned birth move for the galaxy data. The straight line at ESS = 5000 represents the threshold for resampling.



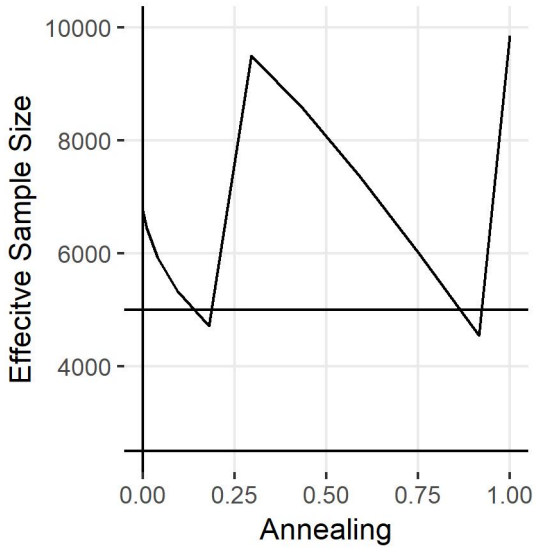
(a) ESS when transitioning from one to two Gaussian components.



(b) ESS when transitioning from two to three Gaussian components.

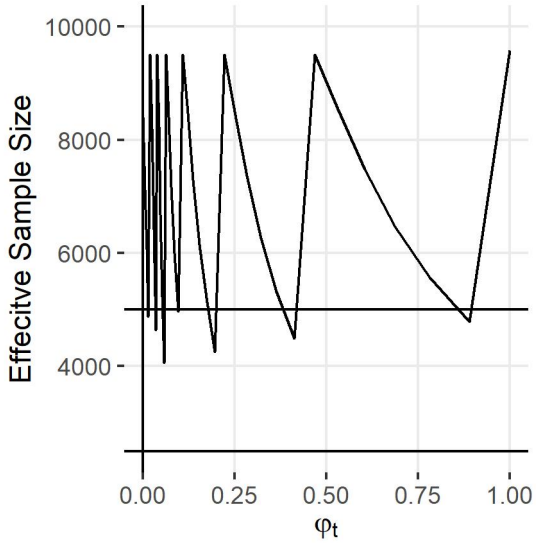


(c) ESS when transitioning from four to five Gaussian components.

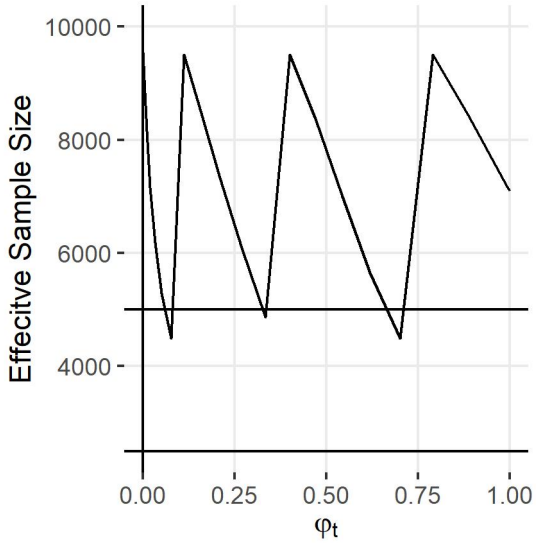


(d) ESS when transitioning from seven to eight Gaussian components.

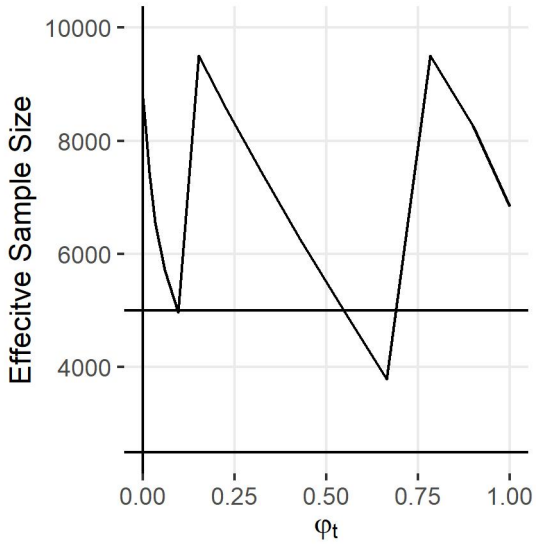
Figure 3.14: Effective sample size plots when applying the deconditioned split move for the enzyme data. The straight line at ESS = 5000 represents the threshold for resampling.



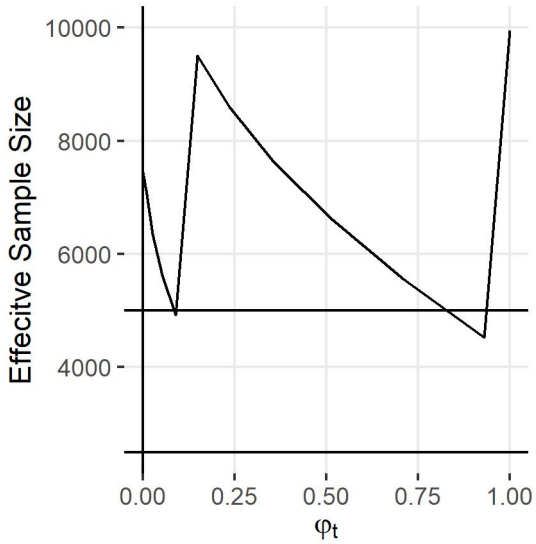
(a) ESS when transitioning from one to two Gaussian components.



(b) ESS when transitioning from two to three Gaussian components.

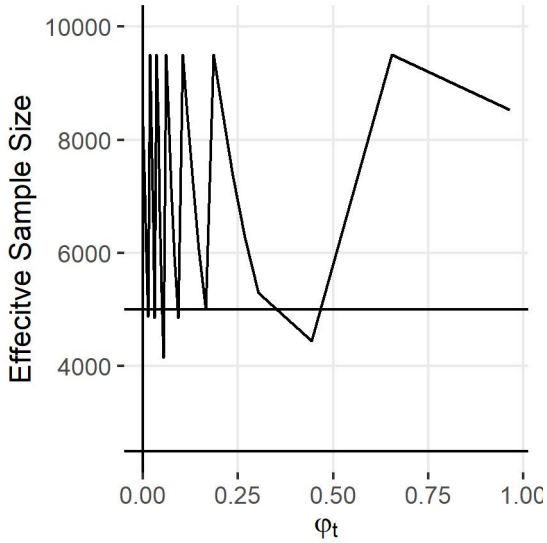


(c) ESS when transitioning from four to five Gaussian components.

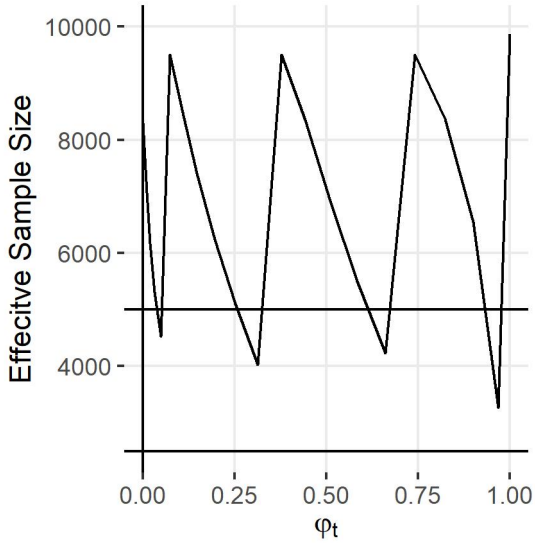


(d) ESS when transitioning from seven to eight Gaussian components.

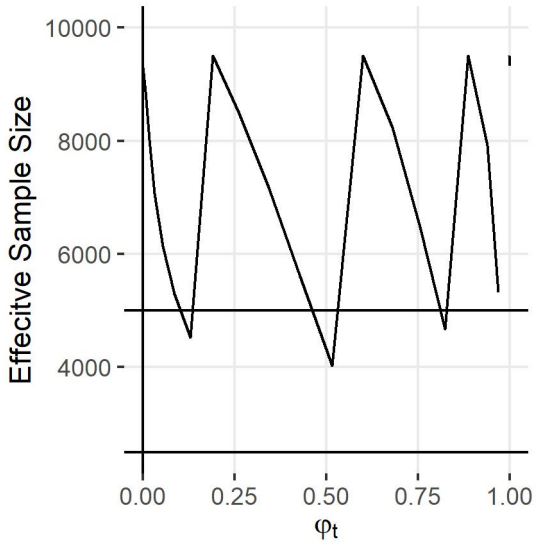
Figure 3.15: Effective sample size plots when applying the deconditioned birth and split move for the enzyme data. The straight line at ESS = 5000 represents the threshold for resampling.



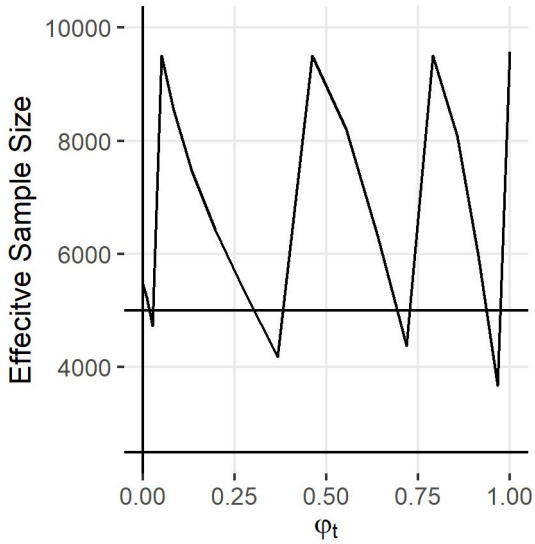
(a) ESS when transitioning from one to two Gaussian components.



(b) ESS when transitioning from two to three Gaussian components.



(c) ESS when transitioning from four to five Gaussian components.



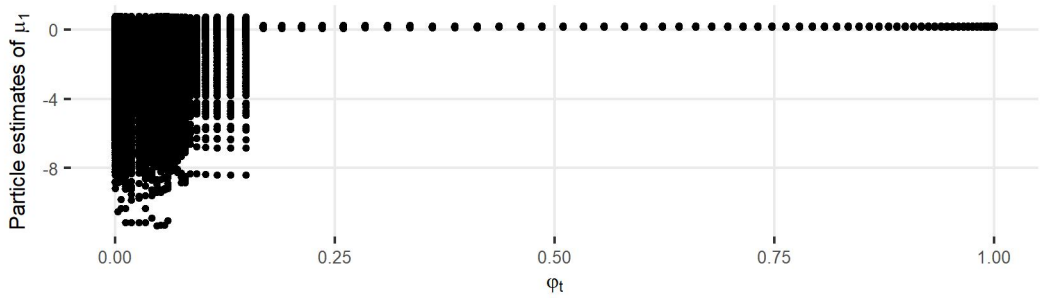
(d) ESS when transitioning from seven to eight Gaussian components.

Figure 3.16: Effective sample size plots when applying the conditioned birth and split move for the enzyme data. The straight line at ESS = 5000 represents the threshold for resampling.

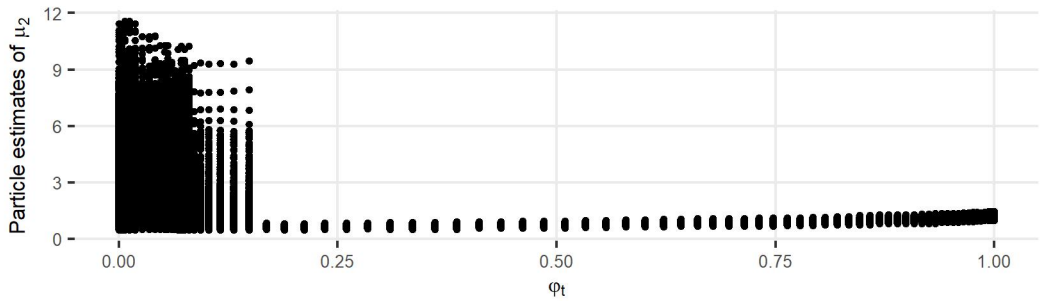
degeneracy as $\varphi_t \to 1$ in comparison to when the transformation proposal was made to the enzyme data. What we note in figures 3.15 and 3.16 is that even though some particles had the birth transformation move applied to them, given that some particle had the split move assigned and otherwise we decontioned over the type of transformation applied to the particles caused the particle degeneracy to not show the same patterns as seen in 3.12. At higher dimensional model transitions the ESS starts to drop less rapidly as $\varphi_t \to 1$ for all six transformation proposals.

We also give particular attention when transitioning from one Gaussian component to two Gaussian components, under both of the deconditioned for the birth and split transformations when the intermediate distributions are adaptively made. From figures 3.17 to 3.20 when performing the birth move on the enzyme data what can be noticed is how the particles representing one of the component means gradually shift into areas that originally did not have any estimated probability mass during the mid to late states of the transition, while the other mean has an estimated distribution that remains roughly the same, as $\varphi_t \to 1$. Furthermore this is accompanied with a large number of reweighting steps due to particle degeneracy, as discussed earlier. The pattern is also repeated for the component precision, and similar results are present for the conditioned transformation variants as well. This highlights that proposing a new component from prior conditions, without any changes to the existing distribution, was not appropiate to estimate a two component univariate Gaussian mixture model, even though it does eventually estimate the posterior density similarly to that of the split move adaptions as seen in figures 3.6 and 3.7 (as well as the particle plots). For the galaxy data there was no sudden shift in the particle estimates, despite requiring more intermediate steps than the enzyme data. In general the split move's performance was superior on both datasets, and converged faster to the posterior distribution of a two component univariate mixture model.

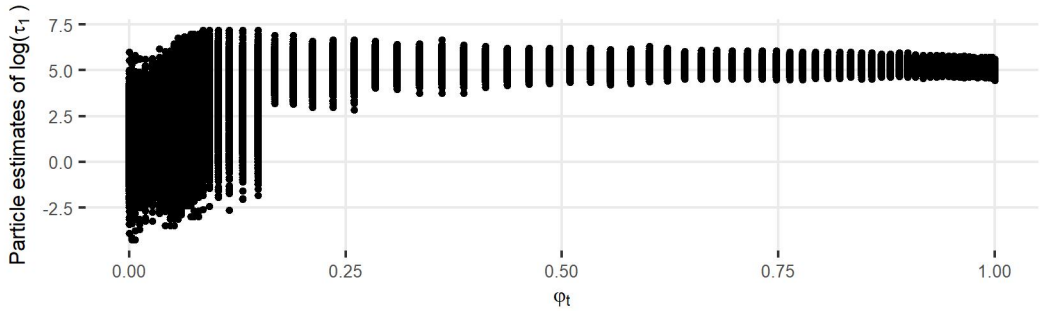
Overall using intermediate steps has proven to have the stated advantage that we predicted would occur, that bad importance sampling proposals from an inappropriate



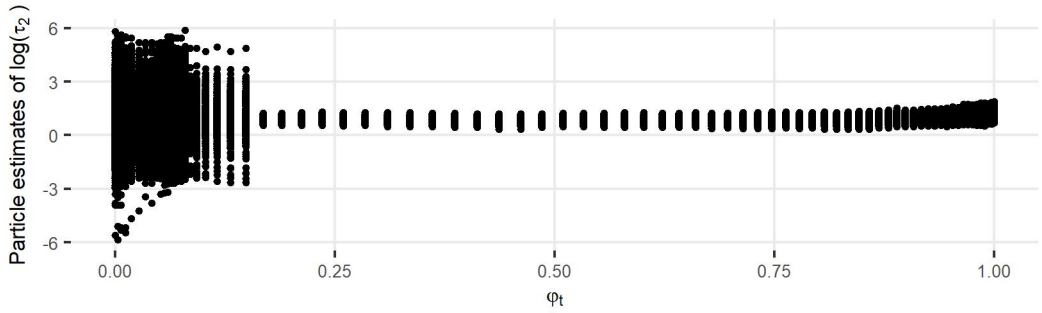
(a) Particle plot of the first ordered mean.



(b) Particle plot of the second ordered mean.

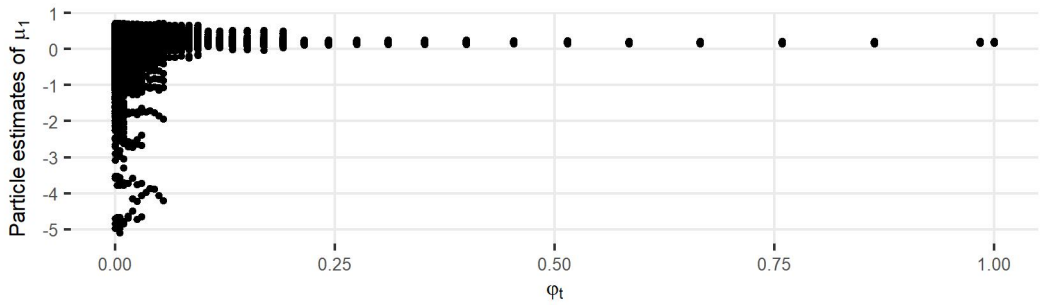


(c) Particle plot of the first ordered precision.

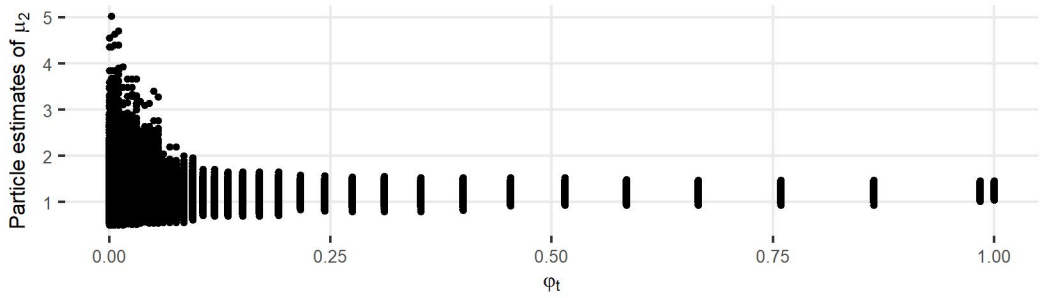


(d) Particle plot of the second ordered precision.

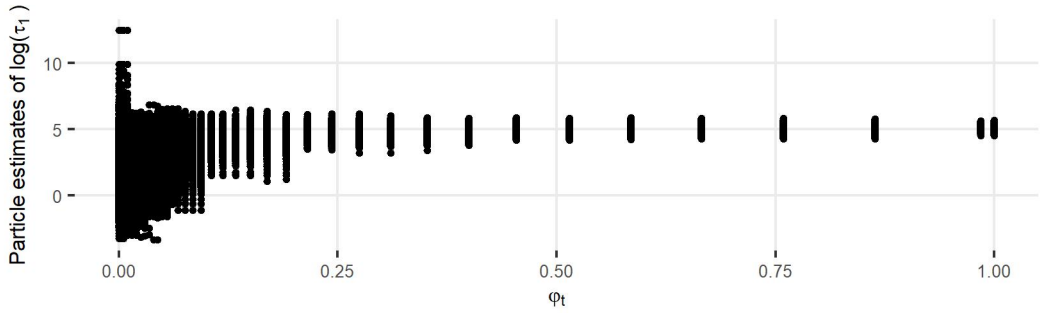
Figure 3.17: Particle plots, for the enzyme data, of the Gaussian means and precisions when transitioning from one to two Gaussian components, using the deconditioned birth transformation.



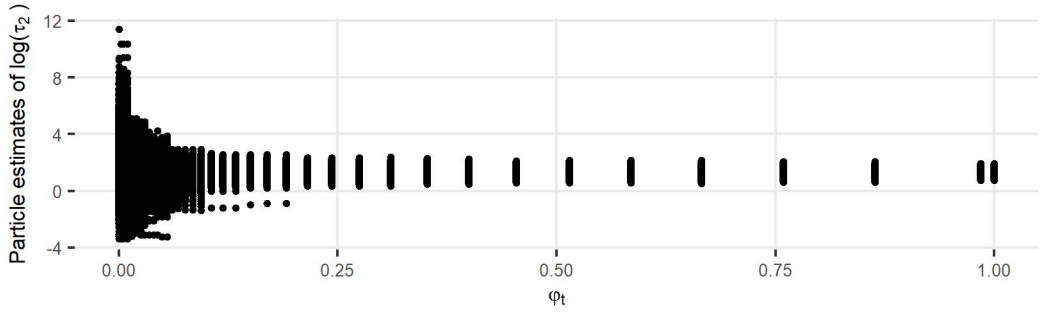
(a) Particle plot of the first ordered mean.



(b) Particle plot of the second ordered mean.

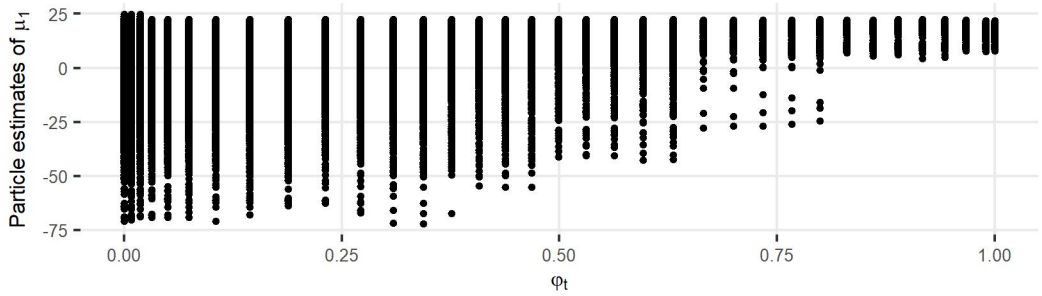


(c) Particle plot of the first ordered precision.

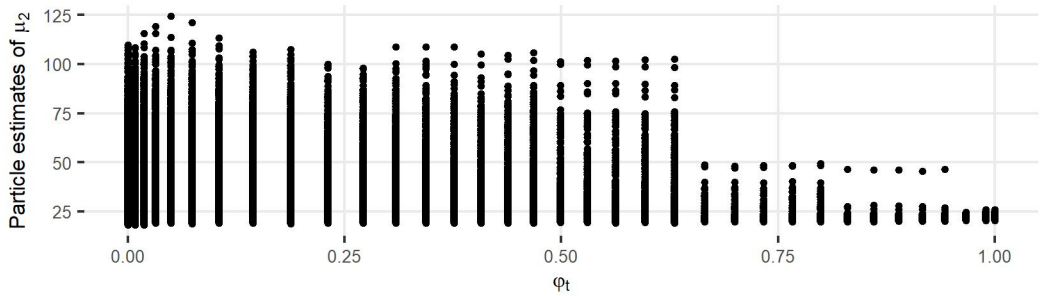


(d) Particle plot of the second ordered precision.

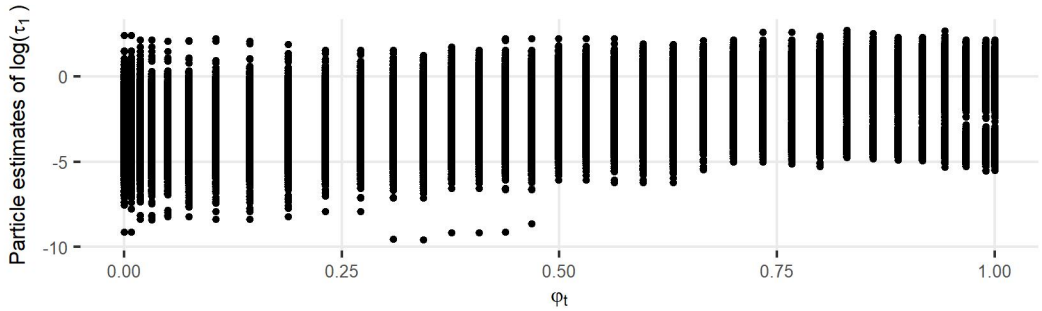
Figure 3.18: Particle plots, for the enzyme data, of the Gaussian means and precisions when transitioning from one to two Gaussian components, using the deconditioned split transformation.



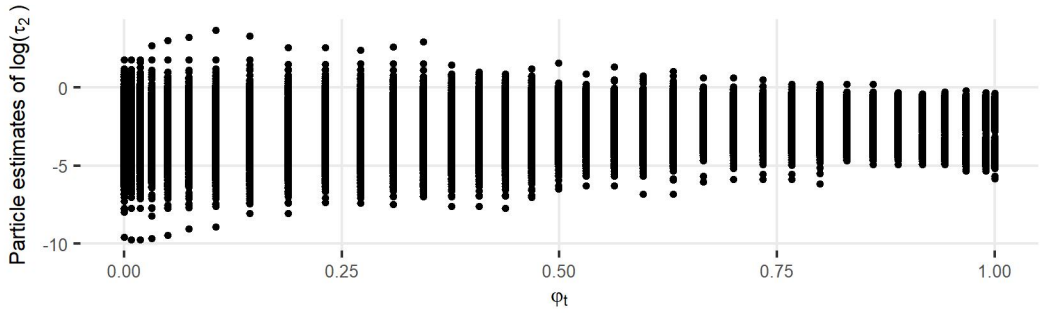
(a) Particle plot of the first ordered mean.



(b) Particle plot of the second ordered mean.

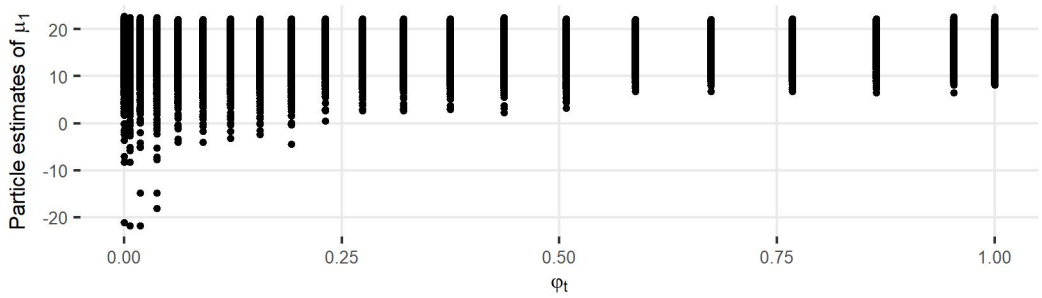


(c) Particle plot of the first ordered precision.

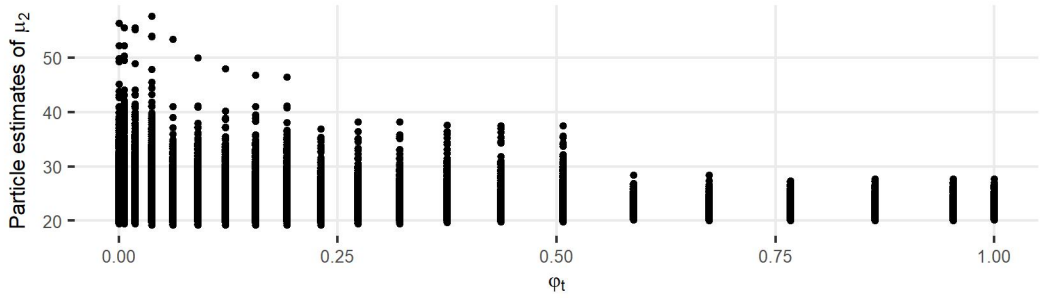


(d) Particle plot of the second ordered precision.

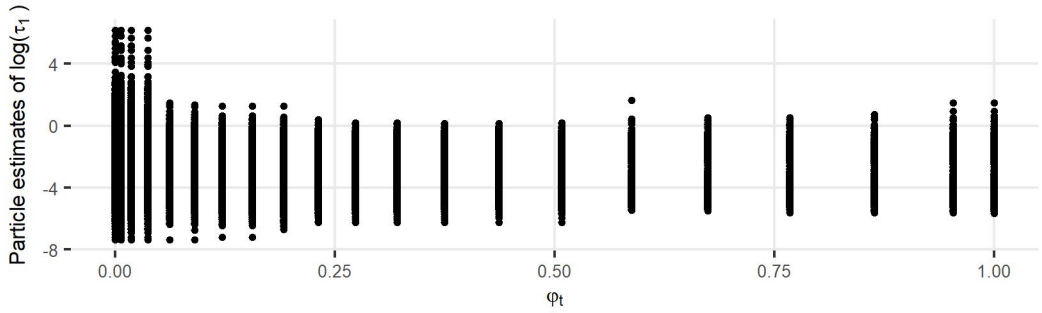
Figure 3.19: Particle plots, for the galaxy data, of the Gaussian means and precisions when transitioning from one to two Gaussian components, using the deconditioned birth transformation.



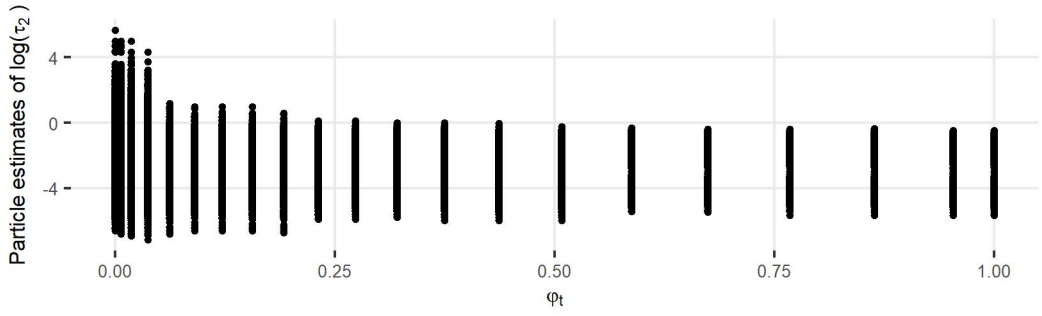
(a) Particle plot of the first ordered mean for the galaxy data.



(b) Particle plot of the second ordered mean for the galaxy data.



(c) Particle plot of the first ordered precision for the galaxy data.



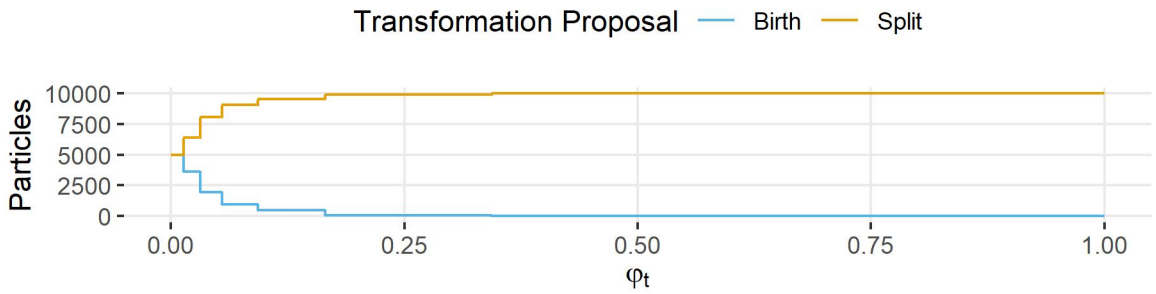
(d) Particle plot of the second ordered precision for the galaxy data.

Figure 3.20: Particle plots, for the galaxy data, of the Gaussian means and precisions when transitioning from one to two Gaussian components, using the deconditioned split transformation.

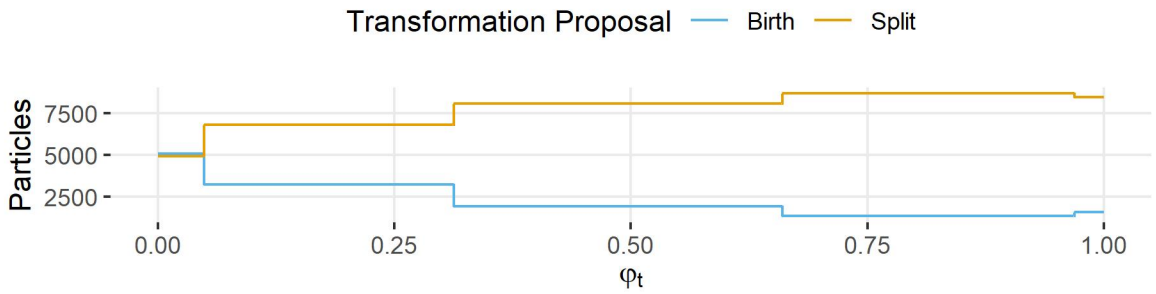
transformation of an existing nested model can be adjusted by an appropriate number of MCMC and reweighting steps at a set of annealed distributions. Otherwise we note that the majority of the particle plots, especially plots that involving transitions when over four components are involved, would initially display wide spread and then narrow down to a smaller interval/subset of values as $\varphi_t \to 1$.

In figures 3.21 and 3.22 we show examples of the distribution of assigned transformations when performing the conditional version of the simultaneous birth and split move over time. For lower dimensional models, the split transformation proposals tended to be far more favored and in the case of the enzyme data would sometimes completely represent all particles. For higher-dimensional models for the enzyme data, the proportion of particles that were generated using the split proposal tended to more dominant in high dimensions. For the galaxy data the distribution is roughly even as $\varphi_t \to 1$ in high dimensional transitions. However despite this transformation proposal favouring the best transformation given the current transition and data, as well as giving an accurate estimate of the posterior distribution, it did not neccessarily give the best estimate of the marginal likelihood which we now discuss.

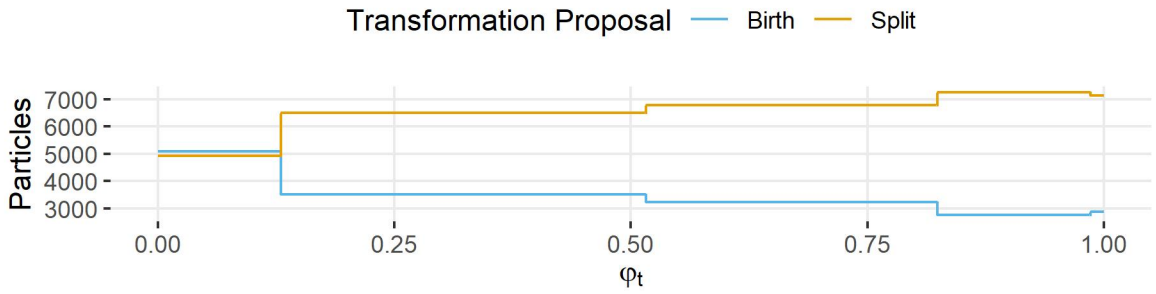
We consider the distribution of the log marginal likelihood estimates, under both the adaptive and fixed annealing schedule for 50 runs, from figures 3.23 to 3.26. As stated in section 3.4 we base the most accurate estimate of the ML from a standard annealed SMC algorithm which uses its corresponding priors as an importance sampler under a very large particle size, which can be seen in figures 3.24 and 3.26. In most of the adaptions applying a fixed annealing schedule gave better estimation to the marginal likelihood and smaller Monte Carlo variance, which implies that when we applied adaptive annealing we set the rate of acceptable CESS loss between intermediate distributions to be too high. For example, figure 3.26 shows that marginal likelihood estimates for the galaxy data under from the deconditioned split move, and the deconditioned birth/split move, was closer to the best possible estimated ML value in comparison to applying adaptive intermediate distributions in figure 3.25.



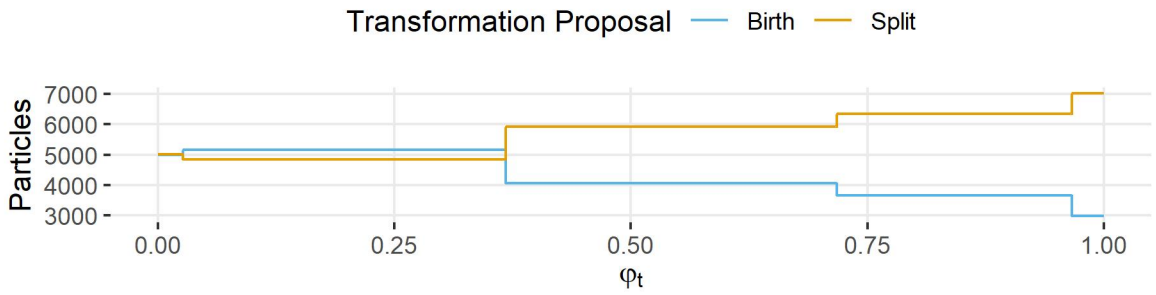
(a) Assigned transformations when transitioning from one to two Gaussian components



(b) Assigned transformations when transitioning from two to three Gaussian components

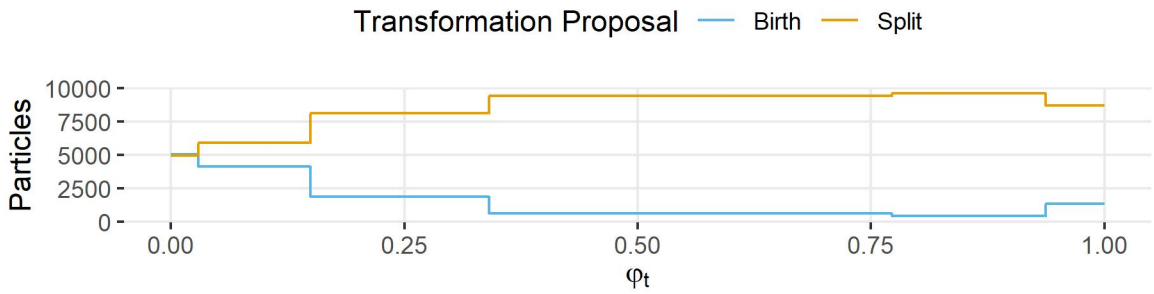


(c) Assigned transformations when transitioning from four to five Gaussian components

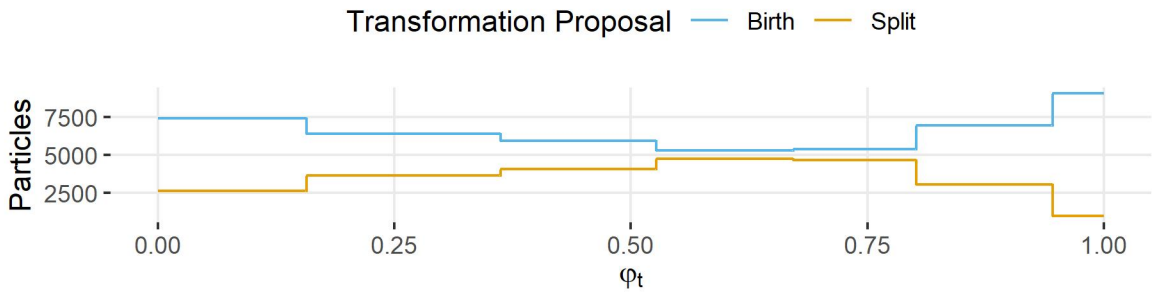


(d) Assigned transformations when transitioning from seven to eight Gaussian components

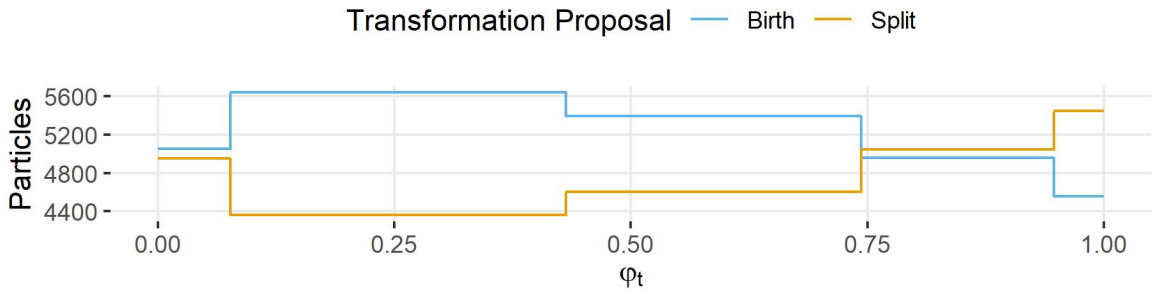
Figure 3.21: Evolution of birth and split moves for the enzyme dataset.



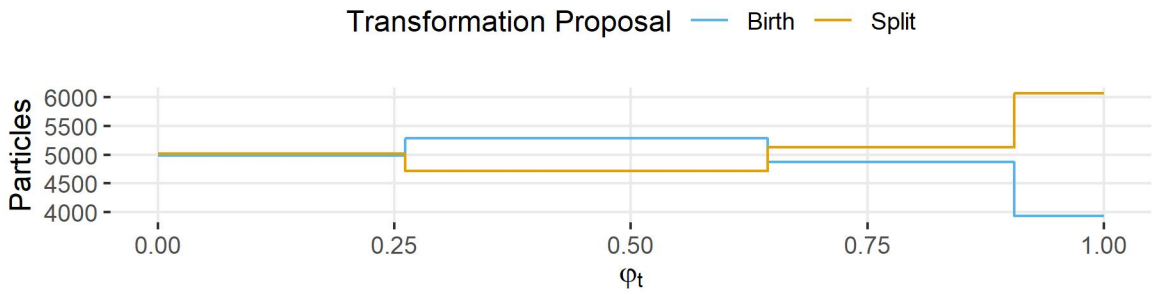
(a) Assigned transformations when transitioning from one to two Gaussian components



(b) Assigned transformations when transitioning from two to three Gaussian components



(c) Assigned transformations when transitioning from four to five Gaussian components



(d) Assigned transformations when transitioning from seven to eight Gaussian components

Figure 3.22: Evolution of birth and split moves for galaxy dataset.

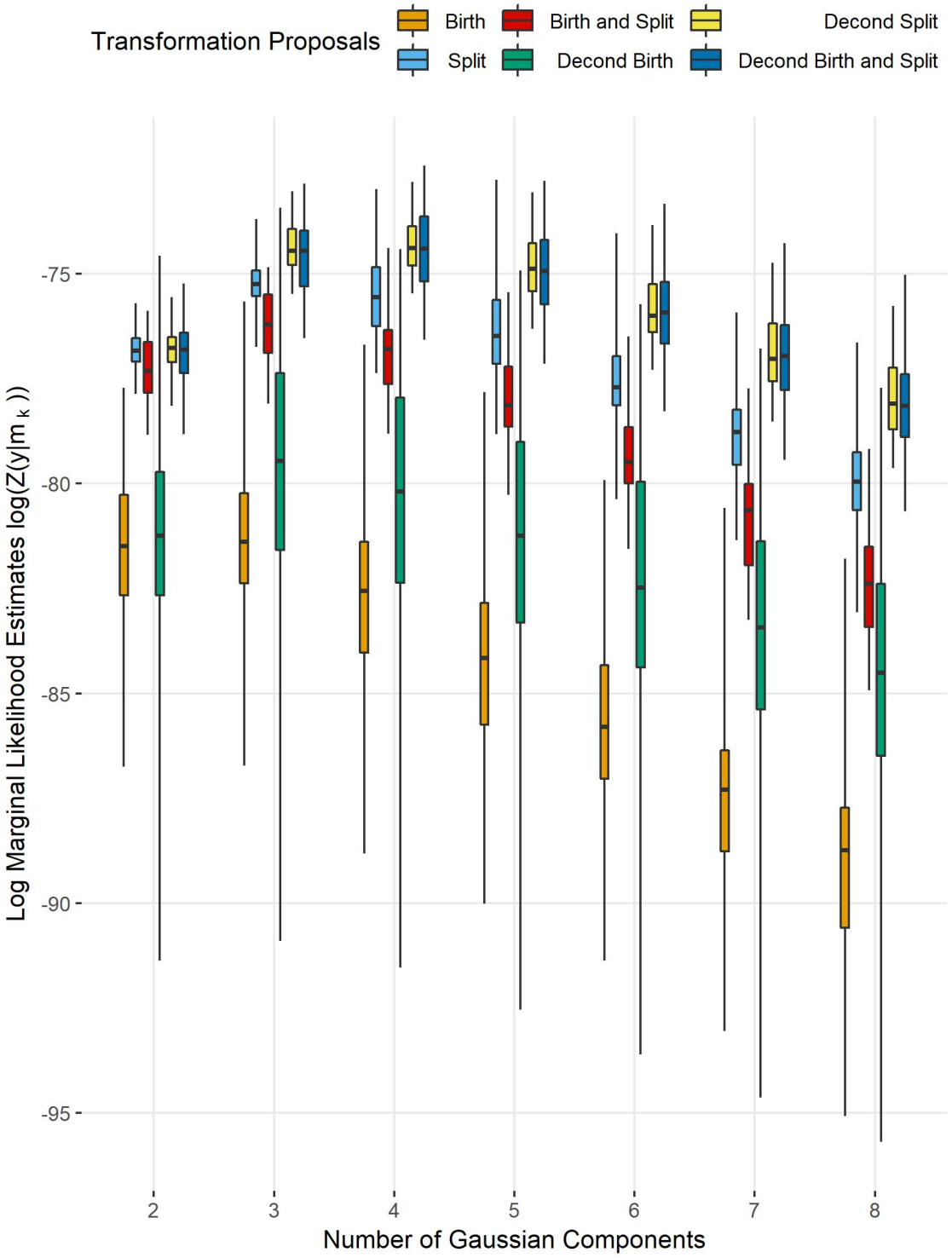


Figure 3.23: Log marginal likelihood plot for the enzyme data under an adaptive intermediate distribution scheme.

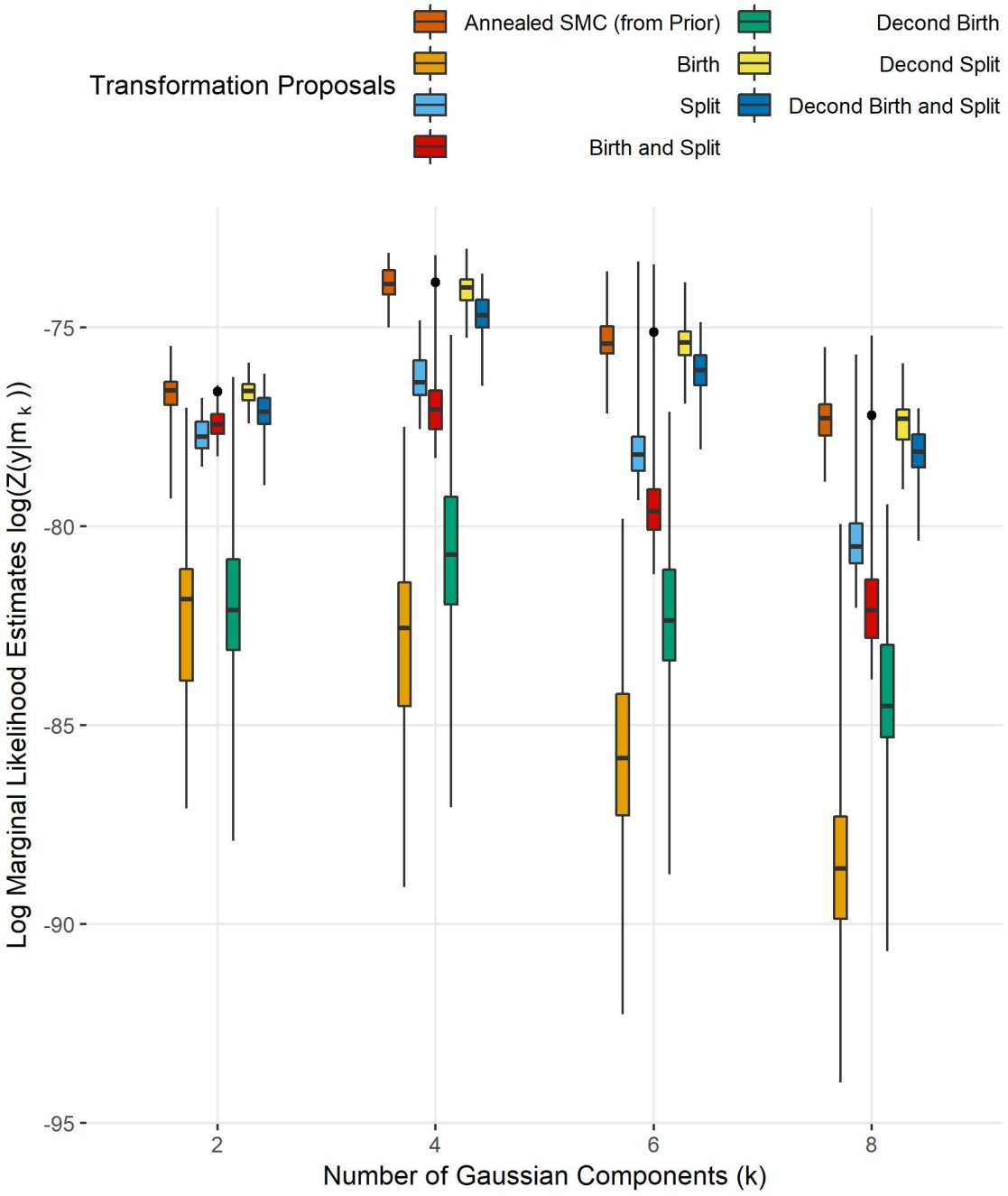


Figure 3.24: Log marginal likelihood plot for the enzyme data when using a fixed number of intermediate distributions. We note that the black point represents our most accurate estimate of the marginal likelihood for each model.

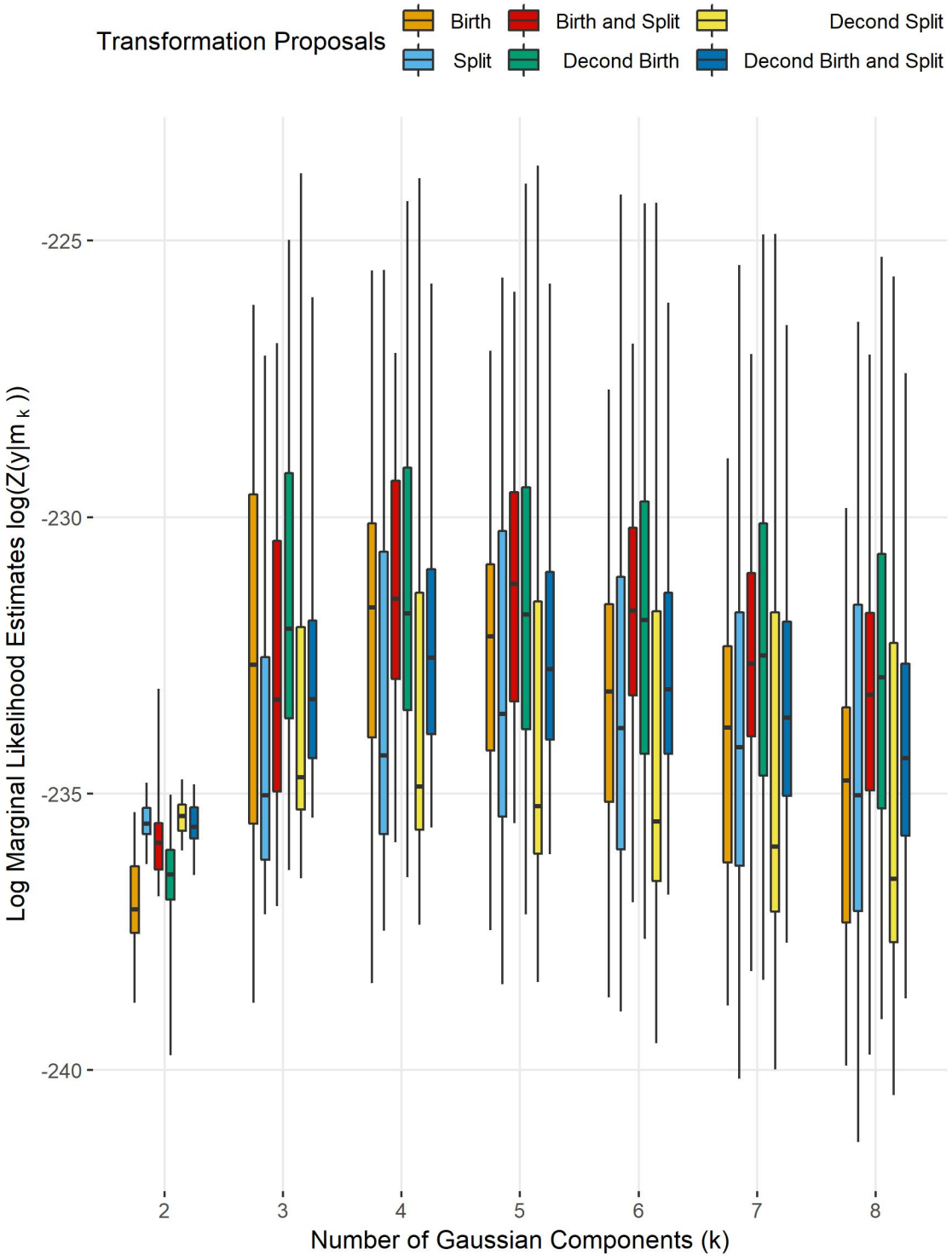


Figure 3.25: Log marginal likelihood plot for the galaxy data under an adaptive intermediate distribution scheme.

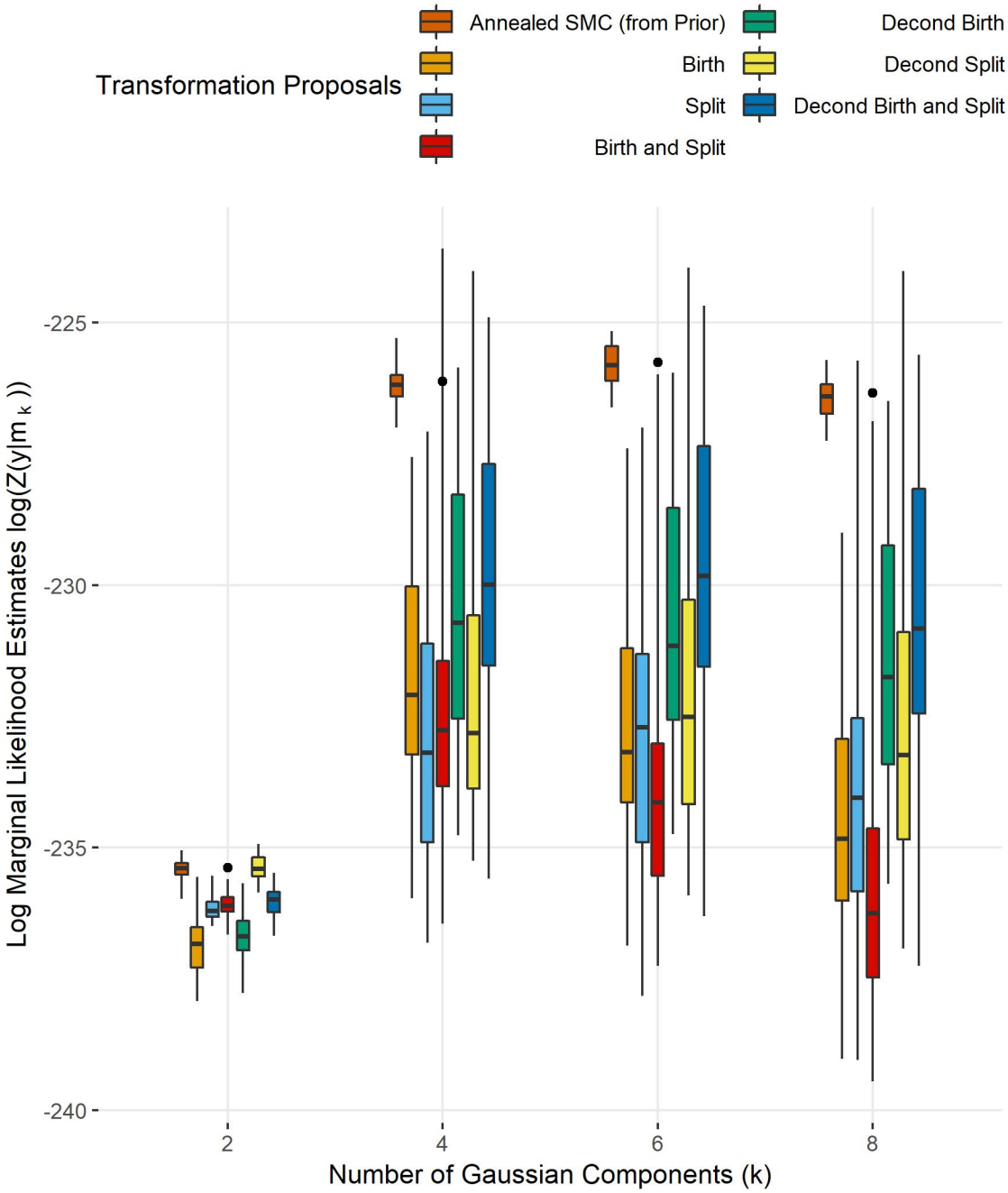


Figure 3.26: Log marginal likelihood plot for the galaxy data when using a fixed number of intermediate distributions. We note that the black point represents our most accurate estimate of the marginal likelihood for each model.

This is expected though as, from chapter 2, more annealed intermediate distributions should reduce the variance of Monte Carlo estimates. Currently the cumulative number of intermediate distributions under all adaptive distribution adaptions is no more than 250, in comparison to the approximately 800 geometric annealed intermediate distributions set in the alternative scheme, and future research should be considered on how setting the decay to some value closer to 1 will reduce these differences while still having a far smaller number of intermediate distributions.

All adaptions displayed marginal increases in the variance of the ML estimates as the dimensional size increased. We were instead hoping for each variance to remain fixed over time as we incrementally move to each of the pairwise models, despite SMC also displaying a similar problem.

When applying a fixed number of intermediate distributions in each run of the tSMC algorithm, the deconditioned versions were superior with a smaller variance and having Monte Carlo averages closer to the best estimated ML. In particular the deconditioned split move was an exceptionally good proposal to transition between different models, at least when modeling the enzyme dataset, when a fixed number of intermediate distributions was applied.

However when tSMC was used to model the galaxy dataset, all of the proposed transformations showed very poor results, something that we go in depth in our discussion. This is despite that we can still obtain good estimates of the posterior distribution when using either transformation proposal. Furthermore there was greater variability on how each transformation proposal performed depending on whether or not

Figures 3.27 and 3.28 shows the Monte Carlo estimates of the log Bayes Factors of adjacent models, under a fixed number of intermediate distributions, for the tSMC adaptions. The figures also consider the log posterior odds estimate of a long running RJMCMC algorithm under similar prior distributions. Note in each of the figures we skip the comparison between model m_2 and model m_1, a two component mixture

model against a one component Gaussian distribution, as the RJMCMC algorithm gave posterior estimates of $\pi(m_1|y) = 0$ for both datasets.

When comparing our results to the posterior odds estimation given by the RJMCMC, each tSMC adaption overestimated the log Bayes factor when comparing high dimensional adjacent models for the enzyme dataset. Although for that matter, the long running SMC result also had overestimated Bayes factors in comparison to the RJMCMC algorithm. We cannot decisively say whether the SMC results were superior to the RJCMCMC result, however given the disadvantages of using RJMCMC in terms of poor mixing in high dimensions and how devising a good proposal specifically for all possible data variation is difficult, as stated in chapter 2, we personally put more trust in the SMC comparisons as seen in figures 3.23 to 3.26.

When considering the log Bayes factor estimates for the galaxy dataset, several transformation proposals gave estimates that matched the posterior odds of the RJMCMC algorithm. Caution is required when analysing posterior odds or a Bayes factor as they regard the evidence of one model directly favouring another model. What they cannot determine is whether these models are wrong or if the marginal likelihoods have been underestimated as what can be seen from figures 3.23 and 3.26.

On a final note we consider the results when applying an alternative form of the intermediate distributions, being arithmetic annealed target distributions as discussed in chapter 2. We analysed these distributions when using the split move under both fixed and adaptive intermediate distributions under the enzyme dataset. We found there were several problems regarding the convergence to each posterior for at least up to a three Gaussian component model. The first issue was predicted from analysing the form of the target distribution, that the very first reweighting step would lead to huge discrepancies between the left hand side of the arithmetic based target distribution of $(1-\varphi_t)(\pi(m_{k-1}, \theta_{m_{k-1}\cdot}|y)\psi_{m_{k-1}\to m_k}(u_{m_{k-1}})J_{m_k\to m_{k-1}})$, in comparison to the right hand side of $\varphi_t(\pi(m_k, \theta_{m_k\cdot}|y)\psi_{m_k\to m_{k-1}}(u_{m_k}))$. As we believe that the particles still received non-zero probability proposals to the next model, despite some particles having zero

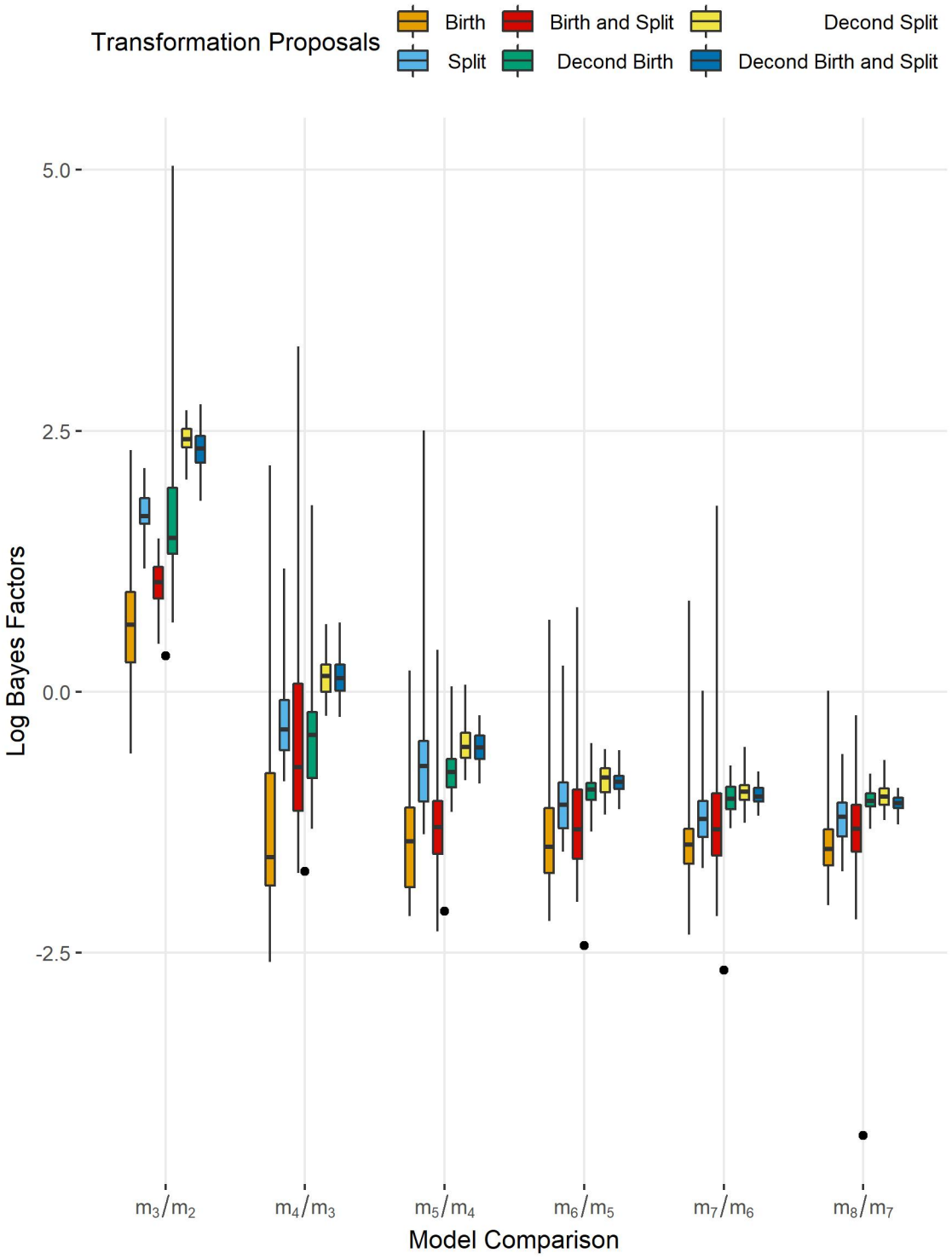


Figure 3.27: Log Bayes factors for the enzyme data when using a fixed number of intermediate distributions. The black dot represents the posterior odds, equivalent to the Bayes factor under prior conditions, for a long running RJMCMC run.

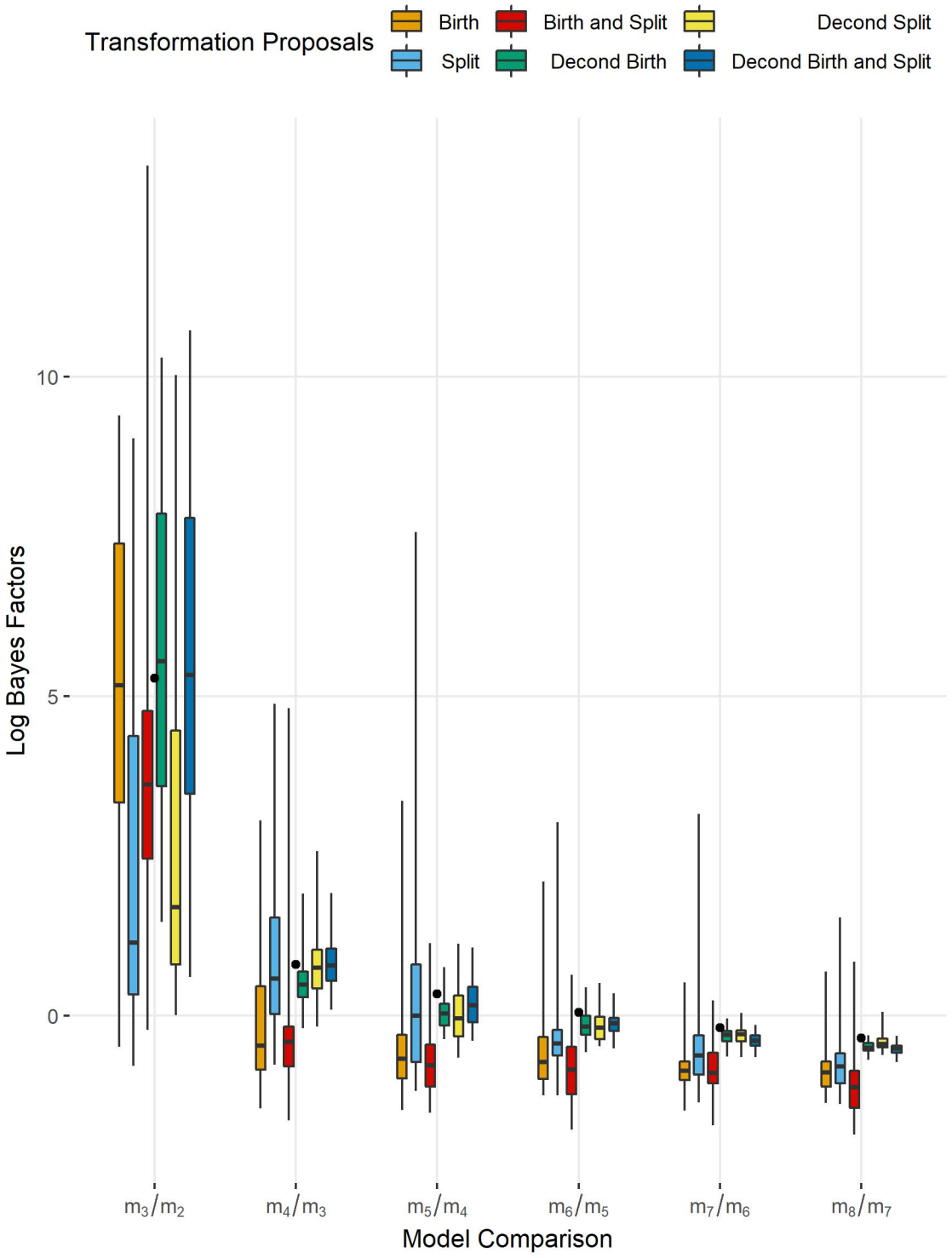


Figure 3.28: Log Bayes factors for the galaxy data when using a fixed number of intermediate distributions. The black dot represents the posterior odds, equivalent to the Bayes factor under prior conditions, for a long running RJMCMC run.

probabilities, the arithmetic annealing caused even greater particle weight variability then the geometric distribution. In many cases only a very small subset of particles represented a resampled set and therefore there was a lack of particle diversity. Also due to the lack of differences between two intermediate distributions whose tempering was dictated by a factor, in comparison to a power when using the geometric bridging intermediate distributions, we found that the adaptive annealing scheme produced very small discrepancies between intermediate distributions. For example when using a CESS target of $0.95N$ it took over 100 times the number of reweighting steps just to simply transition from some model containing only prior assumptions to a Gaussian distribution model, although it could be argued that an appropriate solution would be to set a smaller CESS. Finally, and most importantly, the posterior distribution failed to give approximately the same posterior distributions to a basic SMC algorithm, having probability mass placed in the wrong areas. It is likely that the intermediates distribution attempted to converge between two different models of equal priority for most of the algorithm, which is not what we aim for as we require stronger incremental priority on model m_k when transitioning from model m_{k-1}.

3.6 Discussion

In this chapter we have explained how tSMC may be used for posterior estimation and model selection in the application of simple mixture models. This gives the basic groundwork for how tSMC may be applied to other mixture models, for example when considering multivariate distributions we would recommend split merge adaptions by Zhang *et al.* (2004) and Dellaportas and Papageorgiou (2006) as a starting point.

What we hope to emphasise from this chapter is the advantages of using tSMC, over MCMC or SMC, to estimate high-dimensional mixture models from by applying a subset of nested low-dimensional models. We also showed how adaptive algorithms can be applied within tSMC and still provide similar results to a large fixed annealing

scheme at least with regards to the estimates of a posterior distribution. Thus we will continue to apply this scheme and ignore a fixed annealing scheme in both chapters 4 and 5. What must always be taken into consideration is whether any across model transitions can at least reach some of the modes of the posterior or otherwise convergence would be too dependent on the quality of the MH samplers at each state, as we have seen from figures 3.17 to 3.20.

While we do show that underestimation of the marginal likelihood can be reduced by variance reduction methods such as deconditioning over certain variables, many tSMC adaptions underestimated the ML, based from a long running annealed SMC sampler algorithm, and performed worse then a SMC sampler algorithm that had an equivalent number of likelihood calculations. In particular none of the proposed transformations seemed appropriate for the galaxy data, in comparison to the enzyme data. While the deconditioned split transformation showed to accurately estimate the ML across all analysed mixture models under the enzyme data, the effectiveness of this transformation is still data dependent. Underestimation of the marginal likelihood would still occur if a significant number of particles picked a wrong component to split or otherwise still were a poor match to the posterior distribution of the transitioned model and therefore alternative moves would need to be considered. Naturally an adaption of the birth move that can accurately target probability density unrepresented by the current parameters would be preferable, but such a move will always be limited, unless used as part of a subset of transformations, if we require moves on the other model parameters.

What we also discovered is if a proposal reached any mid-high posterior density on the extended space, then using arithmetic intermediate distributions has the potential to be worse than the geometric target distributions under the initial jump. No further research was considered on this type of intermediate distribution in the other chapters.

We integrated out the allocation variables in the mixture model, as described in section 3.1, despite there existing a strong interest in inferring these variables. If

we were to infer them then what must be noted is that the marginal conditional distribution for at least the allocation variables cannot be obtained when using the geometric bridging scheme in the presence of other continuous variables, despite Gibbs samplers being a popular choice to explore this discrete parameter space (Richardson and Green, 1997). We did consider a solution to infer such allocation variables with a detailed explanation in chapter 5, tested under a different type of mixture model. However our solution only works on the condition that all the priors are conjugate to the likelihood.

Chapter 4

Applications in Genealogy Reconstruction under the Coalescent

This chapter considers an application within population genetics, which is a research area that analyses how a sample of organisms have evolved and where their properties can be expressed through a tree-like diagram. This application presents a scenario to use tSMC where a new set of parameters need to be inferred when a new observation is added to the existing posterior.

We are interested in modeling a joint collection of DNA sequences from haplotype bacteria that reproduce clonally. There is a great interest to analyse the biological processes that govern the evolution of bacteria (Felsenstein, 2004). By constructing the ancestral relationships between sequences we can make conclusions on several properties of the population, see section 4.1 for more detail.

The purpose of our tSMC adaption in this application is not to devise a new model that explains any complex evolutionary behaviour, but to assist with the problems inferring the posterior distribution under simple biological assumptions (see sections 4.2 and 4.3) when a large sample size is present. In this application we plan on using tSMC to update a posterior distribution as new individuals, over some real time span,

are incorporated one at a time and this may be termed as the algorithm being "online".

In section 4.1 we give a basic introduction to the type of data that we are using, including the more specific genetic based terms that we use in the algorithm. We explain what a "tree" is, and how they should be interpreted. Furthermore the differences between "phylogenetics" and "population genetics", and what elements are going to be shared between them are explained. Some of the language explained here is also used in chapter 5.

Section 4.2 gives the relevant model assumptions we apply when inferring the ancestral relationships between sequences, and all parameters that we aim to infer in our Bayes solution.

A brief literature review on the more notable approaches to estimate the posterior distribution and the general flaws of such approaches is given in section 4.3. We explain how our tSMC approach could potentially resolve these drawbacks.

Section 4.4 provides prior assumptions on the parameters and the general form of the posterior distribution. We propose two types of transformation moves, stating their strengths and weaknesses, that allow for a gradual inclusion of observations into the posterior distribution. We explain the MCMC within model moves to be made on the parameters, similar to chapter 3.

Section 4.5 explains what tests were made as well as prior tests on certain model assumptions, with the results explained in section 4.6.

Finally section 4.7 gives concluding thoughts on the results and further research to be made on our proposed adaption of tSMC.

4.1 A Basic Introduction to Genomes, Trees, Phylogenetics and Population Genetics

In this investigation we define several terms relating to how we model a set of genome sequences, an individual's complete set of DNA. The data we use are known

as "aligned sequences" in which we simply term them as "sequences" within the thesis.

In brief, aligned sequences are obtained through first performing sequencing and then afterwards the the retrieved data is aligned. Sequencing is the process of reading the genetic data of an organism, such as a small section of its chromosome, and "alignment" is the process of performing some bioinformatic analysis to "align" the read data by identifying core parts of the sequences and storing the data such that all sequences are lined up (Metzker, 2010).

In a statistical and data-centrist viewpoint we consider one DNA sequence to consist of a series of nucleotides, which can be thought as a basic building block of DNA, that take the following four "nucleobases" which act as multinomial values; Adenine (A), Cytosine (C), Guanine (G) and Thymine (T).

For each organism we either analysis one chromosome, or one sequence, from a haploid genome. Otherwise for diploid organisms we would analysis its two chromosomes, i.e two sequences. Haploid data can be defined by y_{il} where $i \in \{1, ...n\}$ and $l \in \{1, ..., L\}$ for a total of n aligned sequences of sequence length L. We also use the terms "site" and "locus" to refer to the lth nucleotide/site/locus from the beginning of the aligned sequence and contain one of the DNA nucleobases of $\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$. The plural forms of these phrases is simply nucleotides, sites and loci respectively. Similar assumptions are made for diploid data, $y_{il}^{(c)}$, but instead we have an extra dimension that represents the cth chromosome where $c \in \{1, 2\}$. In this investigation we primarily focus on haploid (one chromosome) data, and thus we only use the notation y_{il} when defining the data.

A single nucleotide polymorphism (SNP) is the existence of variation between two sequences at the same site. Furthermore we consider the alleles, which are the set of unique nucleotide types that exist within a site location across all sequences. For example in figure 4.1 with two aligned sequences, at the first site there is no SNPs as the pairwise sites share the same allele $\{\mathrm{A}\}$ but there is an SNP at the second site with two alleles $\{\mathrm{A},\mathrm{T}\}$.

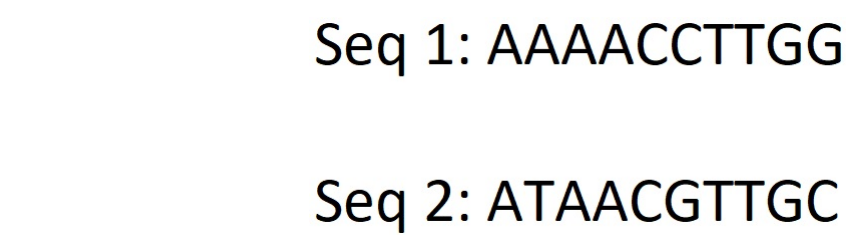


Figure 4.1: An example of three SNPs (or snips) when comparing two haploid sequences of sequence length 10. The SNPs are at the sites $\{2, 6, 10\}$.

The research interest is to use a model to identify patterns in the timing/areas of changes within individuals or discovering alleles that are strongly associated with a certain phenotype, where a phenotype is a type of physical characteristic of a genome. For example when considering the genomes of bacteria we may discover an allele that may be associated with a more toxic strand in comparison to other genomes of the same species (Holder and Lewis, 2003).

We now give an explanation of what a "tree" is, and some of the basic notation that we use when describing its features. We wish to show the "genealogy", or the ancestral relationships, of the sample sequences through a graphical or technical presentation. A "tree" is one way to express the genealogy. A tree consists of a set of vertexes, which most researchers dub as "nodes" when constructing ancestral relationships, and a set of edges (or branches) that link the nodes together. These nodes can either represent the sequences themselves, and thus we define them as "tip/leaf nodes", or they represent some unknown ancestor, alternatively termed as an "inner node", in which a subset of the sample sequences have diverged from said ancestor sequence. The overall relationship between each node, given the connections via branches, we term as the "topology" of the tree. Where the edges are placed, as well as the lengths of each edge, is associated with some genetic distance between each sequence and depends on the model or other researcher beliefs on how the sequences evolved (see section 4.2 for the evolutionary assumptions that we make).

We may rearrange the tree to have a "root" which represents the "most recent

common ancestor" (MRCA) for the entire observation set. Thus the rooted tree takes on a dendrogram/cladogram appearance. Figure 4.2 is an example of a rooted tree, with A_1 being the root node, and we specifically use rooted trees due to the evolutionary assumptions of the sequences that we state in section 4.2. Furthermore a "subtree" (atlernatively termed as "clades") is essentially a tree that considers the genealogy that descends from some other node that is not the true root node, although we classify the entire tree as a type of subtree when defining certain formula, an example being the subtree that has A_2 as the root node and includes the tip nodes of $\{y_1, y_2\}$.

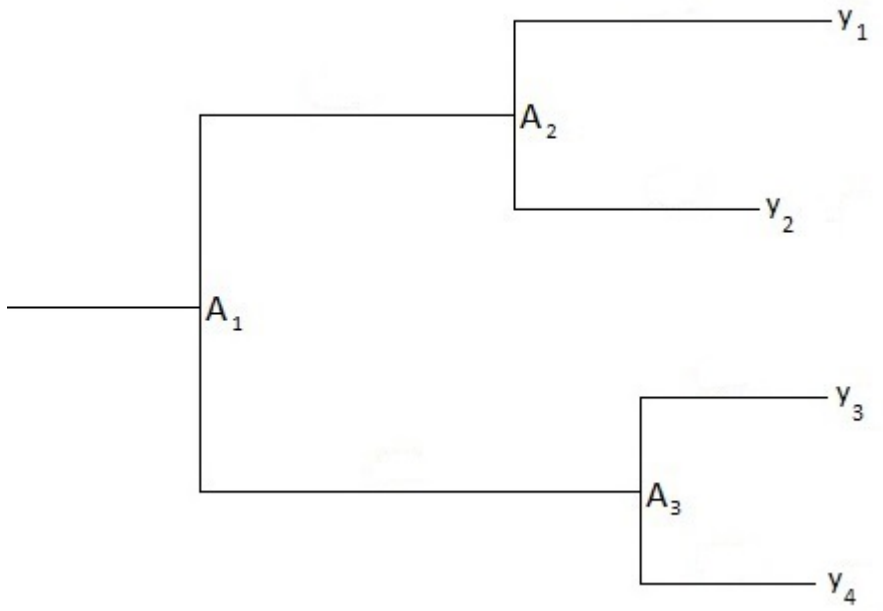


Figure 4.2: A basic rooted tree for the set of sequences $\{y_1, y_2, y_3, y_4\}$. The tree contains three ancestor nodes of $\{A_1, A_2, A_3\}$, with a total of six edges connecting the nodes together.

Finally we briefly mention the difference between "phylogenetics" studies and "population genetics" research, where in this chapter we do cite many references in the field of phylogenetics despite working with a population genetics application. Phylogenetics studies involve constructing how a pattern of organisms, usually representing different species, are related from each other by reconstructing their genetic

relationships in the form of a tree and analysing how each organism diverges from each other. Population genetics mainly focus on the evolutionary and demographic properties within one recorded species/population (Rosenberg and Nordborg, 2002), although the populations can consist of multiple closely related species (Liu *et al.*, 2009; Rannala and Yang, 2003). This includes how certain sites of the sequences within the population are correlated with each other, and this may be due to how they swap/exchange chunks of DNA between themselves (with the process termed as recombination). Alternatively population genetics research could involve analysing the rate of change in the size of the population representing the organisms (Felsenstein, 2004; Yang, 2014). In our application in this chapter there is significant overlap between the two disciplines where we apply likelihood and MC methods on a tree space, see sections 4.3 and 4.4, to sample from the space of a posterior. Note that Chapter 5 considers a population genetics application.

4.2 The Coalescent and the Mutation Rate

4.2.1 Coalescent Theory, Wright-Fisher model and Time Scales

Before we discuss the coalescent model, we give a brief mention about the Wright-Fisher (WF) model (Fisher, 1931; Wright, 1931). This discrete-time Markov chain, was developed during a time period where no genetic data was available. Instead research mostly focused on how theoretically a population of individuals evolve and pass on their genes to the next "generation", a generation representing an indexed state in the Markov chain, under fixed assumptions such as mutation rates.

We first define a number of generations where all individuals within each generation have equal chance of being fit enough for reproduction. Furthermore the size of the population in each generation is constant and the model applies non-overlapping generations representing any past and current populations. As we do not know what

the true constant population size is, and that real populations do not follow the previously stated Wright-Fisher conditions, we instead consider an "effective population size" of $2N_e$ haploid individuals and otherwise we would define N_e individuals (with $2N_e$ chromosomes/genes) for diploid data. This is the idealised Wright-Fisher population size that would show the same magnitude of genetic drift (regarding how the frequency of alleles change after each generation) as the real population, although it will be smaller then the real population as we only consider a population which resulted in the reproduction of each generation and more notably the present generation (Felsenstein, 2004; Hein *et al.*, 2004; Yang, 2014).

The model itself moves forwards in time where at each generation the previous population dies and is replaced by its offspring. This offspring is created by inheriting the genes of a uniformly sampled ancestor, each with probability $1/2N_e$, from the previous generation.

For example suppose that within a population only two alleles exist being A_1 and A_2. Letting i be the number of allele copies of A_1 in the present population then naturally the present frequency of said allele is given by $p = i/2N_e$ and otherwise the frequency for A_2 is $1 - p$. Therefore the Markov transition probability of allele A_1 having j copies in the next generation, given i copies in the present generation, is given by a binomial distribution of,

$$Pr(j|i) = \binom{2N_e}{j} p^j (1-p)^{2N_e - j}. \tag{4.1}$$

An example where we consider the the change in two alleles in a population size of 10 is shown in figure 4.3, note that this is a very simplified example as real world applications could consider an effective population size of 10^4 to 10^8 (Felsenstein, 2004; Hein *et al.*, 2004; Yang, 2014).

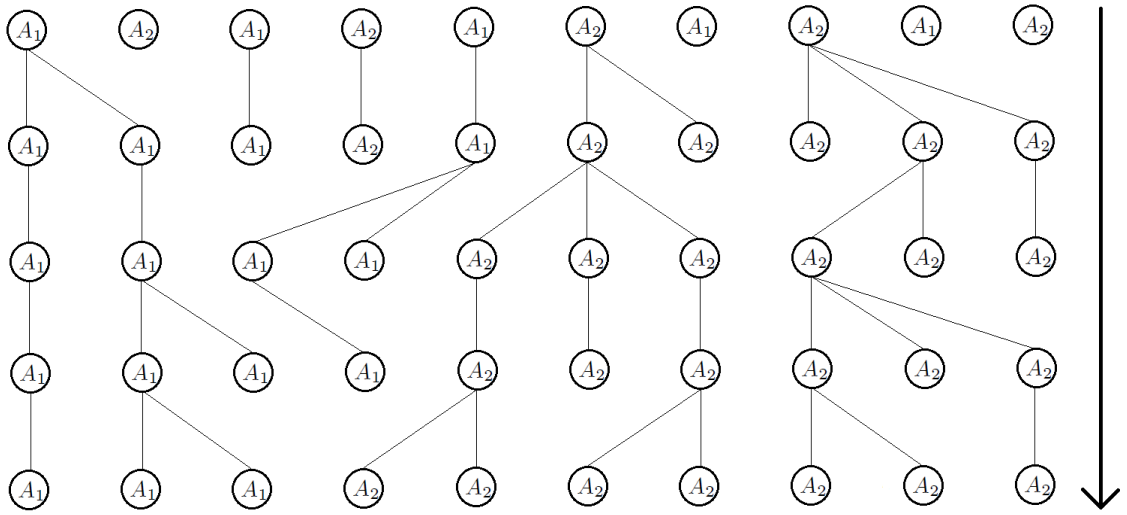


Figure 4.3: Example of the Wright-Fisher Model in practice with a haploid effective population size of $2N_e = 10$ over 5 generations. All crossed lines are removed for easier interpretation. In the first generation the frequency of allele A_1 is 0.5, but in the fifth and present generation the frequency changes to 0.3.

We can expand this to the probability of getting K alleles to have a set of corresponding counts of $(j_1, j_2, ..., j_K)$ given the current counts of $(i_1, i_2, ..., i_K)$ has a Markov transition probability being multinomial distribution (Nagylaki, 1997) defined by

$$Pr((j_1, ..., j_K)|(i_1, ..., i_K)) = (2N_e)! \prod_{k=1}^{K} \frac{1}{j_k!} \left(\frac{i_k}{2N_e}\right)^{j_k}. \tag{4.2}$$

Extensions to diploid data have an individual inherit two uniformly sampled alleles from the previous population. It is also possible to add assumptions to the basic model such as accounting for the presence of mutations on a site that occur at some constant generational rate (see section 4.2.2). Naturally it is possible to trace back a sequence to an ancestor, and even group multiple sequences which share a common ancestor (Felsenstein, 2004; Hein *et al.*, 2004; Yang, 2014).

However if we only want to consider the ancestral relationships between a small sample of n sequences that are from the $2N_e$ sized population then it is ideal to apply the coalescent model instead.

The coalescent, also termed as Kingman's Coalescent or the n-coalescent (Kingman, 1982a,b), was developed in the early 1980s and allows for several advantages when modeling the ancestry of a sample in comparison to other population genetic models like the Wright-Fisher model. The coalescent model represents the genealogy of a small sample of sequences under WF model assumptions in the limit as $2N_e$ approaches infinity. Although in reality $2N_e$ will be a large population not an infinite one, and thus the coalescent is actually an approximation to the WF model, however if $2N_e$ is small then the coalescent does not follow WF assumptions.

The coalescent model works backwards in time and considers how two sequences (sample sequences and/or ancestor sequences) are descended from some unknown ancestor. The backwards joining of two sequences is referred to as "coalescing" and the time of when these nodes are joined can be termed as a "coalescent event". Thus we define the set of "coalescent times" to be the time from each coalescent event to the next coalescent event, which also includes the time from the present to the first coalescent event.

It is far simpler to work backwards in time as we only care about a sample of sequences and the subset of ancestors that are related to them, instead of simulating from a computationally expensive Wright Fisher model where the population to generate can range from 10^4 to 10^8 as well as having a large number of generations (Hein *et al.*, 2004; Yang, 2014).

When simulating a series of coalescent events we generate a set of time periods, $X = \{X_2, ..., X_n\}$ (given n individuals) termed as "coalescent time intervals", between each of the coalescent events. For example the first coalescent event occurs X_n in the past by choosing two lineages to coalesce that are chosen uniformly and independent of generation time. Then a second coalescent event occurs $X_{n-1} + X_n$ back in time from the present. Finally the sample sequences have a MRCA at $X_2 + X_3 + ... + X_{n-1} + X_n$ in the past. Before any coalescent events occur we state that there are $i = n$ lineages left, where lineages are any remaining sample or ancestral sequences whose lineages

have not yet coalesced. After the first coalescent event occurs there are $i = n - 1$ individuals left, which includes the unknown ancestral sequence and the remaining $n - 2$ individuals from the sample. This process continues until there are only two lineages remaining which are naturally the "daughter" sequences of the MRCA for all sequences.

A coalescent model can infer each time interval X_i under different measurements. One such scaling is a "per generation" time, $X_i \subset \mathbb{N}$, where a generation in this context is approximately when some real world individual representing the genome sequence reproduces and this may be referred to as discrete coalescent time. For a sample size of two, the probability that the sequences have a common ancestor one generation ago is $(2N_e)^{-1}$ and with no coalescent event occurring having probability $1 - (2N_e)^{-1}$. This can be interpreted by considering WF assumptions as displayed in figure 4.3 where the probability that one of the individuals shares the same parent with a selected individual is $(2N_e)^{-1}$.

This can be expanded to include the probability of any two genomes coalescing within a set i individuals that have not coalesced yet, with probabilities of $i(i - 1)/2(2N_e)$ and $1 - i(i - 1)/2(2N_e)$ respectively. Therefore the probability that two lineages, out of a sample of $i \leq n$ remaining ancestral lineages, finds a common ancestor j generations ago from the previous coalescent event (if any) is distributed via

$$\Pr(X_i = j) \quad = \quad \text{Geo}\left(p = \left(\frac{i(i-1)}{2}\right)\left(\frac{1}{2N_e}\right)\right), \tag{4.3}$$

with geometric mean of $2(2N_e)/i(i-1)$. An example of this generation time is shown in figure 4.4 which is based from figure 4.3 when considering the first three sequences in the present. In this example the first coalescent event from the present occurs one generation ago, and then the second coalescent event occurs three generations ago from the previous coalescent event.

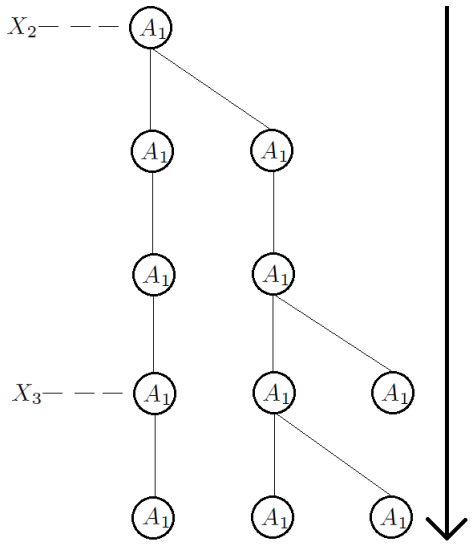


Figure 4.4: Example of identifying where three sequences have diverged from their ancestors based from figure 4.3. The sequences most recent common ancestor (MRCA) can be traced back where two coalescent events have occurred from the present.

However if the population size $2N_e$ is large, which should be true when applying coalescent theory, it is more practical to change the time scale to "per $2N_e$ generations" which is the average time for two lineages to find their specific MRCA. We let this new time scale be $x_i = X_i/2N_e$ where $x_i \subset \mathbb{R}^+$ and $x = \{x_2, ..., x_n\}$. This is derived by considering that the exponential distribution is the limiting case for the geometric distribution. Given that X_i is geometrically distributed with a probability that can be expressed as $p = \lambda/2N_e$ which is small given that $2N_e$ is very large then a random variable $x_i = X_i/2N_e$ has an exponential distribution with mean $2/i(i-1)$ (Hein *et al.*, 2004; Yang, 2014). Given i lineages left we have each coalescent time to the next event to be exponentially distributed by

$$\Pr(x_i) \quad = \quad \text{Exp}\left(\lambda = \frac{i(i-1)}{2}\right). \tag{4.4}$$

An example of a tree generated by the coalescent under this time scale is shown in figure 4.5. When applying coalescent theory, the three inner nodes ($\{A_2$,$A_3\}$ and the root node $\{A_1\}$) represent a coalescent event between two nodes and the branch

lengths are real numbers dictated by the recently stated continuous coalescent time. The tree is also a "bifurcating tree" which means that each inner node, not including the leaves/tip nodes, has three lineages attached being the two offspring nodes and one ancestral node unless it is the root of tree which only has two descendant nodes (Hein *et al.*, 2004).

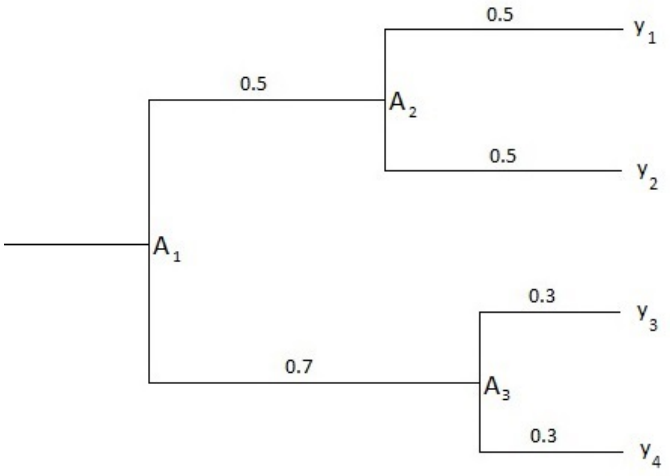


Figure 4.5: A tree, specifically a labeled history tree, represented by $((y_1{:}0.5,y_2{:}0.5){:}0.5,\ (y_3{:}0.3,y_4{:}0.3){:}0.7)$; in the Newick format (Felsenstein, 2004). The first coalescent event occurs at $x_4 = 0.3$ between individual genomes y_3 and y_4. This is followed by a second coalescent between y_1 and y_2 occurring $0.5 - 0.3 = 0.2$ "$2N_e$" generations later from the previous coalescent event. Finally all 4 individuals have a common ancestor $x_4 + x_3 + x_2 = 1$ "$2N_e$" generations in the past.

We also note that applying coalescent theory classifies the rooted tree as a "labeled history" tree. This type of tree occurs as the internal nodes are rank ordered by the most recent coalescent event, with each branch length being dependent on the coalescent events within the entire genealogy. Under the coalescent model we always assume that each possible labeled history tree is equally likely to represent the genealogy of the tree. There are a total of $n!(n-1)!/2^{n-1}$ possible labeled history or coalescent trees. We comment on inferring from this large discrete space of the topology space using standard MCMC methods in section 4.3.

We now describe what the evolutionary processes that dictate the divergence

between sequences, dependent on the branch lengths, in sections 4.2.2 and 4.2.3.

4.2.2 Nucleotide Substitutions and the Population Size Parameter

One possible reason for a particular sequence in the sample to genetically diverge from its ancestor is due to a substitution or a series of substitutions within the genome's DNA sequence, which are the only reasons for divergence from the ancestor that we consider with other factors such as demographic stochasticity are ignored. These types of mutations can also be termed as point mutations and occur when a site in a genome sequence has substituted its nucleobase type for a different type. For example given the four types of nucleobases, $\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$, on a given site of a DNA sequence the nucleobase might change from 'A' to 'T', or 'A' to 'C' etc (Felsenstein, 2004; Hein *et al.*, 2004; Yang, 2014).

It is assumed that the number of substitutions on a particular site is Poisson distributed with a constant Poisson rate, or mutation rate, being $\vartheta/2$ where $\vartheta \in \mathbb{R}^+$ is the population size parameter. To understand the uses of this parameter, suppose that the time scale of the coalescent times are changed such that they are measured in the number of mutations per site. We have an option of using an adjusted continuous coalescent time scale that is defined by $x_i^{'} = \mu^{'} X_i$ where X_i is the coalescent time intervals measured in discrete generations as discussed in section 4.2.1, $\mu^{'}$ is the mutation rate per site per generation where for simplification we assume $\mu^{'}$ is the same and constant among all loci and $x_i^{'}$ is the number of mutations per site that have occurred in-between coalescent events. We note that by substitution

$$\begin{aligned} x_i^{'} &= \mu^{'} X_i \\ &= \mu^{'} \times 2N_e \times (X_i/2N_e) \\ &= x_i\theta/2. \end{aligned} \tag{4.5}$$

Thus the adjusted continuous time scale, being the number of mutations per site, is exponentially distributed by

$$\Pr(x_i^{'}) \quad = \quad \text{Exp}\left(\lambda = \frac{i(i-1)}{2}\left(\frac{2}{\vartheta}\right)\right). \tag{4.6}$$

Overall the population size parameter, ϑ, can be considered as a measure genetic of diversity in the population. Combined with the knowledge of knowing $\mu^{'}$, then ϑ can be used to estimate the effective population size by rearranging the formula of (4.5) (Yang, 2014). We explain how the mutation rate is used in giving the probability of a substitution at a given site within a defined coalescent time in section 4.2.3.

4.2.3 Substitution Models and the JC69 Model

We now describe how to model nucleotide substitutions, in which in our case are dependent on the population size parameter. For example, given a time period or distance between sequences, we may wish to calculate the probability of a nucleotide being subjected to a substitution or simply retaining the nucleobase within this time span. For simplification we consider applying the Jukes-Cantor (JC69) substitution model as the Markov model for nucleotide mutations (Jukes and Cantor, 1969). Otherwise more complex substitution models are ignored as they provide no analytical contribution to the investigation. This model is defined by a distance based probability matrix where the rows represent the original nucleobase of a site and each column represents the possible nucleobases that the site may take. We consider two sequences of s_{il} and s_{jl} at the lth site, in which we will be comparing two ancestor nodes or an ancestor node with a sample sequence when defining the likelihood of a tree.

However we wish to incorporate the mutation rate into the JC69 model. We define the probability of a particular nucleotide changing to a different specific nucleotide, after X number of generations, to be given by $0.25 - 0.25\text{exp}(-4X\mu^{'}/3)$ with the probability of no substitutions occurring within a certain number of generations being

defined by $0.25 + 0.75\exp(-4X\mu^{'}/3)$ (Yang, 2014). By considering the relationship between the different time scales mentioned in (4.5) then by substitution we can define the two said probabilities of a nucleotide mutation occurring as a function of ϑ and per $2N_e$ generation time x. So we define $p_{s_{il}s_{jl}}(x, \vartheta)$ to represent the probability that the site s_{il} can mutate into the nucleotide of site s_{jl} given a certain time period x has passed with a mutation rate shared across the sequences. For example $p_{(s_{il}=\mathrm{C})(s_{jl}=\mathrm{A})}$ is the probability of the nucleotide 'C' having a substitution to become 'A' over some time period. The JC69 model can now be termed as,

$$p_{s_{il}s_{jl}}(x) = \begin{cases} \frac{1}{4} + \frac{3}{4}e^{-2x\vartheta/3} & s_{il} = s_{jl} \\ \frac{1}{4} - \frac{1}{4}e^{-2x\vartheta/3} & s_{il} \neq s_{jl}. \end{cases} \tag{4.7}$$

where the probabilities all sum to one. What can be noticed in (4.7) is that as $x \to \infty$ we reach a limiting distribution of $p_{s_{il}s_{jl}}(x) = 0.25$ in which we assume that so many substitutions have occurred that the nucleotide of s_{jl} has equal probability of being one of the four nucleobases.

4.3 Review of Previous Approaches and tSMC Improvements

The number of algorithms that can construct genealogy trees is too large to summarise within one small section. What we primarily focus on is past approaches that gradually build upon an existing tree given a set of genome sequences, this may include inferring population genetic parameters or not. We also discuss the most basic implementation of MCMC to infer the genealogy, and the difficulties in performing basic inference with it. Finally we consider other SMC approaches that build a tree over time, although they usually consider $\pi(x, \gamma|y)$ with the mutation rate fixed (or otherwise is not accounted for in the model). Many of the algorithms mentioned in this subsection are part of the phylogenetics research field, but nevertheless the tSMC

adaption could assist with inference in this area.

There exists a collection of non-Monte Carlo methods to find the best tree, although usually when coalescent assumptions are not incorporated, usually termed as "Heuristic tree search" methods. In particular, the step-wise addition adaption of a heuristic search is one such method where we usually start with a large number of copies of a tree consisting of two to three individuals and at each tree we randomly graft the ancestor node of a new sequence. Dictating how optimal the tree is could be given by the parsimony score, which we would aim to minimise, where the score represents the minimum number of required changes (such as substitutions) between nodes to explain the history of the individuals and their ancestors (see for example Fitch (1971)). Once the best score is found with the grafted sequence then this is selected to be the best found optimum tree with all other trees discarded and the same process is repeated until all sequences have been grafted. However the downside of such a method is that it does not guarantee that the optimum tree is found, this is due to how certain sub-optimal trees from the past are ignored despite one of them potentially being the precursor for the true optimum tree. Furthermore the genealogy of the final complete tree is strongly affected depending if more diverse sequences are added first or if the sequences that are the most similar take priority to be grafted (Holder and Lewis, 2003; Yang, 2014). Therefore sometimes a mixture of heuristic methods, for example other branch rearrangement proposals as we describe in section 4.4.2, can be applied after or as part of a step-wise addition move (Morrison, 2000).

An alternative to using parsimony scores is the maximum likelihood, with a very popular software package that searches for the maximum likelihood of a tree being PhyML (Guindon *et al.*, 2010). This package and most other maximum likelihood based methods tend to start with some proposed tree, which can be generated via the recently explained heuristic methods for example, that contains the complete set of sample sequences and attempts topology adjustment moves to maximise the likelihood. An example that does including gradual sequence grafting under maximum

likelihood conditions is PUmpER (Izquierdo-carrasco *et al.*, 2014) which does apply a step-wise addition move called Parsimonator as part of a series of multiple steps to optimise the phylogenetic tree when new sequences are grafted.

Although parsimony and maximum likelihood methods can be simple to implement, Bayesian methods can have advantages over the other two classes of algorithms as Bayes' assumptions makes it easier to infer high-dimensional parameter space especially when the joint parameter dimensions exceeds observational size (Holder and Lewis, 2003; Huelsenbeck *et al.*, 2002). Naturally Bayes' methods allow for uncertainty in the model parameters, and certain processes may not be as feasible to model when using maximum likelihood or parsimony methods. One important problem is the estimation of the population size parameter under coalescent model assumptions, in which to identify θ that maximises the likelihood of the sequence data would require integration over all genealogy trees and coalescent time periods which is computationally infeasible (Yang, 2014). Another example is that Bayes methods can accommodate flexible prior assumptions on varying mutation rates across different branches/lineages or sites but unlike non-Bayes methods there is no need to define where these differing rates occur beforehand (Rutschmann, 2006; Yang and Yoder, 2003). Thus applying MCMC with Bayes assumptions on all parameters is a superior option. For a general review of applied MCMC methods we recommend Cheon and Liang (2014); Rutschmann (2006); Yang (2014) regarding general phylogenetic approaches. For examples of Bayes inference on tree space under coalescent assumptions we recommend Didelot and Falush (2007); Felsenstein (2004); Liu *et al.* (2009); Rannala and Yang (2003); Yang (2014).

If MCMC based methods were to be applied then what must be taken into consideration is how MCMC explores a discrete parameter space and how the marginal likelihood is estimated through MCMC output. Independently deciding upon the length of the chain in high dimensional tree space can be daunting, where the number of labeled history based rooted trees is $n!(n-1)!/2^{n-1}$, so to allow for a full exploration

of the topology space then an exceptional chain length may be required even though it can been practically done and MCMC is still the most commonly used methods for ancestral tree reconstruction (Lakner *et al.*, 2008). Appropriate measurements can be made to end the chain if multiple MCMC runs are ran simultaneously and either the average standard deviation, maximum standard deviation or the maximum absolute difference of subtree frequencies between these multiple runs is above a certain cut-off point (Lakner *et al.*, 2008; Whidden and Matsen IV, 2015). Other alterations to the MCMC algorithm should be considered, for example by starting the Markov chain by having the first iteration be drawn from some appropriate reference tree, which could be generated from some of the previously mentioned frequentist methods. Although starting off with a good tree will have a negative effect on some type of convergence diagnostics such as the average standard deviation of split frequencies which depend on initial over-dispersed trees (Holder and Lewis, 2003; Huelsenbeck *et al.*, 2002; Lakner *et al.*, 2008).

While we describe the interesting properties of our SMC approach in this application in section 4.3.1, regarding other existing SMC approaches most research has been focused on agglomerative clustering of genealogy trees. The general themes behind how the clustering method works is that there exists an initial series of subtrees or alternatively called "forests" for each particle and we aim to join them over time to form one genealogy tree. The first step defines n subtrees which represent the tip nodes alone without any common ancestors defined. At each SMC state, two of the subtrees (with the root being an ancestor or sequence node) are selected at random by either uniformly choosing a pairing or otherwise through a more directed proposal. A proposal is made that they have a recent ancestor separated by some time based distance. The joint subtree probabilities of the incomplete trees, with the exception of tip node only subtrees, acts as the target distribution where reweighting and optional resampling steps occur. The process continues, until a tree is formed where all the sample sequences have a defined MRCA with each other.

A coalescent based adaption, although they assume a fixed mutation rate instead of inferring it, was performed by Teh *et al.* (2008). Bouchard-côté *et al.* (2012) and Bouchard-côté (2014) apply their own variant, which they term PosetSMC, that considers non-coalescent assumptions. Wang *et al.* (2015) through their combinatorial SMC framework made further improvements from Bouchard-côté (2014) by considering more advanced assumptions such as non-clock trees, and in comparison to the previous research they suggest MCMC moves may be applied after reweighting or resampling. However we have not identified any previous work that simultaneously , alongside the rest of the genealogy, infers the posterior of population genetic based parameters such as the population size parameter. Furthermore when inferring the target distribution as a set of subtrees that does not have a MRCA defined until the final step, it is questionable to even attempt inferring the population size parameter considering how some particles may have exclusive sample sequences represented in their respective target distributions. Wang *et al.* (2015) suggested a PMCMC algorithm that uses combinational SMC at different values of these evolutionary parameters to obtain a Monte Carlo estimate. However it lacks parallelisation properties (at least on the MCMC component of PMCMC) in comparison to our proposed algorithm tSMC while still allowing for both sequential grafting and inferring the population size parameter simultaneously.

A recent SMC approach was recently given with a theoretical discussion described in Dinh *et al.* (2018), and then applied results in Fourment *et al.* (2018). The idea behind their "Online Phylogenetic SMC" algorithm is that within a set of particles they simply add the observation via a variable transformation and then reweight and resample the particle set. Afterwards they perform a series of Metropolis Hastings moves, although Fourment *et al.* (2018) ignore applying kernels after grafting, which target the new parameter space before grafting another sequence. Fourment *et al.* (2018) only considered inferring the topology and the branch lengths of the genealogy tree and did not consider other evolutionary based parameters. They gave good

proposals to graft a sequence onto a tree, which proved to be effective under a SMC approach.

4.3.1 The tSMC Approach

We are going to use tSMC to sequentially graft sequences one at a time onto trees whose posterior space has been explored. While this is similar to what was done by Fourment *et al.* (2018) we believe the use of intermediate distributions and MCMC kernels should compensate for sudden shifts in the posterior density when a new sequence is added, especially if more complex models were of interest. For tSMC to work efficiently for a standard phylogenetic problem we assume that given that sample sequences are from similar populations there will be very little difference between the topology with $n-1$ sequences and another tree with n sequences. While this condition may hold true for higher dimensional trees, it won't necessarily hold true for low dimensional trees with 3 to 5 sequences but nevertheless genealogies of that size are easy to infer under the biological assumptions made in section 4.2. So providing that that we have generated a tree through posterior inference, then there are $2n-1$ ways to graft the new sequence onto an existing tree with n sequences which is an improvement of doing MCMC and considering a total of $n!(n+1)!/2^n$ possible topologies for $n+1$ sequences.

When trying to estimate the ML many researchers opt for variations of the harmonic mean estimator or path sampling for marginal likelihood estimation, we recommend chapters 3-6 of Chen *et al.* (2014) that repeat said processes for genealogy trees, however the SMC approach does calculate the ML by design. Naturally the implementation of the geometric bridging intermediate distributions in tSMC means that the absolute worst particles are removed first, which allows more mediocre particles to recover through MH moves.

We also believe tSMC allows for more flexibility in the estimation of evolutionary or population based parameters then agglomerative clustering with SMC, for example

the best possible clustering schemes either require the mutation rate to be fixed or inferred under some scheme like a PMCMC but not updated with the rest of the genealogy. However we believe our adaption can update all parameter simultaneously. Furthermore tSMC allows for the posteriors to be updated as data arrives over time.

Fourment *et al.* (2018) suggested that using a larger particle size is more efficient then using a large number of MCMC moves in their adaption. While this may be appropriate when inferring the branch lengths and topology only, if grafting a new sequence proved to have a notable change in the posterior distribution of an evolutionary parameter such as the population size parameter, although this is a factor that we aim to investigate, then no adjustment to the particle size would alleviate this problem. Proposing accurate transformations on these parameters could be a solution, but that might not always be an option available.

4.4 tSMC Adaption and Model Assumptions

In our tSMC adaption we aim to infer a set of models $\{m_1, ..., m_k, ..., m_K\}$, We emphasise that model m_k models the ancestral relationships for $k+1$ sequences, and thus $K = n - 1$ given that we will aim to infer a high dimensional tree containing a total of n sequences. For example model m_2 describes a tree for three sequences with two coalescent time intervals defined by $x_{m_2(2:3)}$. The difference between the posteriors of m_{k-1} and m_k is the inclusion of an additional coalescent time interval and branch in the topology, generated due to the introduction of a new sequence of y_{k+1}. We also incorporate m_0 to represent the proposals for the coalescent time for two sequences, or a two sequence tree, to coalesce as well provide an initial proposal for the population size parameter. We consider the posterior of each

$$\pi_{m_k}(\vartheta, \gamma_{m_k}, x_{m_k}|y_{1:(k+1)}) \quad \propto \quad f(y_{1:(k+1)}|\vartheta, \gamma_{m_k}, x_{m_k})p(\vartheta)p(\gamma_{m_k})p(x_{m_k}), \qquad (4.8)$$

with each marginal parameter previously defined in section 4.2 and $y_{1:k} = \{y_{1\cdot}, ..., y_{k\cdot}\}$. When explaining the transformation proposals and MH kernels to apply within our tSMC algorithm, we hide the notation t dictating the specific intermediate distribution that the parameters correspond to. For example the coalescent time intervals for model m_k that are currently targeting the tth intermediate distribution (where $t \in (0, ..., T)$), within the transition from model m_{k-1} to m_k for example, is technically defined by $x_{m_k(2:(k+1))t}$. However we remove the notation t except briefly when we describe our adaptive MH kernels for the relevent model parameters. This was done for the sake of simplicity when explaining our proposals, despite that in chapter 3 we did incorporate the index of an intermediate distribution. Otherwise we note that $\vartheta \equiv \vartheta_{m_k}$ as we apply no model transformation to this parameter when transitioning between models as described in section 4.4.3.

4.4.1 The Posterior Distribution

4.4.1.1 The Likelihood

We consider the likelihood $f(y_{1:(k+1)}|\vartheta, \gamma_{m_k}, x_{m_k})$ for the specific genealogy of m_k, which includes the overall topology and coalescent time intervals $\{\gamma_{m_k}, x_{m_k}\}$ and other population parameters in which we only infer the population size parameter ϑ.

To illustrate how the likelihood can be calculated, we consider the clonal ancestry tree in figure 4.5 and suppose we analyse the likelihood of receiving this exact genealogy, given a sample of DNA sequences at a particular site l. We define the inner node sequences as $A_{1\cdot}$, $A_{2\cdot}$ and $A_{3\cdot}$ where we assume that their true sequences are unknown and thus they are integrated out within the likelihood formula. The transition probabilities of $p_{s_{i'l}s_{il}}(x_{s_{i'}s_i}, \vartheta)$, are given by the JC69 substitution model described within section 4.2.3 where $x_{s_{i'}s_i}$ relates to the branch length from the sequence s_i to its parent node of $s_{i'}$. Given the JC69 model we assume that each nucleotide type has an equal chance of being the root at that site, and thus given that we can take nucleotide alleles of $\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$ then $p(A_{1l}) = 0.25$ for all nucleotides. Finally we

assume independence across all sites. Therefore the likelihood of site l is defined by

$$\begin{aligned} f(y_{(1:(3+1))l}|\gamma_{m_3}, x_{m_3}, \vartheta) &= \sum_{A_{1l}}\sum_{A_{2l}}\sum_{A_{3l}} p(A_{1l}) p_{A_{1l}A_{2l}}(x_{A_1A_2}, \vartheta) p_{A_{1l}A_{3l}}(x_{A_1A_3}, \vartheta) \\ &\quad \times p_{A_{2l}y_{1l}}(x_{A_2y_1}, \vartheta) p_{A_{2l}y_{2l}}(x_{A_2y_2}, \vartheta) \\ &\quad \times p_{A_{3l}y_{3l}}(x_{A_3y_3}, \vartheta) p_{A_{3l}y_{4l}}(x_{A_3y_4}, \vartheta). \end{aligned} \tag{4.9}$$

where, for example, $\sum_{A_{1l}}$ represents how we sum over the possible multinomial values that A_{1l} can take being $\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$. If there are a large number of lineages in the tree space, then (4.9) can be computationally unfeasible to resolve given it has a cost that is exponential in n, being the total number of sequences to graft, of $O(4^n)$. A useful method to simplify the calculation of the multiple sum terms is through the pruning algorithm which considers the conditional likelihoods of trees. Suppose that $\tilde{f}(s_i)$ represents the conditional probability of sequence s_i having a certain set of nucleobases given the rest of the genealogy that descents it. We consider the sequence node s_i to have daughter nodes of s_D and $s_{D'}$, and thus the conditional probabilities is given by

$$\tilde{f}(s_{il}) = (\sum_{s_{Dl}} p_{s_{il}s_{Dl}}(x_{s_is_D}, \vartheta)\tilde{f}(s_D)) \times (\sum_{s_{D'l}} p_{s_{il}s_{D'l}}(x_{s_is_{D'}}, \vartheta)\tilde{f}(s_{D'})), \tag{4.10}$$

in which if s_i is the root node of the complete genealogy tree then we calculate variants of (4.10) recursively $n-1$ times to give a linear cost of $O(n)$ for the algorithm (Felsenstein, 1981). If the subtree root node is a tip node then any descendant tips will only include the tip itself, for example if $y_{1l} = \mathrm{A}$ then $\tilde{f}(y_{1l} = \mathrm{A}) = 1$ and 0 otherwise ($\tilde{f}(y_{1l} = \mathrm{C}) = 0$ etc). Overall the complete likelihood is then defined by,

$$\begin{aligned} f(y_{(1:(k+1))l}|\gamma_{m_k}, x_{m_k}, \vartheta) &= \prod_{l=1}^{L} f(y_{(1:(k+1))l}|\gamma_{m_k}, x_{m_k}, \vartheta) \\ &= \prod_{l=1}^{L} \sum_{A_{1l}} p(A_{1l})\tilde{f}(A_{1l}). \end{aligned} \tag{4.11}$$

The computational cost of (4.11) can be simplified to $O(NL')$, where L' is the total number of unique trees when only one site is considered for the tip nodes, as any duplicate trees are identified and their respective likelihood values of the genealogy at this site is copied from another site that shares its tree (Yang, 2014).

4.4.1.2 Prior Distributions

The priors depend on the type of coalescent model we infer as explained in 4.2. The "per $2N_e$ generation" time scale, x, is used for the investigation as we do not need to consider the true value and/or interpretation of the effective population size $2N_e$ when using Monte Carlo algorithms to estimate Bayesian posteriors (Hein *et al.*, 2004; Yang, 2014). Overall the joint prior distribution for the set of exponentially distributed coalescent times for model m_k, which has a total of $k+1$ sequences and thus k coalescent time intervals, is given by

$$p(x_{m_k}) = \prod_{i=2}^{k+1} \frac{i(i-1)}{2} \exp\left(-\frac{i(i-1)}{2} x_{m_k i}\right), \tag{4.12}$$

given ordered coalescent events such that $x_{m_k(k+1)}$ is the time to reach the first event and $x_{m_k 2}$ is the final coalescent time period from the sample MRCA and the previous coalescent event.

Under the coalescent all possible labeled tree topologies γ have uniform probability of

$$p(\gamma_{m_k}) = \prod_{i=2}^{k+1} (i(i-1)/2)^{-1}, \tag{4.13}$$

to best represent the genealogy (Yang, 2014). We assign ϑ a gamma distributed prior of

$$\vartheta \sim \text{Ga}(\alpha = 1, \beta = 5), \tag{4.14}$$

which is an appropriate prior given that in this investigation we are analysing a

particular species of bacteria where we do not expect a large ϑ (Takuno *et al.*, 2012; Young *et al.*, 2012).

Note that model m_0 only contains proposals for model m_1, with the proposals being the prior distributions themselves when two sequences are present.

4.4.2 MCMC Kernel Moves

We initiate kernel moves on the individual heights of the inner nodes (including the root), the topology of the tree and the population parameter ϑ in this order. Many of the moves are shared with phylogenetic applications, although they require a few alterations to account for a coalescent prior. We consider proposals for model m_k on the ordered heights of the inner nodes $h = \{h_2, ..., h_{k+1}\}$ where each height represents the cumulative coalescent intervals, for example $h_2 = \sum_{i=2}^{k+1} x_{m_k i}$ or $h_{(i)} = \sum_{i}^{k+1} x_{m_k i}$ for $i \in \{2, ..., (k+1)\}$. These heights will correspond to a certain inner node whose height is subject to change, as described in section 4.4.2.1. The heights themselves are considered temporary variables that implicitly have priors, in comparison to explicit priors on the coalescent time intervals that are incorporated into the posterior. Updating the heights under the coalescent is far more flexible in comparison to making moves on each $x_{m_k i}$ and is common practice in many research papers and software (see for example Didelot and Falush (2007); Drummond *et al.* (2012)).

4.4.2.1 Population Size Parameter and Branch Lengths

Whem making MH moves on the population size parameter we use a log normal proposal, with tuning variance of υ_ϑ, defined by

$$\log(\vartheta^{'}) \sim \text{Normal}(\log(\vartheta), (\upsilon_\vartheta)^{-1}). \tag{4.15}$$

We consider making proposals to at least two branches of the tree simultaneously based on identifying the smallest to largest inner nodes heights of each tree and we

move them by a log-Gaussian random walk, with a tuning variance υ_{h_j} that corresponds to the weighted particle estimates of each ordered height, given by

$$h_i^{'} \sim \text{Normal}(h_i, (\upsilon_{h_i})^{-1}). \tag{4.16}$$

Note that the move is rejected if the proposed inner node that corresponds to the height is moved above its parent node (except if it the root node), or below one of its daughter nodes.

We apply adaptive tuning variances for 4.15 and 4.16, and are changed adaptively at each intermediate distribution such that

$$\begin{aligned} \upsilon_{\vartheta}^{'} &= \text{Wt.Var}(\log(\vartheta), w_{m_k t}) \\ \upsilon_{h_i}^{'} &= \text{Wt.Var}(h_i, w_{m_k t}), \end{aligned} \tag{4.17}$$

where $w_{m_k t}$ are the particle weights for model m_k corresponding to the tth intermediate distribution. If a certain parameter θ (for example, $\theta = \vartheta$ or $\theta = h_2$) has acceptance rates greater than 0.6 then its tuning variance (based on the current weighted particle variance of θ being $\upsilon_{\theta}^{'}$) is readjusted to a constant factor of two such that we instead choose to use a turning variance of $\upsilon_\theta = \upsilon_{\theta}^{'} \times c_{m_k t}$ where $c_{m_k t} = 2 \times c_{m_k (t-1)}$ and $c_{m_k 0} = 1$. Should the acceptance rates go below 0.2 we readjust its tuning variance to the relationship of $\upsilon_\theta = \upsilon_{\theta}^{'} \times c_{m_k t}$ where $c_{m_k t} = 0.5 \times c_{m_k (t-1)}$. Otherwise we let $c_{m_k t} = c_{m_k (t-1)}$. Again these factors do stack with each other, and are reset to one once a tSMC transition from one model (when $\varphi_0 = 0$) to the other ($\varphi_T = 1$) is completed.

However what needs to be considered is the relationship between ϑ and the overall time to the next coalescent event. These two types of parameter are highly dependent a *posteriori*, for example given the best possible tree for n sequences then by decreasing ϑ it is necessary to increase the length of the branch lengths and vice versa to maintain the same number of expected number of mutations between two

nodes. Therefore each υ_{h_i} might not be the right tuning variances. What we consider instead is the conditional variance of the heights of each inner node given ϑ, found by plotting a linear regression (assuming constant variance) of heights against ϑ and calculating the variance of the weighted residuals to be the tuning variance (Raftery and Lewis, 1995), this would naturally give a smaller tuning variance but one that might be appropriate for all trees. There is no certainty that this would offer improved acceptance rates during the initial stages, so we aim to briefly test both forms of the adaptive tuning scheme.

4.4.2.2 Topology

We consider two topology moves with one being a basic subtree pruning and regrafting (SPR) move, with the adaption inspired from the MCMC moves in Didelot and Falush (2007). A second alternative SPR move is based from Wilson and Balding (1998).

A SPR move prunes a subtree, where a subtree can consist of a single leaf node or some ancestry descended from an inner node, and regrafts it on to any of the remaining branches of the tree providing that the move meets certain criterion that we shortly explain. For each non-root node s_i, we consider whether its parent node $s_{i'}$ can be grafted above node s_j and below its corresponding parent node $s_{j'}$. The conditions for this move must be that node $s_{j'}$ is older than s_i, that s_j and s_i must not share the same parent node (i.e $s_{i'} \neq s_{j'}$) and that $s_{i'}$ must not be a daughter node of $s_{j'}$. Each possible move is then randomly selected through some distribution, where we choose to use the discrete uniform distribution. If s_j is not the root then we attach it to some total height sampled from $h_{i'} \sim \text{Unif}(\max(h_i, h_j), h_{j'})$. Otherwise if s_j is the root node then the new height is simulated from some distribution with probability density $q_h(h_{i'})$, where appropriate choices include proposing from a uniform distribution (which we consider) of $h_{i'} \sim \text{Unif}(h_j, 1 + h_j)$ or an alternative method is to sample from an exponential distribution and then graft it above the root

node such that $h_{i'} \sim h_j + \text{Exp}(\cdot)$ (Didelot and Falush, 2007; Yang, 2014). The MH acceptance probability is based upon the densities of where the new branch is placed, with any topology based probabilities having a ratio of one (Höhna and Drummond, 2008). Therefore in order to move the node back to its original position we need to move node $s_{i'}$ between nodes s_r and $s_{r'}$, or above the new root node s_r. The ratio of $q(\gamma|\gamma')/q(\gamma'|\gamma)$ is shown in (4.18),

$$\begin{cases} \dfrac{1/(h_{r'} - \max(h_r, h_i))}{1/(h_{j'} - \max(h_j, h_i))} & \text{if neither } s_{i'} \text{ or } s_j \text{ is the root node in the current state} \\ \dfrac{1/(h_r - \max(h_r, h_i))}{q_h(h_{i'})} & \text{if } s_j \text{ is the root node in the current state} \\ \dfrac{q_h(h_{r'})}{1/(h_{j'} - \max(h_j, h_i))} & \text{if } s_{i'} \text{ is the root node in the current state.} \end{cases} \tag{4.18}$$

A flaw with this first version of this move is that the vast majority of moves will be improbable and are most likely to be rejected, and this issue will be far more common for high dimensional trees. Lakner *et al.* (2008) also showed that a randomised SPR move may perform worse in comparison to other types of moves such as nearest neighbor interchanges (NNI) moves, even though this was tested under non-coalescent assumptions. However since very basic genetic assumptions are made for this investigation we can create a more accurate proposal by temporarily estimating the ancestral states/sequences of each node and then basing the probability of a graft-prune moves on the distance between the pruned node and the node to be grafted above, termed as the Wilson & Balding move (Wilson and Balding, 1998). We consider the probability of proposing to prune and regraft an ancestor node s_i above some ancestor/tip node s_j which is proportional to

$$q(\cdot|s_i, s_j) \quad \propto \quad \frac{1}{1 + D_{s_i s_j}}, \tag{4.19}$$

where $D_{s_i s_j}$ represent the SNP differences between the sequences.

This kernel requires a temporary estimate of the unknown sequences for each

inner node via the forward-backward algorithm. The simplest estimation of an unknown ancestor is the root node A_1 at one locus point l for model $m_{(k-1)}$. We define the specific allele which has the largest marginal probability for the root node via

$$\{\max\}_l \pi(A_{1l}|y_{1l},...,y_{kl},\vartheta,x_{m_{k-1}}) \;=\; \max_l \left\{ \frac{p(A_{1l})\tilde{f}(A_{1l})}{\sum\limits_{A_{1l}} p(A_{1l})\tilde{f}(A_{1l})} \right\}, \qquad (4.20)$$

where the maximum is over the four alleles of $\{\mathrm{A},\mathrm{C},\mathrm{G},\mathrm{T}\}$ and this is repeated for each lth site. Here $\pi(s_{il}|y_{1l},...,y_{kl},\vartheta,x_{m_{k-1}})$ represents the marginal probability of node s_{il} having certain allele types at the lth site given the tip nodes only, although $\pi(s_{il}=y_{jl}|y_{1l},...,y_{kl},\vartheta,x_{m_{k-1}})$ naturally has a probability of one for having its true allele and zero otherwise. Furthermore the $\tilde{f}(s_{il})$ are still the conditional probabilities of having a certain nucleobase given the rest of the genealogy that descents from it as seen in (4.10). Should (4.20) have the maximum in more than one allele type then we randomly choose the allele via a discrete uniform distribution.

To understand how we derive the marginal probabilities of having a certain allele for the other inner nodes that are not the root node, we consider a simple example. We consider a tree with three sequences $\{y_1,y_2,y_3\}$, with a coalescent event (represented by node A_2) between sequences y_1 and y_2 before a final coalescent event occurs represented by the root node A_1. While $\pi(A_{1l}|y_{1l},y_{2l},y_{3l},\vartheta,x_{m_2})$ is given in (4.20) we define the marginal probabilities for inner node A_2 by

$$\begin{aligned}
\pi(A_{2l}|y_{1l},y_{2l},y_{3l},\vartheta,x_{m_2}) &= \sum_{A_{1l}} \left(\frac{\pi(A_{1l},A_{2l},y_{1l},y_{2l},y_{3l},\vartheta,x_{m_2})}{\pi(y_{1l},y_{2l},y_{3l},\vartheta,x_{m_2})} \right) \\
&= \frac{1}{\pi(y_{1l},y_{2l},y_{3l},\vartheta,x_{m_2})} \sum_{A_{1l}} (p(A_{1l})p_{A_{1l}A_{2l}}(x_{A_1A_2}) \\
&\quad \times p_{A_{1l}y_{3l}}(x_{A_1y_3})p_{A_{2l}y_{1l}}(x_{A_2y_1})p_{A_{2l}y_{2l}}(x_{A_2y_2})\tilde{f}(y_{1l}) \\
&\quad \times \tilde{f}(y_{2l})\tilde{f}(y_{3l}))
\end{aligned}$$

$$= \frac{p_{A_{2l}y_{1l}}(x_{A_2y_1})p_{A_{2l}y_{2l}}(x_{A_2y_2})\tilde{f}(y_{1l})\tilde{f}(y_{2l})}{\pi(y_{1l},y_{2l},y_{3l},\vartheta,x_{m_2})} \times\sum_{A_{1l}} p_{A_{1l}A_{2l}}(x_{A_1A_2})\left(p(A_{1l})p_{A_{1l}y_{3l}}(x_{A_1y_3})\tilde{f}(y_{3l})\right). \quad (4.21)$$

Letting $\theta = p(A_{1l})p_{A_{1l}y_{3l}}(x_{A_1y_3})\tilde{f}(y_{3l})$ we note that

$$\theta p_{A_{1l}A_{2l}}(x_{A_1A_2})p_{A_{2l}y_{1l}}(x_{A_2y_1})p_{A_{2l}y_{2l}}(x_{A_2y_2})\tilde{f}(y_{1l})\tilde{f}(y_{2l}) = \pi(A_{1l},A_{2l}|y_{1l},y_{2l},y_{3l},\vartheta,x_{m_2}) \times\pi(y_{1l},y_{2l},y_{3l},\vartheta,x_{m_2})$$

$$\theta\sum_{A_{2l}} p_{A_{1l}A_{2l}}(x_{A_1A_2})p_{A_{2l}y_{1l}}(x_{A_2y_1})p_{A_{2l}y_{2l}}(x_{A_2y_2})\tilde{f}(y_{1l})\tilde{f}(y_{2l}) = \pi(A_{1l}|y_{1l},y_{2l},y_{3l},\vartheta,x_{m_2}) \times\pi(y_{1l},y_{2l},y_{3l}, \vartheta,x_{m_2}), \quad (4.22)$$

in which we now consider that

$$\theta = \frac{\pi(y_{1l}\theta,y_{2l},y_{3l},\vartheta,x_{m_2})\pi(A_{1l}|y_{1l},y_{2l},y_{3l},\vartheta,x_{m_2})}{\sum_{A_{2l}} p_{A_{1l}y_{2l}}(x_{A_1y_2})p_{A_{2l}y_{1l}}(x_{A_2y_1})p_{A_{2l}y_{2l}}(x_{A_2y_2})\tilde{f}(y_{1l})\tilde{f}(y_{2l})}. \quad (4.23)$$

Therefore by substituting (4.23) into (4.21) we receive

$$\begin{aligned}\pi(A_{2l}|y_{1l},y_{2l},y_{3l},\vartheta,x_{m_2}) &= \frac{\pi(y_{1l},y_{2l},y_{3l},\vartheta,x_{m_2})}{\pi(y_{1l},y_{2l},y_{3l},\vartheta,x_{m_2})} \times p_{A_{2l}y_{1l}}(x_{A_2y_1})p_{A_{2l}y_{2l}}(x_{A_2y_2})\tilde{f}(y_{1l})\tilde{f}(y_{2l}) \\ &\quad\times\frac{p_{A_{1l}A_{2l}}(x_{A_1A_2})\pi(A_{1l}|y_{1l},y_{2l},y_{3l},\vartheta,x_{m_2})}{\sum_{A_{2l}} p_{A_{1l}A_{2l}}(x_{A_1A_2})p_{A_{2l}y_{1l}}(x_{A_2y_1})p_{A_{2l}y_{2l}}(x_{A_2y_2})\tilde{f}(y_{1l})\tilde{f}(y_{2l})} \\ &= p(x_{A_2y_1})p_{A_{2l}y_{2l}}(x_{A_2y_2})\tilde{f}(y_{1l})\tilde{f}(y_{2l}) \\ &\quad\times\frac{p_{A_{1l}A_{2l}}(x_{A_1A_2})\pi(A_{1l}|y_{1l},y_{2l},y_{3l},\vartheta,x_{m_2})}{\sum_{A_{2l}} p_{A_{1l}A_{2l}}(x_{A_1A_2})p_{A_{2l}y_{1l}}(x_{A_2y_1})p_{A_{2l}y_{2l}}(x_{A_2y_2})\tilde{f}(y_{1l})\tilde{f}(y_{2l})}.\end{aligned} \quad (4.24)$$

Overall for any ancestor node s_i we consider its parent node $s_{i'}$ and its daughter nodes

s_j and $s_{j'}$, and then identify the allele with the the highest probability via

$$\begin{aligned}
\{\max\}_l \pi(s_{il}|y,\vartheta,x,x_{m_{k-1}}) &= \max_l \left\{ \sum_{s_{(1:(k-1))\neq i)}l} \frac{\pi(s_{(1:(k-1))l}, y_{1l}, ..., y_{kl}, \vartheta, x_{m_{k-1}})}{\pi(y_{1l}, ..., y_{kl}, \vartheta, x_{m_{k-1}})} \right\} \\
&= \max_l \Bigg\{ \rho(s_{il}, s_{jl}, s_{j'l}, y) \\
&\quad \times \sum_{s_{i'l}} \frac{p_{s_{i'l}s_{il}}(x_{s_{i'}s_i})\pi(s_{i'l}|y_{1l}, ..., y_{kl}, \vartheta, x_{m_{k-1}})}{\sum_{s_{il}} p_{s_{i'l}s_{il}}(x_{s_{i'}s_i})\rho(s_{il}, s_{jl}, s_{j'l}, y)} \Bigg\} \quad (4.25) \\
\rho(s_{il}, s_{jl}, s_{j'l}, y) &= p_{s_{il}s_{jl}}(x_{s_is_j}) p_{s_{il}s_{j'l}}(x_{s_is_{j'}}) \tilde{f}(s_{jl}) \tilde{f}(s_{j'l}). \quad (4.26)
\end{aligned}$$

The above formulas to estimate the sequence for each inner node are sum-product formulas in which we estimate a posterior distribution over the inner nodes and identify the allele that maximises the *posteriori* (Jordan, 2004; Kschischang *et al.*, 2001). This is repeated for the daughter nodes s_j and $s_{j'}$, unless they are tip nodes, and that (4.25) should be normalised when selecting the mostly likely allele. Furthermore we move from the root and down towards the tip nodes as we need to calculate each conditional probability of $\pi(s_{il}|y_{1l}, ..., y_{kl}, \vartheta, x_{m_{k-1}})$. However the estimates of each ancestral sequences is not included when calculating the unnormalised posterior distribution, and as stated in section 4.4.1.1 the likelihood considers all possible allele combinations.

The computational cost to at least calculate all viable $q(\cdot|s_i, s_j)$ for a single site for a complete genealogy tree of n sequences is $O(n)$. Although while this cost is linear, it is still higher in comparison to the basic SPR move as we need to calculate both the basic likelihood via the pruning algorithm and also calculate (4.20) and (4.25) for all relevant nodes in which, just like the likelihood, increasing diversity of the sample sequences will increase this cost. Finally we need to re-estimate the ancestral nodes again for at least a subtree of the complete genealogy tree in order to define $q(\cdot|s_i, s_{j'})$ where $s_{j'}$ is the node that we need to place s_i above to return to the original tree.

Therefore the true cost could be many times that of the SPR move.

In conclusion we only make use of SPR-based topology type moves, even if other type of moves such as adaptions of the Nearest Neighbor interchange (Drummond *et al.*, 2002) can be more efficient. There is still no "gold-standard" topology move that provides consistent acceptance rates in high-dimensional parameter space (Lakner *et al.*, 2008). Other research has recommended a total branch length re-scaling move that shrinks or lengthens all branch lengths by some positive factor (Didelot and Falush, 2007; Yang, 2014), however we have opted to ignore this type of move.

4.4.3 Updating the Posterior by Grafting a New Observation

In each of these cases we aim to define which node we graft below the most recent ancestor of a new sequence and on what part of the branch we graft it onto for each model m_k, and that these transformation also define the form of each intermediate distribution. In each of our two proposed cases we make a new proposal for a new height $h_{y_{k+1}}$ from the present day on the new sequence to its next ancestor which also changes the ordering of said heights. These changes simultaneously act as our transformation on the set of coalescent time intervals.

4.4.3.1 Exponential/Uniform Graft Proposal

The first type of move is the least directed of the two moves and takes into account the expected height of the tree. We state that the the expected height/time of the MRCA for all sequences, given current observational size $k+1$ for model m_k, is defined by

$$\begin{aligned} \mathrm{E}[x_{m_k(2:(k+1))}] &= \sum_{i=2}^{k+1} \frac{2}{i(i-1)} \\ &= \frac{2k}{k+1}. \end{aligned} \tag{4.27}$$

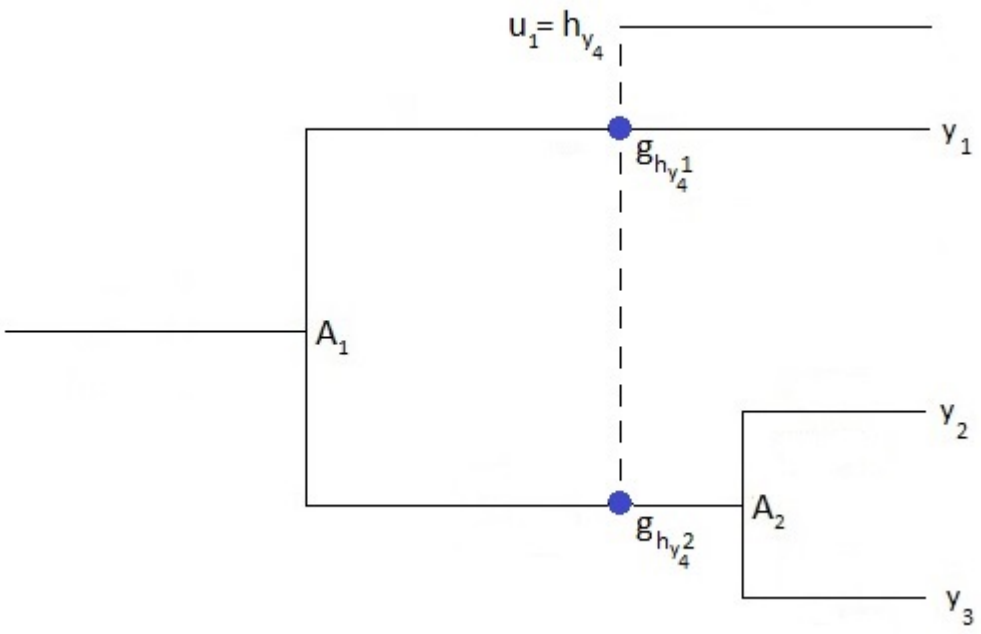


Figure 4.6: Illustration of the exponential/uniform graft proposal, when grafting a fourth individual onto a tree with three individuals. There are two branches where it could be placed conditional on the proposed height.

Therefore we propose a height of the parent node of a new sequence, distributed by $h_{y_{k+1}} = u_1 \sim \psi_{1,m_{k-1}\to m_k} \equiv \text{Exp}\left(\frac{k+1}{2k}\right)$. For example, for three sequences this is an exponential distribution with a rate parameter of 0.75 in which the said parameter eventually converges to 0.5 with increasing number of sequences. The second step uses a discrete uniform distribution to propose which nodes to graft the ancestor node above given the height. Note that we let $g_{h_{y_{k+1}}}$. represent all the possible branches that the ancestor of the sequence y_{k+1} can be grafted to given the proposed height (with a specific ith location given by $g_{h_{y_{k+1}}i}$). A reverse transformation simply involves removing the branch itself. An illustration of the move is shown in figure 4.6.

This gives us an importance weight when we define the intermediate distributions of our tSMC distribution via $\varphi_0 = 0$ and $\varphi_1 = 1$, and letting $x_{m_{k-1}}$ and $\gamma_{m_{k-1}}$ represent

the parameters for model m_{k-1}, of

$$\begin{aligned}\frac{\rho_T(x_{m_k},\gamma_{m_k},\vartheta;m_{k-1}\to m_k)}{\rho_0(x_{m_k},\gamma_{m_k},\vartheta;m_{k-1}\to m_k)} &= \frac{f(y_{1:(k+1)}|x_{m_k},\gamma_{m_k},\vartheta)p(x_{m_k},\gamma_{m_k},\vartheta)}{f(y_{1:k}|x_{m_{k-1}},\gamma_{m_{k-1}},\vartheta)p(x_{m_{k-1}},\gamma_{m_{k-1}},\vartheta)} \\ &\times\frac{1}{\psi_{2,m_{k-1}\to m_k}(g_{h_{y_{k+1}}\cdot}|h_{y_{k+1}})} \\ &\times\frac{1}{\psi_{1,m_{k-1}\to m_k}(h_{y_{k+1}})}. \qquad (4.28)\end{aligned}$$

We can identify the probability density of $\psi_{2,m_{k-1}\to m_k}(g_{h_{y_{k+1}}\cdot}|h_{y_{k+1}})$ by scanning the genealogy tree to determine what positions the ancestor node of the new sequence could have been grafted to, therefore the move is flexible where we could perform a SPR move to move the subtree of the said ancestor node and still be able to evaluate (4.28). While we believe this transformation has the potential to cover all possible probability mass of a posterior distribution, it may require a large number of particles effectively cover all regions of high posterior probability mass.

4.4.3.2 Laplace Approximation Based Proposal

We considered a second transformation which is more directed and takes into account the differences between sequences. The process involves a two-part proposal.

The first part involves defining a tip node in which we can consider a path that starts from the selected sequence and continues towards the MRCA of the tree and then moving towards infinity. Thus we consider a generalisation, described in Li and Stephens (2003), of Ewens formula (Ewens, 1972). Firstly, as described in section 4.2.2, we assume that each locus mutates independently with Poisson rate of $\vartheta/2$ and thus the Poisson mutation rate for the whole sequence is $L\vartheta/2$. If we were to consider one particular locus, the conditional distribution that the locus of the new sequence y_{k+1} to be introduced will differ by $D_{y_i y_{k+1}}$ mutations (or SNP differences) from the same locus of one of the randomly chosen sequences y_i currently grafted onto the tree is given by a geometric distribution with rate $k/(k+\vartheta)$, given that we have k sequences in a tree corresponding to model m_{k-1}. For example the probability of no

mutations on a specific site from one of the k sequences is given by

$$Pr(D_{y_i y_{k+1}} = 0 | y_i, y_{k+1}) = \frac{k}{k+\vartheta}, \tag{4.29}$$

such that it reproduces Ewens sampling formula in the special case of the infinite sites model (Li and Stephens, 2003; Stephens and Donnelly, 2000). When considering the sequence as a whole, the conditional probability of the number of mutations is given by

$$\begin{aligned} Pr(D_{y_i y_{k+1}} | y_i, y_{k+1}) &= \left(\frac{k}{k+L\vartheta}\right)\left(1 - \frac{k}{k+L\vartheta}\right)^{D_{y_i y_{k+1}}} \\ &= \left(\frac{k}{k+L\vartheta}\right)\left(\frac{k+L\vartheta - k}{k+L\vartheta}\right)^{D_{y_i y_{k+1}}} \\ &= \left(\frac{k}{k+L\vartheta}\right)\left(\frac{L\vartheta}{k+L\vartheta}\right)^{D_{y_i y_{k+1}}}. \end{aligned} \tag{4.30}$$

Therefore given a total of k conditional distributions based from (4.30), we choose to construct a discrete distribution to select a sequence y_i with probabilities proportional to (4.30) defined as

$$g_{y_i} \propto \left(\frac{L\vartheta}{k+L\vartheta}\right)^{D_{y_i y_{k+1}}}, \tag{4.31}$$

and thus higher proportional probabilities exist for the smallest $D_{y_i y_{k+1}}$ SNP differences.

Once the tip node has been selected, the second part of the transformation involves choosing to graft the new sequence based on some sampled height, representing the distance between the new sequence and its most recent ancestor, within the path of the chosen tip node. This distance is based on the pairwise likelihood, which is the binomial probability (with the binomial coefficient dropped) of having $D_{y_i y_{k+1}}$ SNP

differences under the JC69 substitution model. This is given by,

$$\begin{aligned}\tilde{L}(h_{y_{k+1}}|y_i, y_{k+1}) &= \left(\frac{3}{4} - \frac{3}{4}\exp\left(-\frac{4\vartheta h_{y_{k+1}}}{3}\right)\right)^{D_{y_i y_{k+1}}} \\ &\quad \times \left(\frac{1}{4} + \frac{3}{4}\exp\left(-\frac{4\vartheta h_{y_{k+1}}}{3}\right)\right)^{L - D_{y_i y_{k+1}}}. \end{aligned} \tag{4.32}$$

Note that we consider an adjustment of this likelihood, in comparison to JC69 sequences distances between two sequences shown in Yang (2014), where the terms $4\vartheta h_{y_{k+1}}$ replace the terms $2\vartheta h_{y_{k+1}}$, in comparison to section 4.2.3, as we have to consider both the distance from the new sequence to some unknown ancestor and then to the selected tip node which doubles the distance. Based on this likelihood we suggest proposing a new height for the sequence using a Laplace approximation of the likelihood given by $h_{y_{k+1}} \sim N(\mu = \tilde{h}, \tau = -\tilde{H})$ where $\tilde{h}$ is the maximum likelihood estimate and $\tilde{H}$ is the Hessian of the log likelihood of 4.32.

To illustrate how we solve the corresponding Laplace approximation, we define some variable g where

$$g \equiv g(h_{y_{k+1}}) = \frac{3}{4} - \frac{3}{4}\exp(-\frac{4\vartheta h_{y_{k+1}}}{3}) \tag{4.33}$$

$$h_{y_{k+1}} = -\frac{3}{4\vartheta}\log(1 - \frac{4g}{3}), \tag{4.34}$$

such that the pairwise log-likelihood is defined by

$$\log(\tilde{L}(g(h_{y_{k+1}})|y_i, y_{k+1})) = D_{y_i y_{k+1}}\log(g) + (L - D_{y_i y_{k+1}})\log(1 - g). \tag{4.35}$$

The MLE of the log-likelihood is obtained by differentiating (4.35) with respect to g and then setting its value to zero, such that

$$\frac{\partial \log(\tilde{L}(g|y_i, y_{k+1}))}{\partial g} = 0 \tag{4.36}$$

$$\begin{aligned}\frac{D_{y_iy_{k+1}}}{g}-\frac{L-D_{y_iy_{k+1}}}{1-g} &= 0\\ (1-g)D_{y_iy_{k+1}} &= g(L-D_{y_iy_{k+1}})\\ \tilde{g} &= \frac{D_{y_iy_{k+1}}}{L}. \end{aligned} \tag{4.37}$$

In regards to defining the Hessian matrix, by considering the chain rule we note that

$$\begin{aligned}\frac{\partial^2\log(\tilde{L}(h_{y_{k+1}}|y_i,y_{k+1}))}{\partial h^2_{y_{k+1}}} &= \frac{\partial}{\partial h_{y_{k+1}}}\left(\frac{\partial\log(\tilde{L}(g|y_i,y_{k+1}))}{\partial g}\frac{\partial g}{\partial h_{y_{k+1}}}\right)\\ &= \left(\frac{\partial g}{\partial h_{y_{k+1}}}\right)^2\frac{\partial^2\log(\tilde{L}(g|y_i,y_{k+1}))}{\partial g^2}\\ &\quad+\frac{\partial^2 g}{\partial h^2_{y_{k+1}}}\left(\frac{\partial\log(\tilde{L}(g|y_i,y_{k+1}))}{\partial g}\right), \end{aligned} \tag{4.38}$$

as $\partial\log(\tilde{L}(\tilde{g}|y_i,y_{k+1}))/\partial\tilde{g}=0$, then the double differential of the log likelihood with respect to the height is equivalent to,

$$\begin{aligned}\left.\frac{\partial^2\log(\tilde{L}(h_{y_{k+1}}|y_i,y_{k+1}))}{\partial h^2_{y_{k+1}}}\right|_{h_{y_{k+1}}=\tilde{h}} &= \left.\frac{\partial^2\log(\tilde{L}(g|y_i,y_{k+1}))}{\partial g^2}\right|_{g=\tilde{g}}\left.\left(\frac{\partial g}{\partial h_{y_{k+1}}}\right)^2\right|_{h_{y_{k+1}}=\tilde{h}}\\ &= -\left(\frac{D_{y_iy_{k+1}}}{\tilde{g}^2}-\frac{L-D_{y_iy_{k+1}}}{(1-\tilde{g})^2}\right)\\ &\quad\times\left(\frac{\partial}{\partial\tilde{h}_{y_{k+1}}}\left(\frac{3}{4}-\frac{3}{4}\left(\exp-\frac{4\vartheta\tilde{h}_{y_{k+1}}}{3}\right)\right)\right)^2\\ &= -\left(\frac{D_{y_iy_{k+1}}}{\tilde{g}^2}-\frac{L-D_{y_iy_{k+1}}}{(1-\tilde{g})^2}\right)\\ &\quad\times\left(\vartheta\exp\left(-\frac{4\vartheta\tilde{h}_{y_{k+1}}}{3}\right)\right)^2. \end{aligned} \tag{4.39}$$

So finally we obtain exact forms of $\tilde{h}$ and $\tilde{H}$

$$\begin{aligned}\tilde{h} &= -\frac{3}{4\vartheta}\log(1-\frac{4\tilde{g}}{3})\\ &= -\frac{3}{4\vartheta}\log(1-\frac{4D_{y_iy_{k+1}}}{3L}) \end{aligned} \tag{4.40}$$

$$\begin{aligned}\tilde{H} &= -\left.\frac{\partial^2 \log(\tilde{L}(h_{y_{k+1}}|y_i, y_{k+1}))}{\partial h_{y_{k+1}}^2}\right|_{h_{y_{k+1}}=\tilde{h}} \\ &= \left(\frac{D_{y_i y_{k+1}}}{\tilde{g}^2} - \frac{L - D_{y_i y_{k+1}}}{(1-\tilde{g})^2}\right)\vartheta^2 \exp\left(-\frac{8\vartheta\tilde{h}_{y_{k+1}}}{3}\right). \end{aligned} \tag{4.41}$$

However we consider a variance stabilising transformation by Reis and Yang (2011), due to the fact the log likelihood has an exponential downward curve (with vastly different gradients between $h_{y_{k+1}} < \tilde{h}$ and $h_{y_{k+1}} > \tilde{h}$) in which larger h are expected to have larger sampling errors. Therefore we use their arcsine transformation suggestion with $u = 2\text{arcsin}(\sqrt{\frac{3}{4} - \frac{3}{4}\exp(-\frac{4\vartheta h_{y_{k+1}}}{3})})$, such that $h_{y_{k+1}} = -\frac{3}{4\vartheta}\log\left(1 - \frac{4\sin^2(u/2)}{3}\right)$. Furthermore we note that

$$g = \sin^2\left(\frac{u}{2}\right). \tag{4.42}$$

The MLE for $\tilde{u}$ is simply given by

$$\begin{aligned}\tilde{u} &= 2\text{arcsin}\left(\sqrt{\frac{3}{4} - \frac{3}{4}\exp(-\frac{4\vartheta\tilde{h}_{y_{k+1}}}{3})}\right) \\ &= 2\text{arcsin}\left(\sqrt{\tilde{p}}\right) \\ &= 2\text{arcsin}\left(\sqrt{\frac{D_{y_i y_{k+1}}}{L}}\right). \end{aligned} \tag{4.43}$$

By applying the chain rule again we can also define the double differential of the log likelihood with respect to $\tilde{u}$ to be

$$\begin{aligned}\left.\frac{\partial^2 \log(\tilde{L}(u|y_i, y_{k+1}))}{\partial u^2}\right|_{u=\tilde{u}} &= \left.\frac{\partial^2 \log(\tilde{L}(g|y_i, y_{k+1}))}{\partial g^2}\right|_{g=\tilde{g}} \left(\frac{\partial g}{\partial u}\right)^2\Bigg|_{u=\tilde{u}} \\ &= -\left(\frac{D_{y_i y_{k+1}}}{\tilde{g}^2} - \frac{L - D_{y_i y_{k+1}}}{(1-\tilde{g})^2}\right)\left(\frac{\partial}{\partial\tilde{u}}\sin^2\left(\frac{\tilde{u}}{2}\right)\right)^2 \\ &= -\left(\frac{D_{y_i y_{k+1}}}{\tilde{g}^2} - \frac{L - D_{y_i y_{k+1}}}{(1-\tilde{g})^2}\right)\left(\sin\left(\frac{\tilde{u}}{2}\right)\cos\left(\frac{\tilde{u}}{2}\right)\right)^2 \end{aligned}$$

$$
\begin{aligned}
&= -\left(\frac{D_{y_iy_{k+1}}}{\left(\frac{D_{y_iy_{k+1}}}{L}\right)^2} - \frac{L - D_{y_iy_{k+1}}}{\left(1-\left(\frac{D_{y_iy_{k+1}}}{L}\right)\right)^2}\right) \\
&\quad \times \left(\sin\left(\arcsin\left(\sqrt{\frac{D_{y_iy_{k+1}}}{L}}\right)\right)\cos\left(\arcsin\left(\sqrt{\frac{D_{y_iy_{k+1}}}{L}}\right)\right)\right)^2 \\
&= -\left(\frac{D_{y_iy_{k+1}}}{\left(\frac{D_{y_iy_{k+1}}}{L}\right)^2} - \frac{L - D_{y_iy_{k+1}}}{\left(1-\left(\frac{D_{y_iy_{k+1}}}{L}\right)\right)^2}\right)\left(\sqrt{\frac{D_{y_iy_{k+1}}}{L}}\sqrt{1-\frac{D_{y_iy_{k+1}}}{L}}\right)^2 \\
&= -\left(\frac{D_{y_iy_{k+1}}\left(1-\frac{D_{y_iy_{k+1}}}{L}\right)}{\left(\frac{D_{y_iy_{k+1}}}{L}\right)} - \frac{\left(L - D_{y_iy_{k+1}}\right)\left(\frac{D_{y_iy_{k+1}}}{L}\right)}{\left(1-\left(\frac{D_{y_iy_{k+1}}}{L}\right)\right)}\right) \\
&= -\left(\frac{D_{y_iy_{k+1}}\left(1-\frac{D_{y_iy_{k+1}}}{L}\right)}{\left(\frac{D_{y_iy_{k+1}}}{L}\right)} - \frac{\left(L - D_{y_iy_{k+1}}\right)\left(\frac{D_{y_iy_{k+1}}}{L}\right)}{\left(1-\left(\frac{D_{y_iy_{k+1}}}{L}\right)\right)}\right) \\
&= -\left(\frac{LD_{y_iy_{k+1}}\left(L - D_{y_iy_{k+1}}\right)}{LD_{y_iy_{k+1}}} - \frac{\left(L - D_{y_iy_{k+1}}\right)\left(D_{y_iy_{k+1}}L\right)}{\left(L - D_{y_iy_{k+1}}\right)L}\right). \\
&= -L
\end{aligned} \tag{4.44}
$$

Finally we define proposal for the new height to be given by,

$$
u^{'} \sim N\left(\mu = 2\arcsin\left(\sqrt{\frac{D_{y_iy_{k+1}}}{L}}\right), \tau = L\right) \tag{4.45}
$$

$$
h_{y_{k+1}} = -\left(\frac{3}{4\vartheta}\right)\log\left(1 - \frac{4\sin^2(u^{'}/2)}{3}\right). \tag{4.46}
$$

Another property from this transformation, in comparison to the untransformed version, is that drawing from (4.45) and (4.46) will always produce a positive real number for the height of the new node. A notable issue with this type of move is if $h_{y_{k+1}}$ is higher than any ancestral node of y_i then it is not possible to trace back its path to the new node placement, for example this could have been generated by starting from the

sister node of y_i. While we could add a label that defines a descendant tip sequence, this could limit the topology exploration by keeping the sequence in a subset of places in the tree. Thus we decondition over the possible number of tip sequences that could of resulted in the proposed tree. The overall move has a Jacobian of 1. Applying this transformation move gives a weight update, when we only consider two intermediate distributions where $\varphi_0 = 0$ and $\varphi_T = 1$, of

$$\begin{aligned}\frac{\rho_T(x_{m_k}, \gamma_{m_k}, \vartheta; m_{k-1} \rightarrow m_k)}{\rho_0(x_{m_k}, \gamma_{m_k}, \vartheta; ; m_{k-1} \rightarrow m_k)} &= \frac{f(y_{1:(k+1)}|x_{m_k}, \gamma_{m_k}, \vartheta)p(x_{m_k}, \gamma_{m_k}, \vartheta)}{f(y_{1:k}|x_{m_{k-1}}, \gamma_{m_{k-1}}, \vartheta)p(x_{m_{k-1}}, \gamma_{m_{k-1}}, \vartheta)} \\ &\quad \times \sum_{y_i \in y_{g_{y_{k+1}}}} (\psi_{1,m_{k-1}\rightarrow m_k}(g_{y_i}|\vartheta) \\ &\quad \times \psi_{2,m_{k-1}\rightarrow m_k}(h_{y_{k+1}}|g_{y_i}, \vartheta))^{-1}, \end{aligned} \tag{4.47}$$

where g_{y_i} is defined via (4.31) and $y_{g_{y_{k+1}}}$ is the set of all the sequences/tip nodes, except for the newly grafted sequence, contained within a subtree where the root of it has a daughter node being the newly grafted sequence. Otherwise the density of $\psi_{2,m_{k-1}\rightarrow m_k}(h_{y_{k+1}}|g_{y_i}, \vartheta)$ is given by

$$\psi_{2,m_{k-1}\rightarrow m_k}(h_{y_{k+1}}|g_{y_i}, \vartheta) = \left(\frac{\partial}{\partial h_{y_{k+1}}} u\right) \sqrt{\frac{\tau}{2\pi}} \exp\left(\frac{\tau(u-\mu)^2}{2}\right), \tag{4.48}$$

where $\mu = 2\text{arcsin}\left(\sqrt{D_{y_i y_{k+1}}/L}\right)$, $\tau = L$ and

$$\begin{aligned}\left(\frac{\partial}{\partial h_{y_{k+1}}} u\right) &= \frac{\partial}{\partial h_{y_{k+1}}} 2\text{arcsin}\left(\sqrt{\frac{3}{4} - \frac{3}{4}\exp(-\frac{4\vartheta h_{y_{k+1}}}{3})}\right) \\ &= 2\left(\frac{\partial}{\partial h_{y_{k+1}}} \sqrt{\frac{3}{4} - \frac{3}{4}\exp(-\frac{4\vartheta h_{y_{k+1}}}{3})}\right) \left(\frac{1}{\sqrt{1 - \left(\frac{3}{4} - \frac{3}{4}\exp(-\frac{4\vartheta h_{y_{k+1}}}{3})\right)}}\right)\end{aligned}$$

$$
\begin{aligned}
&= \left(\frac{2}{\sqrt{1 - \left(\frac{3}{4} - \frac{3}{4}\exp(-\frac{4\vartheta h_{y_{k+1}}}{3}) \right)}} \right) \left(\frac{1}{2\sqrt{\frac{3}{4} - \frac{3}{4}\exp(-\frac{4\vartheta h_{y_{k+1}}}{3})}} \right) \\
&\quad \times \frac{\partial}{\partial h_{y_{k+1}}} \left(\frac{3}{4} - \frac{3}{4}\exp(-\frac{4\vartheta h_{y_{k+1}}}{3}) \right) \\
&= \left(\frac{\vartheta \exp(-\frac{4\vartheta h_{y_{k+1}}}{3})}{\sqrt{1 - \left(\frac{3}{4} - \frac{3}{4}\exp(-\frac{4\vartheta h_{y_{k+1}}}{3}) \right)}} \right) \left(\frac{1}{\sqrt{\frac{3}{4} - \frac{3}{4}\exp(-\frac{4\vartheta h_{y_{k+1}}}{3})}} \right). \qquad (4.49)
\end{aligned}
$$

An example of the transformation in practice is shown in figure 4.7 where a duplicate of the sequence y_1 is being grafted. What can be noticed from the example is the concentration of proposals near the tip node of the currently grafted y_1, with fewer proposals to be grafted above sequences with far few differences such as y_2 and y_5. If it was assigned to be placed above other sequences then the recommended heights are within the range where y_1 shares a MRCA with a certain sequence. We could have stricter or looser grafting probabilities proportional to some function of (4.31), however we believe that the existing probabilities in (4.31) are appropriate enough as seen in figure 4.7 which give some chance for unrelated sequences to follow its path to the root but still prioritise genomes that are more related.

Otherwise for any of the two moves we are strongly dependent on having ϑ converge via the MCMC steps, and although we believe that the parameter is unlikely to vary greatly between large genealogy trees we would still need to analyse if such a transformation on ϑ is needed.

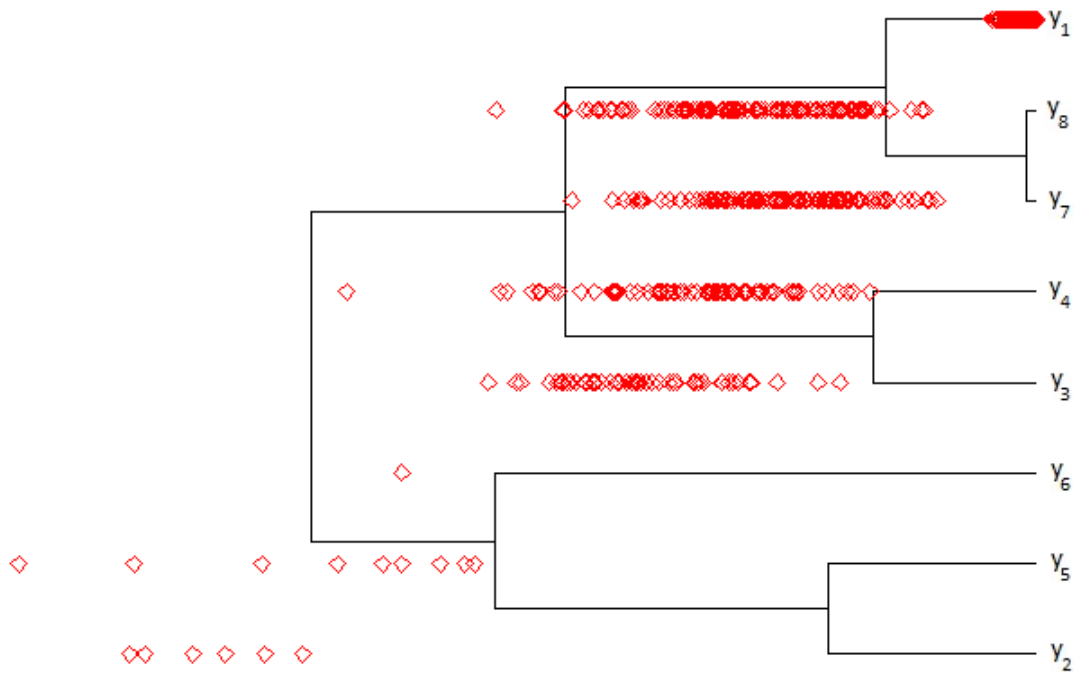


Figure 4.7: An example of multiple proposals made by the Laplace approximation plotted onto the same tree. In this case we are grafting the duplicate sequence of y_1 to the existing tree.

4.5 Diagnostics and Tests for the tSMC Adaption for Genealogy Reconstruction

We attempt to reconstruct the ancestral history of *Staphylococcus aureus* sequences from their multi locus sequence typing genes (Enright *et al.*, 2000). The combined sequences, with a length of 3186 sites, consists of the following housekeeping genes; *arc* (Carbamate kinase), *aro* (Shikimate dehydrogenase), *glp* (Glycerol kinase), *gmk* (Guanylate kinase), *pta* (Phosphate acetyltransferase), *tpi* (Triosephosphate isomerase) and *yqi* (Acetyle coenzyme A acetyltransferase). There exist no missing/unknown alleles at any of the loci. For simplicity it is assumed that the sequences do not exhibit recombination, with the basic concept being that a chromosome may exchange genetic material with other chromosomes (with the processes varying between haploid or diploid genomes) and thus any inference on the genomes ancestry when assuming

SNP changes by substitutions only is very likely to be wrong. However it is known that *Staphylococcus aureus* does go through some form of horizontal gene transfer which is a type of recombination (Everitt *et al.*, 2014), and therefore any biological interpretations from these results should be taken with caution. The MLST types that we use are $\{1, 5, 6, 8, 20, 22, 25, 34, 36, 39, 45, 59, 88, 93, 97, 101, 105, 123, 133, 151, 239, 250, 398\}$.

Unlike in chapter 3 we are not interested in the general properties of different tSMC adaptions, such as the difference between setting either adaptive and fixed geometric bridging intermediate distributions, and focus on application specific tests. We analyse the differences of the estimated posteriors when new sequences were grafted under the exponential/uniform proposal and the Laplace approximation proposal. We also consider the scenario where no topology moves are made to investigate if it is possible to avoid such moves if the transformation proposals alone can target regions of high posterior density for each ordered genealogy of incrementally increasing observational size. (as seen in Fourment *et al.* (2018)). We consider the following;

- We analyse the genealogy tree and the marginal posterior distributions of ϑ. We also analyse how many intermediate distributions were required to convergence under each scheme.

- We consider the differences in the Monte Carlo error per likelihood calculation between the two algorithms, and also analyse the ML under two different orderings to graft the sequences onto a tree.

- On a minor note we consider how appropriate some of our suggested kernel moves. In particular we analyse the two possible tuning schemes for proposing changes to the topology or each of the node heights as described in section 4.4.2.

With regards to the mentioned orderings, one ordering involves grafting the sequences to an existing tree depending on the smallest SNP distance between a new sequence and the existing sequences. In particular this ordering is made by first constructing

a symmetric matrix, with each sequence indexed within the rows and columns, representing the SNP differences between each sequence (with the diagonals naturally equating to zero). We pick the two sequences which have the smallest SNP difference between them and then delete their corresponding rows from the SNP difference matrix. Afterwards we extract all matrix columns who are indexed by the sequences which are currently in the ordering (or currently grafted onto a tree), perform column matrix addition for all said columns and then the next sequence to graft onto a tree is based on the row, given by the matrix representing the added columns, with the smallest SNP distance. This is then followed by deleting its corresponding row in the SNP difference matrix, and the process continues until all observations have been ordered. With this schedule we expect the recent ancestor node of the new sequence to be the new root node or be an ancestor to a large subset of the current tip nodes.

The second ordering considers the largest to the smallest SNP differences, such that new sequences are most likely to have their recent ancestor nodes have one of the other tip nodes as a daughter node. This ordering is defined by the same procedure as the other stated ordering, except we consider the largest SNP difference between sequences in comparison to the smallest.

To visualise the particle representation of the posterior on the tree space we create a weighted 50% majority rule consensus tree (Bryant, 2003; Margush and McMorris, 1981) that involves sampling from the particle set, dependent on the particle weights, and then construct a consensus tree treating all sampled trees as equal. Although there may be some variation in the ancestral topology of each genealogy tree we do expect certain subtrees, such as the subtrees for the daughters of the root node, to contain a consistent subset of tip nodes corresponding to a specific set of sequences. It should be noted that it is not always possible to set up a consensus tree that appears to be represented by coalescent model assumptions, as summary statistics for branch lengths derived from a set of duplicate subtree clades can result in a tree where all the tip nodes do not exist in the present due to uneven branch lengths. Therefore we

present consensus trees where the sample sequences may not match perfectly to the present. We do not apply more advanced consensus trees such as adaptions of the greedy consensus trees or otherwise more refined algorithms (Bryant, 2003; Degnan *et al.*, 2009).

Furthermore pre-testing showed that both transition moves do not initially cause a large number of zero weighted particles, and thus we do not use a scheme which sets $\varphi_1 = 10^{-8}$ as we did in chapter 3 but instead set all $\{\varphi_0 = 0, \varphi_1, \varphi_2, ...\}$ adaptively. All of the stated tests are analysed when using an adaptive scheme to set the number of intermediate distributions as described in chapter 2, and we have them dictated by aiming for the CESS to be equal to $0.95N$ (where again N is the number of particles). Otherwise we apply adaptive MCMC kernels, where at each state we apply 10 SPR MCMC kernel moves and one individual adaptive MCMC kernel for the node heights and population size parameter as given in section 4.4.

For tests involving the differences between the MCMC kernel moves within the tSMC adaption we use 250 particles and up to a subset of 15 sequences, and when investigating the best methods for proposing node heights we analyse this while applying one W&B move and 10 SPR moves respectively. We use these tests to determine what the exact MCMC kernel moves should be applied when analysing the marginal likelihood estimates from our tSMC runs. Although within section 4.6 we do state that based on these results we do apply 10 SPR moves when analysing the consensus trees and the ML, and that there was no notable difference between the two MCMC proposals for the node heights.

For marginal likelihood estimates we use a particle size of 250 with all 23 sequences, and what conditions this is analysed depends on the tests of each kernel.

We display consensus trees which are generated under 1000 particles, with again 10 SPR moves and one MCMC proposal for the population size parameters and each individual height, and analyse whether the consensus topology matches with what is shown with established methods.

We compare the 1000 particle runs of tSMC to MCMC, under the same prior conditions, with an iteration size of 1.5×10^6 and a burn-in period of 10^6. Within this run of the MCMC we analyse whether we obtain similar results. The MCMC algorithm applies the same MH moves as our tSMC algorithm but instead we consider the adaptive metropolis algorithm (see for example Haario *et al.* (2001); Roberts and Rosenthal (2009)), as an alternative algorithm to adaptively give proposals to the population size parameter and node heights. Considering the population size parameter as an example, a proposal, $\tilde{\vartheta}$, for the parameter is given by

$$\begin{aligned} \log(\tilde{\vartheta}) &\sim (0.95 \times \text{Normal}(\mu = \log(\vartheta), \tau = (2.38^2 \upsilon_\vartheta)^{-1})) \\ &+ (0.05 \times \text{Normal}(\mu = \log(\vartheta), \tau = (0.01)^{-1})), \end{aligned} \tag{4.50}$$

where υ_ϑ is defined by the variance of the current Monte Carlo estimates from at least two iterations of the Markov chain. Otherwise we use a proposal of $\log(\tilde{\vartheta}) \sim \text{Normal}(\mu = \log(\vartheta), \tau = (0.01)^{-1})$ in the first 5 iterations of the Markov chain. Although a downside is that the adaptive metropolis algorithm works best if each marginal posterior distribution is expected to be similar to a Gaussian distribution, for example in (4.50) we need $\log(\vartheta)$ to justifiably be defined by a Gaussian distribution, an issue that we discussed in chapter 2. It this is not the case then the adaptive metropolis algorithm is not the most efficient in exploring a parameter space (Haario *et al.*, 2001; Roberts and Rosenthal, 2009).

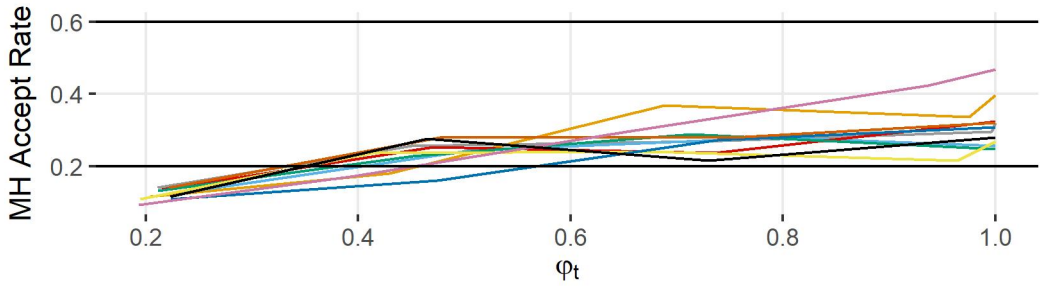
4.6 Results

Before major tests were run we first analysed the rate of successful MH proposals under an adaptive MCMC kernel scheme when the node height tuning variances are dictated via the variance of the ordered heights or by the residuals between the branches of the genealogy and the population size parameter. This was tested under both the height and Laplace based transformation proposals, under 250 particles and

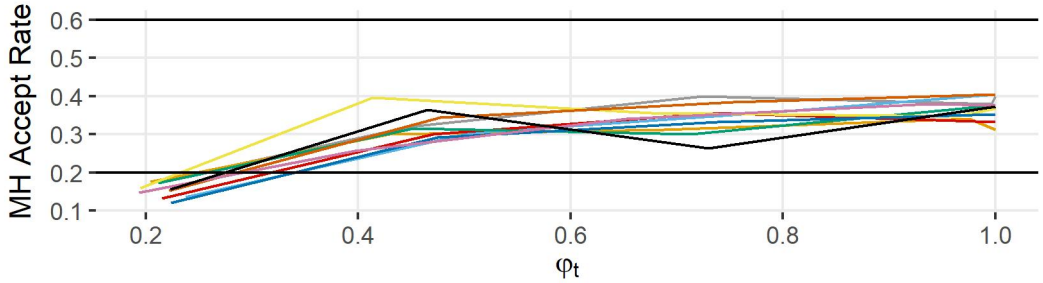
up to a subset of the 15 out of the 23 sequences. As each height scales with the population size, and thus comparisons with how far each height has jumped successfully may give misleading results, we considered analysing the mean square jump (MSJ) distance under the number of mutations per site time scaling as described in section 4.2.2.

Acceptance probabilities are shown in figure 4.8, and we note that they have shown the same pattern and approximate regions of acceptance probability. No scheme gave better initial acceptance rates than the other. Regarding the mean square jump distance, in figures 4.9, 4.10 and 4.11 what can be seen is that there is no consistent difference between the two adaptive tuning variances. Sometimes a certain variance tuning scheme might perform better overall at certain transitions and other times it is roughly the same, and the patterns also vary with each parameter. More importantly, regardless of one scheme being slightly more appropriate for a smaller number of sequences, each scheme appears to give approximately the same marginal likelihood estimates when identical orderings are considered (which we do not show). What was interesting was how the exponential/uniform proposal also gave poorer MSJ the vast majority of the time in comparison to the Laplace approximation. We choose to consider the first scheme that does not use the correlation between the population size parameter and the branches, although using the other scheme should not have a massive impact on the results if inferring high dimensional trees was the key objective. Furthermore we also consider if there was a better way to implement these kernel moves as we discuss in section 4.7.

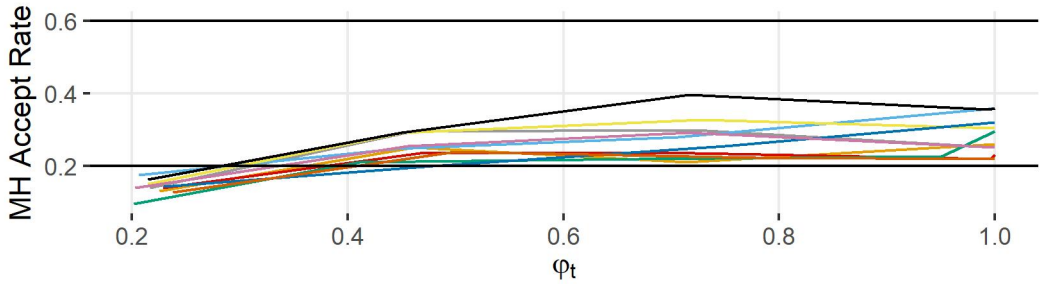
As part of other diagnostics we examined the effectiveness of the SPR moves. We noticed that performing only one iteration of the most basic SPR move as given by (4.18) had very poor acceptance rates, although this was somewhat to be expected as a near identical move given by BEAST also showed similar properties, however this is why we have considered the more advanced variant. We found that using a version of the W&B that considers the estimates of the ancestor nodes did improve the rate of



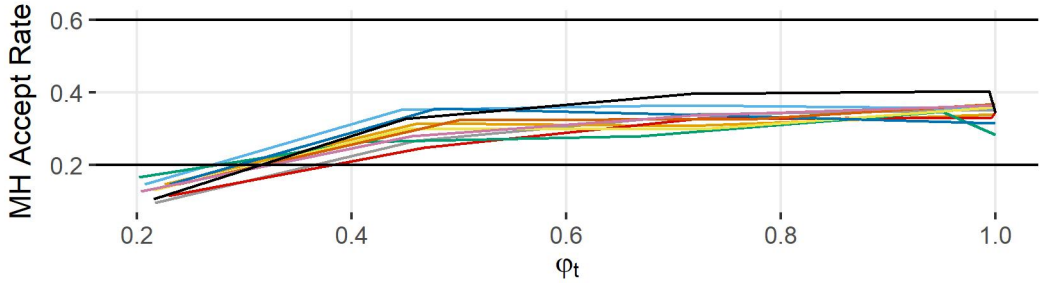
(a) Acceptance probabilities for the height to the first coalescent event under non-residual tuning scheme.



(b) Acceptance probabilities for the Population Size Parameter under non-residual tuning scheme.

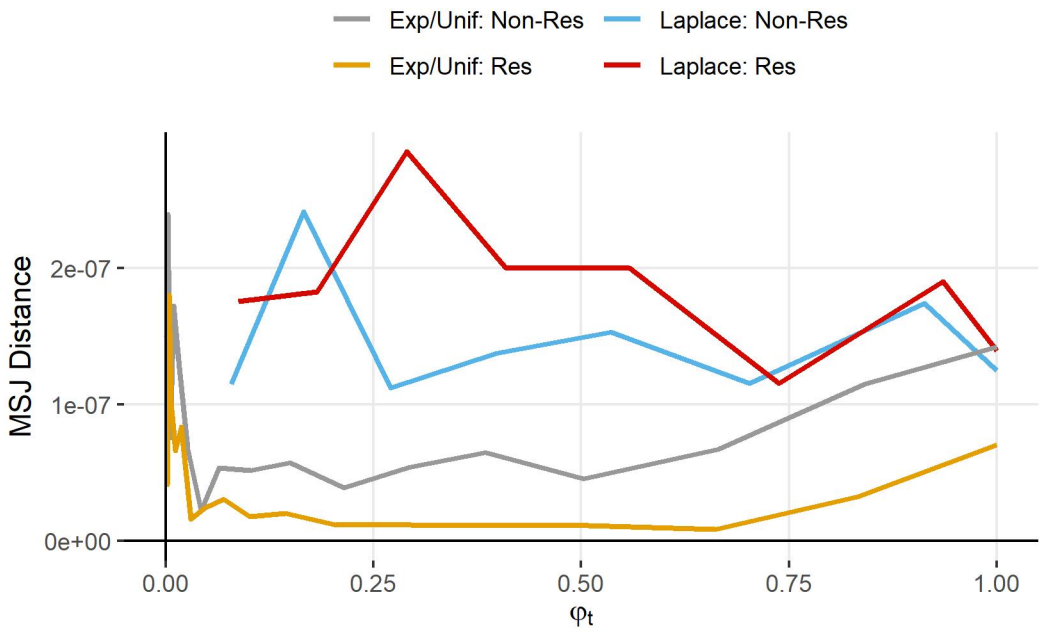


(c) Acceptance probabilities for the height to the first coalescent event under residual tuning scheme.

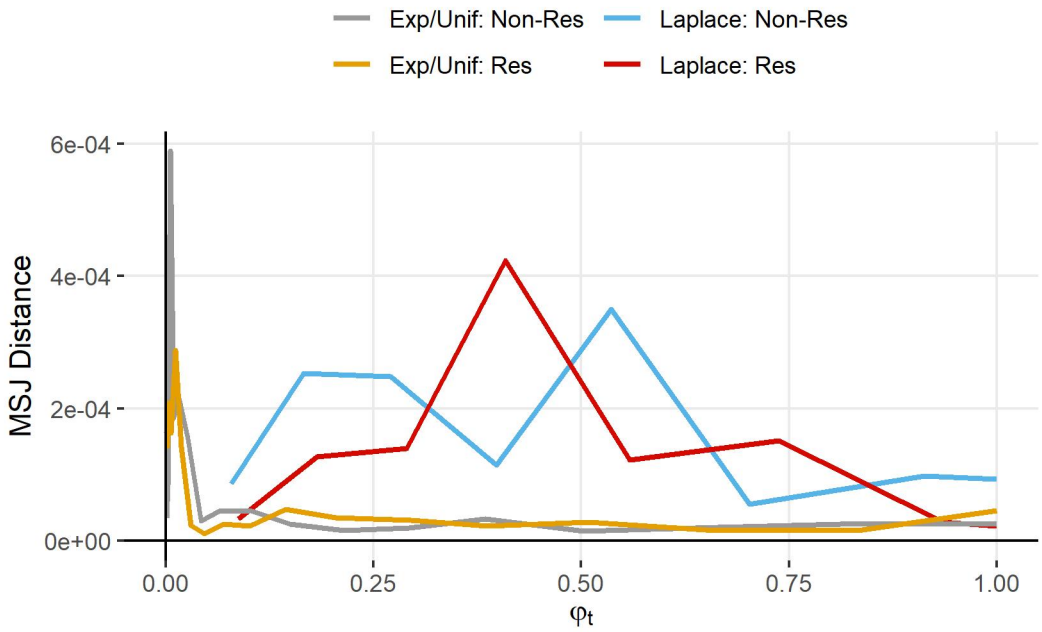


(d) Acceptance probabilities for the Population Size parameter under residual tuning scheme.

Figure 4.8: Acceptance probabilities for the height to the first coalescent event and population size parameter. These represent 10 runs when transitioning from a 2 to 3 sequence genealogy tree.

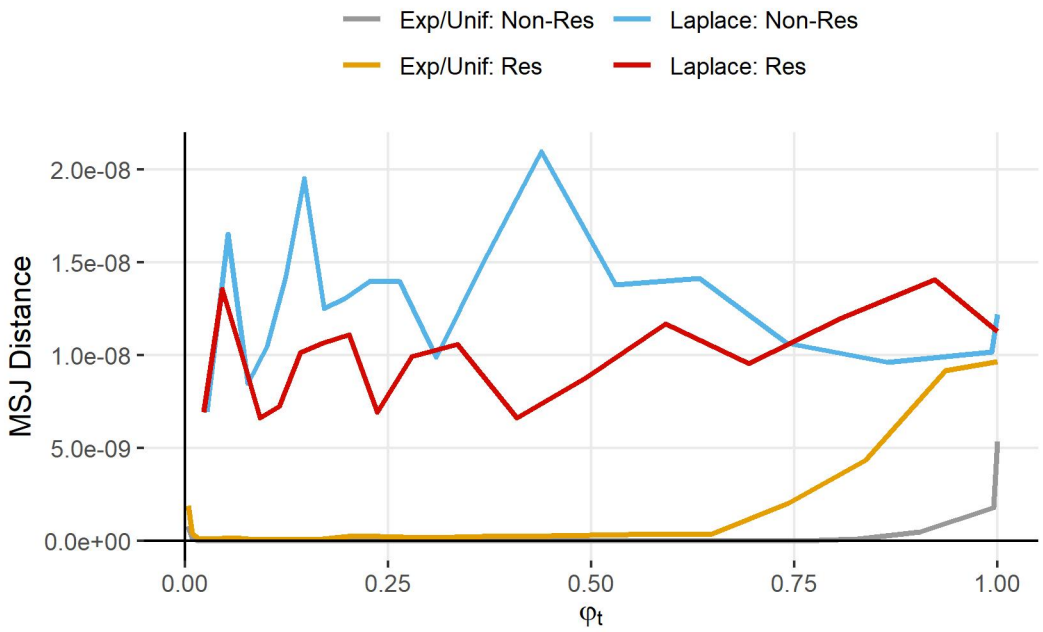


(a) MSJ for the height to the first coalescent event.

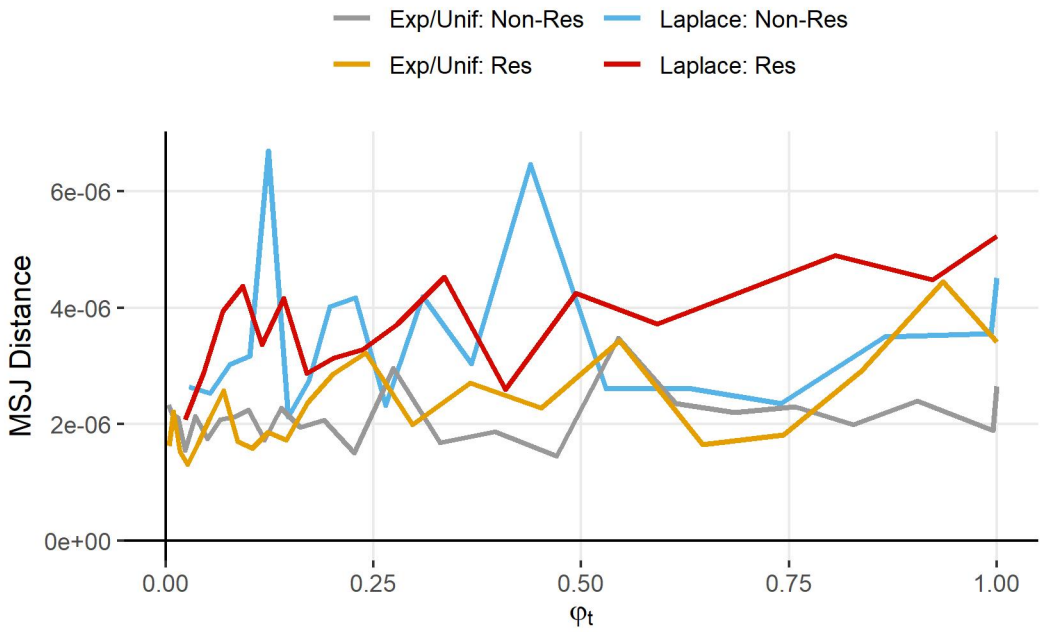


(b) MSJ for the population size parameter.

Figure 4.9: Expected mean square jump distance when transitioning from a 2 to 3 sequence genealogy tree. Analysed under both the exponential/uniform and Laplace approximation proposal.

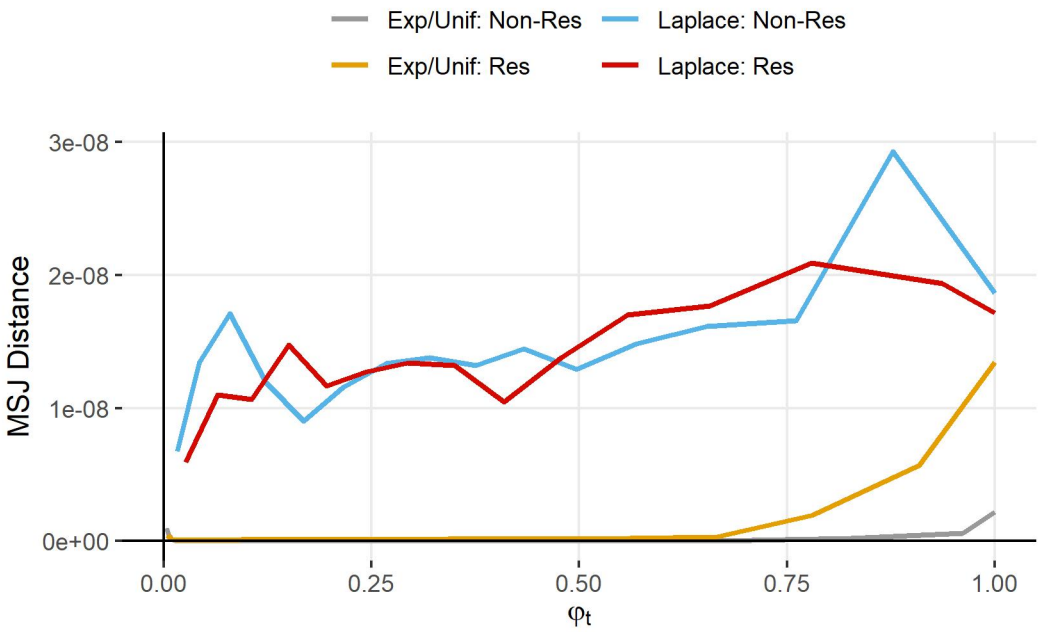


(a) MSJ for the height to the first coalescent event

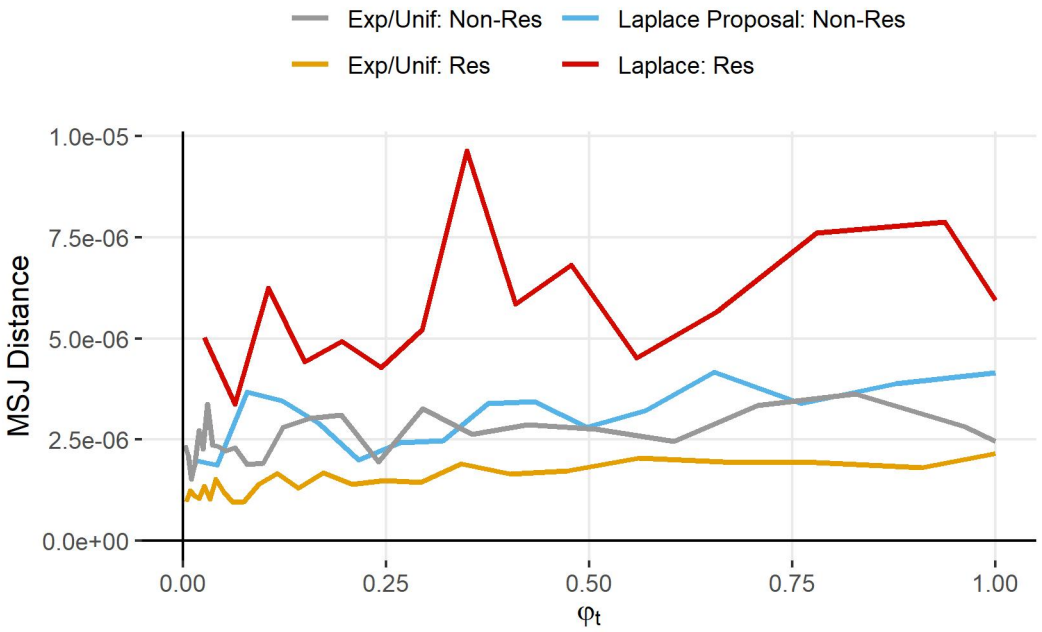


(b) MSJ for the population size parameter

Figure 4.10: Expected mean square jump distance when transitioning from a 10 to 11 sequence genealogy tree. Analysed under both the exponential/uniform and Laplace approximation proposal.



(a) MSJ for the height to the first coalescent event.



(b) MSJ for the population size parameter.

Figure 4.11: Expected mean square jump distance when transitioning from a 10 to 11 sequence genealogy tree. Analysed under both the exponential/uniform and Laplace approximation proposal.

accepted moves by approximately 5-15 times in comparison to one moves, and this was not changed throughout different transitions. There was no notable improvement in the estimates in regards to the posterior distributions or marginal likelihood. However there is still substantial issues when taking into account the computational cost of each move, as discussed in section 4.4.2. While giving an exact number of the cost is tricky as the cost will vary depending on SNP differences based on the ordering of the grafted sequences, performing at least 10 W&B allowed for approximately 5-30% probability for the particles to at least have at least one successful SPR move while still being faster than performing the W&B move. Therefore for our presented results we considered using 10 SPR moves for our presented results.

From figures 4.12 to 4.15 we present the posterior consensus trees under the exponential/uniform and Laplace approximation grafting proposals, with and without topology moves. Otherwise figure 4.16 shows the consensus plot from the MCMC output.

In particular we focus on the two subtrees, which have root nodes being the daughter nodes of the root node for the complete tree, and the subset of tip nodes that are contained within. These are

$$\text{Sequence Set 1} = \{250, 8, 239, 97, 1, 25, 88, 105, 5, 20, 6, 10, 22\} \quad (4.51)$$

$$\text{Sequence Set 2} = \{93, 59, 151, 133, 123, 39, 36, 34, 398, 45\}. \quad (4.52)$$

When we compare this to the maximum likelihood plot generated from Everitt *et al.* (2014), the topologies do have some differences but the two subtrees that descend from the root node contain sequence set 1 and 2 respectively.

What can be seen in figures 4.12 and 4.14 is that under strong topology mixing, regardless of how each branch was grafted onto the tree, the same or similar topology was generated. While there existed some variation when multiple runs of the algorithm were made, at worst they only differed by one-two SPR arrangements in comparison

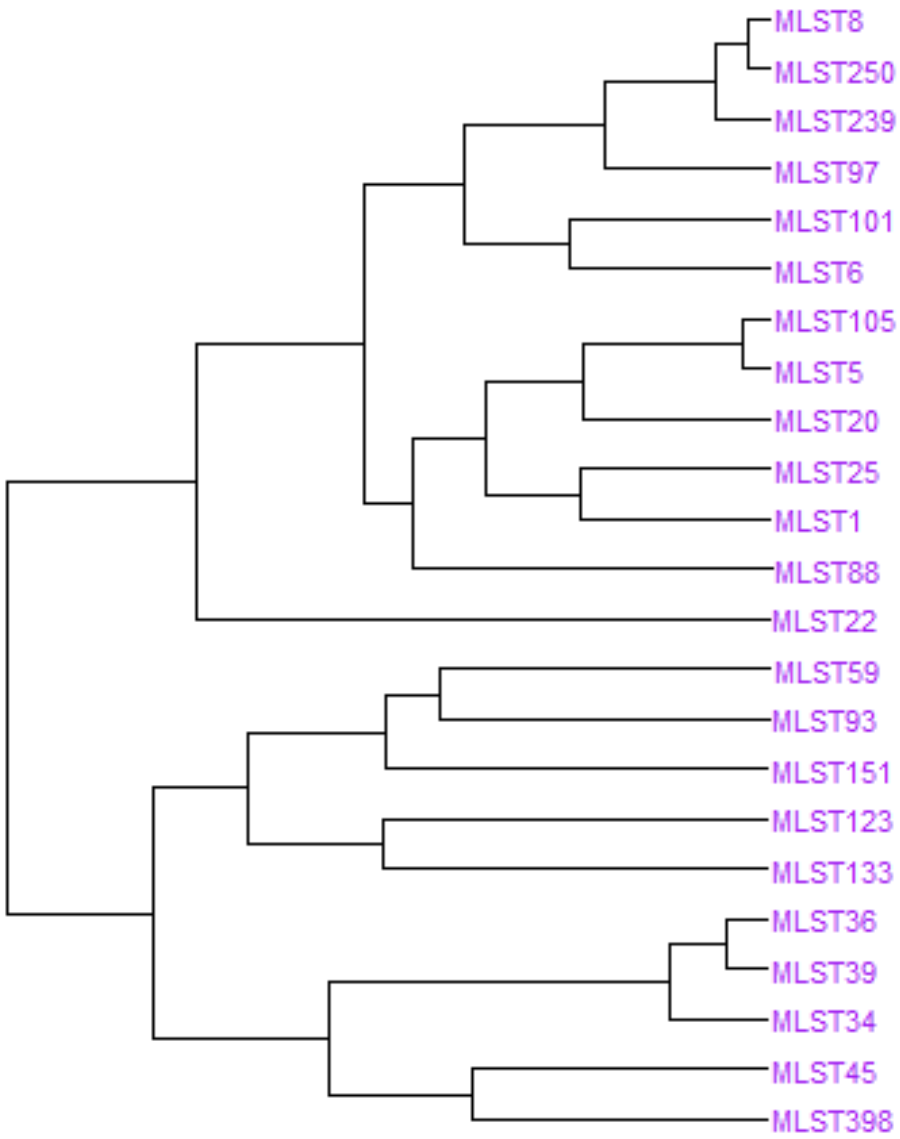


Figure 4.12: Consensus tree for the complete 23 sequence set using the exponential/uniform grafting proposal.

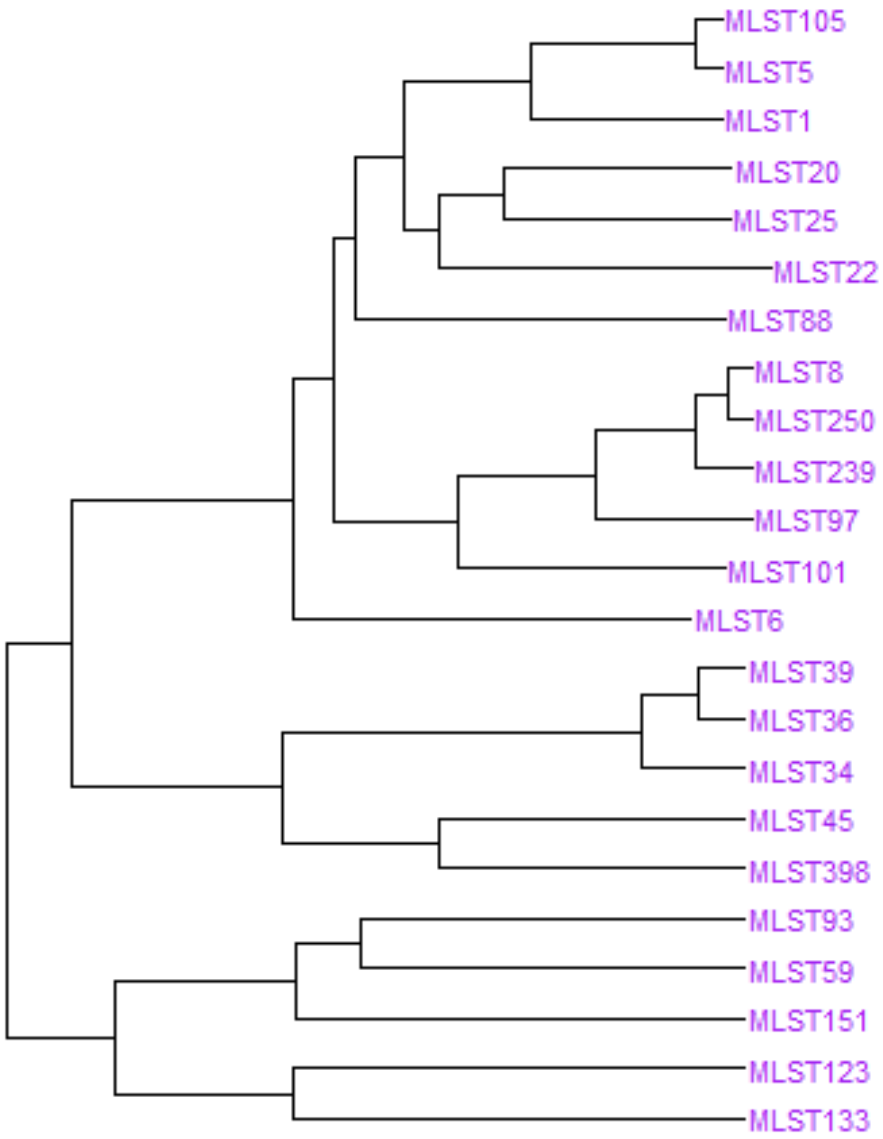


Figure 4.13: Consensus tree for the complete 23 sequence set using the exponential/uniform grafting proposal when no SPR moves were applied.

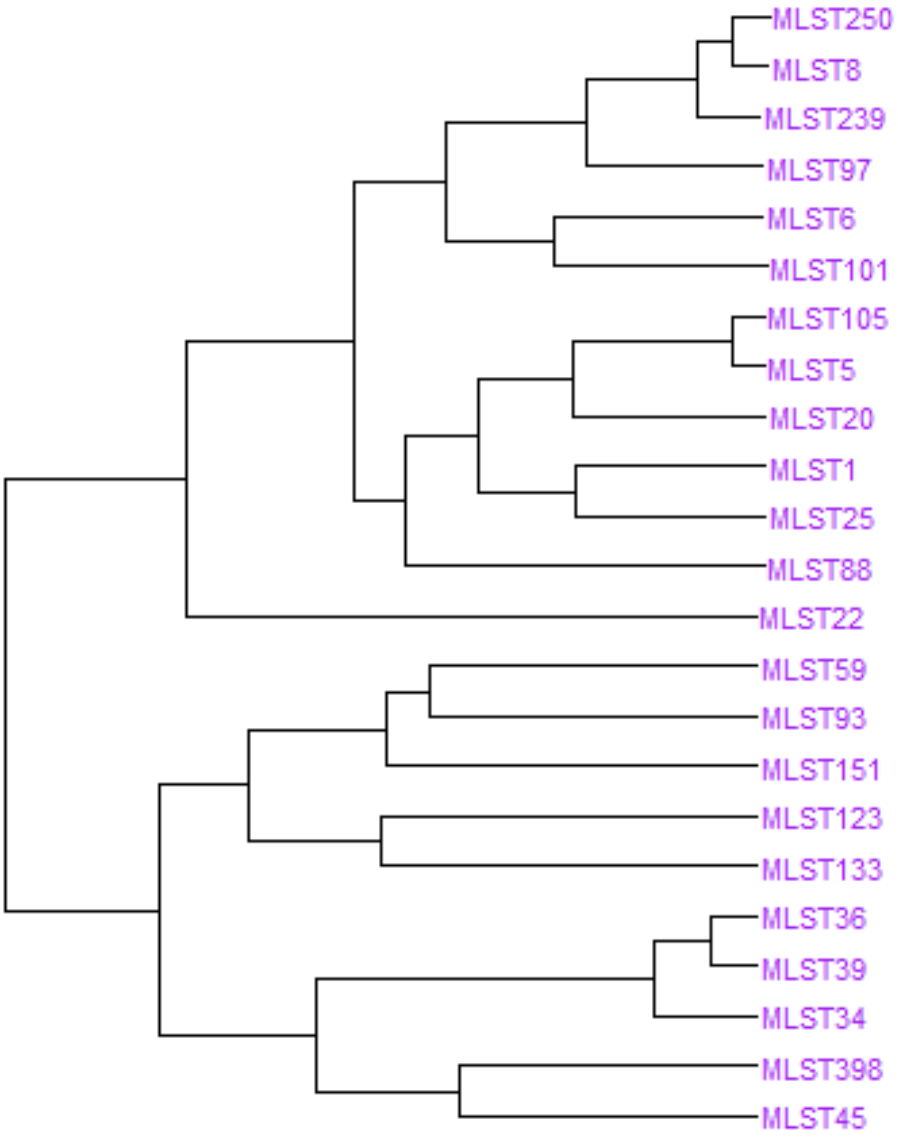


Figure 4.14: Consensus tree for the complete 23 sequence set using the Laplace Approximation grafting proposal.

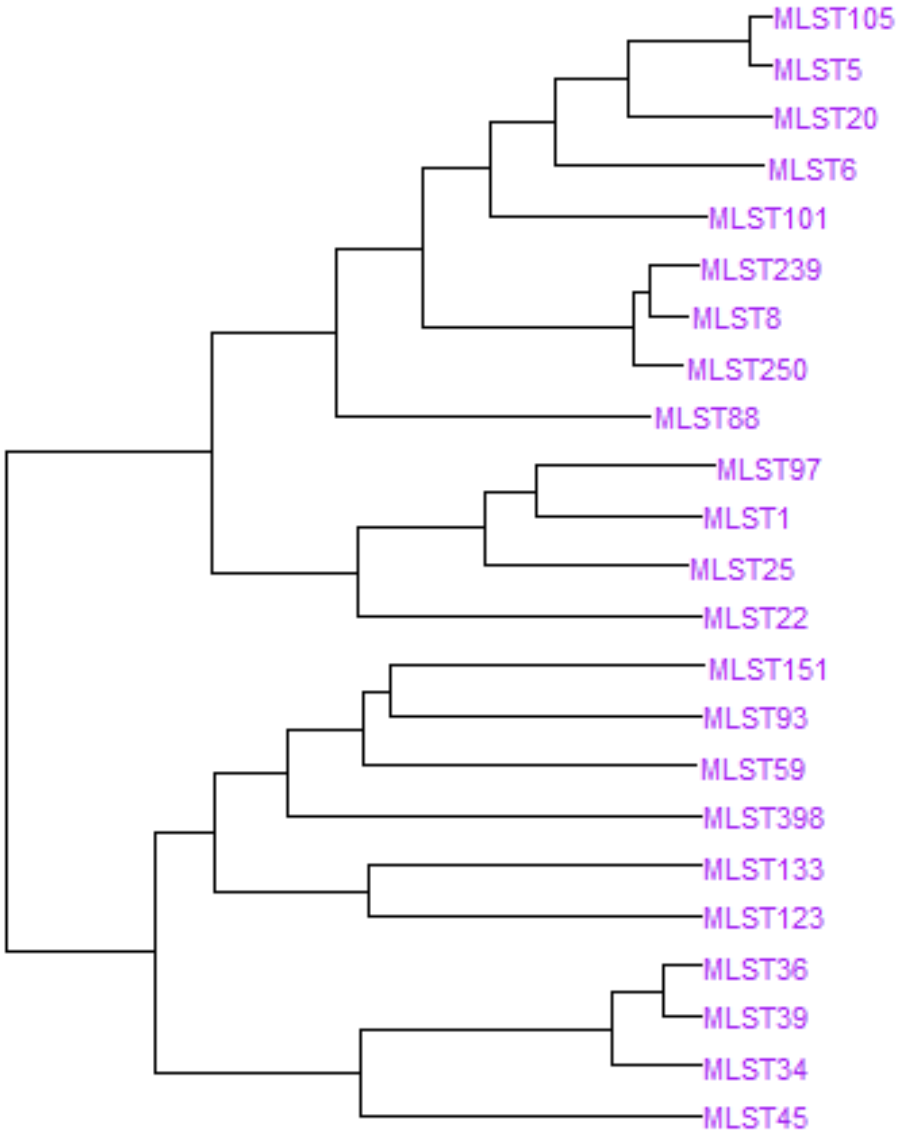


Figure 4.15: Consensus tree for the complete 23 sequence set using the Laplace Approximation grafting proposal when no SPR moves were applied.

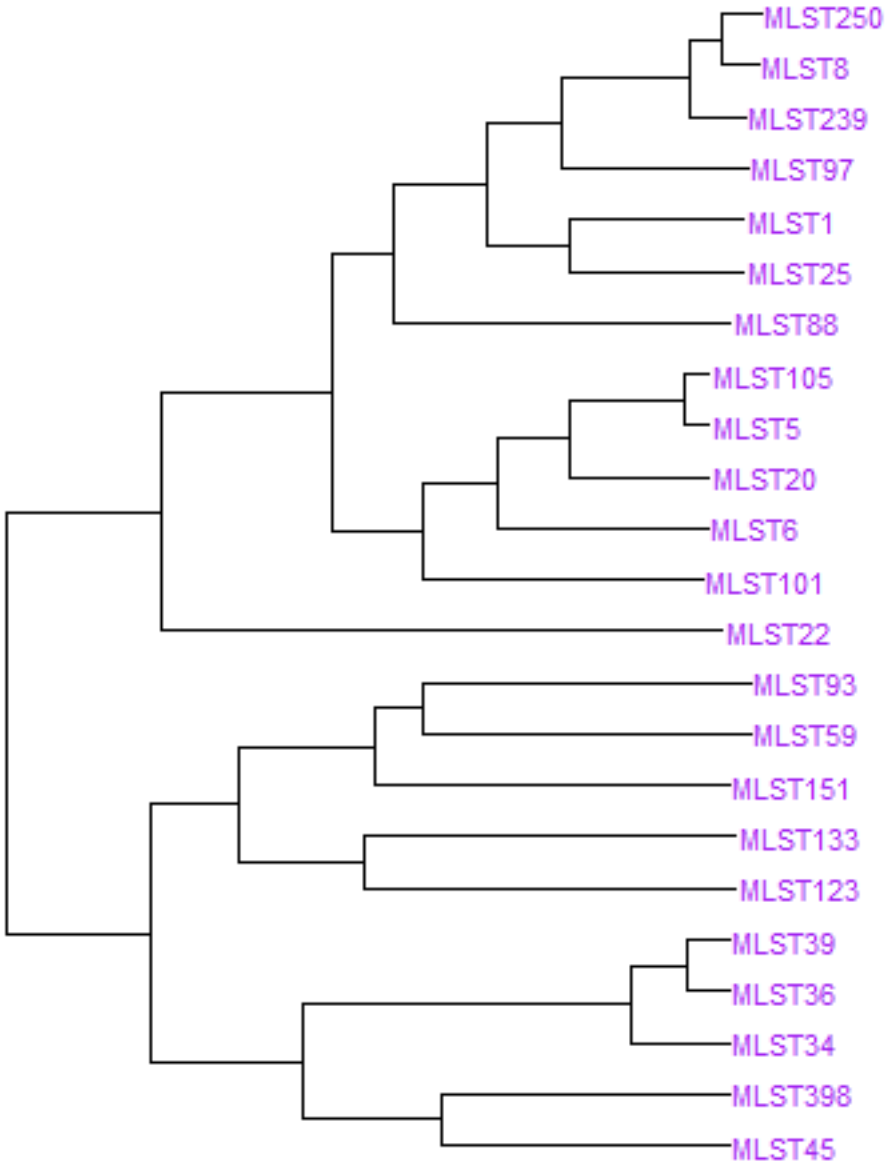


Figure 4.16: Consensus tree for the complete 23 sequence set under generated from MCMC burn-in.

to what is shown in figures 4.12 and 4.14.

The two tSMC adaptions in figures 4.12 and 4.14 had topologies that had similarities to figure 4.16. In the MCMC consensus tree, the subtree representing sequence set 2 in (4.52) had a topology that was identical to the very same subtrees in the tSMC adaptions that represent sequence set 2. Although the other subtree contained the same set of sequences, there were a few differences in the orderings of when each sequence coalesced but still had many similarities such as MLST22 being the last to coalesce with all the other sequences in the set.

When no SPR moves were performed there were multiple deviations from figures 4.12, 4.14 and 4.16. The consensus tree when no SPR moves were made while grafting sequences with the exponential/uniform grafting proposal, was very inaccurate in comparison to other trees. Most notable is that the two subtrees that descend from the root node contain sequences that do not match the sets shown in (4.51) and (4.52). Using no SPR moves while appying the Laplace approximation also had a few deviations, for example MLST45 and MLST398 should coalesce with each other without initially coalescing with other nodes beforehand (however this was not present), but it was not as bad of an estimate in comparison to using the exponential/uniform grafting proposals. From the results, it shows that SPR moves are still required and, unless better transformation proposals are made, relying on the transformation and a large particle size alone is not as efficient in exploring the parameter space of the genealogy.

When analysing the number of intermediate distributions needed to construct the genealogy tree in figure 4.17, when considering particle degeneracy under the current CESS threshold explained in section 4.5, the total number required was shown to be twice as large when using the exponential/uniform grafting proposal in comparison to the Laplace approximation grafting proposal. Although this shows that the exponential/uniform proposal was not a good proposal, which was expected because it was less directed proposal compared to the Laplace approximation proposal, the use

of intermediate distributions combined with MCMC kernel moves in the tSMC algorithm will assist with convergence to the posterior and can give a result that matches with better transformation proposals (as seen in figures 4.12 and 4.14).

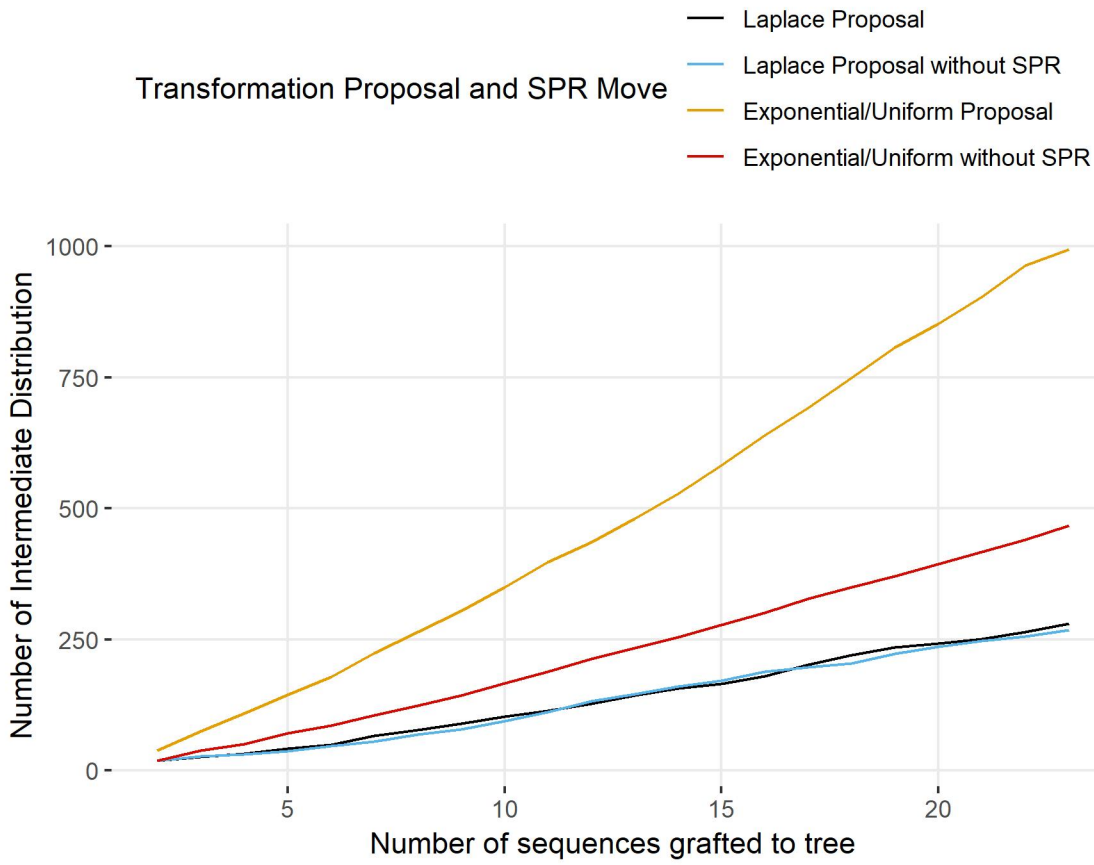


Figure 4.17: The cumulative number of intermediate distributions required to construct the complete genealogy tree when CESS $= 0.95N$.

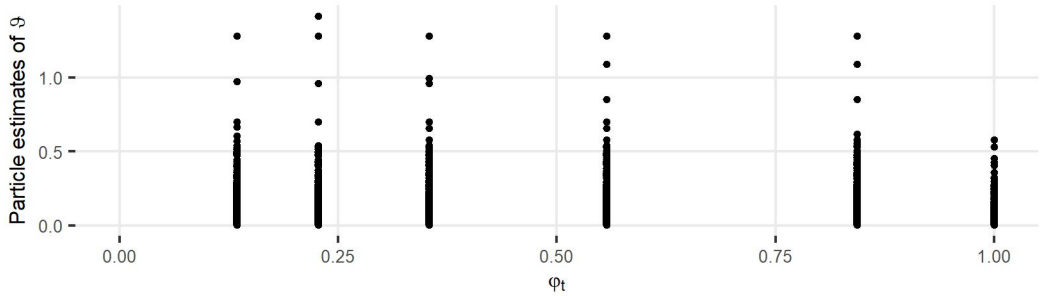
Although when SPR moves are not made when the exponential/uniform grafting proposal is used to transition to the next model the number of intermediate distributions is smaller by a large margin. However the number of intermediate distributions alone does not dictate how appropriate each proposal is. For example if a move only grafts the new branch in fewer appropriate areas this could lead to less diversity from a resampling step and then the resulting decrease of the variance the particle set could lead to large and fewer discrepancies between the intermediate distributions. As dis-

cussed previously, applying no SPR moves gives poorer estimates of the expected topology.

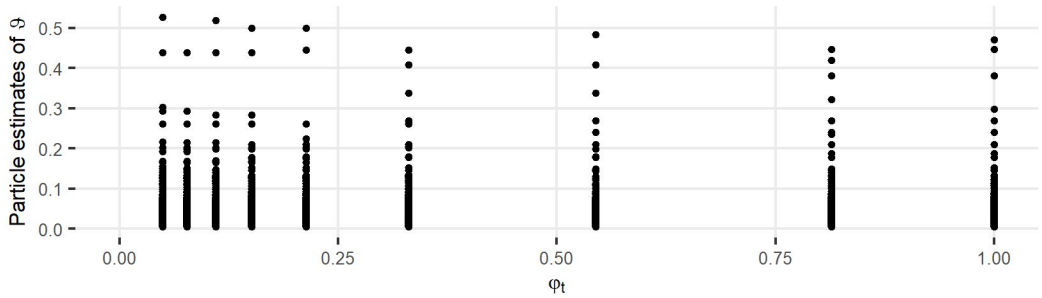
We also analysed the particle plots for the population size parameter and how they have changed over time in figure 4.18. From these figure what can be seen is that the posterior of the population size parameter is initially long-tailed when only a few sequences are present. However the posterior appears to continue contracting but at a decreasing rate with every sequence grafted onto the tree. These results are repeated across multiple runs of the data. From these results we are confident that, at least from this data, that we do not require a transformation on ϑ.

Finally we discuss the effects of the ML depending on patterned or random orderings. From table 4.1, in the scenario where no W&B moves were initiated there was an underestimation of the marginal likelihood under both the Laplace and exponential/uniform transformation moves. Overall the exponential/uniform graft move is a far poorer grafting proposal from the ML results and a higher number of intermediate distributions needed to be applied to allow for convergence. Therefore when considering the ML tests for the fixed orderings, as described in section 4.5, we choose not to analyse the results when an exponential/uniform grafting proposal is applied.

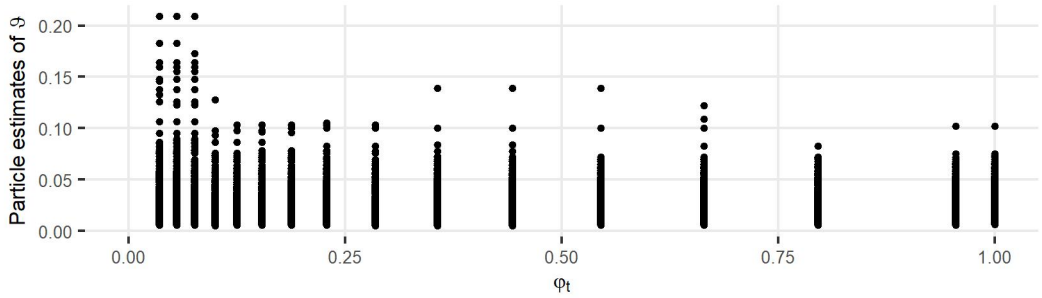
What we have discovered is that depending on the ordering, in table 4.2, different marginal likelihood values are generated. Here we believe there existed some underestimation of when grafting sequences with the largest average SNP differences of the existing grafted sequences to the smallest average differences. It should be noted that this was an issue when using the stepwise addition move, see section 4.3, in which the orderings of the sequences have an effect on the final tree. While we believe tSMC can converge to the correct answer eventually with well mixed topologies and gradual inclusion of sequences, it does not appear to fix the problems related to the marginal likelihood.



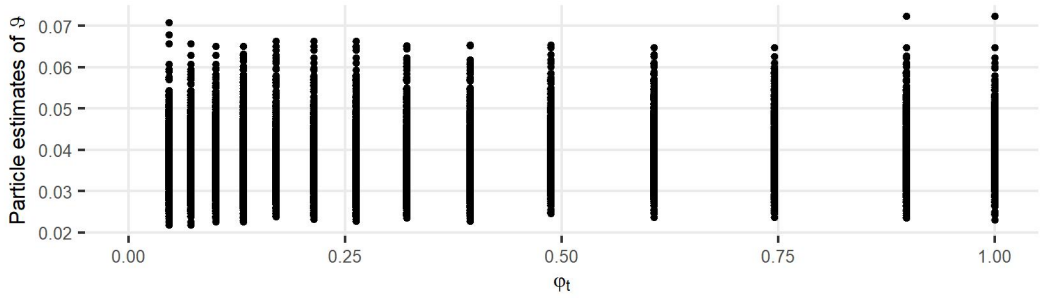
(a) Particle plots of ϑ when transitioning from a 2 to 3 sequence genealogy tree.



(b) Particle plots of ϑ when transitioning from a 4 to 5 sequence genealogy tree.



(c) Particleplots of ϑ when transitioning from a 6 to 7 sequence genealogy tree.



(d) Particle plots of ϑ when transitioning from a 22 to 23 sequence genealogy tree.

Figure 4.18: Particle plots of the population size parameter under multiple transitions.

	Standard	Standard + No SPR
Exponential/Uniform	−6297.278	−6308.481
Laplace Approximation	−6259.373	−6287.086

Table 4.1: First set of log marginal likelihood estimates for the complete genealogy tree. "Standard" scheme refers to running the tSMC algorithm with a particle size of 1000, with the number and type of MCMC kernels applied described in section 4.5.

	Ordering 1	Ordering 2
Laplace approximation	−6258.896 (5.133)	−6262.448 (3.937)

Table 4.2: Second set of log marginal likelihood estimates (+ standard error) for the completed genealogy tree. These were made via 250 particles and 10 SPR Moves moves. "Ordering 1" refers to grafting sequences based on the average smallest SNP differences from the sequences of the current tree, while "Ordering 2" regards grafting sequences based on the largest SNP differences from the sequences of the existing tree.

4.7 Conclusions

This chapter has shown the basic groundwork to estimate a high-dimensional posterior distribution of a phylogenetic tree by sequentially grafting sequences over time, while taking into account the potential changes to the posterior regarding how the sequences are added.

The Laplace approximation grafting proposal proved to be the superior way to graft a new sequence as it was more likely to give better initial approximations regarding how to place each new node in comparison to the other proposal which required a larger number of intermediate distributions under adaptive annealing. However the computational cost for the weight and MCMC updates under this move depends on the number of existing tip nodes within the tree. For example calculating (4.47) is more costly if say a new sequence was to be placed above the root node of a tree which currently has n sequences in comparison to also being above the root but only with $n-1$ sequences. Therefore, when the data is available all at once, it is more desirable

to graft sequences in the ordering from the average longest distance to the shortest distance of the existing sequences, providing that underestimation of the marginal likelihood is accounted for.

Fourment *et al.* (2018) claim that their algorithm, as explained in section 4.3, gave satisfactory convergence to each tree using only transformation moves and only one importance sampling step (along with resampling). However this contradicts our results in section 4.6 where not applying topology MCMC moves gave a more inaccurate estimate of the true genealogy tree and some number of intermediate distributions was still needed to prevent particle degeneracy as dictated by the conditional ESS.

It is difficult to evaluate if our transformation proposals can cover all areas of high posterior density to any transitioned model, and as we discovered with chapter 3 this is an important component for an accurate marginal likelihood. This is despite how tSMC can compensate for this with intermediate distributions and appropriate MCMC kernels. However given that simply trying to construct an accurate interpretation of the ancestral relationship is already a challenge, model comparison is not a major objective in either phylogenetic or population genetics studies at this point in time.

From these tests we do not believe that the population size parameter requires a transformation to shift the posterior space, especially if we do not expect large differences in genome sequences within a population. However it is uncertain if this will remain true for other population genetic based parameters such as the exponential growth rate of a population (Yang, 2014), although given that it is also a static parameter we believe that with more data then the posterior would contract in a similar fashion as well.

We believe a more appropriate way to implement one of the adaptive tuning schemes based on the residuals was to simultaneously give some proposal for the population size ϑ along with a change in the height node. Alternatively a general scaling options on all branching that follows from a change of the population size

parameter could of been more appropriate. This would have to considered as part of future research.

The tSMC adaption could be applied to tree construction under non-coalescent conditions, however when proposing new methods to graft a new sequence we would also need to consider the independent branch length from the new sequence to its ancestor node in comparison to only considering the position of the most recent ancestor node of the new sequence. The Laplace approximation proposal worked on the basis that all individuals are sampled from some present population and the distance from each tip node to the root of the genealogy is the same. This is not an assumption made under non-coalescent model conditions, where some individuals are part of some extinct taxa. Proposals by Fourment *et al.* (2018) should be considered, or an adjustment to the Laplace approximation would be required instead.

What would be of interest, given a particle set of trees with the same sample sequences, is if different substitution models provided a better fit for the data then the cost to answer this question could decrease by considering some appropriate transformation of variables via tSMC. For example the Kimura (K80) model (Kimura, 1980) assumes that substitutions from Thymine to Cytosine or from Guanineas to Adenine, and vice versa with Cytosine to Thymine etc, all termed as *transitions* (not to be confused with how we described transitions in tSMC) occur at a different rates in comparison to all other substitutions which are termed as *transversions*. Therefore we are interested in inferring a new parameter called the *transition/transversion* rate ratio. However more than a parameter generation of substitution model parameters would be required as, depending on the data, the topologies under two different substitution models can be very different should the said ratio greatly diverge from 1. Thus using one transformed model as a type of importance sampler is not recommended. Fixing the problem would require improved topology moves that can better reorganise subtrees, although doing so would make the MCMC a far more viable option and weaken the argument for applying tSMC. Furthermore since our Laplace

approximation was based on the distances of a JC69 model, then we would need to devise alternative methods to graft a node onto a tree if a different substitution model was to be used.

Another challenge is in the application of differing substitution/mutation rates within subsets of the loci where a single mutation rate for all loci, like that was used in this investigation, is not biologically accurate. One possible solution involves grouping sites into "populations" which differ by their substitution rates based on a Dirichlet process prior, see chapter 5 for a full explanation of how the Dirichlet process should be interpreted. Each of the "population" groups have their own topology and branch lengths as well as substitution rates (Huelsenbeck and Andolfatto, 2007; Wu *et al.*, 2012). However the previously discussed adaptions of tSMC are not very efficient at moving such allocation variables around the model space unless all parameters that are not the allocation variables themselves are integrated out, which is a strategy we apply in chapter 5. Nevertheless to perform some integration for all possible labeled history trees is unfeasible, and even if it was possible the number of sites/parameters could range in the hundreds of thousands. An alternative parameterisation that also assigns sites into populations and infers the differences in mutation rates is via point change models (Persing *et al.*, 2015; Suchard *et al.*, 2003). What point change models consider is inferring a number and position of breakpoints which separate the individual groups within a sequence, which is more manageable since the number of discrete parameters will be far smaller. However it is unclear what type of across model move should be used for the breakpoints, whether it is randomly proposing a new breakpoint or splitting one into two like what was proposed in chapter 3, and how the introduction of one breakpoint will affect all other breakpoints and the genealogies between breakpoints in the model.

In summary, regardless of what model we choose to infer if a new sequence changes the genealogy greatly, more notably the topology, then constructing an accurate importance proposal via a transformation will be harder to devise. The algorithm

will be more dependent on MCMC kernels, combined with setting the correct rate of particle degeneracy, to explore the model space. While this strategy might allow for convergence to regions of high posterior probability density it is still very likely that the marginal likelihood will be underestimated given the evidence that we have seen so far in chapter 3.

Chapter 5

Applications in Population Structure under Non-Parametric Model Assumptions

In chapter 3 we discussed inferring a posterior distribution of a mixture distribution where the allocation variables, which are used to represent the assignment of observations to population groups which differ through parameter measured characteristics, were integrated out from the model. This decision was made to simplify the problem, and because we require an efficient method to successfully rearrange the allocation variables when creating an additional component. Here we present a solution to infer the allocation variables when using a set of nested models as proposals, which differ by the number of population clusters in which said variables may be assigned to. However for now it requires a conjugate relationship for all model parameters with the exception of the allocation variables themselves for our transformation proposal to be used. In comparison to the previous applications, we are not transforming the parameters of a previously inferred model. Instead this is an example of using tSMC to gradually explore the parameter space by incrementally increasing the number of states of the allocation variables (or increasing the number of populations) at each transition. On a minor note, Gibbs samplers as a MCMC kernel within tSMC can be

used as we are inferring parameters in a solely discrete parameter space.

In this final chapter we now attempt to perform tSMC under the Structure model (which is used to cluster genome sequences), see section 5.1.1. The Structure application is an example where the within-population parameters can be collapsed out of the model using conjugacy. As this application uses some terms relating to genetics, we recommend referring back to chapter 4 for a brief explanation of certain genetics based concepts.

Section 5.1 describes the basic concepts of the Structure model. We also state the two types of Bayes' prior distributions, including their differences, for the parameter set representing the assignment of observations to populations.

In section 5.2 we give a brief literature review regarding the standard approaches to increase or decrease the number of populations in the posterior distribution, a few note-worthy SMC based adaptions and the contribution that tSMC can give.

We present the posterior distribution of the Structure model in section 5.3. Across model moves and within model MCMC moves are also explained.

Section 5.4 describes the primary objectives for each test and the diagnostics to be applied, with the results presented in section 5.5.

Finally we give a discussion on the limitations to our proposed approach and what to prioritise for future work if we were to continue using the tSMC approach in section 5.6.

5.1 Inference with Structure and Allocation Variables

5.1.1 The Structure Model

In the Structure model we consider aligned sequence data of n individuals, $y = \{y_{1\cdot}, ..., y_{n\cdot}\} \subset Y$, which can be either haploid or diploid. In this thesis we use data of the form $y_{il}^{(c)}$ which represents the ith individual of the cth chromosome where

$c \in \{1, 2\}$. Although we will be analysing diploid data, for simplicity when using y_{il} it equates to $y_{il}^{(1:2)}$ and so we account up to two alleles at each of the l loci for each ith observation.

In comparison to chapter 4 where we only considered the allele type for the lth locus, in this application it is possible to instead consider the alleles of the lth joint loci on a sequence with an example being microsatellite data, which consists of repeated tracts of DNA, as defining the data this way still captures the differences between sequences. Overall we generalise the data to consist of sets of loci, which may consist of only one site or multiple loci. We are interested in inferring the posterior probabilities regarding the allocation of each of the individual sequences, represented by their respective allocation variables of $z = \{z_1, ..., z_n\}$, to one of the k populations. The populations, with $a_p \in \{a_1, ..., a_k\}$, are representative of various genetic properties of the assigned sample individuals. We assume that we do not know the true characteristics of each population, only some prior assumptions.

The structure model considers inferring a population allele matrix $\bar{P}$, being a $k \times l \times \varsigma_l$ object representing the allele frequencies, and we consider the following notation:

- ς_l is the set of unique alleles at the lth loci.
- $\tilde{\varsigma}_l$ is the number of unique alleles at the lth loci.
- ς_{lj} is the jth allele of the set of unique alleles present in the lth loci.
- $\tilde{\varsigma}_{lj}$ is the total number of the the jth allele type in ς_l that exists across all k populations.
- $\tilde{\varsigma}_{l \cdot a_p}$ is the total number of the read alleles at the lth loci for sequences that are in population a_p only.
- $\tilde{\varsigma}_{l j a_p}$ is similarly defined like $\tilde{\varsigma}_{l \cdot a_p}$, except we only record the counts of the jth ordered allele of the lth loci only that are in population a_p only.

- n_{a_p} is the number of individuals, or number of z_i, assigned to population a_p.
- Therefore we define $\bar{P}_{lja_p}$ as the estimated frequency of the jth ordered allele at the lth locus for any individual who belongs to population a_p.
- Otherwise $\bar{P}_{l\cdot a_p}$ is the vector of allele frequencies within the population a_k at the lth loci. Furthermore $\bar{P}_{l\cdot a_p} \in \mathcal{F}$ is integrable over a continuous space.

Furthermore we use an additional '(i)' or '$(-i)$' notation to define whether individual i is to be included in some object. For example $\tilde{\varsigma}^{(-i)}_{lja_p}$ is the number of counts for the ordered jth alleles of the lth loci for all individuals belonging to population a_p with the exception of individual i. Otherwise if we only consider the counts of the ith diploid organism we would use $\tilde{\varsigma}^{(i)}_{lja_p} \in \{0, 1, 2\}$ which depends if the lth loci of the observation either has missing allele data ($\tilde{\varsigma}^{(i)}_{lja_p} = 0$), allele data at one chromosome only (1) or no missing data respectively ($\tilde{\varsigma}^{(i)}_{lja_p} = 2$).

Finally in section 5.3 we use an additional term of m_k, when defining estimated parameters, or otherwise allele counts/frequencies for individual populations, of each model m_k. For example, $z_{m_k i}$ is the ith allocation variable that corresponds to model m_k. In comparison to previous chapters however we will not be defining the specific intermediate distribution, when transitioning between adjacent models, that any estimated parameters correspond too. This decision was made to simplify the notation, and we won't be using it when explaining how our transformation on each model works.

When considering the Structure model we assume that each locus is unlinked with the rest of the sequence and linkage equilibrium is present, which means that any two different allele frequencies $\bar{P}_{l'j'a_p}$ and $\bar{P}_{lja_p}$ do not have a higher or lower frequency of being inherited together by offspring from an individual belonging to population a_p and thus all alleles are independent across sites. Furthermore we assume Hardy-Weinberg conditions in which the population allele frequencies/characteristics will remain unchanged by outside influences, these include immigration/emigration

or mutations that change allele frequency, across multiple generations of offspring. These assumptions are needed such that an allele at each loci for a sequence from population a_k is an independent draw from the allele probabilities of $\bar{P}_{lja_k}$. Therefore we define the distributions for each sequence's chromosome and the population allele frequencies as

$$y_{il}^{(c)} \sim \text{Discrete}(\bar{P}_{l\cdot z_i}) \tag{5.1}$$

$$\bar{P}_{l\cdot a_p} \sim \text{Dirichlet}(\beta_{l1}, ..., \beta_{l\tilde{\varsigma}_l}), \tag{5.2}$$

for $(\beta_{l1}, ..., \beta_{l\tilde{\varsigma}_l}) \subset \mathbb{R}^+$. The basic Structure model, and the class of adapted algorithms that follow from its origins by Pritchard *et al.* (2000), provides a Bayesian-approach which not only attempts to estimate the distribution of the allocation variables but simultaneously estimates what the allele frequencies are for each population. This is usually accomplished through Gibbs sampling as the conditionals of z_i and $\bar{P}_{l\cdot a_p}$ can be defined from appropriate priors (see sections 5.1.2 and 5.3).

Some care is needed when interpreting the biological implications from produced results. By design the Structure algorithm will favor the smallest population size possible that explains the vast majority of genetic variation, and although this should be an ideal property of Bayes models it may not reflect real world populations that the data represents. This would occur if certain genomes from a divergent population do not have a significant sample size and/or a high genetic divergence from the other population groups, and instead the individuals would be mixed with the other groups (Lawson *et al.*, 2018).

5.1.2 Non-Parametric and Parametric Priors on the Allocation Variable

In this subsection we explain two priors for the allocation variables under Structure, one being the Dirichlet process (DP) prior and the other being a finite mixture

prior.

Ferguson (1973) introduced the Dirichlet process, a stochastic discrete time process, and Antoniak (1974) proposed its application in mixture models. This prior assumes that each observation, is generated by first sampling a random distribution G from the Dirichlet process $G \sim \text{DP}(\alpha G_0)$ where α is the concentration parameter and G_0 is a baseline distribution. Then each of the joint parameters θ_i are drawn from G, and finally an observation y_i is sampled from a family of mixture distributions $F(\theta_i)$ corresponding to θ_i. Overall we define a hierarchical model of

$$\begin{aligned} y_i &\sim F(\theta_i) \\ \theta_i|G &\sim G \\ G &\sim \text{DP}(\alpha G_0). \end{aligned} \tag{5.3}$$

For example a Dirichlet process mixture model for the Structure could be defined by

$$\begin{aligned} y_{il}^{(c)} &\sim \text{Discrete}(\bar{P}_{l\cdot z_i}) \\ \bar{P}_{l\cdot z_i} &\sim G \\ G &\sim \text{DP}(\alpha G_0) \\ G_0(\bar{P}_{l\cdot a_p}) &\sim \text{Dirichlet}(\beta_{l1}, ..., \beta_{l\tilde{\varsigma}_l}). \end{aligned} \tag{5.4}$$

To make the link between the DP and mixture models, like in chapter 3, we consider taking the limit as the number of possible population groups goes to infinity for a finite mixture model. Following the example from Huelsenbeck and Suchard (2007);

Lawson *et al.* (2012); Neal (2000) we consider a finite mixture model

$$\begin{aligned} y_i &\sim f(\cdot|z_i, \theta_{z_i}) \\ z_i|\omega &\sim \text{Discrete}(\omega) \\ \theta_{a_p} &\sim G_0 \\ \omega &\sim \text{Dirichlet}(\alpha_1 = \alpha/k, ..., \alpha_k = \alpha/k). \end{aligned} \tag{5.5}$$

Naturally the within within cluster θ_{a_p} correspond to cluster a_p. We set the mixing proportions ω to be distributed by a symmetric Dirichlet prior with concentration parameter α/k, such that the parameters approach zero as k goes to infinity. Depending on the application, additional hierarchical assumptions may be applied which puts further hyper-priors on the concentration parameters of the mixing proportions prior, for example in finite mixture of Gaussian models the mixing proportions could be based on the ordered labeled weights $\{\omega_1, ..., \omega_k\}$ where each ω_j distributed by a beta distribution multiplied by the product of $\prod_{i=1}^{j-1}(1 - \omega_i)$ (Papastamoulis and Iliopoulos, 2013). What we briefly note is that we can integrate out the allocation variables in (5.5), and thus we would be inferring a posterior of $\pi(\omega, \theta|y)$, just like our tSMC adaption of univariate mixture models in chapter 3 (in which we have shown how these can be integrated out in the said chapter). This is termed as being a "without-completion" sampler, where we sum over all the population components for each observation.

However in this chapter we instead integrate over the mixing proportions $\omega \in \Theta$, in addition to integrating out the within component parameters θ. This is known as "collapsing the model". In this section we consider how we define a Dirichlet process by integrating out the mixing proportions and letting the number of population tend to infinity, and otherwise with regards on how to integrate out the population allele frequencies (the only component parameters in the basic Structure model) then refer to section 5.3.1.

Firstly the joint prior for the mixing proportions and the allocation variables is

given by

$$\begin{aligned} p(z|\omega)p(\omega) &= \left(\prod_{i=1}^{k}\omega_i^{n_{a_p}}\right)\left(\frac{\Gamma(\sum_{i=1}^{k}\alpha/k)}{\prod_{i=1}^{k}\Gamma(\alpha/k)}\prod_{i=1}^{k}\omega_i^{\alpha/k-1}\right) \\ &= \frac{\Gamma(\sum_{i=1}^{k}\alpha/k)}{\prod_{i=1}^{k}\Gamma(\alpha/k)}\prod_{i=1}^{k}\omega_i^{\alpha/k+n_{a_p}-1}. \end{aligned} \tag{5.6}$$

However as we can equate the integral of a Dirichlet distribution to

$$\frac{\Gamma(\sum_{i=1}^{k}(\alpha/k+n_{a_p}))}{\prod_{i=1}^{k}\Gamma(\alpha/k+n_{a_p})}\int_\Theta\prod_{i=1}^{k}\omega_i^{\alpha/k+n_{a_p}-1}d\omega = 1$$

$$\int_\Theta\prod_{i=1}^{k}\omega_i^{\alpha/k+n_{a_p}-1}d\omega = \frac{\prod_{i=1}^{k}\Gamma(\alpha/k+n_{a_p})}{\Gamma\left(\sum_{i=1}^{k}\left(\alpha/k+n_{a_p}\right)\right)}. \tag{5.7}$$

Then we can integrate out the mixing proportions by substitution of (5.7) into (5.6) giving

$$\begin{aligned} p(z) &= \int_\Theta p(z|\omega)p(\omega)d\omega \\ &= \frac{\Gamma(\sum_{i=1}^{k}\alpha/k)}{\prod_{i=1}^{k}\Gamma(\alpha/k)}\int_\Theta\prod_{i=1}^{k}\omega_i^{\alpha/k+n_{a_p}-1}d\omega \\ &= \frac{\Gamma(\sum_{i=1}^{k}\alpha/k)}{\prod_{i=1}^{k}\Gamma(\alpha/k)}\frac{\prod_{i=1}^{k}\Gamma(\alpha/k+n_{a_p})}{\Gamma\left(\sum_{i=1}^{k}\left(\alpha/k+n_{a_p}\right)\right)}. \end{aligned} \tag{5.8}$$

What we now consider is that each observation is being introduced one at a time,

and that the allocation for observation y_i is affected by the previous allocations of all other $y_{1:(i-1)}$ but its allocation is not affected by $y_{(i+1):n}$. Overall we consider each conditional distribution of $p(z_i|z_1, ..., z_{i-1})$. The conditional distribution that the ith ordered observation is assigned to population a_p, under a finite mixture model, is defined by

$$\begin{aligned}
p(z_i|z_1, ..., z_{i-1}) &= \frac{p(z_1, ..., z_i)}{p(z_1, ..., z_{i-1})} \\
&= \left(\frac{\Gamma(\sum_{i=1}^{k} \alpha/k)}{\prod_{i=1}^{k} \Gamma(\alpha/k)} \right) \frac{\Gamma(\alpha/k + (n_{a_p} + 1))\Gamma(\alpha/k + n_{a_1})...\Gamma(\alpha/k + n_{a_k})}{\Gamma\left(n_{a_1} + (n_{a_p} + 1) + ...n_{a_k} + \sum_{i=1}^{k} \alpha/k\right)} \\
&\quad \times \left(\left(\frac{\Gamma(\sum_{i=1}^{k} \alpha/k)}{\prod_{i=1}^{k} \Gamma(\alpha/k)} \right) \frac{\Gamma(\alpha/k + n_{a_p})\Gamma(\alpha/k + n_{a_1})...\Gamma(\alpha/k + n_{a_k})}{\Gamma\left(n_{a_1} + ...n_{a_p} + ...n_{a_k} + \sum_{i=1}^{k} \alpha/k\right)} \right)^{-1} \\
&= \frac{(\alpha/k + n_{a_p})\Gamma(\alpha/k + n_{a_p})\Gamma(\alpha/k + n_{a_1})...\Gamma(\alpha/k + n_{a_k})}{(i - 1 + \sum_{i=1}^{k} \alpha/k)\Gamma\left(i + \sum_{i=1}^{k} \alpha/k\right)} \\
&\quad \times \left(\frac{\Gamma(\alpha/k + n_{a_p})\Gamma(\alpha/k + n_{a_1})...\Gamma(\alpha/k + n_{a_k})}{\Gamma\left(i + \sum_{i=1}^{k} \alpha/k\right)} \right)^{-1} \\
&= \frac{(\alpha/k + n_{a_p})}{i - 1 + \alpha}. \qquad (5.9)
\end{aligned}$$

where this is derived by adding one more observation to n_{a_p} in $p(z_1, ..., z_i)$. When we let the number of populations go to infinity we obtain a Dirichlet process, where the conditional probability of assigning an observation to a specific group under the collapsed model converges to

$$p(z_i|z_{i-1}, ..., z_1) = \begin{cases} \frac{n_{a_p}}{(i - 1 + \alpha)} & \text{if placed in an existing group } a_p \\ \frac{\alpha}{(i - 1 + \alpha)} & \text{if placed in a new group} \end{cases}, \qquad (5.10)$$

based from (5.9) and in the case of (5.10) it is the current number of observations from $(y_1, ..., y_{i-1})$ that are allocated to population a_p. Therefore the limit of (5.5) as $k \to \infty$ is equivalent to the Dirichlet process mixture model (Escober and West, 1995; Gershman and Blei, 2012; Neal, 2000).

What we also note is that the joint prior is an exchangeable prior, meaning that the ordering of assigning the observations does not affect the density of the prior as, given in (5.11),

$$\begin{aligned} p(z) &= f(z_n|z_{n-1}, ..., z_1)....f(z_2|z_1)f(z_1) \\ &= (\alpha)^k \frac{\prod_{p=1}^{k} \Gamma(n_{a_p})}{\prod_{i=1}^{n}(\alpha + i - 1)}. \end{aligned} \tag{5.11}$$

Although we assume that there exists infinite components *a priori*, in posterior inference the parameters of a cluster only need to be updated if there is an observation associated with it (Gershman and Blei, 2012; Neal, 2000).

Apart from the Dirichlet process prior we give a brief mention of another type of prior that provides informative or uninformative assumptions regarding how the allocation parameters are concentrated in population groups. The variations of these type of priors, in which we follow Green (2001) in terming them finite mixture priors. This prior distribution firstly considers the maximum number of populations, K, drawn from some distribution often chosen to be uniform (although in some applications, such as Structure, it is chosen to be fixed based from background knowledge). Furthermore for any defined but empty populations, the priors for the parameters of said empty population are still included as part of the posterior distribution in comparison to Dirichlet process priors where empty populations are deleted (along with their parameters). The model shown in 5.5 had priors which are essentially finite mixture priors. Such priors are commonly used in several applications, and naturally this also includes the Structure model (see De Iorio *et al.* (2015); Jasra *et al.* (2008);

Pritchard *et al.* (2000)).

Both types of priors will favor certain posterior results. For example Green (2001) considered a Dirichlet multinomial allocation (DMA) prior, which sets the prior on the label allocations to be a multinomial distribution with Dirichlet distributed hyperpriors for the concentration parameters. They stated how a DMA prior favors more equal allocations by design, in comparison to using the DP prior.

We choose to focus on Dirichlet process prior as we intend for this part of our investigation to provide a groundwork on how these common applications can use tSMC to incrementally increase the number of population groups. Furthermore the type of model transformations and MH proposals that we propose make use of DP assumptions.

5.2 Previous Approaches to Inference of Structure and SMC with Dirichlet Processes

We focus on previous attempts to infer the posterior distribution from the Structure model, as well SMC based approaches to non-parametric models. As stated previously Structure can use a non-parameteric Dirichlet process prior on clusterings, in which the number of clusters may be considered as a model selection problem.

Naturally MCMC does allow for the inference of models with Dirichlet process priors. The simplest adaptions are when the joint prior distribution of the non-allocation variables is conjugate to the likelihood function, as these allow for simple Gibbs moves to be applied to propose new values for each allocation variable. This is a method that we compare with our tSMC adaption, and more details on how these Gibbs probabilities are defined is shown in section 5.3.4.

However single Gibbs updates that move allocation variables one at a time are notorious for getting stuck at local modes as they struggle to cross valleys of low probability density and into a different high probability mode. For example suppose

that the allocation variables are such that all observations are part of one population, but a more likely explanation is that there should exist two populations. However any one observation creating a new population is even more unlikely, and thus Gibbs moves alone are not likely to be effective in exploring the parameter space of the allocation variables (Dahl, 2003; Gershman and Blei, 2012; Neal, 2000).

Therefore there exist proposals that attempt to perform these large jumps by splitting or merging populations into multiple sets. A popular set of such algorithms involve anchoring two allocation variables to either create two new groups, reassign allocation variables between two groups or merge two population groups together (Bouchard-côté and Roth, 2017; Dahl, 2003; Jain and Neal, 2004). We give an in-depth explanation of a split-merge sampler known as the "Sequentially-Allocated Merge-Split" (SAMS) sampler (Dahl, 2003) in section 5.3 and how we use it in out tSMC adaption. Variational methods can also be used (see for example Blei and Jordan (2006)) where we have described the disadvantages of such an approach in chapter 3.

Bouchard-côté and Roth (2017) define their Particle Gibbs Split-Merge sampler, a Particle MCMC algorithm, in which each state considers a series of particles that performs a variation of the split-merge sampler similarly to Dahl (2003). The SMC component of their adaption considers randomly selecting two population labels, in which two labels can take the same value, and then sampling a permutation ordering to rearrange the allocation variables within the labeled population(s). They define a set of ordered intermediate distributions, where there is a difference of an allocation variable between them. Reweighting and resampling steps between the intermediate distributions are made to assess the groupings of the subsetted individuals until they have all been reassigned. Thus the final particle picked out would, assuming the anchors were in two different populations, either have the allocation variables rearranged within the two populations or produce a population containing all observations (a merge move). If the two anchors were from the same population either a split move would happen or the particle would be equal to the previous state of the

Markov chain.

An alternative SMC approach for non-parametric models is introduced by Ulker *et al.* (2010). Their algorithm has the advantage of not only inferring the allocation variables within a current set of observations, but also each SMC state allows for the number of observations to increase at each state. Assuming conjugate conditions, they perform a standard SMC sampler algorithm under increasing sample size, and they use sup-optimal MCMC kernels, like we consider in our tSMC algorithm, that are blocked Gibbs sampling updates that take into account the increasing number of the allocation variables.

Jasra *et al.* (2008) used a trans-dimensional SMC algorithm, which we explained in chapter 2, for inference with the Structure model while assuming a discrete uniform distribution of having an observation assigned to a cluster/population (a finite mixture prior), and inferring both the allele population frequencies and the allocation variables. They proposed two across model proposals that involves generating allele frequencies for the missing population and the complete allocation variable set from some prior distribution, and a second move being a more in-depth birth move which proposes a new state for all population frequency alleles and the allocation variables based on an approximation of the joint posterior in each dimension using an adaption of the Expectation-Maximisation algorithm by Figueiredo *et al.* (2002). They found that the first type of proposal did not give satisfactory convergence, while the second move gave better results.

In section 5.6 we describe other adaptions of the Structure algorithm or non-parametric models, and how we consider applying such work in the future.

5.2.1 The tSMC Approach

To understand how tSMC can be useful consider that the number of unique allocations of the individuals to a fixed number of groups/populations k, with the condition that all populations are non-empty, is naturally given by the Stirling number

in (5.12). If we were to consider all possible permutations for the allocations, that are again non-empty, then this is given by the Bell numbers where

$$S_n^k = \frac{1}{k!}\sum_{j=0}^{k-1}(-1)^j\binom{k}{j}(k-j)^n \tag{5.12}$$

$$B_n = \sum_{i=1}^{n} S_n^i, \tag{5.13}$$

and it is far higher if we go by the additional assumption that populations can be empty. Thus, the number of possible permutations for the allocation variables increases factorially with k. This makes it ideal to start off with a lower population size, ensuring convergence for a smaller population and then gradually insert populations over time. We continue with the concept that to transition to a model of differing population size applying a move in RJMCMC might be very rarely accepted or Gibbs updates might struggle to reach areas of high posterior probability, as also stated in Ulker *et al.* (2010) and Bouchard-côté and Roth (2017), and it may be easier to reach and infer high population size models through tSMC via proposals that connect to each successive parameter space. The tSMC algorithm can be designed such that proposals that split clusters may be used which covers the issue when relying on single Gibbs updates in a MCMC setting alone. By giving posterior estimates under non-parametric modeling we demonstrate how tSMC may be used for other models under similar settings, at least with applications that allow for conjugate distributions (as explained in section 5.3). For non-conjugate cases, which also consider inferring other the other non-allocation parameters simultaneously, see section 5.6 for our discussion on this issue.

5.3 Posterior Distribution, MCMC Kernels and Model Jump Proposals

We now explain in depth the collapsed model adaption of the Structure model, where the posterior to be inferred is

$$\pi(z_{m_k}|y) \quad \propto \quad f(y|z_{m_k})p(z_{m_k}). \tag{5.14}$$

We consider a set of nested models of $m_k \in (m_1, ..., m_K)$, for some maximum number of populations K, in which the difference between each adjacent model is the number of possible population states that the set of allocation variables z_{m_k} can take. In this application we do not define a model m_0 to represent some prior assumptions on model m_1, although we may consider it if we were considering a model that does not integrate out any within cluster parameters (in which m_0 would then represent a proposal for said cluster parameters), and we consider that model m_1 has each $z_i = a_1$ for $i \in \{1, ..., n\}$. However, we strongly emphasise that because we are starting with a model that clearly has a marginal likelihood that is not equal to one, then instead of calculating the ML for each model we instead formulate the Bayes Factor in favour of each model m_k against model m_1.

In this section we explain the form of the likelihood, the "transformation" to increase the number of populations at each tSMC transition and how we plan on exploring the discrete parameter space of the allocation variables based on this transformation.

5.3.1 Priors and Likelihood

The joint exchangeable prior of the allocation variables, $p(z)$, is stated in (5.11) where we set $\alpha = 1$ thus giving a symmetric uniform DP prior. Regarding the Structure model, the probability of a particular set of alleles being present in a chromosome of an observation and the prior for the population allele frequencies at each site/loci

is shown in (5.1) and (5.2) respectively. For each $\bar{P}_{l\cdot a_p}$ we are assuming an uninformative prior which has hyperparameters ($\beta_{l1} = ... = \beta_{l\tilde{\varsigma}_l} = 1$). The likelihood for one observation for one diploid sequence, assuming a non collapsed model, is

$$\begin{aligned} f(y_{i\cdot}|\bar{P}, z_i) &= \prod_{l=1}^{L}\prod_{j=1}^{\tilde{\varsigma}_l} f(y_{il}|z_i, \bar{P}_{l\cdot z_i}) \\ &= \prod_{l=1}^{L}\prod_{j=1}^{\tilde{\varsigma}_l} \left(\bar{P}_{ljz_i}\right)^{\bar{\varsigma}^{(i)}_{ljz_i}}. \end{aligned} \tag{5.15}$$

We give a reminder that $\bar{\varsigma}^{(i)}_{ljz_i}$ represents the counts of the jth allele of the lth loci that is contained within the diploid sequence y_i only, with $\bar{\varsigma}^{(i)}_{l\cdot z_i} \in \{0, 1, 2\}$. We note that the complete likelihood of the observations does not need to take into account the orderings regarding when they were assigned to population groups (Bouchard-côté and Roth, 2017; Gershman and Blei, 2012; Lawson *et al.*, 2012). Suppose we let I_{a_p} represent the indexes of the sequences assigned to population a_p, the observations corresponding to the allocation variables as $z_{I_{a_p}}$ and the set of lth loci of all observations assigned to a_p defined by $y_{I_{a_p}l}$. Then the site/loci likelihood of a cluster, for a non-collapsed model, is simply given by

$$f(y_{I_{a_p}l}|\bar{P}_{l\cdot a_p}, z_{I_{a_p}} = a_p) = \prod_{j=1}^{\tilde{\varsigma}_l} \left(\bar{P}_{lja_p}\right)^{\varsigma_{lja_p}}. \tag{5.16}$$

Overall the likelihood for the collapsed model is termed by

$$\begin{aligned} f(y|z) &= \prod_{l=1}^{L}\left(\prod_{p=1}^{k} f(y_{I_{a_p}l}|a_p)\right) \\ &= \prod_{l=1}^{L}\left(\prod_{p=1}^{k}\int_{\mathcal{F}} f(y_{I_{a_p}l}|a_p, \bar{P}_{l\cdot a_p})p(\bar{P}_{l\cdot a_p})d\bar{P}_{a_p\varsigma_l}\right) \\ &= \prod_{l=1}^{L}\left(\prod_{p=1}^{k}\frac{f(y_{I_{a_p}l}|a_p, \bar{P}_{l\cdot a_p})p(\bar{P}_{l\cdot a_p})}{\pi(\bar{P}_{l\cdot a_p}|y_{I_{a_p}l}, a_p)}\right), \end{aligned} \tag{5.17}$$

where $y_{.l}$ represents the sequences at the site l. The numerator terms follow from (5.15) and (5.2). Considering that

$$f(y_{I_{a_p}l}|a_p, \bar{P}_{l\cdot a_p})p(\bar{P}_{l\cdot a_p}) \;\propto\; \prod_{j=1}^{\tilde{\varsigma}_l} \left(\bar{P}_{lja_p}\right)^{\beta_{lj}-1+\tilde{\varsigma}_{lja_p}}. \tag{5.18}$$

Then $\pi(\bar{P}_{l\cdot a_p}|y_{I_{a_p}l}, a_p)$ can be obtained by considering the conjugate relationship between the likelihood and the prior of Dirichlet$(\beta_{lj} + \tilde{\varsigma}_{lja_p})$, defined by

$$\pi(\bar{P}_{l\cdot a_p}|y_{I_{a_p}l}, a_p) \;=\; \frac{\Gamma(\sum_{j=1}^{\tilde{\varsigma}_l}(\beta_{lj} + \tilde{\varsigma}_{lja_p}))}{\prod_{j=1}^{\tilde{\varsigma}_l}\Gamma(\beta_{lj} + \tilde{\varsigma}_{lja_p})} \prod_{j=1}^{\tilde{\varsigma}_l} \left(\bar{P}_{lja_p}\right)^{\beta_{lj}-1+\tilde{\varsigma}_{lja_p}}. \tag{5.19}$$

Therefore by substitution into (5.17) we receive the complete likelihood below

$$\begin{aligned}
f(y|z) &= \prod_{l=1}^{L}\left(\prod_{p=1}^{k}\frac{f(y_{I_{a_p}l}|a_p, \bar{P}_{l\cdot a_p})p(\bar{P}_{l\cdot a_p})}{\pi(\bar{P}_{l\cdot a_p}|y_{I_{a_p}l}, a_p)}\right) \\
&= \prod_{l=1}^{L}\left(\prod_{p=1}^{k}\frac{\left(\frac{\Gamma(\sum_{j=1}^{\tilde{\varsigma}_l}\beta_{lj})}{\prod_{j=1}^{\tilde{\varsigma}_l}\Gamma(\beta_{lj})}\right)\prod_{j=1}^{\tilde{\varsigma}_l}\left(\bar{P}_{lj\varsigma_l}\right)^{\tilde{\varsigma}_{lja_p}+\beta_{lj}-1}}{\left(\frac{\Gamma(\sum_{j=1}^{\tilde{\varsigma}_l}(\beta_{lj} + \tilde{\varsigma}_{lja_p}))}{\prod_{j=1}^{\tilde{\varsigma}_l}\Gamma(\beta_{lj} + \tilde{\varsigma}_{lja_p})}\right)\prod_{j=1}^{\tilde{\varsigma}_l}\left(\bar{P}_{lja_p}\right)^{\beta_{lj}-1+\tilde{\varsigma}_{lja_p}}}\right) \\
&= \prod_{l=1}^{L}\left(\prod_{p=1}^{k}\frac{\left(\Gamma(\sum_{j=1}^{\tilde{\varsigma}_l}\beta_{lj})\right)\left(\prod_{j=1}^{\tilde{\varsigma}_l}\Gamma(\beta_{lj} + \tilde{\varsigma}_{lja_p})\right)}{\left(\Gamma(\sum_{j=1}^{\tilde{\varsigma}_l}(\beta_{lj} + \tilde{\varsigma}_{lja_p}))\right)\left(\prod_{j=1}^{\tilde{\varsigma}_l}\Gamma(\beta_{lj})\right)}\right) \\
&= \prod_{l=1}^{L}\left(\prod_{p=1}^{k}\frac{\Gamma(\sum_{j=1}^{\tilde{\varsigma}_l}\beta_{lj})}{\Gamma(\sum_{j=1}^{\tilde{\varsigma}_l}(\tilde{\varsigma}_{lja_p} + \beta_{lj}))}\prod_{j=1}^{\tilde{\varsigma}_l}\frac{\Gamma(\beta_{lj} + \tilde{\varsigma}_{lja_p})}{\Gamma(\beta_{lj})}\right).
\end{aligned} \tag{5.20}$$

5.3.2 Inferring the Posterior of Allocation Variables

To infer each $\pi(z_{m_k}|y)$ we first consider all allocation variables belonging to the same population, i.e. $k = 1$. Unlike in the previous chapters where each model transition involves the addition of new parameters, each incremental increase of k only increases the existing parameter space of the allocation variables by one additional state. We define $z_{m_k i}$ as the allocation variable for the ith observation given that it can be assigned to k possible clusters. We are interested in using a proposal based on the SAMS proposal (Dahl, 2003) to split a population. However, we cannot use each variable $z_{m_k i}$ as a proposal for each corresponding variable $z_{m_{k+1} i}$, since this proposal would have no chance of proposing the new $(k+1)$th state of the variable $z_{m_{k+1} i}$ with $k+1$ possible states. Thus we instead consider a sequence of distributions between the two parameter spaces in the form of

$$\rho_{0(k+1)} = f(y|z_{m_k})p(z_{m_k})q(z_{m_{k+1}}|z_{m_k}) \tag{5.21}$$

$$\rho_{T(k+1)} = f(y|z_{m_{k+1}})p(z_{m_{k+1}})q(z_{m_k}|z_{m_{k+1}}), \tag{5.22}$$

and how the above is defined will depend on the type of across model proposal that increases the number of states. In the next subsection we now explain how we designing each q to transition between the two stated joint distributions.

5.3.3 The Across Model Move based on the SAMS Proposal

We present one proposal to increase the number of states in the population and reallocate a subset of the allocation variables. Before we explain the sole transformation proposal, we briefly mention that we planned a move which considered generating $z_{m_{k+1}}$ from the DP prior, and simultaneously sampling z_{m_k} using a similar prior as part of a inverse move, and then generating the population allele frequencies conditional on $z_{m_{k+1}}$. However strictly relying on priors for transformations was very unlikely to

be successful.

We consider an adaption of the "Sequentially-Allocated Merge-Split Sampler" in Dahl (2003) which considers multiple ratios of Gibbs sampler probabilities to divide a single population into two groups through an MH ratio. First we randomly select a population through some distribution of

$$\psi_{J,(m_k)\to(m_{k+1})}(\cdot), \tag{5.23}$$

on the condition that said population has more than two labels associated with it. Afterwards we randomly choose two observations from this population to act as our "reference indexes" or anchors. Let $A_{m_{k+1}1}$ and $A_{m_{k+1}2}$, be the indices of these two anchor points where $A_{m_{k+1}1} \neq A_{m_{k+1}2}$ and can only take values of observation that are only contained within the selected population $\bar{a}_{m_{k+1}}$ sampled via

$$\psi_{A,(m_k)\to(m_{k+1})}(\cdot \mid \bar{a}_{m_{k+1}}), \tag{5.24}$$

with $\tilde{a}_1$ and $\tilde{a}_2$ being the index indices of the two new populations. The pair of anchors could be selected via a discrete uniform proposal with each pair of anchors having equal probability of $2/n_{\bar{a}_{m_{k+1}}}(n_{\bar{a}_{m_{k+1}}} - 1)$. An alternative is to consider all the distances between each allocation variable and then use a discrete distribution based on the normalised distances. In this case, pairs of of individuals which have a greater genetic variation are more likely to act as our two anchors. Then for all remaining allocation variables we sample a random permutation $O_{m_{k+1}}$ on the ordering of how they are assigned to the two new groups, where for now it is given by a discrete uniform distribution,

$$\psi_{O,(m_k)\to(m_{k+1})}(\cdot \mid A_{m_{k+1}1}, A_{m_{k+1}2}, \bar{a}_{m_{k+1}}). \tag{5.25}$$

We define the auxiliary variables as $u_k = \{\bar{a}_{m_{k+1}}, A_{m_{k+1}1}, A_{m_{k+1}2}, O_{m_{k+1}}\}$. The population to be selected, the anchors and the allocation ordering variables do not change

throughout the set of intermediate distributions involved in the transition from z_{m_k} to $z_{m_{k+1}}$.

We define $y_{I^{(-i)}_{\tilde{a}_1}}$ to be set of observations that are currently assigned to the first split cluster that does not include the ith ordered (given the permutation from $O_{m_{k+1}}$) observation, which should at least contain the anchor observation, and $y_{I^{(-i)}_{\tilde{a}_2}}$ is the set of observations currently assigned to the second split cluster which again does not contain the ith ordered observation. From this the proportional probability to assign the ith ordered individual from population $\bar{a}_{m_{k+1}}$ to belong to one of the two new population clusters, in this case population $\tilde{a}_1$, that are seeded by the anchor variables is stated below,

$$Pr(z_{m_{k+1}i} = \tilde{a}_1 | z_{m_{k+1}(-i)}, y_i, y_{(-i)}) = \frac{\pi(z_{m_{k+1}i} = \tilde{a}_1 | z_{m_{k+1}(-i)}, y_i, y_{I^{(-i)}_{\tilde{a}_1}})}{\sum_{j=1}^{2} \pi(z_{m_{k+1}i} = \tilde{a}_j | z_{m_{k+1}(-i)}, y_i, y_{I^{(-i)}_{\tilde{a}_j}})} \quad (5.26)$$

$$\pi(z_{m_{k+1}i} = \tilde{a}_1 | z_{m_{k+1}(-i)}, y_i, y_{(-i)}) \propto \left(n_{\tilde{a}_1} \times \int_{\mathcal{F}} f(y_i | z_{m_{k+1}i} = \tilde{a}_1, \bar{P}_{l \cdot \tilde{a}_1 \varsigma_l}) \times \pi(\bar{P}_{l \cdot \tilde{a}_1} | y_{I^{(-i)}_{\tilde{a}_1}}) d\bar{P}_{l \cdot \tilde{a}_1} \right) \quad (5.27)$$

$$\pi(z_{m_{k+1}i} = \tilde{a}_2 | z_{m_{k+1}(-i)}, y_i, y_{(-i)}) \propto \left(n_{\tilde{a}_2} \times \int_{\mathcal{F}} f(y_i | z_{m_{k+1}i} = \tilde{a}_2, \bar{P}_{l \cdot \tilde{a}_2}) \times \pi(\bar{P}_{l \cdot \tilde{a}_2} | y_{I^{(-i)}_{\tilde{a}_2}}) d\bar{P}_{l \cdot \tilde{a}_2} \right), \quad (5.28)$$

and furthermore the normalisation constants in (5.26) also cancel out. We note that in (5.27) for example is simply the conditional likelihood

$$\int_{\mathcal{F}} f(y_i | z_{m_{k+1}i} = \tilde{a}_1, \bar{P}_{l \cdot \tilde{a}_1}) \pi(\bar{P}_{l \cdot \tilde{a}_1} | y_{I^{(-i)}_{\tilde{a}_1}}) d\bar{P}_{l \cdot \tilde{a}_1}, \quad (5.29)$$

multiplied by its corresponding conditional prior probability, with cancellations leading to just the term $n_{\tilde{a}_1}$, and we do not need to account for the normalisation constant

of the data. The conditional likelihood of the observation is defined by

$$\prod_{l=1}^{L}\left(\frac{\Gamma(\tilde{\varsigma}_{l\cdot\tilde{a}_1}^{(-i)}+\sum_{j=1}^{\tilde{\varsigma}_l}\beta_{lj})}{\Gamma(\tilde{\varsigma}_{l\cdot\tilde{a}_1}^{(i)}+\tilde{\varsigma}_{l\cdot\tilde{a}_1}^{(-i)}+\sum_{j=1}^{\tilde{\varsigma}_l}\beta_{lj})}\prod_{j=1}^{\tilde{\varsigma}_l}\left(\frac{\Gamma(\tilde{\varsigma}_{lja_1'}^{(i)}+\beta_{lj}+\tilde{\varsigma}_{lja_1'}^{(-i)})}{\Gamma(\beta_{lj}+\tilde{\varsigma}_{lja_1'}^{(-i)})}\right)\right). \tag{5.30}$$

and how we calculate (5.29), or (5.30), is identical to how we defined the likelihood in section 5.3.1 or alternatively how we integrated out the mixture proportions in section 5.1.2.

For each ith observation we calculate the two unnormalised probabilities to access if $z_{m_{k+1}i}=\tilde{a}_1$ or $z_{m_{k+1}i}=\tilde{a}_2$, normalise the probabilities and then draw from them. We furthermore update either $\tilde{\varsigma}_{lja_1'}$ or $\tilde{\varsigma}_{lja_2'}$ depending on which population the new observation is assigned to. Conditional on the auxiliary variables, the proposal is represented by $\psi_{z,(m_k)\rightarrow(m_{k+1})}\left(z_{m_{k+1}}\mid z_{m_k},y,u_k\right)$, where this is the product of the SAMS probabilities representing successful allocations. For example if two observations, indexed by i_1 and i_2, were to be assigned to the two split groups, but they were assigned to different groups then the density of the proposal is given by

$$\begin{aligned}\psi_{z,(m_k)\rightarrow(m_{k+1})}\left(z_{m_{k+1}}\mid z_{m_k},y,u_k\right) &= Pr(z_{m_{k+1}i_1}=\tilde{a}_1|z_{m_{k+1}(-i_1)},y_{i_1},y_{(-i_1)})\\ &\quad\times Pr(z_{m_{k+1}i_2}=\tilde{a}_2|z_{m_{k+1}(-i_2)},\\ &\quad y_{i_2},y_{(-i_2)}).\end{aligned} \tag{5.31}$$

This procedure we have introduced defines $q(z_{m_{k+1}}|z_{m_k})$. The distribution $q(z_{m_k}|z_{m_{k+1}})$ is defined by considering that the auxiliary distributions

$$\psi_{J,(m_{k+1})\rightarrow(m_k)}(\bar{a}_{m_{k+1}}) \tag{5.32}$$

$$\psi_{A,(m_{k+1})\rightarrow(m_k)}(A_{m_{k+1}1},A_{m_{k+1}2}\mid\bar{a}_{m_{k+1}}) \tag{5.33}$$

$$\psi_{O,(m_{k+1})\rightarrow(m_k)}(O_{m_{k+1}}\mid A_{m_{k+1}1},A_{m_{k+1}2},\bar{a}_{m_{k+1}}), \tag{5.34}$$

must be defined on the auxiliary variables used, and we set it to be the same as their corresponding distributions in the proposal. Then to complete the definition $\psi_{z,(m_{k+1})\to(m_k)}\left(z_{m_k} \mid z_{m_{k+1}}, y, u_k\right)$ is equal to 1 since there is only one way of merging the proposed clusters.

5.3.3.1 Joint Space Representation given by the SAMS Proposal

We now consider the form of the joint space representation of each intermediate distribution distribution given the unique problem stated in section 5.3.2. This means that our true target distribution is constructed on the extended $z_{m_{(1:K)}}$ space and what the kth iteration looks like. In fact we consider that we are just given z_{m_k} but require all other states we would need to construct a transformation proposal on all $z_{m_{(1:K)}}$. Therefore an unnormalised posterior distribution $\tilde{\pi}(\cdot)$ of z_{m_k} would take the form of

$$\begin{aligned}
\tilde{\pi}(z_{m_{(1:K)}}|m_k) \quad = \quad & p\left(z_{m_k}\right) f\left(y \mid z_{m_k}\right) \\
& \times \prod_{k'=k}^{K} \Big(\psi_{z,(m_{k'})\to(m_{k'+1})}\left(z_{m_{k'+1}} \mid z_{m_{k'}}, y, u_k\right) \\
& \times \psi_{J,(m_{k'})\to(m_{k'+1})}(\bar{a}_{m_{k'+1}})\psi_{A,(m_{k'})\to(m_{k'+1})}(A_{m_{k'+1}1}, A_{m_{k'+1}2} \mid \bar{a}_{m_{k'+1}}) \\
& \times \psi_{O,(m_{k'})\to(m_{k'+1})}(O_{m_{k'+1}} \mid A_{m_{k'+1}1}, A_{m_{k'+1}2}, \bar{a}_{m_{k'+1}})\Big) \\
& \times \prod_{k'=1}^{k-1} \Big(\psi_{z,(m_{k'+1})\to(m_{k'})}\left(z_{m_{k'}} \mid z_{m_{k'+1}}, y, u_k\right) \\
& \times \psi_{J,(m_{k'+1})\to(m_{k'})}(\bar{a}_{m_{k'+1}})\psi_{A,(m_{k'+1})\to(m_{k'})}(A_{m_{k'+1}1}, A_{m_{k'+1}2} \mid \bar{a}_{m_{k'+1}}) \\
& \times \psi_{O,(m_{k'+1})\to(m_{k'})}(O_{m_{k'+1}} \mid A_{m_{k'+1}1}, A_{m_{k'+1}2}, \bar{a}_{m_{k'+1}})\Big). \qquad (5.35)
\end{aligned}$$

However we note that the proportional normalised weight update between two models, each differing by allowing for an addition state to exist, simplifies to

$$w_{m_{k+1}} \quad \propto \quad w_{m_k} \frac{\tilde{\pi}(z_{m_{(1:K)}}|m_{k+1})}{\tilde{\pi}(z_{m_{(1:K)}}|m_k)}$$

$$
\begin{aligned}
w_{m_k}\frac{\tilde{\pi}(z_{m_{(1:K)}}|m_{k+1})}{\tilde{\pi}(z_{m_{(1:K)}}|m_k)} &= \frac{p\left(z_{m_{k+1}}\right) f\left(y \mid z_{m_{k+1}}\right)}{p\left(z_{m_k}\right) f\left(y \mid z_{m_k}\right)} \\
&\times \frac{\psi_{z,(m_{k+1})\to(m_k)}\left(z_{\cdot k} \mid z_{m_{k+1}}, y, u_k\right)\psi_{J,(m_{k+1})\to(m_k)}(\bar{a}_{m_{k+1}})}{\psi_{z,(m_k)\to(m_{k+1})}\left(z_{m_{k+1}} \mid z_{\cdot k}, y, u_k\right)\psi_{J,(m_k)\to(m_{k+1})}(\bar{a}_{m_{k+1}})} \\
&\times \frac{\psi_{A,(m_{k+1})\to(m_k)}(A_{m_{k+1}1}, A_{m_{k+1}2} \mid \bar{a}_{m_{k+1}})}{\psi_{A,(m_k)\to(m_{k+1})}(A_{m_{k+1}1}, A_{m_{k+1}2} \mid \bar{a}_{m_{k+1}})} \\
&\times \frac{\psi_{O,(m_{k+1})\to(m_k)}(O_{m_{k+1}} \mid A_{m_{k+1}1}, A_{m_{k+1}2}, \bar{a}_{m_{k+1}})}{\psi_{O,(m_k)\to(m_{k+1})}(O_{m_{k+1}} \mid A_{m_{k+1}1}, A_{m_{k+1}2}, \bar{a}_{m_{k+1}})}. \qquad (5.36)
\end{aligned}
$$

Due to the cancellation of the auxiliary distributions (since we specify the distributions to be equal in the numerator and denominator) and how the density $\psi_{z,(m_{k+1})\to(m_k)}$ is equal to one then we can simplify the form of the intermediate distribution between model transitions and express it as

$$
\begin{aligned}
\rho_t(z_{m_{k+1}}; m_k \to m_{k+1}) &= (\rho_{0(k+1)})^{1-\varphi_t}(\rho_{T(k+1)})^{\varphi_t} \qquad (5.37) \\
\rho_{0(k+1)}(z_{m_{k+1}}; m_k \to m_{k+1}) &= p\left(z_{\cdot k}\right) f\left(y \mid z_{m_k}\right) \\
&\quad \times \psi_{z,(m_k)\to(m_{k+1})}\left(z_{m_{k+1}} \mid z_{m_k}, y, u_k\right) \qquad (5.38) \\
\rho_{T(k+1)}(z_{m_{k+1}}; m_k \to m_{k+1}) &\propto p\left(z_{m_{k+1}}\right) f\left(y \mid z_{m_{k+1}}\right). \qquad (5.39)
\end{aligned}
$$

It is clear that the main flaw with this adaption is the large number of auxiliary variables in order to jump between models. The labeling, anchors and ordering cause the exploration of the parameter space to be limited as allocation variables either have to always not be a part of the split populations or between the two split groups as we explain in the next subsection.

5.3.4 General Within Model MCMC Moves

Given that the form of the intermediate distributions as defined in the previous section, we consider MCMC kernels that have separate updates for the allocation variables in the split populations of $\{\tilde{a}_1, \tilde{a}_2\}$ and the other non split populations. If we were to move the ith allocation variable that is currently in one of the split

populations to the other split population based on a Gibbs sampler move (see for example Escober and West (1995)), then the probability of the observation being in population group $\tilde{a}_1$ is proportional to

$$Pr(z_{m_{k+1}i} = \tilde{a}_1 | z_{(-i)}, y) = \frac{\rho_t(z_{m_{k+1}(-i)}, z_{m_{k+1}i} = a_{1'}; m_k \to m_{k+1})}{\sum_{a_{j'} \in \{\tilde{a}_1, \tilde{a}_2\}} \rho_t(z_{m_{k+1}(-i)}, z_{m_{k+1}i} = a_{j'}; m_k \to m_{k+1})}, \quad (5.40)$$

where

$$\begin{aligned} \rho_t(z_{m_{k+1}(-i)}, z_{m_{k+1}i} = a_{j'}; m_k \to m_{k+1}) &= (f(y|z_{m_k})p(z_{m_k}))^{1-\varphi_t} \\ &\times (\psi_{z,(m_k)\to(m_{k+1})}(z_{m_{k+1}(-i)}, \\ &z_{m_{k+1}(-i)} = a_{j'} \mid z_{m_k}, y, u_k)^{1-\varphi_t} \\ &\times \Big(f(y|z_{m_{k+1}(-i)}, z_{m_{k+1}i} = a_{j'}) \\ &\times p(z_{m_{k+1}(-i)}, z_{m_{k+1}i} = a_{j'})\Big)^{\varphi_t}, \quad (5.41) \end{aligned}$$

where we can further simplify $\psi_{z,(m_k)\to(m_{k+1})}\left(z_{m_{k+1}} \mid z_{m_k}, y, u_k\right)$ by removing the joint probabilities that precede from the location of the ith observation in the SAMS ordering $O_{m_{k+1}}$. However the subset of these probabilities that come after $z_{m_k i}$ in said ordering will change depending on the proposed values of $z_{m_k i}$, and we emphasise that they must be recalculated according to (5.26). Furthermore we could trim the likelihoods and priors to only incorporate the observations within the split groups only. We cannot apply this Gibbs kernel to the anchors as they are needed to define the split groups, and we cannot move any observations to any other clusters except for the two spit groups.

If the ith observation is within one of the non-split groups then the proportional probability of it being moved to a different cluster a_j, which is not one of the split

groups, is defined by

$$Pr(z_{m_{k+1}i} = a_j | z_{m_{k+1}(-i)}, y) = \frac{\rho_t(z_{m_{k+1}(-i)}, z_{m_{k+1}i} = a_j; m_k \to m_{k+1})}{\sum_{a_{j'} \notin \{\tilde{a}_1, \tilde{a}_2\}} \rho_t(z_{m_{k+1}(-i)}, z_{m_{k+1}i} = a_{j'}; m_k \to m_{k+1})} \quad (5.42)$$

$$\begin{aligned} \rho_t(z_{m_{k+1}(-i)}, z_{m_{k+1}i} = a_{j'}; m_k \to m_{k+1}) &= (f(y|z_{m_k(-i)}, z_{m_k i} = a_{j'}) \\ &\quad \times p(z_{m_{k+1}(-i)}, z_{m_{k+1}i} = a_{j'}))^{1-\varphi_t} \\ &\quad \times (f(y|z_{m_{k+1}(-i)}, z_{m_{k+1}i} = a_{j'}) \\ &\quad \times p(z_{m_{k+1}}))^{\varphi_t}, \end{aligned} \quad (5.43)$$

where $\psi_{z,(m_k)\to(m_{k+1})}(\cdot)$ cancels out. Note that cannot move this observation to $\tilde{a}_1$ or $\tilde{a}_2$ if it is not already contained in of those groups and furthermore we cannot move the observation if it causes one population to be empty.

An alternative move is to perform a reverse transformation and then initiate completely new SAMS move, as explained in the previous subsection, on the parameters as our MH move (in summary, an independent sampler for all allocations). What this type of move allows for is if a bad choice for the populations to split is made, or a bad choice of anchors, then this move allows us to essentially propose a completely new transformation. It is debatable that such as move is necessary, for example if it would be more appropriate to consider more particles to ensure that all populations have some probability of being split. Therefore we later test this and compare this move to at least the Gibbs sampler only counterpart.

We could apply kernel moves to the auxiliary variables such as which population to split or which anchors should be used to represent the two new populations, however as we are more interested in satisfactory diversity over z and z' so we ignore such moves. While Ulker *et al.* (2010) applied blocked Gibbs updates on the allocation variables, we have not presented considered such a move and how to perform them for the split and non-split groups.

5.4 Data and Diagnostics

We examine the tSMC adaption on allele data for Taita Thrush (*Turdus Helleri*), introduced in Galbusera *et al.* (2000), and consists of 155 diploid individuals and 7 microsatellite loci. This bird species was sampled at four locations, but this may not be representative of the population structure, and missing alleles are present in some of the loci.

We evaluate the results for this dataset up to eight unique population groups. For both datasets we examine the effect on,

- Performing No MCMC moves, which we use to determine if the SAMS auxiliary proposal generates enough diversity to ensure good performance.
- Using the single Gibbs sampling proposal on each of the allocation variables, with the tSMC specific conditional distributions stated in section 5.3.4.
- Using the above stated Gibbs move, and an additional MH move where we inverse the original SAMS clustering and retry the original SAMS move.

We continue to use the same adaptive annealing scheme as the previous two chapters and base the intermediate geometric distributions on 0.95CESS. A particle size N of 250 is used for summarising properties of the posterior distribution, using the diagnostics below. All prior settings were stated in section 5.3.

We identify the weighted mean permutation of the posterior results, based on Gusfield (2002), which considers the partition distance of $D^{'}(\{z_{m_k}\}^i, \{z_{m_k}\}^j)$ for each particle set of model m_k. The partition distance can be be described as the minimum number of individuals that must be deleted, or the number of allocation variables that must move around, to make the allocation variable $\{z_{m_k}\}^i$ from the ith particle be identical to $\{z_{m_k}\}^j$. The weighted mean partition $\overline{z}_{m_k}$ for model m_k is the partition that minimises the squared distance of $\sum_{i=1}^{P} w_{m_k i} \left(D^{'}(\{z_{m_k}\}^i, \overline{z}_{m_k})\right)^2$, where w_{m_k} are the normalised particle weights corresponding to model m_k. To find the mean partition

we first pick a random particle to act as our current $\overline{z}_{m_k}$. Given an ordering of the allocation variables we move each of the ordered variables, one at a time, and check if there has been a decrease in the squared distance when placed in any of the populations. Note that the allocation variables are moved while still retaining the same number of non-empty populations. Should there be a decrease in the squared distance then we move this allocation variable to the new population which becomes our new $\overline{z}_{m_k}$, revert back to the original ordering and repeat the same process. The final estimate of $\overline{z}_{m_k}$ is defined if there is no decrease in the squared distance after the collection of possible single permutations for each allocation variable have been checked (Huelsenbeck and Suchard, 2007). An alternative plot we also consider are plaid plots which is a $n \times n$ matrix object displaying the weighted probabilities of the ith individual and jth individual being in the same population group.

We also analyse the number of intermediate distributions that occurred when adaptively choosing the number of intermediate distributions to converge to each posterior and the best possible number of populations as given by the BF. Given the results from the previous chapters we expect the true values of the BF to be underestimated given how the inclusion of auxiliary variables lead to underestimation of the marginal likelihood, and thus Bayes factors, as it prevented other parts of the posterior to be sufficiently explored.

We also compare tSMC results with a standard Gibbs sampler algorithm, which again infers a posterior distribution of a collapsed Structure model that only infers the allocation variables. Defining z_i^j to be the ith allocation variable at the jth state in the Markov chain, then the Gibbs probability that we move z_i to population a_p is given by

$$\Pr(z_i^j = a_p | z_{(-i)}^j, y, \bar{P}) \quad \propto \quad n_{a_p} \int_{\mathcal{F}} f(y_i | z_i^j = a_p, \bar{P}_{l \cdot a_p}) \pi(\bar{P}_{l \cdot a_p} | y_{I_{a_p}^{-i}}) d\bar{P}_{l \cdot a_p}, \quad (5.44)$$

with the exact form of the integral shown in (5.30). However we also state that the

probability that it will move to a new group, say a_{k+1}, is defined by

$$\Pr(z_i^j = a_{k+1}|z_{(-i)}^j, y, \bar{P}) \quad \propto \quad \alpha \int_{\mathcal{F}} f(y_i|z_i^j = a_{k+1}, \bar{P}_{l\cdot a_{k+1}}) p(P_{l\cdot a_{k+1}}) d\bar{P}_{l\cdot a_{k+1}}, \tag{5.45}$$

in which the integral is defined by (5.30) with $\tilde{\varsigma}_{lja_{k+1}}^{(-i)}$ and $\tilde{\varsigma}_{l\cdot a_{k+1}}^{(-i)}$ equal to zero (Huelsenbeck and Suchard, 2007). Naturally the complete Gibbs probabilities are defined by dividing (5.44) and (5.45) by the sum of (5.45) and each (5.44) for all $p \in \{1, ..., k\}$. Finally we compare our tSMC adaption to a Gibbs sampler combined with the SAMS Metropolis-Hastings Proposal as given by (Dahl, 2003). One difference between the original algorithm and how we implemented a portion of it to act as a transformation proposal for the tSMC algorithm is that they choose two allocation variables at random instead of picking a population to split. Should the two chosen allocation variables belong to different populations groups then the two populations are merged and if they are in the same population then a split is proposed with the two variables as the anchors. Otherwise the way the probabilities to allocate each individual to one of the two split populations is identical to (5.26). We would then accept the proposed allocation variable set of $\tilde{z}^j$ with MH probability of

$$\min\left\{1, \frac{f(y|\tilde{z}^{(j)})p(\tilde{z}^{(j)})}{f(y|z^{(j-1)})p(z^{(j-1)})} \frac{q(z^{(j-1)} \mid \tilde{z}^{(j)})}{q(\tilde{z}^{(j)} \mid z^{(j-1)})}\right\} \tag{5.46}$$

where each $q(\cdot|z)$ depends on whether a split or merge move occurred. For example if the two anchors indicate a split move then $q(\tilde{z}^{(j)} \mid z^{(j-1)})$ will be a product of Gibbs probabilities shown in (5.26).

5.5 Results

In figures 5.1 and 5.2, we display the weighted mean partitions for the thrush data. Under the thrush data when transitioning to three populations, all three adaptions gave identical results to Huelsenbeck and Suchard (2007) when they applied a Dirichlet process prior to the very same thrush dataset. This was also backed up by results

(a) Mean partition when a single Gibbs kernel is applied.

(b) Mean partition when no Gibbs kernels are applied.

(c) Mean partition when a single Gibbs kernel + SAMS kernel is applied.

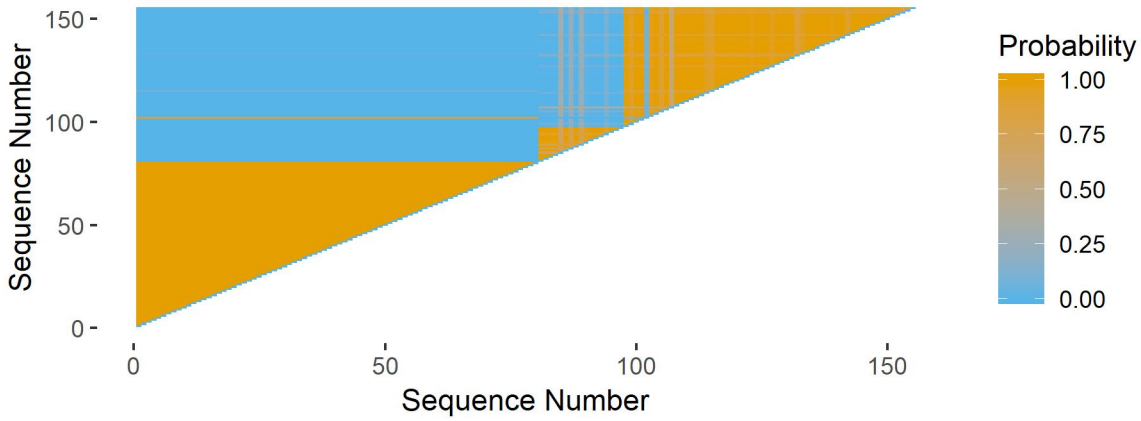
Figure 5.1: Mean partitions of the thrush data under a population size of three.

(a) Mean Partition when a single Gibbs kernel is applied.

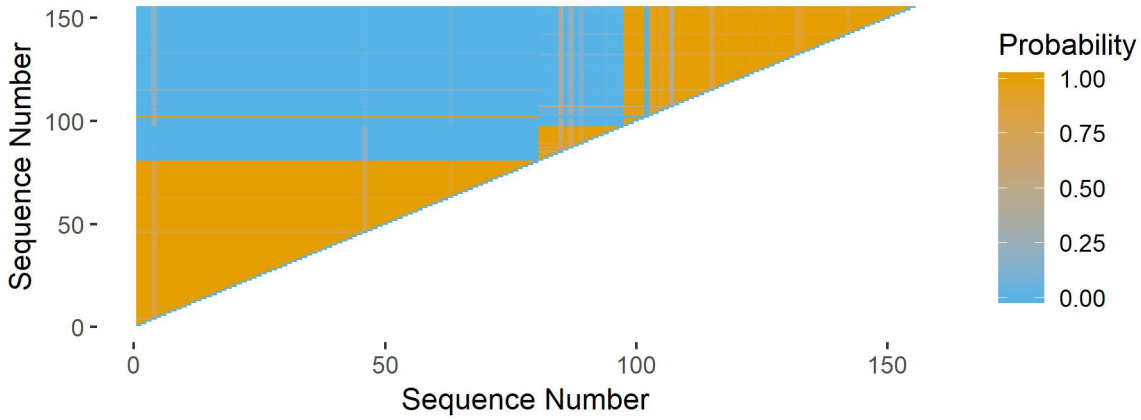
(b) Mean partition when no Gibbs kernel is applied.

(c) Mean partition when a single Gibbs kernel + SAMS kernel is applied.

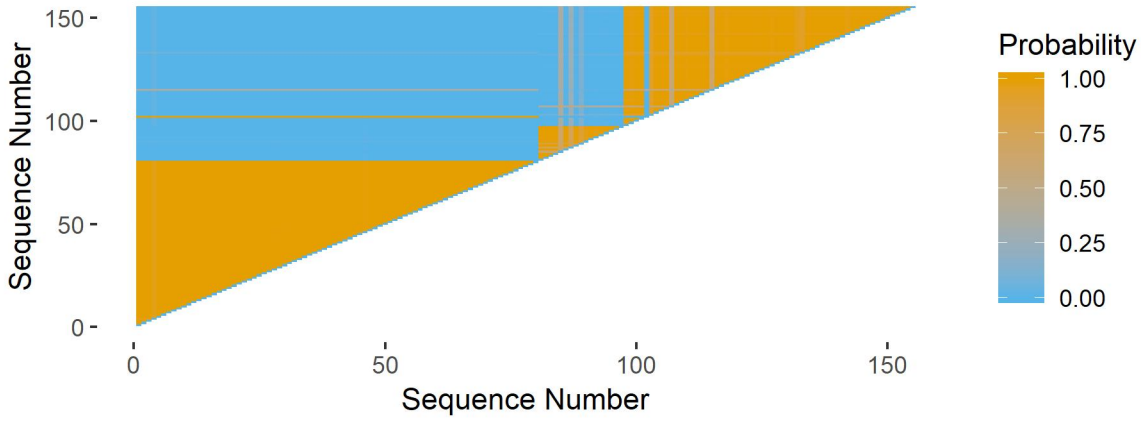
Figure 5.2: Mean partitions of the thrush data under a population size of four.



(a) Plaid plot when no MCMC kernels are applied.



(b) Plaid plot when a single Gibbs kernel was applied.



(c) Plaid plot when a single Gibbs kernel + SAMS kernel is applied.

Figure 5.3: Plaid plots for the thrush data under population size of three.

from our Gibbs samplers algorithm and the SAMS + Gibbs sampler algorithm which also gave the same mean partition for a run that uses 20,000 particles with a burn-in of 5000 particles.

However when forcing the model to transition to a model that considered a population group size of four, it did not match the results in comparison to Huelsenbeck and Suchard (2007) when a population size of four was fixed (although this under a finite mixture prior, and not a DP prior). This arguably could of been due to how we are using a Dirichlet process prior which favours smaller and compact groups. This pattern continues for all increasing number of population states added to the model. Under three populations we had a mean squared distance of 2.3296 under the no MCMC kernel scheme, 6.5282 under the Gibbs only kernel scheme and 4.48405 for the Gibbs + SAMS Kernel. As seen from the plaid plots in figure 5.3, the vast majority of particles displayed very little variation with a high concentration of certain individuals pairings sharing the same cluster. For example the 102nd read observation was close to a probability of one to be part of the same cluster as the first set of 80 observations.

Figures 5.4 and 5.5 show the log Bayes factors, up to 8 populations, and the cumulative intermediate distributions that was used to reach each population size respectively. The Thrush data gave the highest Bayes factor, for model m_k against model m_1, at three populations in which Huelsenbeck and Suchard (2007) also stated a similar pattern for their ML estimates when fixing the population sizes to a constant, although the estimates were made for a finite mixture allocation prior so the ML values should not be compared. Naturally we need to account for some Monte Carlo variance in the estimation of the BF in figure 5.4. In regards to the number of intermediate distributions needed for each transitions, there was no notable difference in the speed of convergence whether using an additional SAMS kernel or not. Applying no MCMC kernels in the tSMC algorithm showed to have far fewer intermediate steps, but this might be expected as given an adaptive scheme and the mean squared distance there

existed less variation in the particles in comparison to the other schemes.

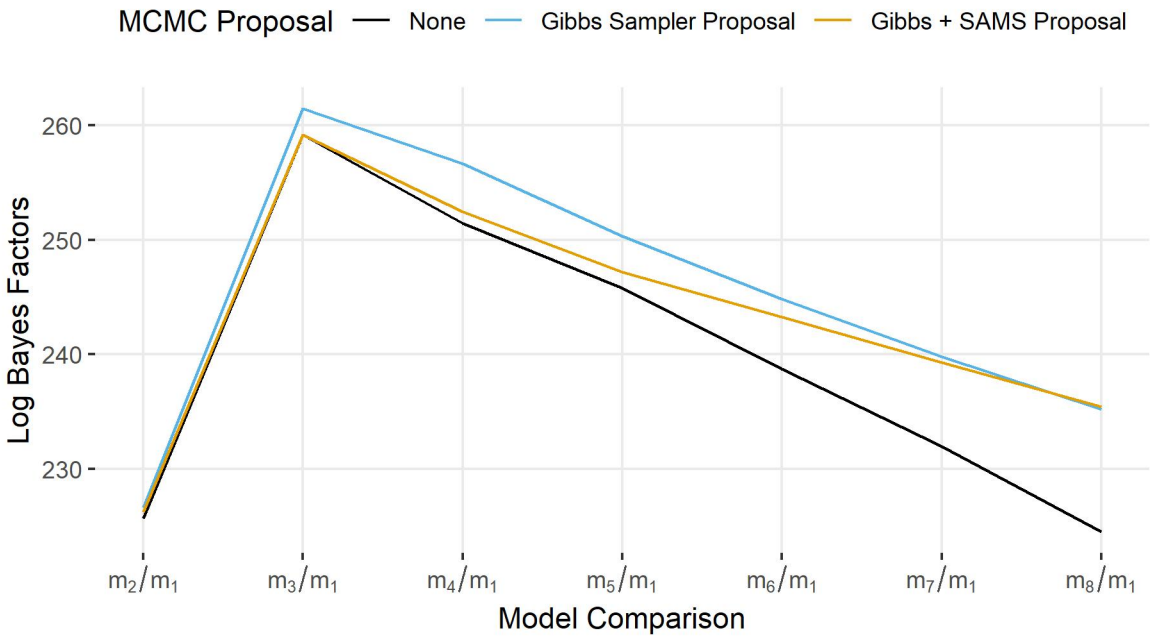


Figure 5.4: Log Bayes factors, of model m_k against model m_1, for the thrush data under three different MCMC kernel schemes within tSMC.

However we found that the tSMC adaption in this application does not perform any better then either the Gibbs Sampler or the SAMS + Gibbs Sampler algorithms, at least under the thrush dataset. We ran both algorithms for under 20,000 iterations, which is more than the maximum number of MCMC moves that was applied in the three tSMC runs (being the particles multiplied by the number of intermediate distributions) to reach the mean partition with a total of three populations as seen in figure 5.6. When plotting the unnormalised posterior densities over the iterations, shown in figure 5.6, what can be seen is that both runs converge to the posterior mode that gives the partition mean shown in figure 5.6 after 500 iterations have passed. This is significantly faster than tSMC. This was repeated multiple times with similar results. Both of the diagnostic algorithms have a smaller computational cost of $\mathrm{O}(n)$ in comparison to tSMC. Furthermore since our algorithm required that the product of Gibbs probabilities be recalculated in each intermediate distribution, see (5.37),

when performing single Gibbs move on individual allocation variables, means that our tSMC adaption is more computationally complex in comparison to having each target distribution have no bridging between two models like in the Gibbs sampler algorithm.

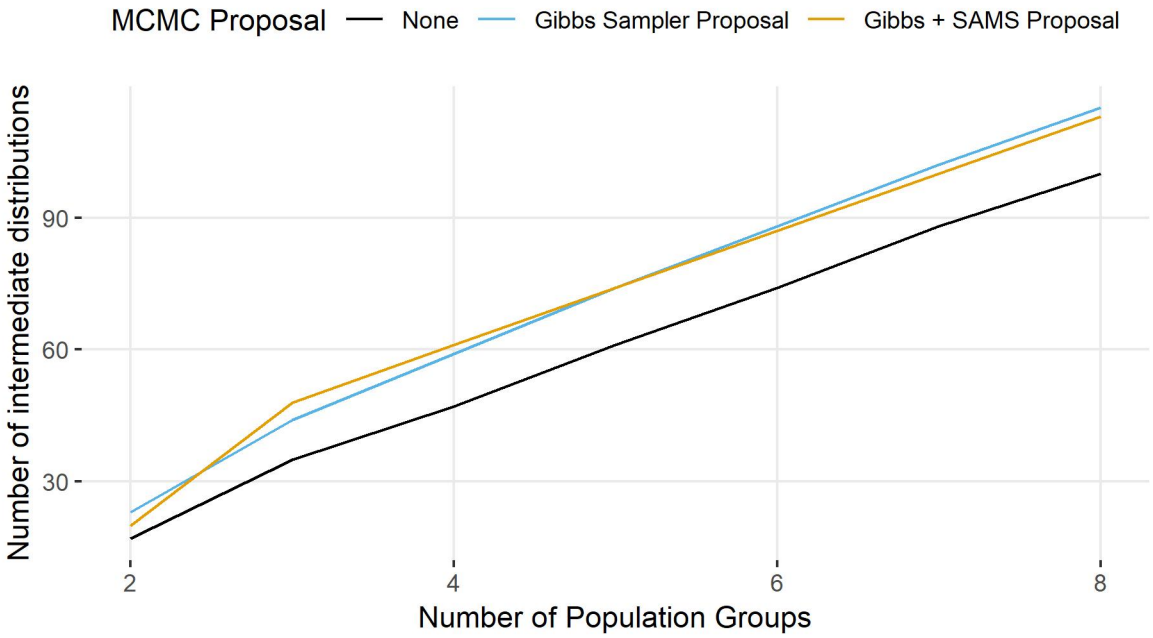


Figure 5.5: Intermediate distributions for the thrush data under three different MCMC kernels within tSMC.

More complex population data, or known but complex simulated allele data, is needed to understand how tSMC might have advantages or disadvantages over the tested established methods. Nevertheless, as we will now discuss in section 5.6, we have only tested the tSMC adaption on a model in which there exists a conjugate relationship between the likelihood and all other model parameters leaving the allocation variables to infer. Therefore we need to equally consider whether it is possible to use tSMC to infer population clustering under more complex model assumptions, as well as how to reduce the complexity of calculating each target distribution when using the SAMS transformation proposal.

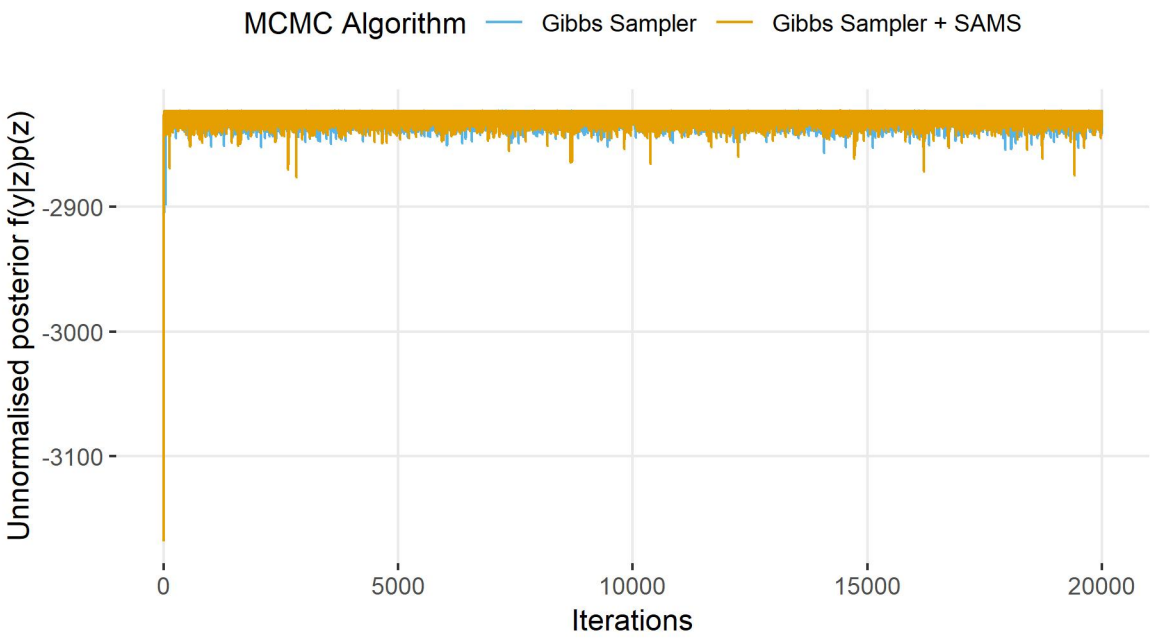


Figure 5.6: Plot of the unnormalised posterior densities over the iterations for the Gibbs sampler algorithm and the SAMS + Gibbs sampler algorithm.

5.6 Discussion

We have given a method to infer the allocation variables on conjugate based mixture models under a Dirichlet process prior. We had good performance on the thrush dataset. Nevertheless there are still some uncertainties to our adaption, as we explain within this section.

What would have been desirable was to decondition the model with regards to the label for the population to split, the allocation variable anchors to define the two split group and the orderings to sort the remaining observations. This is not possible as the parameter space for the discrete variables can be huge where for example if we were to consider the discrete orderings $O_{m_{k+1}}$ for 10 observations then there are over 3 million possible joint Gibbs based probabilities sets to calculate, and that is not including the possible number of anchors pairings that the orderings condition on. Nevertheless this should not affect the estimation of model posteriors since the posterior is the same

for each auxiliary variable. If we were able to then apply appropriate deconditoning then the loss of dependence would of allowed for more flexible MCMC moves where we could propose each allocation variable to be assigned to any population group regardless of whether they are a part of a split group or not.

In terms of different moves to create new population groups, the "Restricted Gibbs Sampling Split Merge" algorithm for conjugate posteriors by Jain and Neal (2004) is an alternative move that splits a population into two population groups. They randomly choose two of the observations, and should they belong in the same population group they perform a split move. Afterwards the remaining observations are randomly assigned between the two new split populations. Then multiple Gibbs scans are made on the same observations (except for the anchors) within the two populations. Finally the split population is accepted via a Metropolis-Hastings probability. We have let to devise whether the inverse transformation is possible, and should it be applied to tSMC, as we have let to come up with a solution that allows us to identify the auxiliary variables (consisting of Gibbs probabilities) that would of led to the proposed allocation variables. When no Gibbs Scans are used after reassignment then it doesn't give any notable advantages over Dahl (2003) as the initial allocations to the two split groups have equal probability instead of a conditional probability and it becomes similar to another split move by Nobile and Fearnside, 2007.

There was a proposed split transformation move in Nobile and Fearnside (2007) that integrates out the auxiliary variables responsible for assigning each observation to one of the split groups. Given some population to split they propose to move each individual to a new population, with a shared probability generated from a beta distribution that is defined so that it is highly likely that the new population will not be empty, or otherwise stay within their original population group. Unlike the split-merge move by Dahl, 2003, this probability does not use the similarities between the new sequences and the existing sequences between groups. Therefore while this move clearly does not have to be concerned about the anchors or orderings, we did

not use this move as the allocations are based on basic random probabilities instead of the properties of the data. It would only be appropriate for a small population size, and we would still need to change their adaption to include anchors such that we absolutely guarantee an increased number of populations in comparison to a marginal chance of an empty population like they have done. Otherwise they do suggest a few MH and Gibbs based moves which can allocate a set of individual from one population to a different population, or exchanges the allocations between populations, without changing the total number of states.

Otherwise the second across model move by Jasra *et al.* (2008), as stated in section 5.2, did show a strong potential to work. In particular this proposal is quite similar to that of a split move in chapter 3 in how it adjusts all population allele frequencies based on some deterministic function. Nevertheless since the transformation is based on the Expectation-Maximisation algorithm, there is no clear transformation relationship between a target distribution and a transformed importance sampler that is required for tSMC.

In future work we should consider how to use tSMC in the non-conjugate case, where we cannot use the collapsed model (i.e where the within cluster parameters remain in the model). While Jain and Neal (2007) proposed a variation of the split-merge sampler, their method only works for conditional conjugate models, where conjugate relationships exists for a parameter set providing that a separate subset of parameters is fixed to allow for a conjugate relationship. Otherwise their algorithm approximately shares the same algorithm to Jain and Neal (2004), and there is still some uncertainty whether we can apply it due to the multiple Gibbs Scans as stated earlier. Alternatively Dahl (2005) suggested that instead of integrating out the parameters to obtain (5.26), we instead consider replacing the conditional probabilities given in (5.27) and (5.28) with $n_{\tilde{a}_1} \times f(y_i|z_{m_{k+1}i} = \tilde{a}_1, \bar{P}_{l\cdot\tilde{a}_1})$ and $n_{\tilde{a}_2} \times f(y_i|z_{m_{k+1}i} = \tilde{a}_2, \bar{P}_{l\cdot\tilde{a}_2})$ respectively where initial values $\bar{P}_{l\cdot\tilde{a}_1}$ and $\bar{P}_{l\cdot\tilde{a}_2}$ are generated based on the prior and a single anchor observation. When an observation gets assigned to one of the split

groups then the selected populations corresponding parameters, $\bar{P}_{l\cdot\tilde{a}_1}$ or $\bar{P}_{l\cdot\tilde{a}_2}$, are updated.

We could very easily apply our tSMC adaption to univariate Gaussian mixture models by integrating out all weights, means and precisions for each component by setting conjugate priors, where there exist several choices, for the joint parameter set (Görür and Rasmussen, 2010; Tadesse *et al.*, 2005). Further research is required regarding how we could fuse our two strategies for model exploration together, one that splits the components and another that splits a clustering group.

In regards to how we infer the Structure algorithm, we may also want to consider if admixture is present in the populations. In Pritchard *et al.* (2000) they assume that for each individual a certain proportion of their loci was inherited from different populations, this being a basic definition of admixture, instead of a single group like we have assumed in the investigation. Falush *et al.* (2003) extended the admixture assumptions by Pritchard *et al.* (2000) to account for admixture linkage disequilibrium where a set of unbroken combined loci is usually inherited together through many generations. These segments of unbroken loci on each sequence are defined by breakpoints given by a Poisson process, where higher Poisson rates would mean more breakpoints and an infinite rate implies independence of all loci. Under these assumptions the allocation variables take the form of which population did the combined loci within these breakpoints come from. Alternatively we may want to account for the existence of correlated alleles. The assumption is that all the populations diverged from a most recent common ancestor population, and as each population would have different magnitudes of genetic drift the allele frequencies in the ancestral population may give information on each descendant population (Falush *et al.*, 2003; Nicholson *et al.*, 2002). To translate both types of extensions into the Structure algorithm under non-parameter model assumptions we need to consider hierarchical Dirichlet processes (Teh *et al.*, 2005). How such model assumptions can be given in both parameter and non-parametric (Dirichlet process) form is stated in De Iorio *et al.* (2015). An al-

ternative parameterisation that does account for admixed individuals while avoiding hierarchical DP would be an adaption of the fineStructure algorithm by Lawson *et al.* (2012). They consider using an alteration of the data that takes the form of a $N \times N$ co-ancestry matrix which accounts for the admixture between all individuals. They infer the allocation variables and a $K \times K$ object being the population co-ancestry matrix. This would be the easiest to adapt as Lawson *et al.* (2012) already applies a SAMS like proposal to split a population into two, which is again accomplished by integrating out the population co-ancestry matrix from the posterior.

In our analysis we fixed α within $p(z)$, see (5.10) and (5.11), to some constant. The concentration parameter α affects the size of each population group with a decreasing α leading to great variability between groups with one or two groups being dominant in population size. Inferring the posterior of this parameter is possible under basic MCMC schemes (see Bouchard-côté and Roth (2017); Escober and West (1995)). Appropriate kernel moves that can be applicable under a geometrically bridged intermediate distribution would need to be considered. Huelsenbeck and Suchard (2007) showed that under an MCMC scheme where Dirichlet process priors were applied to the allocation variables, varying set values of α did not have an effect on the mean partitions of the thrush data (as well as in other datasets). This differed from their simulated data, in which the simulated data was constructed with α such that a certain number of populations are expected to be present in each dataset. When misclassifying α to be larger than its true value value the average distance between the mean partitions and true partitions of the simulated datasets increased. On average for all simulated runs a higher posterior mean for the number of populations, in comparison to the true population size, was given under a higher misclassified concentration parameter. Nevertheless both overestimation decreased with increasing size of the loci. Overall only practical analysis would show whether α differs greatly between parameter spaces, and this would have an affect on the mean partitions, but inferring this parameter is not an intermediate interest.

The priority for future adaptions of tSMC should be to make it applicable for general mixture models which include non-conjugate posteriors. In regards to the posterior estimation, there are still uncertainties how the restrictions on the MCMC kernels can harm the exploration of the parameter space or if using a large particle size can compensate for this problem. We also have to consider transformations that have a lower computational complexity then our adaption of the SAMS proposal. Therefore we wish to decondition our existing adaption in regards to the labeling given by the auxiliary variables, although it is not clear how this can be accomplished.

Final Discussion

In this investigation we have introduced transformation Sequential Monte Carlo, an adaption of the SMC algorithm that specialises in across-model simulation. This adaption should be applied when a series of ordered models can be defined, which differ by dimensional size or both a difference of observational and dimensional size, and transformation proposals can be made between each adjacent model. The key concept is how a proposal that is based on a transformation of some existing model, that targets the new parameter space, has the potential to be more efficient then some prior distribution. The quality of the proposal is then accessed through a series of annealed intermediate distributions, with proposals of low probability either adjusted via MCMC kernels or removed via a resampling algorithm. Furthermore the use of adaptive schemes means that there is flexible control of the number, and type, of intermediate distributions and the overall exploration of the parameter space. We have assessed the performance of tSMC when inferring the posterior distribution of a standard univariate mixture model, a mixture model under a Dirichlet process prior and another application in the field of population genetics. We considered application specific research questions, and if they can be answered under our current tSMC scheme. In this final discussion we summarise the general tSMC algorithm. We refer to application specific points in the discussion sections of each chapter, and conclude with the following general key points.

If the primary aim was to simply model the posterior where attempting to provide an initial estimate proves to be challenging and a series of subsetted models exist, such

as reconstruction of the ancestral history of a series of genomes where the space can become very large, then tSMC is really useful. In particular we are positive with our work on the reconstruction of the genealogy under coalescent model conditions in chapter 4 which has the best potential for further research. We were still satisfied with some of the results in chapters 3 and 5 even though there exists problems with our tSMC adaptions in comparison to other established methods.

With regards to future research in the application of genealogy reconstruction we hope to use tSMC under JavaScript to be compatible with Beast phylogenetic software (Drummond *et al.*, 2012). Their software offers faster posterior density evaluation then what we have used for this investigation due to the nature of Java v.s R. However the addition of "online" posterior updates and data inclusion, as well as the parallelisation properties, are key traits that tSMC has over the current MCMC based ancestral reconstruction software. Nevertheless in the long term this would require implementation of other population genetic parameters and under non-coalescent phylogenetic assumptions. More notably we would require additional changes to the proposals to graft a new node to a tree. Caution is required with how the type of ordering for gradual data inclusion can affect the ML. An analysis is required to consider if there exist scenarios where, over a real life time period, we would expect more differences in the newer sequences in comparison to the existing gnomes and thus likely to be placed as recent coalescent events.

What we initially believed was through the use of tempered annealing to gradually converge to posterior distribution, be it through a large number of intermediate distributions or an adaptive scheme dictating the variability of the particles, the marginal likelihood would also similarly converge to its true value. We have discovered that this is not the case, and in many cases underestimation of the marginal likelihood was present despite good posterior results in chapters 3-5. The asymptotic properties of a SMC algorithm means that increasing the number of intermediate distributions eventually gives an estimate that would resemble the true ML (with a minuscule bias

due to adaptive kernels and normalisation of the weights). However we never identified a way to find this safe/guaranteed number of intermediate distributions that would prevent underestimation of the ML. For example, even larger runs under each scheme were made on the mixture models in chapter 3 (which we do not show in this thesis) however they still didn't match the ML on the best run for an eight component Gaussian mixture model for the galaxy dataset despite exceeding their 1st quartile of their corresponding MC estimates shown in the marginal likelihood boxplots.

An issue is whether any transformation we apply is appropriately long-tailed. Failing to reach high probability regions of the posterior of an extended parameter space for most particles on the initial transformation, and not from the combined use of intermediate distribution and MCMC kernels like what we have seen, would cause an underestimation of the marginal likelihood. This is a problem that can occur in many of the algorithms mentioned in chapter 2, such as RJMCMC and importance sampling methods such as the harmonic mean estimator, and tSMC has not removed this issue.

How to tell if a transformation is long-tailed is a difficult task. This can be daunting if transformations, that are not identity functions, need to be applied on every single parameter in order reach posterior modes on high-dimensional space. The point of applying MCMC kernels was that small changes that were required on parameters not involved in the model transformation itself can be conducted through these kernels. This has proven to be successful as even with poor results on the marginal likelihood, generated by an inappropriate transformation, we can still obtain good approximations to the posterior.

For applications involving non-increasing data size we did not consider stopping conditions for inferring an increasing higher dimensional space, and only considered an ideal number of likely components to halt the algorithm. A possible solution would be stop the algorithm after a series of multiple drops in the estimated marginal likelihood, or due to a significant single drop in the ML when transitioning between

two models. However this requires us to account for the existence of variance of the ML estimate within one run and not stop the algorithm too soon due to Monte Carlo variance. We have let to identify how to estimate this variability in one run, although if we could we would simply define some confidence intervals for each ML estimate and stop the algorithm if some threshold has been breached.

Another minor objective would be to find a way to apply Gibbs samplers in tSMC, for the small subset of applications in which using Gibbs samplers is highly recommended over Metropolis-Hastings, while still giving us the option to not integrate out continuous variables. This would remove multiple obstacles that we have highlighted in the previous chapters, as well as any unrealised problems for any future applications. Alternate target distributions would need to be identified which allow for Gibbs probabilities to be applied for all parameters. However these intermediate distributions would also need to be close to each other such that there are no sudden jumps with each annealed state, as we found when using the arithmetic annealed target distribution.

References

Agapiou, S., Papaspiliopoulos, O., Sanz-alonso, D., and Stuart, A. M. (2017). Importance Sampling: Intrinsic Dimension and Computational Cost. *Statistical Science 32*(3), 405–431.

Alquier, P., Friel, N., Everitt, R. G., and Boland, A. (2016). Noisy Monte Carlo: convergence of Markov chains with approximate transition kernels. *Statistics and Computing 26*(1-2), 29–47.

Andrieu, C., de Freitas, N., Doucet, A., and Jordan, M. I. (2003). An Introduction to MCMC for Machine Learning. *Machine Learning 50*((1-2)), 5–43.

Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 72*(3), 269–342.

Andrieu, C. and Roberts, G. O. (2009). The pseudo-marginal approach for efficient Monte Carlo computations. *Annals of Statistics 37*(2), 697–725.

Antoniak, C. E. (1974). Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems. *The Annals of Statistics 2*(6), 1152–1174.

Attias, H. (1999). Inferring Parameters and Structure of Latent Variable Models by Variational Bayes. In K. Laskey and H. Prade (Eds.), *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pp. 21–30. Morgan Kaufmann Publishers Inc.

Baele, G., Lemey, P., Bedford, T., Rambaut, A., Suchard, M. A., and Alekseyenko, A. V. (2012). Improving the Accuracy of Demographic and Molecular Clock Model Comparison While Accommodating Phylogenetic Uncertainty Research article. *Molecular Biology and Evolution 29*(9), 2157–2167.

Bayes, T. and Price, R. (1743). An Essay towards Solving a Problem in the Doctrine of Bayes. By the late Rev. Mr. Bayes, frs cinnybucatd by Mr. Price, in a letter to Jon Canton, A.M.R.F.S. *Philosophical Transactions*, **53**, 370–418.

Bechtel, Y. C., Bonaiti-Pellie, C., Poisson, N., Magnette, J., and Bechtel, P. R. (1993). A population and family study of *N*-acetyltransferase using caffeine urinary metabolites. *Clinical Pharmacology & Therapeutics 54*(2), 134–141.

Berger, J. (2006). The Case for Objective Bayesian Analysis. *Bayesian Analysis 1*(3), 385–402.

Bernardo, J. M. and Smith, A. F. M. (1994). *Bayesian Theory*. John Wiley & Sons.

Beskos, A., Crisan, Danand Jasra, A., and Whiteley, N. (2014b). Error Bounds and Normalizing Constants for Sequential Monte Carlo Samplers in High Dimensions. *Advances in Applied Probability 46*(1), 279–306.

Beskos, A., Crisan, D., and Jasra, A. (2014a). On the stability of sequential Monte Carlo methods in high dimensions. *The Annals of Applied Probability 24*(4), 1396 – 1445.

Blei, D. M. and Jordan, M. I. (2006). Variational Inference for Dirichlet Process Mixtures. *Bayesian Analysis 1*(1), 121–144.

Bouchard-côté, A. (2014). SMC (sequential Monte Carlo) for Bayesian phylogenetics. In M.-H. Chen, L. Kuo, and P. O. Lewis (Eds.), *Bayesian Phylogenetics: Methods, Algorithms and Applications*, Chapter 8, pp. 163–186. CRC Press.

Bouchard-côté, A. and Roth, A. (2017). Particle Gibbs Split-Merge Sampling for Bayesian Inference in Mixture Models. *Journal of Machine Learning Research 18*(28), 1–39.

Bouchard-côté, A., Sankararaman, S., and Jordan, M. I. (2012). Phylogenetic Inference via Sequential Monte Carlo. *Systematic Biology 65*(1), 579–593.

Brooks, S. P., Giudici, P., and Roberts, G. O. (2003). Efficient construction of reversible jump Markov chain Monte Carlo proposal distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 65*(1), 3–55.

Bryant, D. (2003). A Classification of Consensus Methods for Phylogenetics. *DIMACS series in Discrete Mathematics and Theoretical Computer Science*, **61**, 163–184.

Cappé, O., Robert, C. P., and Ryd\'{e}n, T. (2003). Reversible jump , birth-and-death and more general continuous time Markov chain Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 65*(3), 679–700.

Chen, M.-H., Kuo, L., and Lewis, P. O. (Eds.) (2014). *Bayesian Phyogenetics: Methods, Algorithms and Applications*. CRC Press.

Cheon, S. and Liang, F. (2014). Bayesian phylogeny analysis. In M.-H. Chen, L. Kuo, and P. O. Lewis (Eds.), *Bayesian Phylogenetics: Methods, Algorithms and Applications*, Chapter 7, pp. 129–162. CRC Press.

Chib, S. and Greenberg, E. (1995). Understanding the Metropolis-Hastings Algorithm. *Journal of the American Statistical Association 49*(4), 327–335.

Chopin, N. (2002). A Sequential Particle Filter Method for Static Models. *Biometrika 89*(3), 539–551.

Chopin, N. (2004). Central limit theorem for Sequential Monte Carlo Methods and Its Application to Bayesian Inference. *Annals of Statistics 32*(6), 2385–2411.

Cron, A. J. and West, M. (2011). Efficient Classification-Based Relabeling in Mixture Models. *American Statistician 65*(1), 16–20.

Dahl, D. B. (2003). An improved merge-split sampler for conjugate Dirichlet process mixture models. Technical report, University of Wisconsin - Madison.

Dahl, D. B. (2005). Sequentially-Allocated Merge-Split Sampler for Conjugate and Nonconjugate Dirichlet Process Mixture Models. *Journal of Computational and Graphical Statistics 11*(1).

De Iorio, M., Favaro, S., and Teh, Y. W. (2015). Bayesian inference on population structure: from parametric to nonparametric modeling. In R. Mitra and P. Müller (Eds.), *Nonparametric Bayesian Inference in Biostatistics*, pp. 135–151. Springer Cham.

Degnan, J. H., Degiorgio, M., Bryant, D., and Rosenberg, N. A. (2009). Properties of consensus methods for inferring species trees from gene trees. *Systematic Biology 58*(1), 35–54.

Del Moral, P., Doucet, A., and Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68*(3), 411–436.

Del Moral, P., Doucet, A., and Jasra, A. (2012). An adaptive sequential Monte Carlo method for approximate Bayesian computation. *Statistics and Computing 22*(5), 1009–1020.

Dellaportas, P. and Papageorgiou, I. (2006). Multivariate mixtures of normals with unknown number of components. *Statistics and Computing 16*(1), 57–68.

Didelot, X. and Falush, D. (2007). Inference of Bacterial Microevolution Using Multilocus Sequence Data. *Genetics 175*(3), 1251–1266.

Dinh, V., Darling, A. E., and Matsen IV, F. A. (2018). Online Bayesian phylogenetic inference: Theoretical foundations via sequential Monte Carlo. *Systematic Biology 67*(3), 503–517.

Douc, R. and Cappé, O. (2005). Comparison of Resampling Schemes for Particle Filtering Randal. In *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium*, pp. 64–69.

Douc, R., Guillin, A., Marin, J. M., and Robert, C. P. (2007). Convergence of adaptive mixtures of importance sampling schemes. *Annals of Statistics 35*(1), 420–448.

Doucet, A., de Freitas, N., and Gordon, N. J. (Eds.) (2001). *Sequential Monte Carlo Methods in Practice*. Springer-Verlag New York.

Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering.

Doucet, A. and Johansen, A. M. (2011). A Tutorial on Particle Filtering and Smoothing: Fifteen years later. In D. Crisan and B. Rozovskii (Eds.), *The Oxford Handbook of Nonlinear Filtering*, pp. 656–704. Oxford University Press.

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.

Drummond, A. J., Suchard, M. A., Xie, D., and Rambaut, A. (2012). Bayesian Phylogenetics with BEAUti and the BEAST 1.7 Research article. *Molecular Biology and Evolution 29*(8), 1969–1973.

Enright, M. C., Day, N. P. J., Davies, C. E., Peacock, S. J., and Spratt, B. G. (2000). Multilocus Sequence Typing for Characterization of Methicillin-Resistant and Methicillin-Susceptible Clones of *Staphylococcus aureus* . *Journal of Clinical Microbiology 38*(3), 1008–1015.

Escober, M. D. and West, M. (1995). Bayesian Density Estimation and Inference Using Mixtures. *Journal of the American Statistical Association 90*(430), 577–588.

Everitt, R. G., Didelot, X., Batty, E. M., Miller, R. R., Knox, K., Young, B. C., Bowden, R., Auton, A., Votintseva, A., Larner-Svensson, H., Charlesworth, J., Golubchik, T., Ip, C. L. C., Godwin, H., Fung, R., Peto, T. E. A., Walker, A. S., Crook, D. W., and Wilson, D. J. (2014). Mobile elements drive recombination hotspot in the core genome of *Staphylococcus aureus* . *Nature Communications*, **5**.

Ewens, W. J. (1972). The Sampling Theory of Selectively Neutral Alleles. *Theoretical Population Biology 3*(1), 87–112.

Falush, D., Stephens, M., and Pritchard, J. K. (2003). Inference of Population Structure Using Multilocus Genotype Data: Linked Loci and Correlated Allele Frequencies. *Genetics 164*(4), 1567–1587.

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

Felsenstein, J. (2004). *Inferring Phylogenies*. Sinauer Associates.

Feng, D. (2018). Miscellaneous Functions (Package 'miscF').

Ferguson, T. S. (1973). A Bayesian Analysis of Some Nonparametric Problems. *The Annals of Statistics 1*(2), 209–230.

Figueiredo, M. A. T., Member, S., and Jain, A. K. (2002). Unsupervised Learning of Finite Mixture Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*(3), 381–396.

Fisher, R. A. (1931). The Distribution of Gene Ratios for Rare Mutations. *Proceedings of the Royal Society of Edinburgh*, **50**, 204–219.

Fitch, W. M. (1971). Toward defining the course of evolution: minimum change for a specific tree topology. *Systematic Biology 20*(4), 406–416.

Fourment, M., Claywell, B. C., Dinh, V., McCoy, C., Matsen IV, F. A., and Darling, A. E. (2018). Effective online Bayesian phylogenetics via sequential Monte Carlo with guided proposals. *Systematic Biology 67*(3), 490–502.

Friel, N. and Pettitt, A. N. (2008). Marginal likelihood estimation via power posteriors. *Journal of the Royal Statistical Society: Series B (Methodological) 70*(3), 589–607.

Friel, N. and Wyse, J. (2012). Estimating the evidence - a review. *Statistica Neerlandica 66*(3), 288–308.

Frühwirth-Schnatter, S. (1994). Data Augmentation and Dynamic Linear Models. *Journal of Time Series Analysis 15*(2), 183–202.

Galbusera, P., Lens, L., Waiyaki, E., Schenck, T., and Mattysen, E. (2000). Effective population size and gene flow in the globally, critically endangered Taita thrush, *Turdus helleri* . *Conservation Genetics*, **1**, 45–55.

Gelfand, A. E. and Smith, A. F. M. (1990). Sampling-Based Approaches to Calculating Marginal Densities. *Journal of the American Statistical Association 85*(410), 398–409.

Gelman, A. and Meng, X.-L. (1998). Simulating Normalizing Constants: From Importance Sampling to Bridge Sampling to Path Sampling. *Statistical Science 13*(2), 163–185.

Gelman, A., Roberts, G. O., and Gilks, W. R. (1996). Efficient Metropolis jumping rules. In J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith (Eds.), *Bayesian Statistics 5*, pp. 599–607. Oxford University Press.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence 6*(6), 721–741.

Gerber, M., Chopin, N., and Whiteley, N. (2017). Negative association, ordering and convergence of resampling methods. *arxiv*.

Gershman, S. J. and Blei, D. M. (2012). A tutorial on Bayesian nonparametric models. *Journal of Mathematical Psychology 56*(1), 1–12.

Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing) 140*(2), 107–113.

Görür and Rasmussen, C. E. (2010). Dirichlet Process Gaussian Mixture Models : Choice of the Base Distribution. *Journal of Computer Science and Technology 25*(4), 653–664.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika 82*(4), 711–732.

Green, P. J. (2001). Modelling Heterogeneity With and Without the Dirichlet Process. *Scandinavian Journal of Statistics 28*(2), 355–375.

Grosse, R. B., Ghahramani, Z., and Adams, R. P. (2015). Sandwiching the marginal likelihood using bidirectional Monte Carlo. *arxiv*.

Guindon, S., Dufayard, J.-F., Lefort, V., Anisimova, M., Hordijk, W., and Gascuel, O. (2010). New Algorithms and Methods to Estimate Maximum-Likelihood Phy-

logenies: Assessing the Performance of PhyML 3 .0. *Systematic Biology 59*(3), 307–321.

Gusfield, D. (2002). Partition-distance: A problem and class of perfect graphs arising in clustering. *Information Processing Letters 82*(3), 159–164.

Haario, H., Saksman, E., and Tamminen, J. (2001). An adaptive Metropolis algorithm. *Bernoulli 7*(2), 223–242.

Hastie, D. I. (2005). *Towards Automatic Reversible Jump Markov Chain Monte Carlo*. Ph. D. thesis, University of Bristol.

Hastie, D. I. and Green, P. J. (2012). Model choice using reversible jump Markov Chain Monte Carlo. *Statistica Neerlandica 66*(3), 309–338.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika 57*(1), 97–109.

Hein, J., Schierup, M. H., and Wiuf, C. (2004). *Gene genealogies, variation and evolution: a primer in coalescent theory*. Oxford University Press.

Hitchcock, D. B. (2003). A History of the Metropolis-Hastings Algorithm. *American Statistician 57*(4), 254–257.

Höhna, S. and Drummond, A. J. (2008). Evaluation of proposal distributions on clock- constrained trees in Bayesian phylogenetic inference. *Proceedings of the New Zealand Computer Science Research Student Conference*, 41–48.

Hol, J. D., Schön, T. B., and Gustafsson, F. (2006). On resampling algorithms for particle filters. *Nonlinear Statistical Signal Processing Workshop*, 79–82.

Holder, M. and Lewis, P. O. (2003). Phylogeny estimation: traditional and Bayesian approaches. *Nature Reviews Genetics 4*(4), 275–284.

Huelsenbeck, J. P. and Andolfatto, P. (2007). Inference of population structure under a dirichlet process model. *Genetics 175*(4), 1787–1802.

Huelsenbeck, J. P., Larget, B., Miller, R. E., and Ronquist, F. (2002). Potential Applications and Pitfalls of Bayesian Inference of Phylogeny. *Systematic Biology 51*(5), 673–688.

Huelsenbeck, J. P. and Suchard, M. A. (2007). A Nonparametric Method for Accommodating and Testing Across-Site Rate Variation. *Systematic Biology 56*(6), 975–987.

Hyvärinen, A. (2005). Estimation of Non-Normalized Statistical Models by Score Matching. *Journal of Machine Learning Research*, **6**, 695–708.

Izquierdo-carrasco, F., Cazes, J., Smith, S. A., and Stamatakis, A. (2014). PUmPER: phylogenies updated perpetually. *Bioinformatics 30*(10), 1476–1477.

Jain, S. and Neal, R. M. (2004). A Split-Merge Markov chain Monte Carlo Procedure for the Dirichlet Process Mixture Model. *Journal of Computational and Graphical Statistics 13*(1), 158–182.

Jain, S. and Neal, R. M. (2007). Splitting and Merging Components of a Nonconjugate Dirichlet Process Mixture Model. *Bayesian Analysis 2*(3), 445–472.

Jasra, A. (2006). *Bayesian Inference for Mixture Models via Monte Carlo Computation*. Ph. D. thesis, Imperial College London (University of London).

Jasra, A., Doucet, A., Stephens, D. A., and Holmes, C. C. (2008). Interacting sequential Monte Carlo samplers for trans-dimensional simulation. *Computational Statistics and Data Analysis 52*(4), 1765–1791.

Jasra, A., Holmes, C. C., and Stephens, D. A. (2005). Markov Chain Monte Carlo Methods and the Label Switching Problem in Bayesian Mixture Modeling. *Statistical Science 20*(1), 50–67.

Jasra, A., Stephens, D. A., Doucet, A., and Tsagaris, T. (2011). Inference for Lévy-Driven Stochastic Volatility Models via Adaptive Sequential Monte Carlo. *Scandinavian Journal of Statistics 38*(1), 1–22.

Jeffreys, H. (1946). An Invariant Form for the Prior Probability in Estimation Problems. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences 186*(1007), 453–461.

Jeffreys, H. (1998). *Theory of Probability* (3rd ed.). Oxford University Press.

Johansen, A. M. (2009). SMCTC: Sequential Monte Carlo in C ++. *Journal of Statistic Software 30*(6), 1–41.

Jones, G. L. (2004). On the Markov chain central limit theorem. *Probability Surveys*, **1**, 299–320.

Jordan, M. I. (2004). Graphical Models. *Statistical Science 19*(1), 140–155.

Jukes, T. H. and Cantor, C. R. (1969). Evolution of protein molecules. In H. N. Munro (Ed.), *Mammalian protein metabolism 3*, pp. 21–132. Academic Press.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering 82*(1), 35–45.

Karagiannis, G. and Andrieu, C. (2013). Annealed Importance Sampling Reversible Jump MCMC Algorithms. *Journal of Computational and Graphical Statistics 22*(3), 623–648.

Kimura, M. (1980). A Simple Method for Estimating Evolutionary Rates of Base Substitutions Through Comparative Studies of Nucleotide Sequences. *Journal of Molecular Evolution 16*(2), 111–120.

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

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

Kong, A., Liu, J. S., and Wong, W. H. (1994). Sequential Imputations and Bayesian Missing Data Problems. *Journal of the American Statistical Association 89*(245), 278–288.

Kschischang, F. R., Frey, B. J., and Loeliger, H.-A. (2001). Factor Graphs and the Sum-Product Algorithm. *IEEE Transactions on Information Theory 47*(2), 498–519.

Lakner, C., Van Der Mark, P., Huelsenbeck, J. P., Larget, B., and Ronquist, F. (2008). Efficiency of Markov Chain Monte Carlo Tree Proposals in Bayesian Phylogenetics. *Systematic Biology 57*(1), 86–103.

Lartillot, N. and Philippe, H. (2006). Computing Bayes factors using thermodynamic integration. *Systematic Biology 55*(2), 195–207.

Lawson, D. J., Hellenthal, G., Myers, S., and Falush, D. (2012). Inference of population structure using dense haplotype data. *PLoS Genetics 8*(1).

Lawson, D. J., van Dorp, L., and Falush, D. (2018). A tutorial on how (not) to over-interpret STRUCTURE/ADMIXTURE bar plots. *Nature Communications 9*(1).

Lee, A. and Whiteley, N. (2018). Variance estimation in the particle filter. *Biometrika 105*(3), 609–625.

Li, N. and Stephens, M. (2003). Modeling Linkage Disequilibrium and Identifying Recombination Hotspots Using Single-Nucleotide Polymorphism Data. *Genetics 165*(4), 2213–2233.

Liang, F., Paulo, R., Molina, G., Clyde, M. A., and Berger, J. O. (2008). Mixtures of g Priors for Bayesian Variable Selection. *Journal of the American Statistical Association 103*(481), 410–423.

Liu, J. S. (2004). Theory of Sequential Monte Carlo. In *Monte Carlo Strategies in Scientific Computing*, pp. 53–77. Springer-Verlag New York.

Liu, J. S., Wong, W. H., and Kong, A. (1994). Covariance Structure of the Gibbs Sampler with Applications to the Comparison of Estimators and Augmentation Schemes. *Biometrika 81*(1), 27–40.

Liu, L., Yu, L., Kubatko, L., Pearl, D. K., and Edwards, S. V. (2009). Coalescent methods for estimating phylogenetic trees. *Molecular Phylogenetics and Evolution 53*(1), 320–328.

Lunn, D., Spiegelhalter, D., Thomas, A., and Best, N. (2009). The BUGS project: Evolution , critique and future directions. *Statistics in medicine 28*(25), 3049–3067.

Mackay, D. J. C. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press.

Margush, T. and McMorris, F. R. (1981). Consensus n-trees. *Bulletin of Mathematical Biology 43*(2), 239–244.

McGrory, C. A., Pettitt, A. N., Titterington, D. M., Alston, C. L., and Kelly, M. (2016). Transdimensional sequential Monte Carlo using variational Bayes - SMCVB. *Computational Statistics and Data Analysis*, **93**, 246–254.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of State Calculations by Fast Computing Machines. *The journal of chemical physics 21*(6), 1087–1092.

Metropolis, N. and Ulam, S. (1949). The Monte Carlo Method. *Journal of the American Statistical Association 44*(237), 335–341.

Metzker, M. L. (2010). Sequencing technologies - the next generation. *Nature Reviews Genetics 11*(1), 31–46.

Morrison, D. A. (2000). Increasing the Efficiency of Searches for the Maximum Likelihood Tree in a Phylogenetic Analysis of up to 150 Nucleotide Sequences. *Systematic Biology 28*(6), 988–1010.

Nagylaki, T. (1997). Multinomial-Sampling Models for Random Genetic Drift. *Genetics 145*(2), 485–491.

Neal, R. M. (2000). Markov Chain Sampling Methods for Dirichlet Process Mixture Models. *Journal of Computational and Graphical Statistics 9*(2), 249–265.

Neal, R. M. (2001). Annealed importance sampling. *Statistics and Computing 11*(2), 125–139.

Newton, M. A. and Raftery, A. E. (1994). Approximate Bayesian Inference with the Weighted Likelihood Bootstrap. *Journal of the Royal Statistical Society: Series B (Methodological) 56*(1), 3–48.

Nicholson, G., Smith, A. V., Jónsson, F., Gústafsso, O., Stefánsson, K., and Donnelly, P. (2002). Assessing population differentiation and isolation from single-nucleotide polymorphism data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 64*(4), 695–715.

Nobile, A. and Fearnside, A. T. (2007). Bayesian finite mixtures with an unknown number of components: The allocation sampler. *Statistics and Computing 17*(2), 147–162.

Ormerod, J. T. and Wand, M. P. (2010). Explaining variational approximations. *American Statistician 64*(2), 140–153.

Papastamoulis, P. and Iliopoulos, G. (2009). Reversible Jump MCMC in mixtures of normal distributions with the same component means. *Computational Statistics and Data Analysis 53*(4), 900–911.

Papastamoulis, P. and Iliopoulos, G. (2010). An Artificial Allocations Based Solution to the Label Switching Problem in Bayesian Analysis of Mixtures of Distributions. *Journal of Computational and Graphical Statistics 19*(2), 313–331.

Papastamoulis, P. and Iliopoulos, G. (2013). On the Convergence Rate of Random Permutation Sampler and ECR Algorithm in Missing Data Models. *Methodology and Computing in Applied Probability 15*(2), 293–304.

Paulin, D., Jasra, A., and Thiery, A. (2019). Error Bounds for Sequential Monte Carlo Samplers for Multimodal Distributions. *Bernoulli 25*(1), 310–340.

Persing, A., Jasra, A., Beskos, A., Balding, D. J., and De Iorio, M. (2015). A Simulation Approach for Change-Points on Phylogenetic Trees. *Journal of Computational Biology 22*(1), 10–24.

Phillips, D. B. and Smith, A. F. M. (1995). Bayesian Model Comparison Via Jump Diffusions. In W. R. Gilks, S. Richardson, and D. Spiegelhalter (Eds.), *Markov Chain Monte Carlo in Practice*, Chapter 13, pp. 215–239. Chapman & Hall.

Prangle, D., Everitt, R. G., and Kypraios, T. (2018). A rare event approach to high-dimensional approximate Bayesian computation. *Statistics and Computing 28*(4), 819–834.

Pritchard, J. K., Stephens, M., and Donnelly, P. (2000). Inference of Population Structure Using Multilocus Genotype Data. *Genetics 155*(2), 945–959.

R Core Team (2019). R: A Language for Data Analysis and Graphics.

Raftery, A. E. and Lewis, S. M. (1995). The Number of Iterations, Convergence Diagnostics and Generic Metropolis Algorithms. *Practical Markov Chain Monte Carlo 7*(98), 763–773.

Raftery, A. E., Newton, M. A., Satagopan, J. M., and Krivitsky, P. N. (2007). Estimating the Integrated Likelihood via Posterior Simulation Using the Harmonic Mean Identity. In J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith, and M. West (Eds.), *Bayesian Statistics 8*, pp. 1–45. Oxford University Press.

Rannala, B. and Yang, Z. (2003). Bayes Estimation of Species Divergence Times and Ancestral Populations Sizes Using DNA Sequences From Multiple Loci. *Genetics 164*(4), 1645–1656.

Reis, M. D. and Yang, Z. (2011). Approximate Likelihood Calculation on a Phylogeny for Bayesian Estimation of Divergence Times. *Molecular Biology and Evolution 28*(7), 2161–2172.

Richardson, S. and Green, P. J. (1997). On Bayesian Analysis of Mixtures with an Unknown Number of Components. *Journal of the Royal Statistical Society: Series B (Methodological) 59*(4), 731–792.

Robert, C. P. and Casella, G. (2004). *Monte Carlo Statistical Methods* (2nd ed.). Springer-Verlag New York.

Robert, C. P. and Casella, G. (2011). A Short History of Markov Chain Monte Carlo: Subjective Recollections from Data. *Statistical Science 26*(1), 102–115.

Roberts, G. O. and Rosenthal, J. S. (2001). Optimal Scaling for Various Metropolis-Hastings Algorithms. *Statistical Science 16*(4), 351–367.

Roberts, G. O. and Rosenthal, J. S. (2004). General state space Markov chains and MCMC algorithms. *Probability Surveys*, **1**, 20–71.

Roberts, G. O. and Rosenthal, J. S. (2009). Examples of Adaptive MCMC. *Journal of Computational and Graphical Statistics 18*(2), 349–367.

Rodríguez, C. E. and Walker, S. G. (2014). Label Switching in Bayesian Mixture Models: Deterministic Relabeling Strategies. *Journal of Computational and Graphical Statistics 23*(1), 25–45.

Roeder, K. (1990). Density Estimation with Confidence Sets Exemplified by Superclusters and Voids in the Galaxies. *Journal of the American Statistical Association 85*(411), 617–624.

Rosenberg, N. A. and Nordborg, M. (2002). Genealogical trees, coalescent, theory and the analysis of genetic polymorphisms. *Nature Reviews Genetics 3*(5), 380–390.

Rutschmann, F. (2006). Molecular dating of phylogenetic trees: A brief review of current methods that estimate divergence times. *Diversity and Distributions 12*(1), 35–48.

Särkkä, S. (2013). *Bayesian Filtering and smoothing*. Cambridge University Press.

Shao, S., Jacob, P. E., Ding, J., and Tarokh, V. (2018). Bayesian model comparison with the Hyvärinen score: computation and consistency. *Journal of the American Statistical Association*.

Sisson, S. A. (2005). Transdimensional Markov Chains: A Decade of Progress and Future Perspectives. *Journal of the American Statistical Association 100*(471), 1077–1089.

Stephens, M. (1999). Dealingwith multimodal posteriors and non-identifiability in mixture models. Technical report, University of Oxford.

Stephens, M. (2000a). Bayesian Analysis of Mixture Models with an Unknown Number of Components - An Alternative to Reversible Jump Methods. *The Annals of Statistics 28*(1), 40–74.

Stephens, M. (2000b). Dealing with Label Switching in Mixture Models. *Journal of the Royal Statistical Society: Series B (Methodological) 62*(4), 795–809.

Stephens, M. and Donnelly, P. (2000). Inference in molecular population genetics. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 62*(4), 605–635.

Suchard, M. A., Weiss, R. E., Dorman, K. S., and Sinsheimer, J. S. (2003). Inferring Spatial Phylogenetic Variation Along Nucleotide Sequences: A Multiple Change-point Model. *Journal of the American Statistical Association 98*(462), 427–437.

Tadesse, M. G., Sha, N., and Vannucci, M. (2005). Bayesian Variable Selection in Clustering High-Dimensional Data. *Journal of the American Statistical Association 100*(470), 602–617.

Takuno, S., Kado, T., Sugino, R. P., Nakhleh, L., and Innan, H. (2012). Population Genomics in Bacteria: A Case Study of *Staphylococcus aureus* . *Molecular Biology and Evolution 29*(2), 797–809.

Tanner, M. A. and Wong, W. H. (2010). From EM to Data Augmentation: The Emergence of MCMC Bayesian Computation in the 1980s. *Statistical Science 25*(4), 506–516.

Teh, Y. W., Daumé III, H., and Roy, D. (2008). Bayesian Agglomerative Clustering with Coalescents. *Advances in Neural Information Processing Systems*, 1473–1480.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2005). Sharing Clusters Among Related Groups : Hierarchical Dirichlet Processes. *Advances in Neural Information Processing Systems*, 1385–1392.

Tierney, L. (1994). Markov Chains for Exploring Posterior Distributions. *The Annals of Statistics 22*(4), 1701–1728.

Tokdar, S. T. and Kass, R. E. (2010). Importance sampling: a review. *Wiley Interdisciplinary Reviews: Computational Statistics 2*(1), 54–60.

Ulker, Y., Gunsel, B., and Cemgil, A. T. (2010). Sequential Monte Carlo Samplers for Dirichlet Process Mixtures. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 876–883.

Wang, L., Bouchard-côté, A., and Doucet, A. (2015). Bayesian Phylogenetic Inference using a Combinatorial Sequential Monte Carlo Method. *Journal of the American Statistical Association 110*(512), 1362–1374.

Whidden, C. and Matsen IV, F. A. (2015). Quantifying MCMC Exploration of Phylogenetic Tree Space. *Systematic Biology 64*(3), 472–491.

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.

Wu, B., Mcgrory, C. A., and Pettitt, A. N. (2012). A new variational Bayesian algorithm with application to human mobility pattern modeling. *Statistics and Computing 30*(3), 185–203.

Wu, C.-h., Suchard, M. A., and Drummond, A. J. (2012). Bayesian Selection of Nucleotide Substitution Models and Their Site Assignments. *Molecular Biology and Evolution 30*(3), 669–688.

Xie, W., Lewis, P. O., Fan, Y., Kuo, L., and Chen, M.-H. (2011). Improving Marginal Likelihood Estimation for Bayesian Phylogenetic Model Selection. *Systematic Biology 60*(2), 150–160.

Yang, Z. (2014). *Molecular Evolution: A Statistical Approach*. Oxford University Press.

Yang, Z. and Yoder, A. D. (2003). Comparison of Likelihood and Bayesian Methods for Estimating Divergence Times Using Multiple Gene Loci and Calibration Points , with Application to a Radiation of Cute-Looking Mouse Lemur Species. *Systematic Biology 52*(5), 705–716.

Young, B. C., Golubchik, T., Batty, E. M., Fung, R., Larner-Svensson, H., Votintseva, A. A., Miller, R. R., Godwin, H., Knox, K., Everitt, R. G., Iqbal, Z., Rimmer, A. J., Cule, M., Ip, C. L. C., Didelot, X., Harding, R. M., Donnelly, P., Peto, T. E. A., Crook, D. W., Bowden, R., and Wilson, D. J. (2012). Evolutionary dynamics of *Staphylococcus aureus during progression from carriage to disease. Proceedings of the National Academy of Sciences 109*(12), 4550–4555.

Zhang, Z., Chan, K. L., Wu, Y., and Chen, C. (2004). Learning a multivariate Gaussian mixture model with the reversible jump MCMC algorithms. *Statistics and Computing 14*(4), 343–355.

Zhou, Y., Johansen, A. M., and Aston, J. A. D. (2016). Towards Automatic Model Comparison: An Adaptive Sequential Monte Carlo Approach. *Journal of Computational and Graphical Statistics 25*(3), 701–726.