Sequential Monte Carlo methods for demographic inference

Donna Henderson
Lincoln College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy
Trinity 2017

Sequential Monte Carlo methods for demographic inference

Donna Henderson, Lincoln College

DPhil thesis, Trinity Term 2017

Abstract

Patterns of mutations in the DNA of modern-day individuals have been shaped by the demographic history of our ancestors. Inferring the demographic history from these patterns is a challenging problem due to complex dependencies along the genome. Several recent methods have adopted McVean's sequentially Markovian coalescent (SMC′) to model these dependencies [56, 49, 74]. However, these methods involve simplifying assumptions that preclude the inference of rates of migration between populations.

We have developed the first method to infer directional migration rates as a function of time. To do this, we employ sequential Monte Carlo (SMC) methods, also known as particle filters, to infer parameters in the SMC′ model. To improve the sampling from the state space of SMC′ we have developed a sophisticated sampling technique that shows better performance than the standard bootstrap filter. We apply our algorithm, SMC^2, to Neanderthal data and are able to infer the time and extent of migration from the Vindija Neanderthal population into Europeans.

With the large volume of sequencing data being produced from diverse populations, both modern and ancient, there is high demand for methods to interrogate this data. SMC^2 provides a flexible algorithm, which can be modified to suit many data applications. For instance, we show that our method performs well when the phasing of the samples is unknown, which is often the case in practice.

The long runtime of SMC^2 is the main limiting factor in the adoption of the method. We have started to explore ways to improve the runtime, by developing an adaptive online expectation maximisation (EM) procedure.

Acknowledgements

Firstly, I would like to express my gratitude to my supervisor Prof. Gerton Lunter for the support during my studies, and for his enthusiasm and extensive knowledge. His guidance helped me throughout the research and writing of this thesis. Moreover, his optimism consistently countered my own pessimism, and without it this thesis would not have come together.

I thank my fellow labmates, both old and new, for their limitless support. A special thanks to Joe Zhu for his training and advice throughout the project; I do not have the words to describe the importance of the skills and support Joe provided. I would like to thank Anna and Erika for the many long chats and pep talks. I would like express my appreciation to the Lunter Group more broadly for the walks, tea breaks, and games of table tennis. I have no doubt these distractions kept me on track and saved me from the much more destructive distraction of checking the news.

I thank my family for their constant faith in me and genuine interest in my research. Their patience and understanding were invaluable when I would get too wrapped up in work and forget to call. I would like to thank my friends both near and far. Those in the U.S. have kept me in touch with the world outside, encouraging me to visit and remember that my life exists outside of graduate school. Chelsey, Meri, and Radhika, I have not seen you nearly enough, but your reliable support has meant so much to me. My friends in Oxford have reminded me that I have hobbies and interests beyond statistics, and I am certain regular games nights did wonders for my happiness over the last four years. Which brings me to my partner Phil, whose dedication to his interests has always been an inspiration, and who has ensured I continue to eat well throughout the manic thesis write up. I am truly grateful to you all.

Contents

Chapter 1

Introduction

The genomes in a population encode a vast amount of data on not only the individuals they help define, but thousands of ancestors of those individuals. Patterns exist in the DNA of modern-day individuals which provide insight into population dynamics going back tens of thousands of generations. Strikingly, the two genomes from a single diploid European individual reveal the population size bottleneck that occurred when anatomically modern humans first left Africa [49].

Recent improvements in sequencing technology have resulted in a deluge of genomic data. This data has the ability to reshape our understanding of the history of our species. To this end, researchers have developed dozens of methods to model the processes that led to modern-day genomes. By modelling ancestry these methods are able to infer the historic demography of populations.

We add to this field by developing a novel method for demographic inference with a focus on rates of migration between populations through time. Our method stands apart from the others by explicitly modelling the migration process on the coalescent with recombination. This provides estimates of the directional migration rates as a function of time, which elude other methods due to their various simplifying assumptions.

In this thesis I will describe our inference method, including novel sampling and inference techniques we have developed within the context of sequential Monte Carlo. I start with a brief description of the mathematical models underlying our algorithm, a review of the field of demographic inference, and a brief summary of sequential importance sampling with resampling. Chapter 2 describes our method, SMC^2, in detail and provides results from analyses on simulated data. Chapter 3 provides an application of our method to human and Neanderthal data, where we detect signals of migration from Neanderthals to the non-African population at the time of the Out of Africa event. Chapter 4 details an adaptive version of online expectation maximisation which we developed with the intention to decrease the runtimes of our algorithm.

1.1 Modelling ancestry

1.1.1 The coalescent

Demographic inference from genomic data relies on the passing of DNA from parent to offspring and the mutations that arise during this process. The core mathematical model of this process is Kingman's coalescent [46]. This model considers a single locus of n haploid indiviuals under the following assumptions:

- the samples are from a single panmictic population
- this population has a constant population size through time
- the haplotypes come from a neutral region of the genome

We will assume our haploid samples come from a diploid population with effective population size $2N_e$. At the present time each sample will form a distinct lineage, however at some set of times in the past each pair of lineages will coalesce due to

a common ancestor. This genealogy, or tree, defines the relatedness of the modern-day individuals at this locus. The process of coalescing lineages can be framed as a continuous time Markov process operating backwards in time.

The transition probabilities of the coalescent Markov process are defined by two independent components. First, a continuous-time death process which describes the decreasing number of unique lineages. Second, a discrete-time embedded Markov chain which describes the ordering of coalescence with respect to the lineage indices. The first process determines when coalescences occur, the second determines which two of the remaining lineages coalesce.

When time is scaled by the effective population size $2N_e$ the transition rate of the death process is

$$q_k = \binom{k}{2},$$

where k is the number of unique lineages in the current state. In this scaling, the time spent in the state with k lineages has density

$$\binom{k}{2} \exp\left(- \binom{k}{2} t \right) \qquad \text{for} \quad t > 0.$$

The discrete-time jump chain is defined by a transition from k to $k-1$ distinct lineages, where all possible pairs are equally likely to coalesce.

In addition to the coalescent process, there is a mutational process acting on the lineages (see Figure 1.1). As DNA is passed from parent to offspring there is a chance of mutation. This is modeled on the coalescent by a Poisson process with rate $\frac{\Theta}{2} \cdot l$ where $\Theta = 4N_e\mu$ and l is the length of the locus. The mutation process on the coalescent provides a link between observable DNA sequences and the demographic history which shapes the coalescent. Hence, observed mutations can be used to infer N_e.

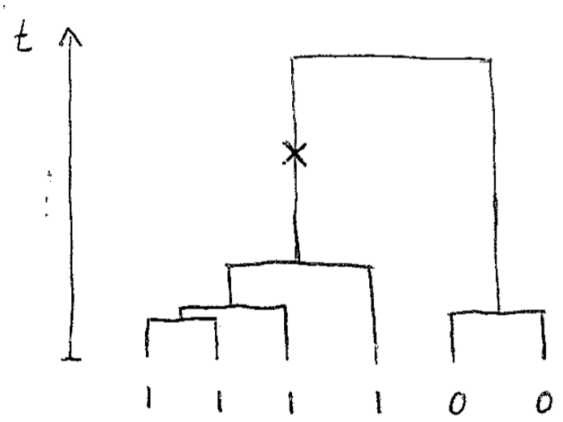


Figure 1.1: A mutation occurring on an ancestral lineage affects the mutation pattern of modern samples.

1.1.2 The structured coalescent

So far we have imposed rather strict constraints on the demography. Notably, assuming the individuals are from a single panmictic population with constant effective population size. The assumption of constant population size of $2N_e$ can easily be relaxed to an assumption of piecewise constant [16, 20]. The driving purpose of this work is to infer migration rates between populations. To do this, we must relax the assumption of all lineages belonging to a single panmictic population, and instead consider multiple panmictic populations. Now at time t each lineage $i \in 1, \ldots, k_t$ has an associated population pop_i. The Markov process can now transition either by a coalescence of two lineages, as before, or by a change of pop_i for some lineage i. The model for coalescences has also changed, as we only allow lineages within the same population to coalesce. If there are k_p lineages in population p and population p has effective population size $N_e^{(p)}$, then the rate of coalescence among these lineages is $\binom{k_p}{2} \cdot \frac{N_e}{N_e^{(p)}}$, and so the total rate of a transition due to coalescence is

$$q_{\text{coal}} = \sum_{p=1}^{P} \binom{k_p}{2} \cdot \frac{N_e}{N_e^{(p)}},$$

where P is the number of populations and N_e is the sum of $N_e^{(p)}$. The transition of lineage $\{i,\ pop_i = p\}$ to $\{i,\ pop_i = p'\}$ occurs at rate $m_{p,p'}$, where $m_{p,p} = 0$. We follow the convention that $m_{p,p'}$ describes the migration process in coalescent time, i.e. backwards in natural time [16]. We define our migration rate $m_{p,p'} = \frac{N_e^{(p')}\gamma_{p',p}}{N_e^{(p)}}$ where $\gamma_{p',p}$ is defined as the proportion of individuals born in population p' who migrate to population p. Thus, the total transition rate is

$$\begin{aligned} q_{k_1,\ldots,k_P} &= q_{\text{coal}} + q_{\text{migr}} \\ &= \left(\sum_{p=1}^{P} \binom{k_p}{2} \cdot \frac{N_e}{N_e^{(p)}}\right) + \left(\sum_{i=1}^{k}\sum_{p'=1}^{P} m_{pop_i,p'}\right). \end{aligned} \tag{1.1}$$

The structured coalescent models a specified number of populations through time. In reality, populations are often structured which violates the model assumption of panmixia within each population. Mazet et al. show that inferred population expansions or contractions can be artifacts of unmodelled population structure [55]. This caveat will apply to all of our results; we infer effective population sizes which may differ from census population sizes for a variety of reasons, including violations of the panmixia assumption.

1.1.3 The coalescent with recombination

For many species of interest, including humans, a single locus is not particularly informative of demography. The variance in genealogy given N_e is high, as is the variance in mutation pattern given the genealogy. The latter is less of a concern for large loci as the mutation rate is proportional to the length of the locus. However the coalescent model breaks down if we consider an arbitrarily long sequence of the autosomal genome. This is due to historic chromosomal crossover events, called crossover recombination. Recombination is a meiotic process where homologous maternally inherited and paternally inherited regions of the chromosome are exchanged, so the chromo-

some passed to the offspring will be partially from its grandmother and partially from its grandfather. To model ancestry in a region where recombination has occurred, a single genealogy does not suffice as a lineage can have multiple ancestors and hence can split as it is traced back through time. This problem led to the development of the coalescent with recombination [39, 30, 31].

The coalescent with recombination is similar to Kingman's coalescent in that they are both backward in time processes with mutation occurring on lineages and coalescence occurring between pairs of lineages. The coalescent with recombination has the additional process of a single lineage splitting. This split reflects the fact that some portion of an individual's haplotype was inherited from its grandmother and the rest from its grandfather due to a recombination event. These two inherited sequences have independent ancestors, conditional on having been drawn from the same population. Recombination is modeled as a Markovian process with rate $4N_e\rho l$ per lineage. When a splitting occurs, the position of the recombination, u, is uniformly sampled from the interval $(0, l)$. For that lineage, the genetic material to the left of this position was inheritted from a different ancestor than the genetic material to the right of this position. With this additional process, the model is no longer a tree, but a graph (see Figure 1.2). The ancestral recombination graph (ARG) is of a much higher dimension than the genealogical tree of a single locus, and hence more challenging to model. However, the ARG is far more informative of demography than a single genealogy. Genealogies are highly stochastic and as such a single inferred tree, say from the non-recombining mitochondrial DNA, provides little information on the underlying demographic parameters. On the other hand, a set of genealogies from along the nuclear genome, provide sufficient data to infer the N_e curve.

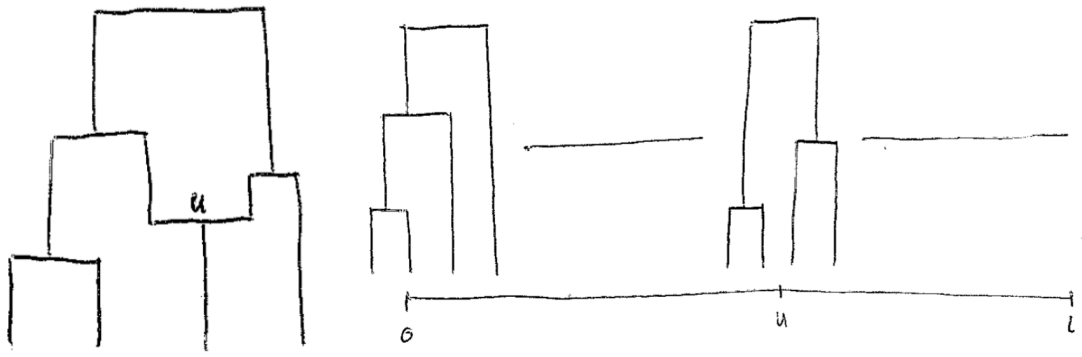


Figure 1.2: The ancestral recombination graph and its decomposition into a sequence of trees.

1.1.4 The sequentially Markovian coalescent

So far we have considered only the Markovian structure of the backwards in time process. If we put one major restriction on the ARG, we can ensure a Markovian structure along the sequence as well, which will reduce the size of the state-space [56]. The restriction is to disallow coalescence between lineages which do not share ancestral material. Recall that when a lineage splits, one resulting lineage will match genetic material over the positions $(0, u)$ and the other lineage will match over (u, l). Thus, lineage i will contain genetic material ancestral to the sample(s) over some non-overlapping intervals $\mathbf{x_i}$. The restriction states lineages i and j may only coalesce if $\mathbf{x_i} \cap \mathbf{x_j} \neq \varnothing$.

With the additional restriction, we can model the ARG using the sequentially Markovian coalescent (SMC′). The SMC′ is produced by simulating a coalescent tree at genome position 0. If the samples are from multiple populations, this coalescent will be structured. The genealogy is extended along the sequence according to a Markov process with transition rate $4N_eB_s$ where B_s is the total branch length of the tree at position s. When a transition occurs, a recombination is sampled uniformly on the branches. The Markovian backwards in time process is used to resolve the recombination. From the recombination point, which we will say has been sampled

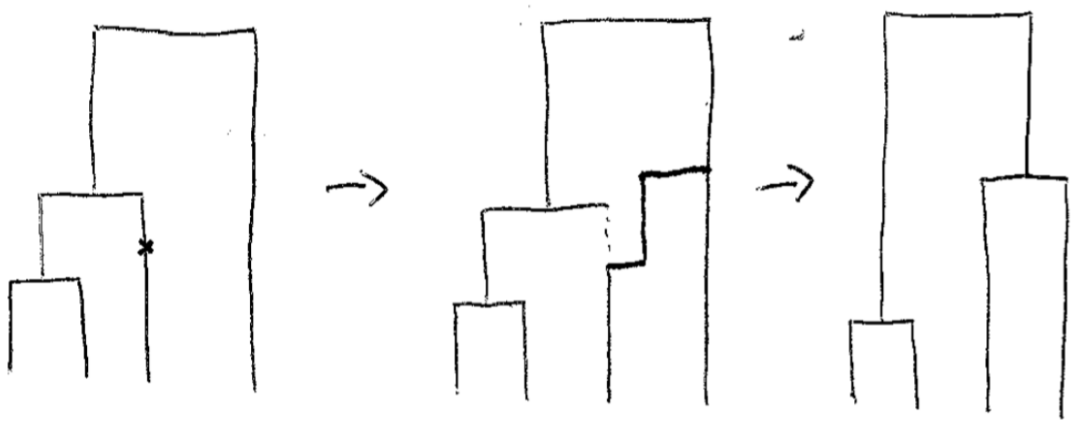


Figure 1.3: A recombination causes a branch movement.

on lineage i in population p, the lineage is evolved backwards in time according to a coalescence and migration process. The resulting transition rate is

$$q_i = q_{\text{coal}} + q_{\text{migr}} \tag{1.2}$$

$$= \left(k_p \cdot \frac{N_e}{N_e^{(p)}}\right) + \left(\sum_{p'=1}^{P} m_{p,p'}\right). \tag{1.3}$$

If a migration occurs first, the process of evolving the lineage continues with parameters appropriate for its new population. Once a coalescence occurs the process is completed, and the broken branch which was previously above the recombination point is removed as it no longer relates to the samples (see Figure 1.3).

Note that the evolving lineage is able to coalesce into the branch it broke from, resulting in no change to the genealogy. This is the feature which distinguishes SMC′ [53] from its original form SMC [56]. The SMC′ has been shown to be a good approximation of the ARG [81].

Now that we have covered many of the basics for modelling the ancestry of genomic data, we will take a broader look at how demographic parameters are inferred.

1.2 Existing methods for demographic inference

Demographic inference methods broadly fall into two categories: those that do not consider linkage, and those that do. Generally, methods of the former type are able to handle data from a large number of samples, whilst methods of the latter type are restricted to fewer samples due to increased complexity of the underlying model.

1.2.1 Unlinked loci

1.2.1.1 SFS based methods

Central to the use of large numbers of samples is the simplification of data to summary statistics. The site frequency spectrum (SFS) is the most widely used statistic for summarising sequence data of populations. The SFS is a matrix representing the derived allele count across populations. Each dimension of the matrix corresponds to a population. In an example with three populations, element $M_{i,j,k}$ is the number of variants where the derived allele is observed in i individuals from population 1, j individuals from population 2, and k individuals from population 3. Hence the SFS summarises sharing of derived alleles between populations.

Many methods make use of the SFS to assess the viability of certain demographic models. The poster child of this methodology is diffusion approximation demographic inference ($\delta a \delta i$) [33]. $\delta a \delta i$ has the ability to infer population size changes, split times, and migration between populations for up to three populations. A similar method *momi* is able to handle a large number of populations, but relies on a less flexible model and therefore cannot infer continuous migration [42].

Another noteworthy method which relies on the SFS is $fastsimcoal2$ [21]. This method estimates the likelihood of a proposed demographic model by simulating coalescent trees under the model and creating a SFS for the proposed model. This method was shown to outperform $\delta a \delta i$ for inference of sufficiently complex demo-

graphic models including multiple populations with migration.

There is much interest in the field in the potential and limitations of inferring demography using the SFS. There are many reasons that working with a single matrix rather than genome data itself is appealing, including processing speed, data storage, and scope of samples. However, the reduction in data does potentially have a major cost. Myers et al. showed that using only the SFS of unlinked loci to infer demography results in non-uniqueness of solutions in some cases [59]. In particular they give an example of a demography where a large bottleneck cannot be inferred as the resulting signal in the SFS can be explained by an alternative population size history. On the other hand, Bhaskar and Song argue that under the common assumption of piecewise-defined population sizes and with a sufficient number of samples, analysis of the expected SFS does lead to a unique solution [2].

1.2.1.2 MCMCcoal and G-PhoCS

An alternative data source to the SNP data used by the SFS is a set of approximately independent haplotypes from across the genome. MCMCcoal jointly infers population sizes and split times using haplotypes at unlinked loci [70, 5]. G-PhoCS was derived from MCMCcoal and is additionally able to infer asymmetric migration at user-specified time bands [32]. G-PhoCS simulates a coalescent tree for each loci under an initial demographic model. Metropolis-Hastings is then used to iteratively update the locus-specific genealogies and the demographic parameter estimates. The main limitation of this method is that the model is restricted to piece-wise constant parameters over large epochs, e.g. a constant migration rate since the split of two populations. This is done to keep the size of the parameter space down and reduce the number of necessary simulations. G-PhoCS possesses two nice features missing from many related methods; it can work with unphased data and infer the age of a sample. Many haplotype-based methods are restricted by their need for perfectly phased

haplotypes. G-PhoCS avoids potential bias due to imperfect phasing by integrating over all possible phasings, at a reasonable computational cost. For applications with ancient samples, the ability to infer sample ages can provide a huge advantage.

The above methods avoid the complexity of the recombination process by treating loci as independent. In most cases, the loci must be selected to be positioned far away from each other and small enough that no recombination occurs within each locus. The former condition somewhat limits the useable data as large sections of the genome must be thrown away. Fortunately, for species like humans, the genome is large enough that this loss of data is not restrictive. The latter condition is more prohibitive as loci must be fairly small to designate them non-recombining. If the loci are chopped too small, they will be uninformative of demography as insufficient numbers of mutations will occur within the regions. A preliminary step to all unlinked methods is the selection of these loci and the accompanying balancing of data quantity and model assumption validity.

1.2.2 Model positional dependence

In order to fully utilise the available data, many methods attempt to retain linkage information. This can be thought of as analysing each site of the genome in the context of its surrounding sites. The crux of the problem is determining what qualifies as surrounding, which is complexly intertwined with the recombination history.

1.2.2.1 IBD and IBS

Identity by descent (IBD) methods form an intriguing alternative to the previously mentioned methods. Whilst the unlinked-data methods avoid recombination boundaries, IBD methods base their inference on these recombination boundaries. The distribution of IBD lengths can be used to infer the distribution of times to the most recent common ancestor (TMRCAs), which are informative of the demographic pa-

rameters. These methods perform particularly well in the recent past as they are aware of long-range correlations which are ignored by the methods in Section 1.2.1 [63].

The main challenge of IBD methods for demographic inference is to identify the IBD segments [4]. Harris and Nielsen recently developed an approach which is similar in spirit, but uses identity by state (IBS) tracts which are more straightforward to obtain [36]. The IBS method considers a wide range of tract sizes which enables inference in both the recent and distant past. The summary statistic IBS, like IBD, captures some but not all of the information present in linkage data. All pairs of haplotypes are considered and a histogram of lengths of identical by state blocks is created. This distribution of lengths of identical tracts is descriptive of recombination and hence tree size. The IBS summary statistic retains information on the linkage only within the tracts. There is additional information in the sequence data which is discarded when neighbouring tracts are treated as independent.

IBD and IBS analysis leverages linkage to a certain extent, but strong correlations exist between regions separated by recombination. To utilize these correlations, we must shift our thinking away from single genealogies and toward the ancestral recombination graph (ARG). Whereas previously mentioned methods, such as G-PhoCS, sought to infer over possible genealogies, the methods we now consider seek to infer over possible ARGs.

1.2.2.2 PSMC

Unquestionably the most widely used method for inferring population size history by modelling the ARG is the pairwise sequentially Markovian coalescent (PSMC) [49]. PSMC approximates the ARG by a Markovian sequence of genealogies, one genealogy for each site in the genome. With this Markovian approximation as well as the discretisation of the time dimension, the problem is reduced to a hidden Markov model

(HMM). In the PSMC HMM the sequence data at a genomic position is the observed variable and the underlying genealogy for that position is the unobserved variable. In order to keep the state space of the unobserved variables computationally manageable, this particular formulation is restricted to comparing only two haploid individuals (or more commonly, the two haplotypes within a single diploid individual, in which case phasing information is not required). With this restriction, the unobserved variable, the genealogy, is simply a positive real number representing the TMRCA of the samples at that site. In this HMM framework, the forward backward equations are employed to infer maximum likelihood estimates (MLEs) of the historic population size.

1.2.2.3 MSMC

Whilst PSMC has been embraced by the community for population size inference, its restriction to two samples renders it unusable for analyses involving multiple interacting populations. The extension of PSMC to multiple samples (MSMC) addresses this issue and enables the inference of the effective sizes of multiple populations as well as a measure of the mixing of those populations [74]. MSMC simplifies the problem of modelling the genealogies relating multiple individuals by instead modelling the TMRCA of any two of the samples along with the indices of the two samples involved. This method can handle data from up to four diploid individuals, however the inclusion of many samples results in smaller times to the most recent coalescence. An important consequence of this is that power for inference in the distant past is decreased as more samples are added to the analysis. To an extent, this can be circumvented by running the analysis over subsets of the samples and combining the results.

MSMC provides MLEs of the effective population sizes as well as the cross-coalescent rate for the populations. This cross coalescent rate estimate is particularly

exciting as it provides some insight into the relationship of populations at different points in time. Unfortunately, this insight is limited. The cross-coalescent rate provides a glimpse at migration history, but no detail. If we seek information on the magnitude and direction of migration between populations, we must model more than this most recent coalescence event.

1.2.2.4 SMC++

An alternative to modelling the first coalescence between any two of the total pool of samples is to select two samples and use summary statistics, like the SFS, to inform the coalescent time of the chosen samples. This is what SMC++ does [80]. Leveraging information from hundreds of samples gives SMC++ higher resolution in the recent past than those ARG-modelling methods which are restricted to a few samples.

I introduced this section as a set of models which model the ARG, but thus far have only detailed methods which model a simplified version of the ARG. This is because modelling the ARG is an incredibly ambitious task. The size of the ARG state space is massive, even with the reduction in size due to the SMC$'$ restriction (as detailed in Section 1.1.4). We now explore a method which, as far as I have seen, models the structured ARG with fewer simplifications than any other method.

1.2.2.5 diCal

Whilst MSMC can analyse multiple samples, for practical reasons it is restricted to about eight haploid individuals. This is a result of state space size restriction as well as the choice of hidden state. An alternate method Demographic Inference using Composite Approximation Likelihood (diCal) considers the entire genealogy and therefore can handle more samples without any loss of information [76, 78, 79]. diCal iterates over the samples modelling the lineage absorption into a trunk genealogy which is an approximation of the genealogy relating all other samples. The main drawback of

using the trunk genealogy approximation is that inference of migration rates then relies on symmetry of migration and thus models with asymmetric migration can easily derail the inference [78].

1.2.2.6 ARGweaver

A promising method for modelling the ARG builds up the graph by weaving in each sample, hence the name ARGweaver [71]. This has been shown to infer ARGs with dozens of samples using MCMC. The inferred ARGs can be used to identify regions of interest in the genome, such as those under natural selection. Unfortunately, in its current implementation ARGweaver does not have sufficient resolution for inferring the N_e and migration curves. The method could be extended to incorporate a complex demographic model, but the computation time would be impractical. To date, the potential of ARG inference for inferring demographic parameters has not been fully harnessed.

1.3 The Sequential Importance Sampling with Resampling algorithm

We employ Monte Carlo and sequential importance sampling methods to tackle the problem of modelling the ARG. The details of applying these approaches to ARGs will be covered in Chapter 2, but first it is useful to review these methods in a general context.

1.3.1 Sequential Importance Sampling

There are many situations where sampling directly from a distribution of interest is infeasible. The common solution to this problem is importance sampling where a sample $\{X^{(i)}\}_{i=1,\ldots,N}$ is drawn from an alternative distribution, called the instrumental

distribution (in our case, $\{X^{(i)}\}$ will be an ARG). The samples generated using the instrumental distribution are reweighted to reflect the distribution of interest. Denote the target distribution by $p(X)$ and define an instrumental distribution $q(X)$ with the same support, i.e. $q(x) > 0$ for all x such that $p(x) > 0$. Then drawing a sample $\{X^{(i)}\}_{i=1,\dots,N}$ from $q(\cdot)$ provides the approximation

$$\hat{p}(X) = \sum_{i=1}^{N} w(X^{(i)}) \cdot \delta(X^{(i)}),$$

where $w(X^{(i)}) = p(X^{(i)})/q(X^{(i)})$ and δ is the Dirac delta function.

We focus on models with a sequential structure. In particular, we will consider general state-space models which are comprised of a sequence of unobserved variables $\{X_k\}_{k=1,\dots,t} := X_{1:t}$ and a sequence of observed variable $Y_{1:t}$. Here the unobserved sequence has a Markovian structure and the observed variables are independent conditional on the unobserved sequence. The model can be fully defined by the transition distribution $f(x_k|x_{k-1})$ and emission distribution $g(y_k|x_k)$. The parameters of the transition and emission distribution are discussed in Section 1.3.4.

The goal is to create a sample of particles $\{X_{1:k}^{(i)}\}_{i=1,\dots,N}$ to approximate the posterior distribution

$$p(x_{1:t}, y_{1:t}) = \mu(x_1) \cdot \prod_{k=2}^{t} f(x_k|x_{k-1}) \cdot \prod_{k=1}^{t} g(y_k|x_k).$$

These methods are often referred to as particle smoothing methods, or particle filters if the marginal distributions $\{p(x_k|y_{1:k})\}_{k=1,\dots}$ are the distributions of interest. Defining an instrumental distribution $q(X_{1:t})$ which can be sampled from directly can be challenging, so constructing the sample using a sequence of instrumental distributions $q_k(x_k|x_{k-1})$ is recommended (Algorithm 1.1).

Algorithm 1.1 Sequential Importance Sampling

for $k = 1$ **do**
  for $i = 1, \ldots, N$ **do**
    Sample $X_1^{(i)} \sim q_1(\cdot)$
    $w_1(X_1^{(i)}) = p_1(X_1^{(i)}|y_1)/q_1(X_1^{(i)})$
  end for
  $W_1^{(i)} \propto w_1(X_1^{(i)}), \sum_{i=1}^N W_1^{(i)} = 1$
end for
for $k = 2, ..., t$ **do**
  for $i = 1, \ldots, N$ **do**
    Sample $X_k^{(i)} \sim q_k(\cdot|X_{k-1}^{(i)})$
    $X_{1:k}^{(i)} = (X_{1:k-1}^{(i)}, X_k^{(i)})$
    $w_k(X_{1:k}^{(i)}) = W_{k-1}^{(i)} \cdot p_k(X_k^{(i)}|X_{k-1}^{(i)}, y_k)/q_k(X_k^{(i)}|X_{k-1}^{(i)})$
  end for
  $W_k^{(i)} \propto w_k(X_{1:k}^{(i)}), \sum_{i=1}^N W_k^{(i)} = 1$
end for

1.3.2 Resampling

The sequential importance sampling (SIS) procedure is inefficient for large t. As the sample is sequentially constructed the majority of the particle weights will go to 0; this is the problem of particle weight degeneracy. A partial solution to this problem was proposed by Gordon et al. [28]. By resampling the particles proportional to their weights, and then setting all of the weights to $1/N$, the algorithm avoids relying on a single particle.

Resampling at every step k is often computationally inefficient and can needlessly increase the stochastic variability. For this reason it is common practice to resample only when the weights of the particles become skewed. The skew of the particles is measured by the effective sample size, defined as $ESS = [\sum_{i=1}^N (W_k^{(i)})^{-2}]^{-1}$. The particle resampling step is then implemented only when the ESS drops below a chosen threshold, typically $N/2$ (Algorithm 1.2).

The SISR procedure prevents constructing an inefficient sample where many particles have tiny densities under the target distribution. However resampling introduces another type of inefficiency where many of the particles take the same values over

Algorithm 1.2 Sequential Importance Sampling with Resampling (SISR)

for $k = 1$ **do**
  for $i = 1, \ldots, N$ **do**
    Sample $X_1^{(i)} \sim q_1(\cdot)$
    $w_1(X_1^{(i)}) = p_1(X_1^{(i)}|y_1)/q_1(X_1^{(i)})$
  end for
  $W_1^{(i)} \propto w_1(X_1^{(i)}), \sum_{i=1}^{N} W_1^{(i)} = 1$
  if $ESS \leq N/2$ **then**
    Resample $\{W_1^{(i)}, X_1^{(i)}\}$ to obtain N equally-weighted particles
    $\{W_1^{(i)} = \frac{1}{N}, \bar{X}_1^{(i)}\}$
  else
    $\{W_1^{(i)}, \bar{X}_1^{(i)} = X_1^{(i)}\}$
  end if
end for
for $k = 2, ..., t$ **do**
  for $i = 1, \ldots, N$ **do**
    Sample $X_k^{(i)} \sim q_k(\cdot|\bar{X}_{k-1}^{(i)})$
    $X_{1:k}^{(i)} = (\bar{X}_{1:k-1}^{(i)}, X_k^{(i)})$
    $w_k(X_{1:k}^{(i)}) = W_{k-1}^{(i)} \cdot p_k(X_k^{(i)}|X_{k-1}^{(i)}, y_k)/q_k(X_k^{(i)}|X_{k-1}^{(i)})$
  end for
  $W_k^{(i)} \propto w_k(X_{1:k}^{(i)}), \sum_{i=1}^{N} W_k^{(i)} = 1$
  if $ESS \leq N/2$ **then**
    Resample $\{W_k^{(i)}, X_{1:k}^{(i)}\}$ to obtain N equally-weighted particles
    $\{W_k^{(i)} = \frac{1}{N}, \bar{X}_{1:k}^{(i)}\}$
  else
    $\{W_k^{(i)}, \bar{X}_{1:k}^{(i)} = X_{1:k}^{(i)}\}$
  end if
end for

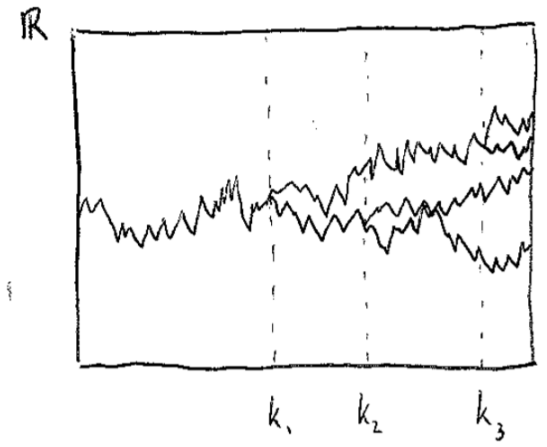


Figure 1.4: An example of shared particle trajectories due to resampling at positions k_1, k_2, and k_3. At step t of the algorithm, the four particles $X_{0:t}^{(1)}$, $X_{0:t}^{(2)}$, $X_{0:t}^{(3)}$, and $X_{0:t}^{(4)}$ are the results of particle extension and resampling. Here $X_{0:k_1}^{(i)} = X_{0:k_1}^{(j)}$ for all i, j due to the resampling that has occurred.

the early dimensions of the state space (see Figure 1.4). This lack of independence between particles is referred to as sample impoverishment.

1.3.3 Fixed lag approximation

For models with good forgetting properties the problem of sample impoverishment can be lessened. In such models there exists Δ such that

$$p(X_{0:k}|Y_{0:t}) \approx p(X_{0:k}|Y_{0:min(k+\Delta,t)}). \tag{1.4}$$

This approximation suggests independence of particles can be maintained by fixing the k-th component of the particles after $k + \Delta$ steps. However, this procedure does introduce an asymptotic bias as $p(x_{0:k}|y_{0:t}) \neq p(x_{0:k}|y_{0:min(k+\Delta,t)})$, even as $N \to \infty$ [43].

Moreover, the forgetting rate Δ of the model is often unknown and must be approximated by the user. Choosing a lag L too large will not solve the particle dependency issue. Choosing L too small will result in a poor approximation in equa-

tion (1.4).

1.3.4 Parameter inference using SISR

The methods discussed thus far have assumed the transition and emission distributions are known. In fact, we are creating a Monte Carlo sample in order to infer parameters θ of the posterior distribution

$$p_\theta(x_{1:t}, y_{1:t}) = \mu_\theta(x_1) \cdot \prod_{k=2}^{t} f_\theta(x_k|x_{k-1}) \cdot \prod_{k=1}^{t} g_\theta(y_k|x_k).$$

In our application of modelling the ARG X, the only parameter of the emission distribution is the mutation rate μ which is assumed to be known. However, the parameters of the transition distribution are unknown. These unknown parameters are the recombination rate ρ, the effective population size N_e for each population and each epoch, and migration rates $M_{p,p'}$ for each ordered pair of populations and each epoch. A variety of techniques could be used to estimate the unknown parameters.

Kantas et al. detail a few approaches to the problem of parameter inference using SISR [43]. We want to use the genomic sequence dependencies of the ARG to infer the parameters. These dependencies can span a long distance, and so the number of genomic positions modelled, t, will need to be large. Online inference methods are preferred for large t, as a good approximation of $p_\theta(x_{1:t}, y_{1:t})$ cannot be obtained due to sample degeneracy and sample impoverishment. The choice to use an online method precludes the popular class of techniques particle MCMC. There remains a choice between employing maximum likelihood or Bayesian parameter estimation. We opt for an online maximum likelihood estimation procedure.

The marginal likelihood of our observed data takes the form

$$p_\theta(y_{0:t}) = \int p_\theta(x_{0:t}, y_{0:t}) dx_{0:t}.$$

The standard Expectation-Maximisation (EM) algorithm finds the maximum likelihood estimate (MLE) by alternating between calculating the expectation of the likelihood with respect to the hidden variables as a function of θ (the Q-function) and setting $\hat{\theta}$ to the parameter which maximises this function.

Expectation step (E step)

$$Q(\theta|\hat{\theta}_j) = \mathbb{E}_{X|Y,\hat{\theta}_j}[\log L(\theta; X, Y)]$$

Maximisation step (M step)

$$\hat{\theta}_{j+1} = \underset{\theta}{\operatorname{argmax}} Q(\theta|\hat{\theta}_j)$$

In the cases we consider

$$\begin{aligned} Q(\theta|\hat{\theta}_j) &= \int \log p_\theta(x_{0:t}, y_{0:t}) p_{\hat{\theta}_j}(x_{0:t}|y_{0:t}) dx_{0:t} \\ &= \mathbb{E}_{\hat{\theta}_j}[\log \mu_\theta | y_{0:t}] + \sum_{k=2}^{t} \mathbb{E}_{\hat{\theta}_j}[\log f_\theta(X_k|X_{k-1})|y_{0:t}] + \sum_{k=1}^{t} \mathbb{E}_{\hat{\theta}_j}[\log g_\theta(y_k|X_k)|y_{0:t}], \end{aligned}$$

and the E step cannot be solved analytically. Instead, we must use Monte Carlo approximations. In fact, we employ stochastic approximation EM (SAEM) rather than the more typical Monte Carlo EM (MCEM) as MCEM is comparatively computationally inefficient [9, 8]. This choice is discussed further in Chapter 4.

In certain situations, primarily when $p_\theta(x_{0:t}, y_{0:t})$ belongs to the exponential family, the M step can be done using summary statistics. Essentially, the features of the posterior distribution which affect $Q(\theta|\hat{\theta}_j)$ can be extracted in a step-wise fashion and these can be used to calculate $\hat{\theta}_{j+1}$ which maximises $Q(\theta|\hat{\theta}_j)$. Define additive functionals of the form

$$S_t^{(l)}(x_{0:t}, y_{0:t}) = \sum_{k=2}^{t} s(x_k, x_{k-1}, y_k).$$

Then the summary statistics are

$$S^{(l)}_{\hat{\theta}_j,t} = \mathbb{E}_{X_{0:t}|Y_{0:t},\hat{\theta}_j}[S^{(l)}_t(x_{0:t}, y_{0:t})]. \tag{1.5}$$

The parameter estimate is updated by a known function $\Lambda()$ mapping from the summary statistics to the parameter which maximises $Q(\theta|\hat{\theta}_j)$

$$\hat{\theta}_{j+1} = \Lambda\Big((t-1)^{-1} \cdot \{S^{(l)}_{\hat{\theta}_j,t}\}_{l=1,\ldots,n}\Big).$$

In practice, we cannot compute (1.5) directly, and so we will use Monte Carlo approximations of the summary statistics obtained through SISR. The simplest approach to this approximation would be

$$\hat{S}^{(l)}_{\hat{\theta}_j,t} = \sum_{i=1}^{N} w_t(X^{(i)}_{0:t}) \sum_{k=2}^{t} s(X^{(i)}_k, X^{(i)}_{k-1}, y_k).$$

However, sample impoverishment would lead to a high variance for this approximation. We use an alternative approximation using the fixed lag technique proposed by Cappé and Moulines

$$\hat{S}^{(l)}_{\hat{\theta}_j,t} = \sum_{i=1}^{N} \Big(\sum_{k=2}^{t-L} w_{k+L}(X^{(i)}_{0:k+L}) s(X^{(i)}_k, X^{(i)}_{k-1}, y_k) + \sum_{k=t-L+1}^{t} w_t(X^{(i)}_{0:t}) s(X^{(i)}_k, X^{(i)}_{k-1}, y_k) \Big). \tag{1.6}$$

The summary statistics can be calculated in an online manner because of their additive functional form. This provides a substantial practical benefit when t is large as it reduces the required computational memory needed to store particles. When using the fixed-lag technique discussed earlier, the components of a particle that have already contributed to the summary statistics can be discarded.

The online computation of summary statistics naturally lends itself to online updates of the parameter estimates. We will explore this subject more thoroughly in

Chapter 4. For now, we will use batch EM (BEM) which is an online parameter estimation procedure where the summary statistics are updated as described above and the parameter estimate is updated once a pre-specified amount of data has been processed.

In practice, we often process hundreds of megabases of genomic data per EM iteration. To increase speed we take advantage of the additive funcional form of the summary statistics. The data is divided into 20Mb chunks, each of which is processed on its own CPU in parallel. The approximation in (1.6) is obtained by summing the statistics of the chunks. This technique equates to a composite likelihood approach.

1.4 Discussion

It is widely accepted that genomes contain enough information to infer recent demographic history. However current methods are somewhat limited when it comes to inferring complex demographic histories. In particular, histories with time-varying asymmetric migration present a challenge. We endeavoured to develop a flexible method which is capable of inferring migration rates by modelling the ARG.

We apply sequential Monte Carlo methods to the problem of estimating the posterior distribution of ARGs. This work was started in 2013 by Gerton Lunter and Joe Zhu. Section 2.1 details the basics of this method and the starting point of this thesis. The remainder of Chapter 2 describes my contribution to the algorithm, including the development of a calibration technique for the fixed-lag and adjustments to the sampler to improve efficiency. The simulation results presented confirm the ability of SMC^2 to infer demographic parameters in a range of demographic models.

Chapter 3 details applications of our new method to real data, with a focus on inferring Neanderthal history. We explore the relationship of African and non-African populations with the Vindija Neanderthal population. Additionally, we assess the

effect of violations to the neutrality assumption. We end this chapter with a description of potential future works (Section 3.5.1), which we feel would increase the feasible applications of SMC^2.

With an eye on improving the runtimes of SMC^2, Chapter 4 explores alternative EM procedures. In the process of researching various online EM (OEM) techniques, we found efficient implementation of OEM requires a deep understanding of the model in order to set an optimal value for a tuning parameter. We developed an adaptive version without the required tuning parameter. In a variety of toy models, our introspective online EM (IOEM) shows convergence comparable to the optimal choice of tuning parameter in standard OEM techniques.

We have developed software for inferring demographic parameters using sequential Monte Carlo methods applied to the sequentially Markovian coalescent. The algorithm is flexible and can easily be adapted to account for biological complications. We hope SMC^2 will help to move the field of demographic inference from trees to more informative ARGs.

Chapter 2

SMC^2: A method for demographic inference

This chapter will detail how the sequential Monte Carlo (SMC) methods described in the Introduction can be applied to the problem of inferring demographic parameters. We will cover two versions of the algorithm and analyse its performance using data simulated under known demographic scenarios.

2.1 Applying sequential Monte Carlo to the sequentially Markovian coalescent

Recall that the SMC′ model is a general state-space model with the sequence of genealogies as unobserved variables $\{X_k\}_{k=1,\ldots,m}$, and the alleles of the haploid samples as observations, denoted by $\{Y_k\}_{k=1,\ldots,m}$. X_k is a genealogy which details the TMRCA for each pair of samples at genomic position k. Additionally, X_k contains the time and lineage associated with and migration or recombination event. The observations we represent as a sequence of 0s and 1s indicating the allele state of each sample. The transition distribution $f(x_k|x_{k-1})$ describes the approximately Markovian process of

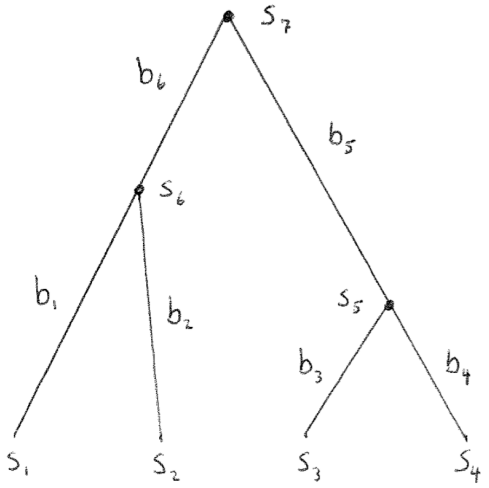


Figure 2.1: An example tree with allele states s and branch lengths b.

tree changes due to recombination. The emission distribution $g(y_k|x_k)$ describes mutations arising in the ancestral lineages of the samples. We will use the following notation throughout: $B(X_k)$ is the total branch length of the k-th genealogy, ρ is the recombination rate, and μ is the mutation rate. Both ρ and μ are assumed constant along the sequence and through time unless otherwise stated.

The emission distribution can be decomposed into a process that places the mutation somewhere along the sequence, and a process on the tree conditional on a mutation occurring at a sequence position which places the mutation on a lineage and at a particular time. Conditional on a mutation occurring at position k, the mutation placement process is simply a uniform random variable on the tree x_k. To apply SISR (Algorithm 1.2) we need to define $g(y_k|x_k)$. If $y_k \neq 0000$ or 1111 indicating a mutation has occurred, then a simple algorithm similar to Felsenstein's tree-pruning algorithm [22, 23] is used to calculate $g(y_k|x_k)$, as follows.

Consider the tree in Figure 2.1 where s_i denotes the allele of node i and b_i denotes the length of the branch directly above node i. If the alleles of the common ancestors

(s_5, s_6, s_7) were known, then the likelihood of the data would simply be

$$g(y = s_1 s_2 s_3 s_4 | x) = P_{s_7 s_6}(b_6) P_{s_7 s_5}(b_5) P_{s_6 s_1}(b_1) P_{s_6 s_2}(b_2) P_{s_5 s_3}(b_3) P_{s_5 s_4}(b_4),$$

where $P_{ss'}(b)$ is the probability that a lineage will move from state s to state s' in b time units. We assume only two possible alleles, and so

$$P_{ss'}(b) = e^{-\mu b}\delta_{ss'} + (1 - e^{-\mu b})(1 - \delta_{ss'}). \tag{2.1}$$

Note equation (2.1) relies on μ and $B(x_k)$ small enough that the likelihood of multiple mutations occuring on a single branch is negligible. We assume this to be true and ignore the possibility of back mutation.

As the states s_5, s_6, s_7 are unknown, Felsenstein's pruning algorithm calculates the probability of the data given the tree by summing the likelihoods of all possible states for the inner nodes.

$$g(y|x) = \sum_{s_7} \pi_{s_7} \Big\{ \Big[\sum_{s_6} P_{s_7 s_6}(b_6) \Big(P_{s_6 s_1}(b_1) \Big) \Big(P_{s_6 s_2}(b_2) \Big) \Big] \Big[\sum_{s_5} P_{s_7 s_5}(b_5) \Big(P_{s_5 s_3}(b_3) \Big) \Big(P_{s_5 s_4}(b_4) \Big) \Big] \Big\},$$

where π_{s_7} is a prior on the ancestral allele. We assume no prior knowledge of the ancestral allele and use the prior $\pi_0 = \pi_1 = \frac{1}{2}$. Note that this formulation does allow for multiple mutations on the tree and hence does not use the infinitely many sites model. We prefer this formulation as it prevents particle weights from becoming exactly 0 due to an inconsistent topology. The mutation rate is small enough that a tree consistent with a single mutation will have a much larger density than a tree requiring multiple mutations. However, if the scenario arises that none of the particles have a topology consistent with a single mutation at a variant site, this emission distribution will favour particles that are more consistent with multiple mutations.

This will effectively favour tall trees. If the inifitely many sites model were used, each particle weight would be multiplied by 0, discarding the information gained from earlier data. Allowing for multiple mutations at a single site has its advantages, but we should remain mindful of the potential bias towards tall trees.

We have chosen a simple model of mutation which only allows for two alleles, and where 0-to-1 and 1-to-0 mutation are equally likely. We could incorporate a more complex model, possibly distinguishing between transitions and transversions or increasing the mutation rate at CpG sites.

The process of mutation along the sequence can be modelled by a geometric random variable with $p = 1 - e^{-B(x_k)\mu}$. As $p \to 0$ the discrete geometric distribution converges to the exponential distribution with rate $B(x_k)\mu$, its continuous analogue. In the human genome a polymorphism arises every thousand base pairs (kb) on average. As p is expected to be small, we can approximate the geometric process using an exponential process with rate $B(x_k)\mu$. This approximation is needed for computational efficiency.

The transition distribution can also be decomposed into an exponential process along the sequence and a backwards in time process, which we will refer to as the sequence-process and the time-process respectively. The sequence-process models recombinations along the sequence as an exponential random variable with rate $B(x_k)\rho$. The time-process is then applied when a recombination has occurred. The recombination is placed uniformly on the tree and the coalescence above this point is removed. The branch is then grown backwards from the recombination point according to the coalescent (and optionally migration) rates as defined in equation (1.3). For convenience, I will refer to this breakage, growth, and coalescence of a branch as a branch movement.

By combining the sequence-process and the time-process we get the transition

distribution between sequence positions k and l

$$p(X_{k:l}|X_k = x_k) = \Big(\prod_{j=1}^{|x|} \rho B(x_{k_j})\ e^{-\rho B(x_{k_j})(k_{j+1}-k_j)} \frac{1}{B(x_{k_j})} \Big) \cdot \Big(\prod_{j=1}^{|x|} C_{h_j^{\text{coal}}} b_{h_j^{\text{coal}}}(x_{k_{j-1}}) e^{-\sum_{t=1}^{T-1} b_{h_t^j}(x_{k_{j-1}}) C_{h_t^j}(h_{t+1}^j - h_t^j)} \Big) \tag{2.2}$$

for $l > k$ in a single population model (i.e. no migration). The first term reflects the placement of the recombination in the sequence and on the tree. The second term reflects the new coalescent event. Here $|x|$ is the number of recombinations in $X_{k:l}$. For $j = 1, \ldots, |x|$ recombination j occurs at genomic position k_j, hence x_{k_j} is the genealogy created by the j-th recombination. k_0 and $k_{|x|+1}$ are specially defined to be $k_0 := k$ and $k_{|x|+1} := l$. The coalescent rate at time u is $C_u := 1/2N_e(u)$. The number of branches in topology x at time u is $b_u(x)$. h_j^{coal} is the height of the coalescence caused by the j-th recombination. $h_1^j := h_j^{\text{rec}}$ is the height of the j-th recombination. $h_T^j := h_j^{\text{coal}}$, and $h_2^j, \ldots, h_{T-1}^j$ are the heights at which either an existing coalescence occurs or an epoch boundary between h_1^j and h_T^j, satisfying $h_1^j < h_2^j < \ldots < h_T^j$.

Using the SISR algorithm described in Section 1.3.2 and the ARG simulation software SCRM [77] we can generate a Monte Carlo sample of ARGs from the posterior distribution $p_{\hat{\theta}}(X_{1:T}|y_{1:T})$. Of course, for a fixed number of samples N this approximation will deteriorate due to particle degeneracy and sample impoverishment. However, we are interested primarily in inferring demographic parameters which can be done by combining the fixed-lag technique (Section 1.3.3) and online EM (Section 1.3.4).

The online EM procedure consists of collecting summary statistics as we extend the particles. In the context of SMC^2 events occur according to a Poisson process. The summary statistics are the count of how many times an event occurs as well as the opportunity available for that event to occur (proof in Appendix A). This is simplest for the recombination rate parameter where the opportunity for an event is

the total branch length of the tree, $B(x_k)$, times the length of the genome segment. The summary statistics used to infer recombination rate are then the weighted branch length and the weighted count of recombinations.

$$\hat{\rho} = \frac{\sum_{i=1}^{N} w_i |x|^{(i)}}{\sum_{i=1}^{N} w_i \sum_{j=1}^{|x|^{(i)}} B(x_{k_j}^{(i)})(k_{j+1}^{(i)} - k_j^{(i)})}.$$

To simplify the text we will not always emphasise that the statistics are weighted by the weight of the particle they are associated with, although of course this remains the case.

For the coalescence rate parameter, which is inversely proportional to N_e, we also have an opportunity and a count summary statistic. The opportunity for coalescence only arises once a recombination has triggered a branch movement. When growing a branch from a recombination there is opportunity for coalescence into any contemporary lineage in the same population. And so the opportunity for coalescence on tree x_j is the total branch length of the tree between the height of the new coalescence, h^{coal}, and the height of the recombination h^{rec}, which we will denote by $B(x_j, h^{\text{rec}}, h^{\text{coal}})$. However, we are interested in inferring N_e per epoch, and so we split the opportunity depending on the epoch boundaries (Figure 2.2). The count of one coalescent event is then added to the epoch which contains the time of the coalescent.

The migration rate parameter is similar to the coalescent rate parameter as opportunity arises due to recombination events. If a recombination occurs in population A then there is opportunity for backwards in time migration from population A to population B, a statistic for the parameter $m_{A,B}$. The opportunity however is not the branch length between the height of the recombination and the height of the new coalescence. If no migration occurs, the opportunity is $h^{\text{coal}} - h^{\text{rec}}$, as it was the grown branch that had the possibility of migrating to population B. If a single migration occurs, the opportunity for A to B migration is $h^{\text{migr}} - h^{\text{rec}}$. In this case there is

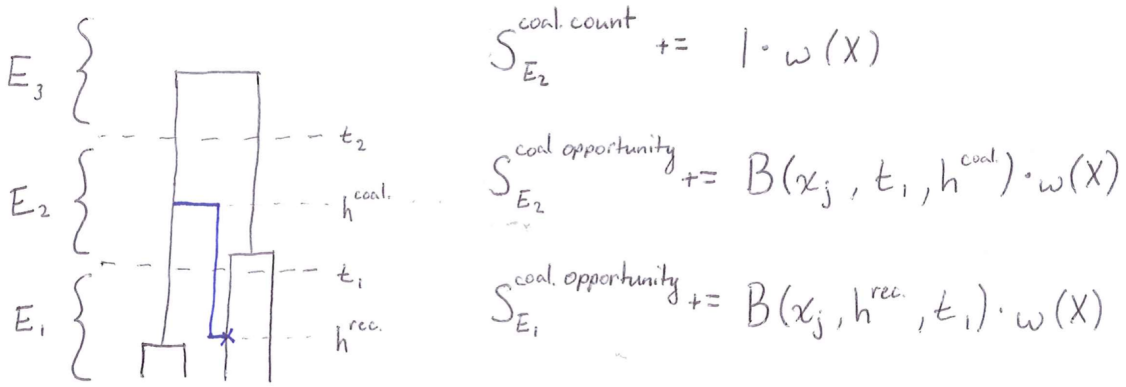


Figure 2.2: An example of the coalescent summary statistics of a branch movement where the cross represents a recombination and the blue line represents the new lineage.

additionally opportunity for B to A migration of $h^{\text{coal}} - h^{\text{migr}}$. Again, this must be split between epochs and the event is attributed to the epoch it occurs in.

We employ an online EM approach in collecting the sufficient statistics as this is computationally efficient. For simplicity we use the batch version of online EM where the statistics are collected as the data is scanned, but the parameter is updated after a predetermined batch of data. We use the full data as a batch and update the parameter before rescanning the data.

2.2 Parameter-specific fixed lag

The main challenge in applying fixed-lag techniques is determining an appropriate lag value L. The optimal value Δ depends on the forgetting rate of the underlying true transition distribution and on the collapsing time of the resampling process. The larger the forgetting rate of the transition distribution, the smaller Δ can be and still produce a good approximation

$$p(X_{0:k}|Y_{0:T}) \approx p(X_{0:k}|Y_{0:min(k+\Delta,T)}). \tag{2.3}$$

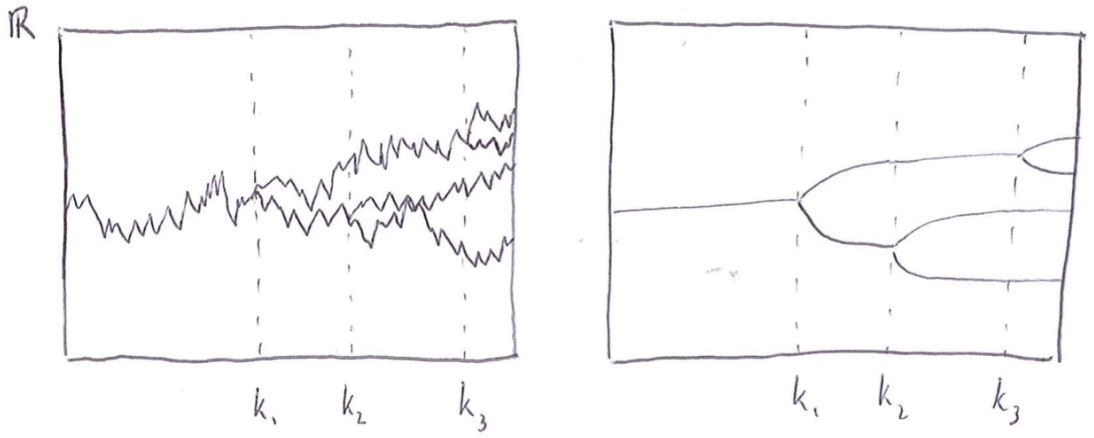


Figure 2.3: Typical particle trajectories in SISR. (*Left*): Trajectories when the state space is $\mathbb{R}$. (*Right*): An abstract representation for a more complex state space, like the tree space.

So the forgetting rate imposes an unknown lower bound on suitable L. On the other hand, the collapsing time of the resampling process imposes an unknown upper bound on L needed to keep the sample sufficiently independent.

The left panel of Figure 2.3 shows particle histories in SISR applied to a typical model, for example the stochastic volatility model. There are $N = 4$ 'independent' particles at the current component of the state space, but for the earlier parts of the sequential state space many particles will share values as the particles have been resampled as they were extended. Hence, the particle collapse due to resampling imposes an upper bound on effective L due to large Monte Carlo variance. The tree state space cannot be as easily visualised as the real numbers, but the same principle applies. The right panel of Figure 2.3 shows an abstract representation of the particle histories, which can be used for our tree state space. Here the values of the particles are not shown, but the relationship between particles is clear. We are now considering a tree describing the particle inheritance structure. For clarity, the genealogies relating the sequenced individuals will always be drawn vertically, and particle inheritance will be drawn horizontally.

For the ARG state space the Monte Carlo variance introduced by resampling is

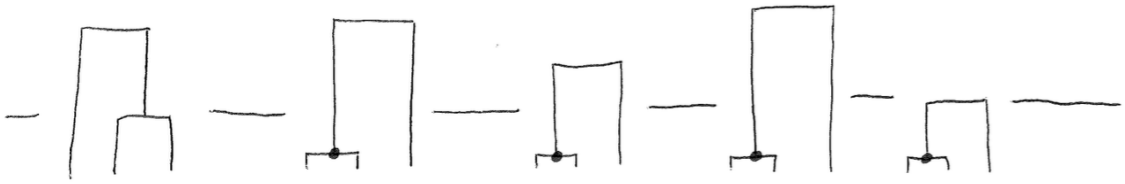
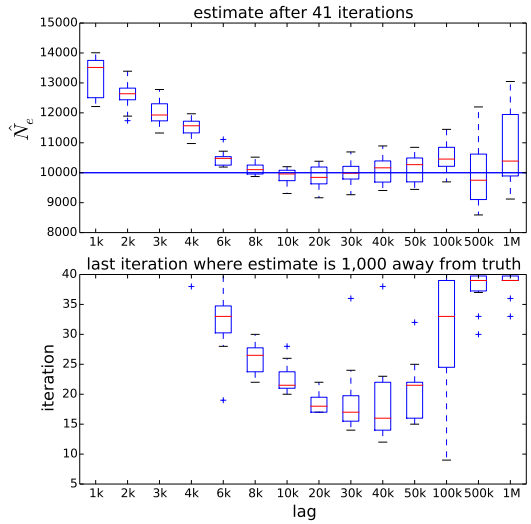


Figure 2.4: Events low in the tree are maintained for a greater distance along the genome.

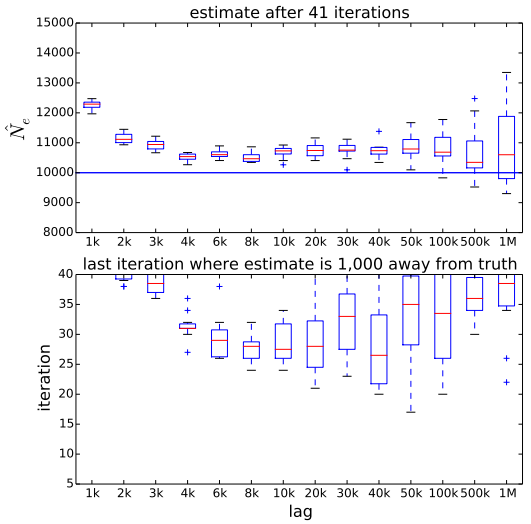
more severe than suggested by the particle trajectories as shown in Figure 2.3. In many models, two particles that are sampled from the same parental particle quickly diverge from each other and become sufficiently independent. This is not the case for ARGs due to the long distance dependencies.

The especially interesting challenge for SMC^2 involves the lower bound on effective L due to the extent of the bias introduced by the approximation in equation (2.3). The unique structural dependencies of SMC$'$ are most easily conveyed by an example. Figure 2.4 shows a typical particle under the SMC^2 model. The bottom (recent) section of the trees retain dependence along the genome for a greater distance than the top (ancient) section of the trees do. This suggests the forgetting rate of a particular coalescence depends on the height of the coalescence.

To investigate this problem, I tested a range of lag values for inferring N_e. Two haplotypes were simulated under a constant population size model of $N_e = 10,000$ using the coalescent simulator SCRM [77] (see Appendix B). For inference, the model was parameterised with 17 epochs, and I chose a single epoch j to investigate. SMC^2 was initialised at $\hat{N}_{ej}^{\,0} = 20,000$ and $\hat{N}_{ek}^{\,0} = N_e$ for $k \neq j$, and the algorithm was altered so only $\hat{N}_{ej}$ was updated during the M-step. Fixing all but one parameter to the true value removes confounding affects and allows for a deep look at the optimal lag for an epoch. When investigating the 7th epoch (0.14-0.2 in time scaled by $4N_e$ generations), we found a lag of approximately 30,000 to be optimal. The top panel of Figure 2.5a shows the final inferred population size after 41 EM iterations, $\hat{N}_{e7}^{\,41}$, for the different lag choices. With too small a lag, the estimates are slow to move from



(a) Inference for epoch 7, $\hat{N}_{e7}$



(b) Inference for epoch 11, $\hat{N}_{e11}$

Figure 2.5: Inference under different choices of lag L; 2 samples; 1,000 particles.

their initial values due to a bad approximation $p(X_{0:k}|Y_{0:T}) \approx p(X_{0:k}|Y_{0:min(k+\Delta,T)})$; this bad approximation also introduces bias. With too large a lag, the estimates suffer from high variance due to sample impoverishment. The choice of 41 EM iterations is fairly arbitrary, and so we should not overly rely on the estimates at this point. To analyse convergence of parameter estimated under different lag choices, the bottom panel of Figure 2.5a plots the latest iteration i where $\hat{N}_{e7}^{\ i}$ is outside of $(0.9 \cdot N_e, 1.1 \cdot N_e)$.

We conduct the same analysis on the more ancient 11th epoch (0.45-0.58) and display the results in Figure 2.5b. Comparing the results of these two separate analyses, we see that estimation for the more ancient epoch does relatively well with a smaller lag. We conclude that a large lag is optimal for recent epochs and a small lag is optimal for ancient epochs.

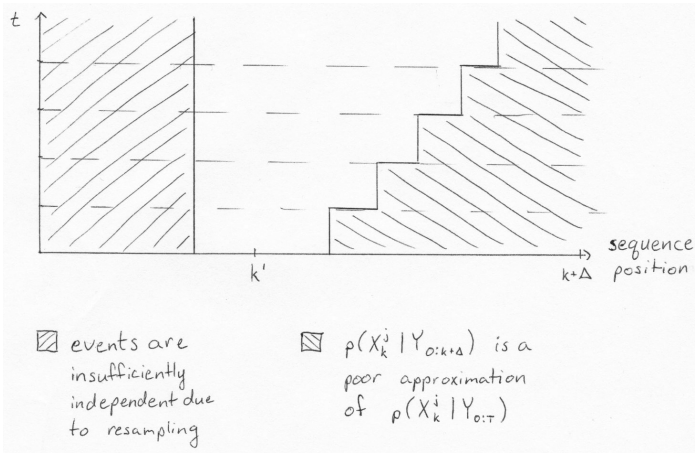
This finding motivated our use of parameter-specific, or more precisely epoch-specific, lag. The optimal lag will be model specific as the demography affects the strength of the dependence at each epoch. Hence we need to understand why these lags appear to be optimal.

When implementing the SISR algorithm, the optimal lag depends on the forgetting rate of the model, with a large lag needed for a small forgetting rate [11]. Due to the complexities of the ARG state space, the transition distribution is not a member of the exponential family, and so does not have a typical forgetting rate. However the SMC′ model is built up from multiple exponential processes acting on different regions of the trees. We will use X_k^j to denote the epoch j component of the tree at position k. As described earlier, the recent epochs in the bottom of the tree have a longer memory, or smaller forgetting rate. A small forgetting rate suggests it is better to use a large L, and so our findings are consistent with the theoretical argument.

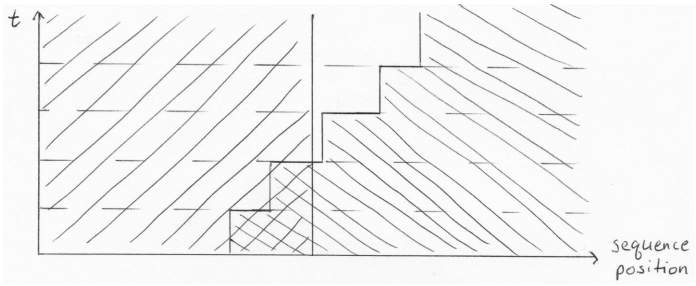
The ARG has a particularly large state space and as such retaining many independent particles with sufficiently large weights to survive resampling is a challenge. However, the analysis in Figure 2.5 shows certain choices of lag can yield estimates with a reasonable variance and bias. We must remain mindful of effect our choice of lag has, as described in Figure 2.6.

We introduce a model specific lag-calibration step before the E step of our EM procedure. To do this we simulate ARGs using the current estimates of the demographic parameters. We measure the survival distance of each node in the trees. For each epoch, we then calculated the median node survival distance, v_j, and we set the lag equal to this value. This empirical value ensures the events which contribute to the summary statistics have been weighted using the data which would provide the most evidence for or against their existence.

Scaling the parameter-specific lag by the memory of the parameter is intuitive and seems to work. However, the argument that only the data upto the removal of the associated coalescence provides support for or against the event is flawed. The possible tree transitions due to a recombination event are limited to certain topologies. As a result dependencies on a particular coalescence event outlast the coalescence itself. For this reason a lag larger than the survival distance may result in



(a) There is a region where particles are sufficiently independent and the posterior is well approximated. If Δ is chosen such that $k = k'$, then neither the bias nor the variance will be large. If Δ is small enough that k falls in the right zone, the bias will be large. If Δ is large enough that k falls in the left zone, the variance will be large.

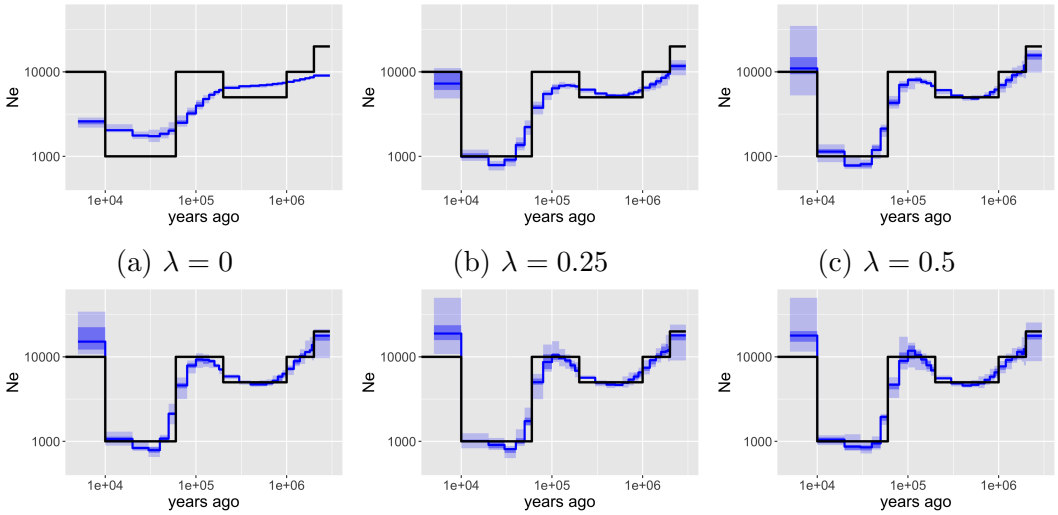


(b) In this case, for lower epochs no choice of lag produces sufficiently independent events and a good approximation of equation (2.3).

Figure 2.6: An illustration of the regions of the ARG space that have sufficient particle diversity and informative data contributing to the particle weights. This is a simplification with hard boundaries dictating where bias or variance become an issue. In reality, this effect is a spectrum, and there is a bias/variance tradeoff for any choice of lag.

improved inference. Alternatively, a lag smaller than survival distance may be better to avoid the increase in variance problem which could arise in the situation described in Figure 2.6b. In this situation, utilising all the pertinent data could diminish the sample. And so, while we are happy to set the lag for epoch j to $L_j := \lambda \cdot v_j$, we need to consider the possible values for λ.

We ran SMC^2 with a range of lag scaling factors $\lambda = 0, 0.25, 0.5, 1, 2, 4$ on data simulated under the bottleneck model (Appendix B). For each λ we ran 10 replicates



(a) $\lambda = 0$ (b) $\lambda = 0.25$ (c) $\lambda = 0.5$

(d) $\lambda = 1$ (e) $\lambda = 2$ (f) $\lambda = 4$

Figure 2.7: Lag scaling factor comparison; bottleneck model; 4 samples; the black line is the true N_e.

and let these run for 30 EM iterations. We plot the quantiles of our 10 estimates for each parameter in Figure 2.7. The light blue band spans the 0th to 100th quantile, the dark blue band spans the 25th to 75th quantile, and the solid blue line is the median estimate. The results show that $\lambda < 1$ results in relatively slow convergence. All lag factors greater than or equal to 1 seem to produce stable results without substantial bias from too smal a lag. with higher variance with the larger lag factors. We proceed using $\lambda = 1$ because of the reduced variance compared to larger choices of λ. In addition, the smaller the lag the more efficient the programme in computational memory, and so $\lambda = 1$ is a more practical choice than $\lambda > 1$.

2.3 Simulated data results for version 1

With the core of the algorithm in place and the lag factor chosen, we now analyse the performance of SMC^2 in a range of demographic models.

The simplest model to consider is one with a single population of constant size. Running SMC^2 over data simulated under this model produces the estimates in Figure 2.8. Each plot shows the results of 10 runs of SMC^2 provided with 10 different data sets. The estimates of N_e for the most recent epoch clearly diverge from the true value. This effect is most pronounced in the case of 8 haploid samples, where 10,000 particles is not sufficient to explore the state space. Suppose no particle has a compatible topology at the site of a single mutation that is observed in two individuals. The emission distribution will need two mutations, one on each of the singleton branches, to fit the data. This will put a large weight on particles with tall singleton branches, effectively biasing against recent coalescences, and hence inflating the estimate of the most recent N_e.

The bottleneck model (see Appendix B for the full model description) is a more interesting case as it is representative of the demographic history of modern day non-African populations. Figure 2.9 shows the sequence of estimates produced by 30 EM iterations of SMC^2 initialised at $\hat{N}_{ej}^{\,0} = 10,000 \;\forall\; j$. The results are comparable to those obtained by PSMC [49] (Figure 2.10). Here SMC^2 has, with the exception of the most recent epoch, correctly inferred the demography using 8 haploid samples rather

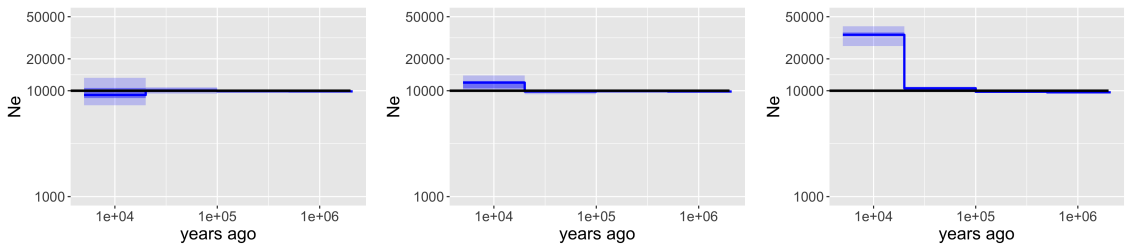


Figure 2.8: Constant model; 2, 4, 8 samples; initialised at the truth; run for 10 EM iterations over 200Mb of data; the black line is the true N_e.

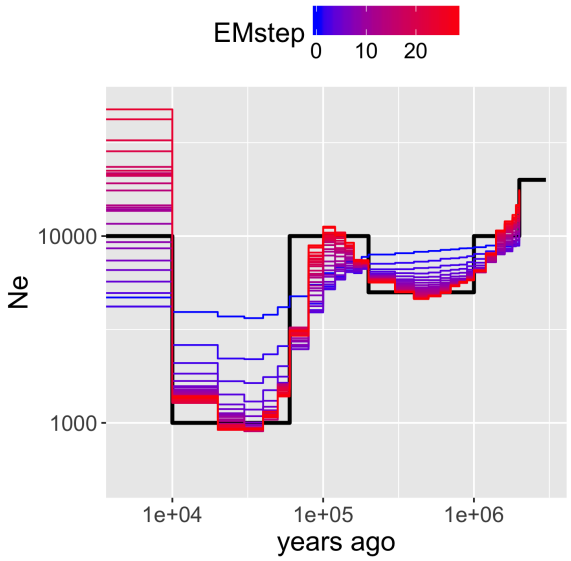


Figure 2.9: Bottleneck model parameter estimate evolution; 8 samples; a single replicate run for 30 EM iterations; the black line is the true N_e.

than the 2 used by PSMC. At face value using more data and getting comparable results is not an achievement, but considering the additional complexity of the 8 sample state-space this is a significant step forward, as it allows inference of migration rates, which is not possible if only 2 haploid samples are used (for PSMC), or when approximate models are used (for MSMC).

Figure 2.11 shows the influence of the number of particles ($N = \{100, 1000, 10000\}$) and the length of the genomic data ($T = \{10\text{Mb}, 50\text{Mb}, 100\text{Mb}\}$). Any estimates that exceed the boundaries of the plots are plotted at that boundary. As expected, increasing either N or the amount of data results in less Monte Carlo variance and standard error.

The above analyses use 10 replicates each with a new set of data simulated from the same demographic model. To compare the extent of estimate variance resulting from the diverse data and the Monte Carlo variance, we run 10 replicates of SMC^2 on a single dataset (Figure 2.12). The single dataset replicates show only slightly less variability than the unique data replicates. It seems the bulk of the variation is due to the Monte Carlo approximations, rather than variation in the data itself. For the

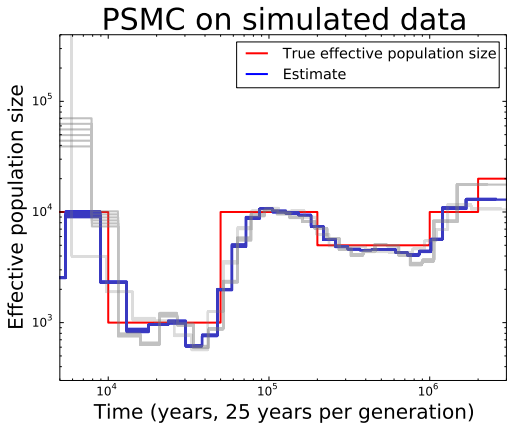


Figure 2.10: PSMC on bottleneck model; 2 samples; run for 40 EM iterations; the grey lines are inferred curves from bootstrapped data.

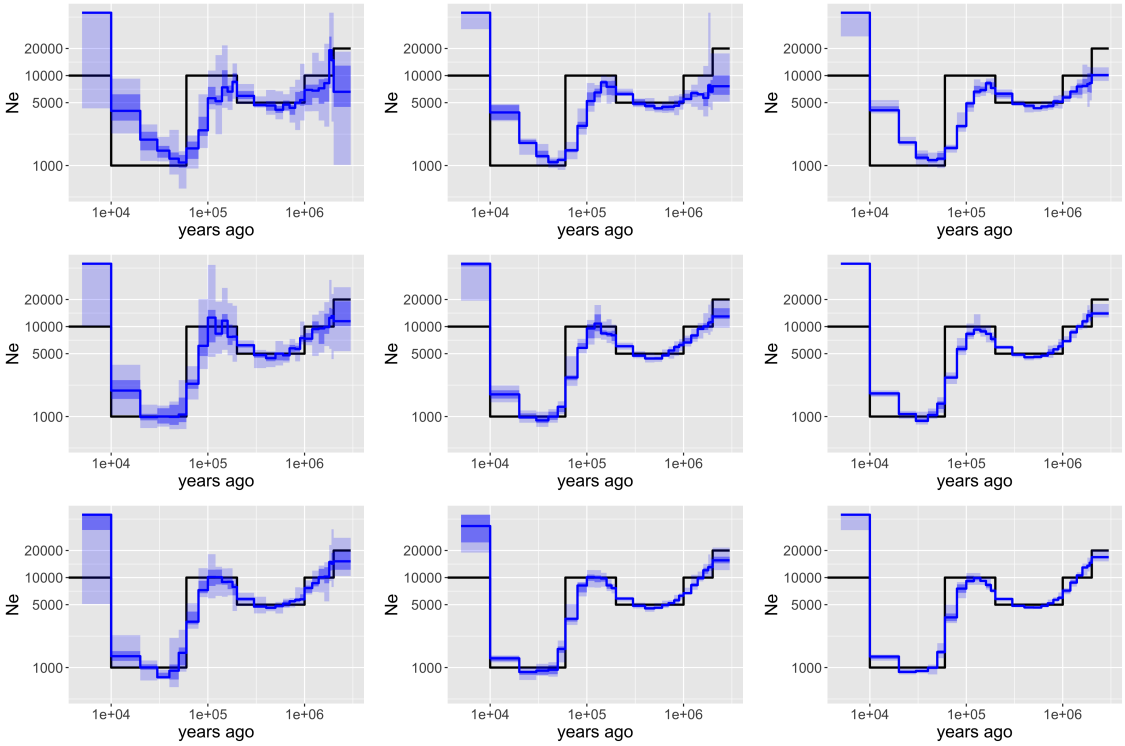


Figure 2.11: Bottleneck model; 8 samples; from top to bottom $N = 100, 1\text{k}, 10\text{k}$; from left to right $T = 10\text{Mb}, 50\text{Mb}, 100\text{Mb}$; the black line is the true N_e.

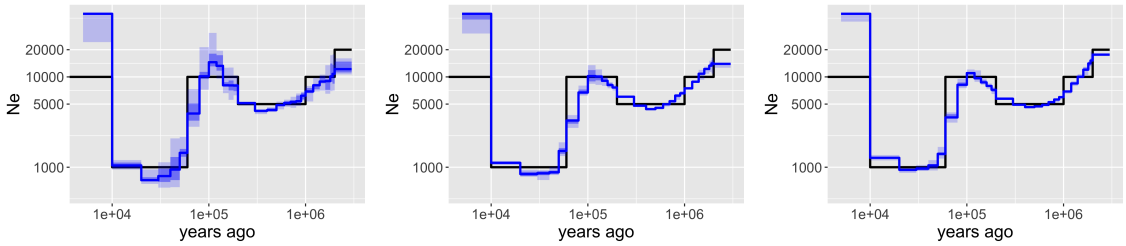


Figure 2.12: Bottleneck model; 10 replicates on a single simulated dataset; 8 samples; $N = 10$k; from left to right $T = 10\text{Mb}, 50\text{Mb}, 100\text{Mb}$; the black line is the true N_e.

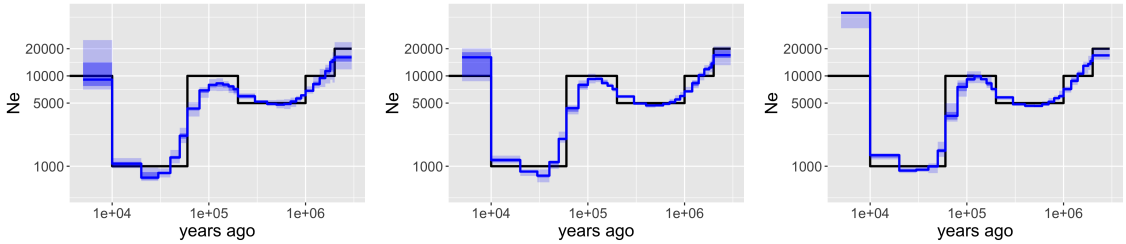


Figure 2.13: Bottleneck model; 2, 4, and 8 samples; the black line is the true N_e.

remainder of this chapter, we use replicates with unique data.

Fixing $N = 10,000$ and $T = 100\text{Mb}$ we can compare the inference results from analysing 2, 4, or 8 haploid samples. Overall these results are similar, but, as in the constant model, the 8 sample case struggles to infer N_e for the most recent epoch (Figure 2.13). If we look at the estimate evolution for a single replicate for each case, it is clear the 8 sample case is not the only one struggling with inferring effective population size for 0-10kya. Figure 2.14 shows that the apparent good estimates for the 2 and 4 sample cases after 30 EM steps are actually diverging from the true value, but more slowly than the 8 sample case. We will revisit this shortcoming in the next section.

While this version of SMC^2 does fairly well at inferring N_e, there are alternative methods to achieve this which do not require modelling the entire ARG. We aim to model the ARG in order to infer parameters that are unobtainable from existing methods. In particular we are interested in estimates of directional migration rates (backwards-in-time). To test inference of migration rates, we introduce two simple de-

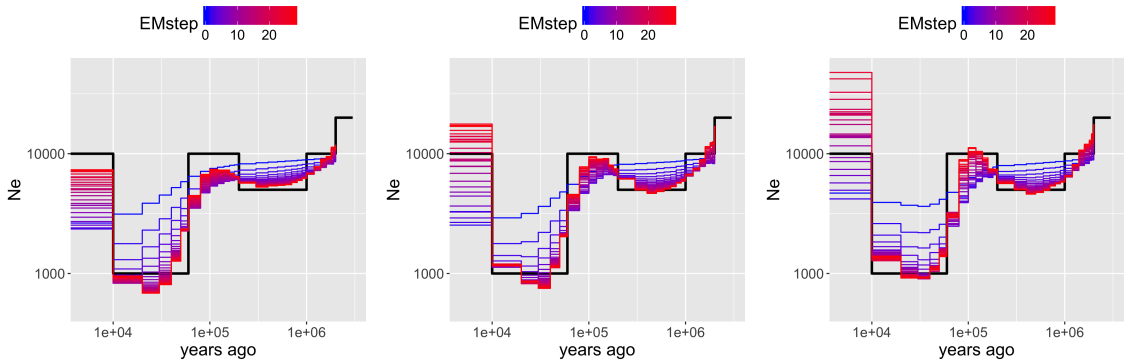


Figure 2.14: Bottleneck model parameter estimate evolution; 2, 4, and 8 samples; the black line is the true N_e.

mographic models: uni-direction migration and split no migration (see Appendix B).

The uni-directional migration model imposes a population split at .5 (in $4N_e$ generations) followed by migration from population 0 to 1 at a rate of 5e-6 migrations per lineage per generation. SMC^2 is able to accurately infer migration in the case of 4 samples using 10,000 particles (Figure 2.15), but in cases with fewer particles or more samples the migration in the most recent epoch is falsely inferred to be 0. This suggests the underestimation of recent migration is caused by an insufficient exploration of the state space. The particles with the correct migrations and topologies are not reliably proposed. When no particles contain a migration near the site of a true migration, and therefore do not have the correct topology at mutation sites, the particles with large singleton branches will be upweighted as they fit the data best conditional on there being more than one mutation.

Of course we also need to be able to infer a lack of migration. To test this we use the split no migration model, where an ancestral population splits into two constant size populations. Here we falsely infer a non-zero rate of migration (Figure 2.16). This bias is more pronounced in the case of 4 samples than 8 samples. The 4 sample case has a smaller state space and so is easier to model, which indicates the increase in bias is not due to sampling difficulties. It is likely the increased bias is due to a lack of power in the data when using 4 samples.

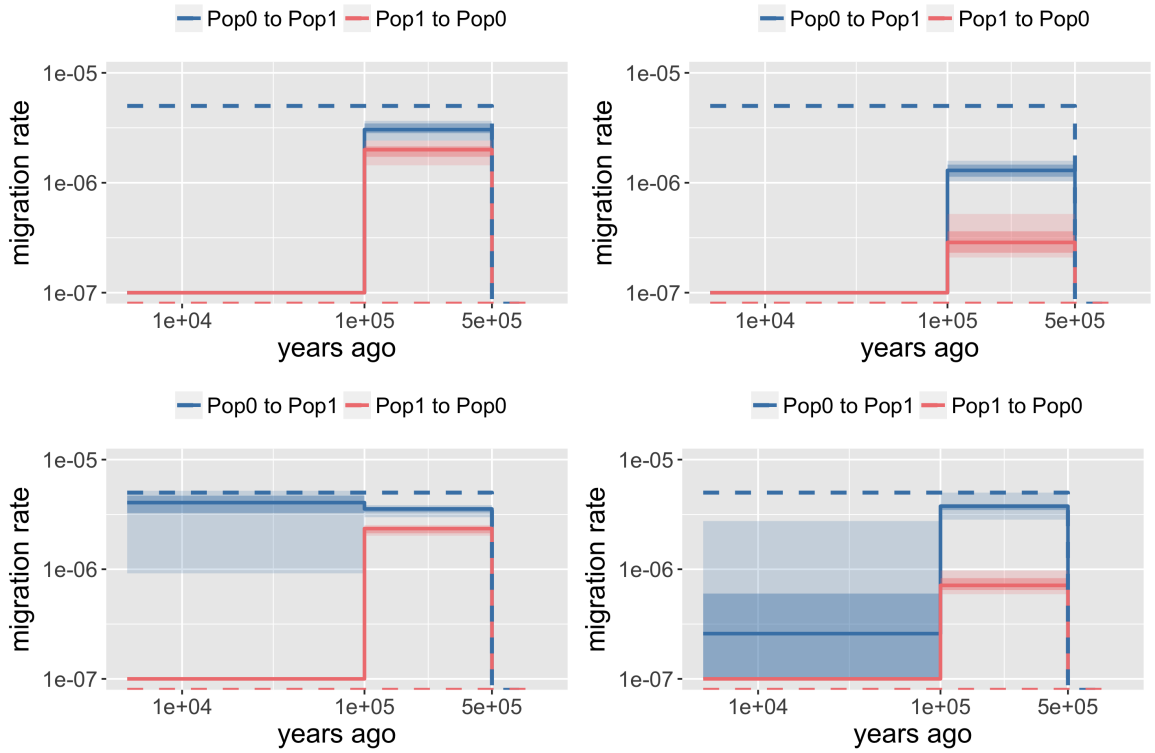


Figure 2.15: Uni-directional migration model; from left to right 4, 8 samples; from top to bottom 1k, 10k particles; the truth is represented by the dashed lines.

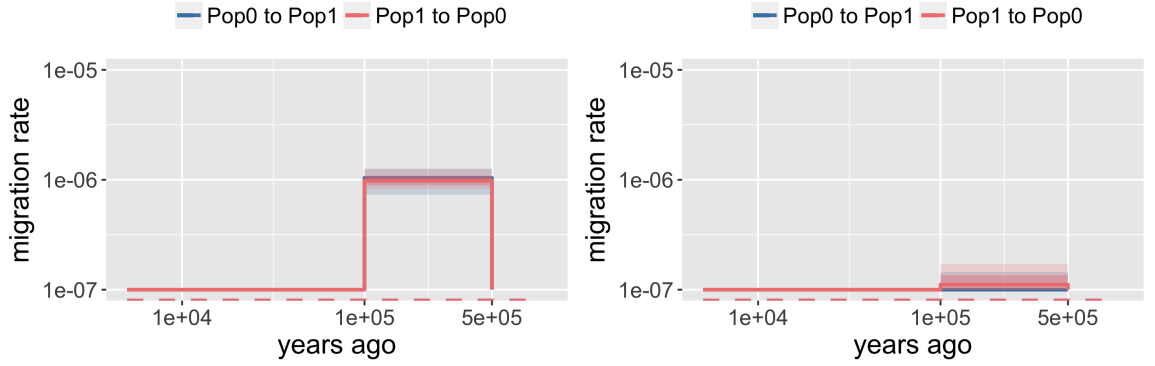


Figure 2.16: Split no migration model; 4, 8 samples; 10k particles; the truth is represented by the dashed lines.

2.4 Focusing the sampler

Propogating particles according to our estimated transition density is not sufficient for inferring migration parameters or particularly recent epoch parameters. Asymptotically, the estimate of the transition density would be the optimal choice of proposal distribution [19]. However, in our case of limited particles this is no longer true. Under the true transition distribution, branch movements which result in coalescent events low in the tree occur sparsely along the sequence. These events have happened somewhere in the data and they are needed to infer the recent demography. As such, sampling based on even the true transition distribution with a computationally feasible number of particles will rarely propose these events, and the sampler is not likely to propose them near sequence positions where they actually occurred. For this reason we have chosen to broaden the tails of the proposal distribution by focusing the sampler on rare transitions.

To focus the sampler, we alter the proposal distribution $q()$. Recall that the transition density from position 0 to k in equation (2.2) is

$$p(X_{0:k}|X_0 = x_0) = \Big(\prod_{j=1}^{|x|} \rho B(x_{k_j})\ e^{-\rho B(x_{k_j})(k_{j+1}-k_j)} \frac{1}{B(x_{k_j})} \Big) \cdot \Big(\prod_{j=1}^{|x|} C_{h_j^{\text{coal}}} b_{h_j^{\text{coal}}}(x_{k_{j-1}}) e^{-\sum_{t=1}^{T-1} b_{h_t^j}(x_{k_j-1}) C_{h_t^j}(h_{t+1}^j - h_t^j)} \Big).$$

Now consider a segment of a particle such that there is no recombination on $[0, K)$ followed by a recombination at position K at height h^{rec}. The transition density for this particle is

$$p(X_{0:K}|X_0 = x_0) = \frac{1}{B(x_0)} \rho B(x_0) e^{-\rho B(x_0)(K)} \cdot A,$$

where

$$A := C_{h^{\text{coal}}} b_{h^{\text{coal}}}(x_0) e^{-\sum_{t=1}^{T-1} b_{h_t}(x_0) C_{h_t}(h_{t+1}-h_t)}.$$

The recombination term $\frac{1}{B(x_0)}\rho B(x_0) e^{-\rho B(x_0)(K)}$ is independent of the placement of the recombination on the genealogy. In the transition distribution the tree-position of the recombination is uniformly distributed over the tree. Our altered proposal distribution includes a vector of heights $\{h_l\}$ and a vector of focus strengths $\{s_l\}$ which determine the rate of recombination placement on the tree. To focus on a particular time section of the tree $[h_l, h_{l+1})$, in this case the bottom of the tree, we set $s_l > 1$. The proposal distribution for a recombination placed in $[h_l, h_{l+1})$ is then

$$q(X_{0:K}|X_0 = x_0) = \frac{s_l}{C}\rho B(x_0) e^{-\rho B(x_0)(K)} \cdot A,$$

where C is the normalising constant

$$C := \sum_{l=1}^{L} s_l B(x_0, h_l, h_{l+1}),$$

and $B(x_0, h_l, h_{l+1})$ is the total branch length of tree x_0 between heights h_l and h_{l+1}.

The weight for such a particle generated using the new sampler is

$$\begin{aligned} w_{0,K}(X^{(i)}) &= g(y_{0:K}|X^{(i)}) \cdot \frac{f()}{q()} \\ &= g(y_{0:K}|X^{(i)}) \cdot \frac{1/B(x_0^{(i)})}{s_l^{(i)} / \sum_{l=1}^{L} s_l B(x_0^{(i)}, h_l, h_{l+1})} \\ &= g(y_{0:K}|X^{(i)}) \cdot \frac{\sum_{l=1}^{L} s_l B(x_0^{(i)}, h_l, h_{l+1})}{s_l^{(i)} B(x_0^{(i)})}. \end{aligned}$$

More generally, the weight update is

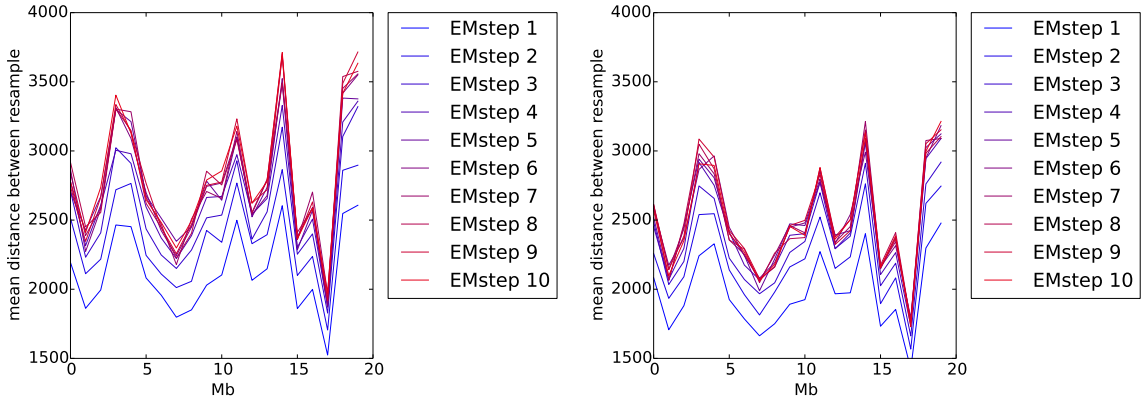
$$w_{k,K}(X^{(i)}) = g(y_{k:K}|X^{(i)}) \cdot \prod_{j=1}^{|x|^{(i)}} \frac{\sum_{l=1}^{L} s_l B(x_{k_j}^{(i)}, h_l, h_{l+1})}{s_{l,j}^{(i)} B(x_{k_j}^{(i)})}.$$

We define

$$\frac{f}{q}(X_{k:K}^{(i)}) := \prod_{j=1}^{|x|^{(i)}} \frac{\sum_{l=1}^{L} s_l B(x_{k_j}^{(i)}, h_l, h_{l+1})}{s_{l,j}^{(i)} B(x_{k_j}^{(i)})}.$$

We implemented the focused sampling (FS) and then revisited the migration models. The broad proposal distribution provided only modest improvements in our parameter estimates (shown in Section 2.5). The importance weight updates $w_{k,K}()$ are detrimental to the particles that propose events in the broadened tails of the proposal. Of course this must be the case to convert our sample from the proposal distribution to the target distribution. We hypothesised that while we were sampling a more diverse range of particles, those with the rare transitions we intended to boost in our sample were killed by resampling almost immediately due to the penalty $w_{k,K}()$. To test this hypothesis we ran SMC^2 with the naive and the focused sampler and compared the density of resampling events along sequence. Figure 2.17 shows the per-Mb average distance between positions where resampling is called. The decreased distance for the FS indicates resampling is indeed occuring more often. We cannot say for certain if this increased resampling is disposing of the desired particles, or if the desired particles are creating a skew in the weights which causes more resampling.

Proposing particles which will quickly be culled by the resampling process is a waste of computational resources. We could reduce the number of such cases by setting all focus strengths close to 1, but this would bring us back to the shortcomings of the naive sampler. We would like for rare transitions to be proposed frequently enough that when the data supports such a transition, there is at least one representative particle available. To do this with a reasonable number of particles the resampling process must be altered.



(a) naive sampler

(b) focused sampler

Figure 2.17: The average distance between resampling per-Mb.

The difficulty is due to the long-distance linkage in the ARG and the noisyness of the emission distribution $g()$. When a particle undergoes a rare transition which is similar to the true transition, the data will eventually support the particle by increasing its importance weight through a relatively large $g()$. The supporting data is incrementally incorporated into the importance weight as the data is scanned. At the same time the resampling process is acting to dispose of particles with small importance weights. We alter the resampling process to allow the data to the right of a transition to contribute to the weight before the particle is removed due to the transition factor $\frac{f}{q}()$. We refer to this method as focused sampling with delayed resampling (FSDR).

It is important to note that we are changing only the resampling process. The importance weights, which are used for online EM, remain the same. For the purpose of resampling we introduce different weights w^{res} which map the particles to what we call the delayed-resampling distribution. When we resample, we sample the particles proportional to the w^{res} instead of the importance weight. This means immediately after resampling the ESS will not equal N as before. If we choose the delayed-resampling distribution wisely, we can maintain more diversity in the particles by sacrificing some resolution in the peakiest region of the transition.

Another way to view the delayed resampling process is as a two-step importance sampling procedure. We use SISR to obtain a sample from the delayed-resampling distribution. The delayed-resampling distribution then acts as the proposal distribution for our second importance sampling which yields the approximation of the target distribution. When a resample occurs at position K, the particles are resampled proportional to w^{res}, and the resampled particles are distributed according to the delayed-resampling distribution. Then the second importance sampling step approximates the target distribution by

$$\hat{p}(X_{0:K}) = \sum_{i=1}^{N} w'_K(X^{(i)}_{0:K}) \cdot \delta(X^{(i)}_{0:K})$$

where

$$w'_K(X^{(i)}) \propto \frac{w_{0,K}(X^{(i)})}{w^{\text{res}}_{0,K}(X^{(i)})} \text{ such that } \sum_{i=1}^{N} w'_K(X^{(i)}) = 1.$$

The role of w^{res} is to force the resampler to consider data to the right of a transition before applying $\frac{f}{q}()$, which will be a penalty for the oversampled rare transitions (see Figure 2.18). As such, we want w^{res} to take the form

$$w^{\text{res}}_{k,K}(x) = g(x_{k:K}|y_{k:K}) \cdot \frac{f(x_{k:K-D})}{q(x_{k:K-D})}$$

with $D > 0$ and $k < K-D$. This is similar to, but distinct from, a class of SIS methods called lookahead methods which we will discuss in more detail in Section 3.5.1.

The delay D should be chosen to ensure sufficient data contributes to w^{res} before the transition factor $\frac{f}{q}()$ is applied. The notion of how much data is sufficient to provide evidence for or against a transition relies on the forgetting rate of the transition distribution. We revisit the ideas explored in Section 2.2 where we concluded that the lower region of the genealogy has a smaller forgetting rate than the upper region.

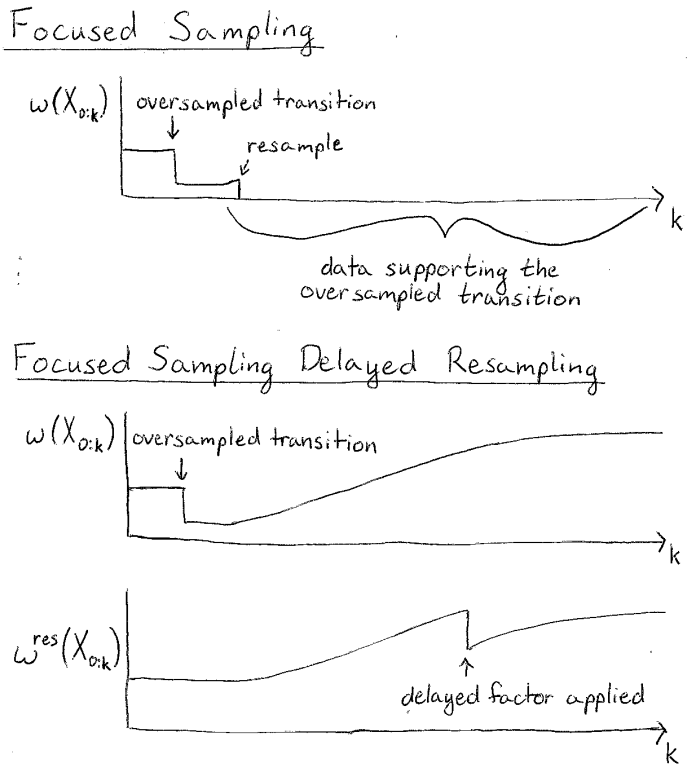


Figure 2.18: **FSDR**. In this example the focused sampler proposes a transition with a low probability. In the case of FS, the particle is quickly removed by resampling (i.e. its weight becomes 0) before the data which would support the transition is incorporated into the weight. In the case of FSDR, the data contributes to the particle weight before resampling removes it.

Just as the lag L_j was chosen to be epoch-dependent, so too is the resampling delay D_j.

Recall that the lag for online EM is assigned based on the epoch of the summary statistic (an event or opportunity that occurs in epoch j and hence affects $\hat{N}_{ej}$ is assigned lag L_j). The $\frac{f}{q}()$ factor of the importance weight is not as easily attributed to a single epoch. The branch movement involves a recombination in epoch j_{rec}, a coalescence in epoch j_{coal}, and possibly a migration in epoch j_{mig} with $j_{\text{rec}} \leq j_{\text{mig}} \leq j_{\text{coal}}$ (for simplicity of notation we assume a maximum of one migration although more are possible). We have chosen to assign a delay of $D_{j_{\text{rec}}}$ as the recombination rate is independent of the epoch. We fear introducing a dependence between the buffer and the inferred demography could lead to feedback loops.

We would like to scale the delay according to the epoch of the recombination. To do this we again rely on the pre-E step of SMC^2 to calculate the median node survival for each epoch and set the delay to be half of this value (and so half of the lag, since $\lambda = 1$). This ensures some of the relevant data has been considered before the resampler can dispose of a particle due to a rare transition.

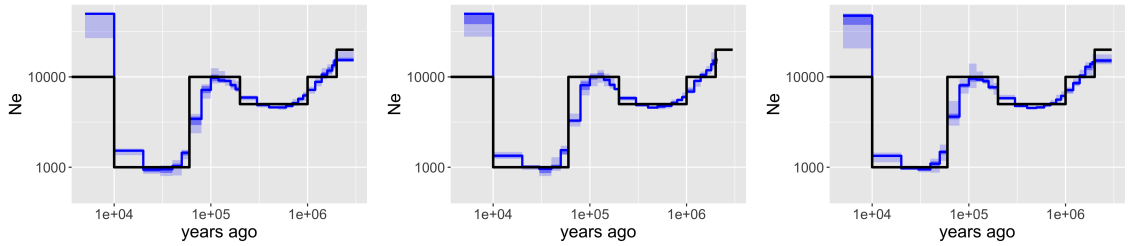
Although there is logic behind applying the factor $\frac{f}{q}_{K-D_{j_{\text{rec}}}} := \frac{f}{q}(x_{K-D_{j_{\text{rec}}}-1:K-D_{j_{\text{rec}}}})$ at position K, it remains a fairly arbitrary choice for a possibly large change to the w^{res}. As such we decided to apply this delayed factor in three stages. For a recombination that occurs at genomic position $K - D_{j_{\text{rec}}}$, the entire factor is applied by position K. The factor is applied gradually by setting $w^{\text{res}} = w^{\text{res}} \cdot (\frac{f}{q}_{K-D_{j_{\text{rec}}}})^{1/3}$ at positions $K - \frac{6}{7}D_{j_{\text{rec}}}$, $K - \frac{4}{7}D_{j_{\text{rec}}}$, and K.

The focused sampler has introduced quite a few tuning parameters. Unless otherwise stated we set $h = \{0, 800\}$ (in generations) and $s = \{3, 1\}$. This default oversamples recombinations in the bottom 800 generations (or equivalently 20ky) of the tree. The next section discusses the inference of version 2 of SMC^2 which includes FSDR.

2.5 Simulated data results for version 2

The FSDR implementation should increase the power of inferring parameters associated with recent epochs. In the bottleneck model we see marginal improvements in $\hat{N}_e$ for epoch 0 and 1 when analysing 8 samples (Figure 2.19). When using FS, fewer of the replicates show the extreme bias in $\hat{N}_e{}^0$ which is prominent with the naive sampler. FSDR shows further modest improvements over FS.

Returning to the uni-directional migration model we see an improvement in the migration estimates when the FSDR is employed (Figure 2.20). If we consider a slighlty more complex model, with population size changes and periods with and



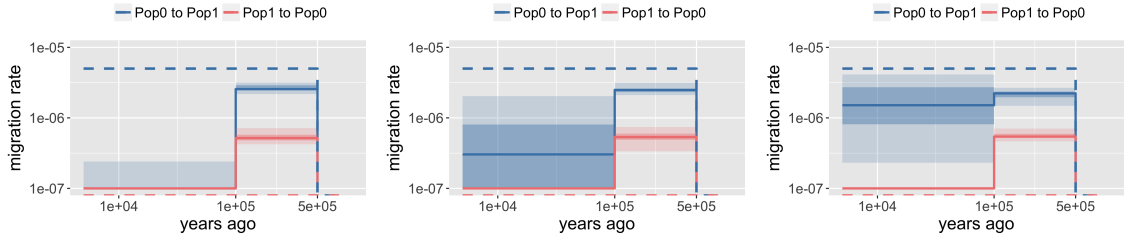
(a) naive sampler (b) FS (c) FSDR

Figure 2.19: Bottleneck model; 8 samples; 3k particles; the black line is the true N_e.

without migration (period of migration model in Appendix B), we again see improved estimates with the FSDR (Figure 2.21). As the naive sampler failed to infer migration over a large time scale, we focus on the period 0-56kya (2240 generations) for the two population models.

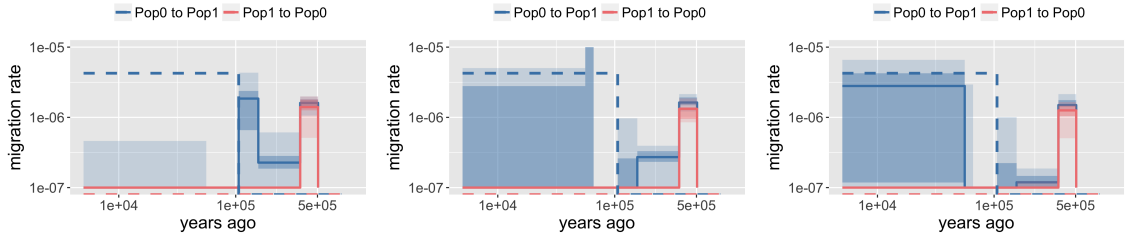
The focused sampler is designed to increase the number of recombinations in a selected time section in order to increase both the opportunity and the event count. By increasing both, we increase the power to infer the parameter, without biasing the estimates. The uni-directional migration and period of migration models show an increase in power to infer the presence of migration. The split no migration model provides reassurance that SMC^2 with the FSDR does not lose the ability to infer a period of no migration (Figure 2.22).

The experiments above show that the FSDR improves the estimates of migration rates in the recent epochs (Figure 2.20 and Figure 2.21). We do not see an improvement in the epoch immediately following the split, hereby refered to as the post-split epoch, as the FSDR focuses on transitions in the bottom of the tree. We believe the overestimation of migration rates in the post-split epoch is due to a shallow likelihood function. The data cannot easily distinguish between no migration and a migration in the post-split epoch with a resulting coalescence before the split. To address this problem we tried using the FSDR to focus on the time section around the split time as well as the bottom of the tree. We refer to this as multi-tiered focusing. In this case we focus on the times 0-56kya and the 56ky since the split time 450-506kya. Fig-



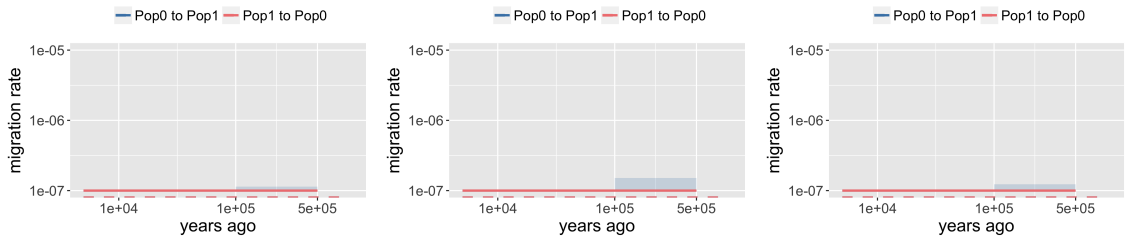
(a) naive sampler (b) FS (c) FSDR

Figure 2.20: Uni-directional migration model; 8 samples; 3k particles; the truth is represented by the dashed lines.



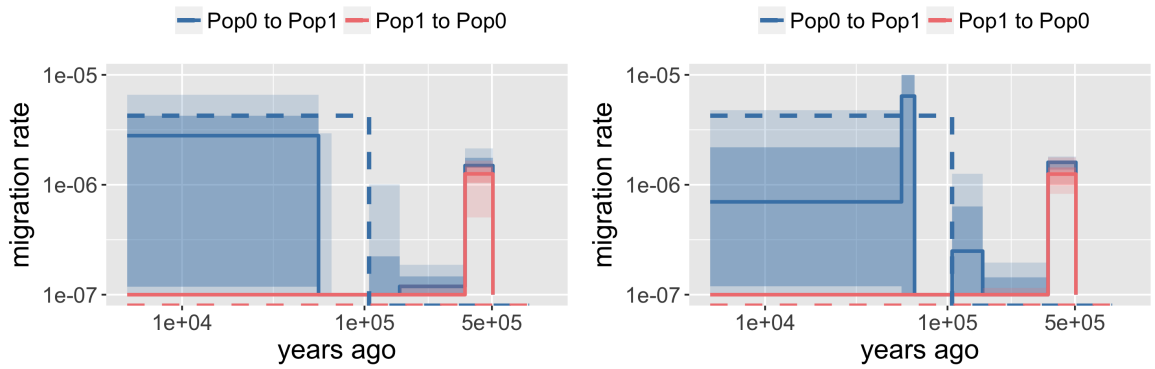
(a) naive sampler (b) FS (c) FSDR

Figure 2.21: period of migration model; 8 samples; 3k particles; the truth is represented by the dashed lines.



(a) naive sampler (b) FS (c) FSDR

Figure 2.22: Split no migration model; 8 samples; 3k particles; the truth is represented by the dashed lines.



(a) FSDR

(b) FSDR multi-tiered focusing

Figure 2.23: Focus on only the bottom of the tree, or both the bottom and the post-split epoch; the truth is represented by the dashed lines.

ure 2.23 shows the multi-tiered FSDR does not improve the estimates of migration in the post-split epoch. While the multi-tiered FSDR should increase the diversity of transitions in this region, it seems the difference in $g(y|x)$ is not great enough to accurately infer the absense of migration. The issue of inaccurate post-split epoch migration inference remains an unsolved problem which we will discuss further in the next chapter.

2.6 Guided recombination sampling

Another approach to improving the sampler is to utilise data from previous EM iterations. The FSDR broadens the tails of the proposal distribution to better explore the state space, but this is a naive data-agnostic exploration. A more sophisticated proposal distribution would utilise the available information on the location of likely recombinations, to increase the particle coverage in high-posterior density regions of the state space.

If we record the posterior density of recombinations along the sequence, we get a sense of where the algorithm would benefit from oversampling recombinations. This may be particularly advantageous with real data where the recombinations will not be evenly dispersed along the genome, but cluster in recombination hotspots. We discuss incorporating known recombination hotspots in the transition distribution in Section 2.7, here we are considering using them for the proposal distribution only and letting the data support their existence, rather than using a hotspot map.

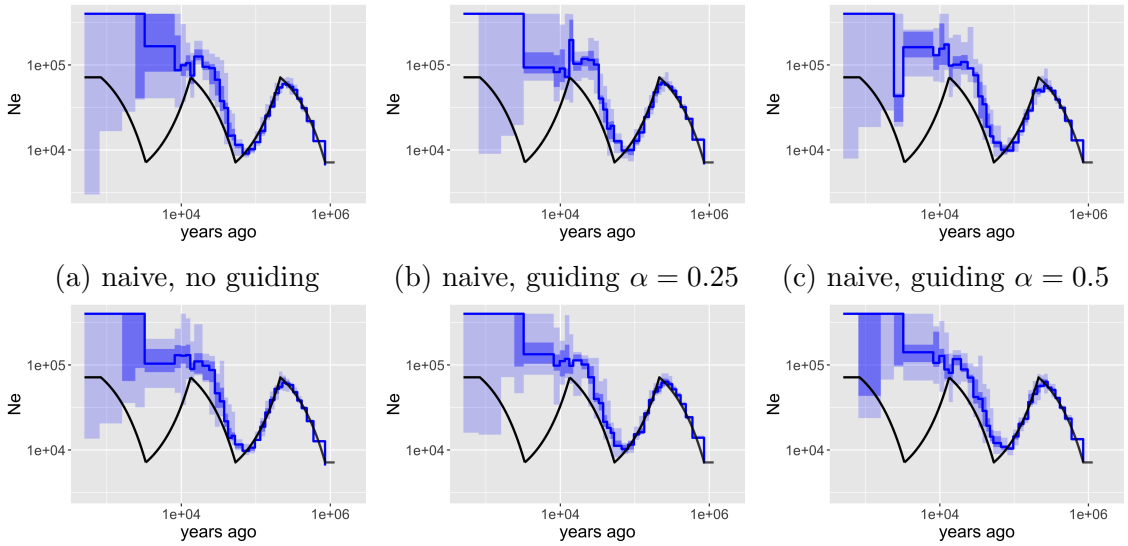
In addition to learning the sequence positions of recombinations, we can learn where in the tree space these high-support recombinations occurred. Once one EM iteration has been completed, we have a set of recombination events, each with a weight reflecting the particles which support that event. Thus far we have used these recombination events only to update $\hat{\rho}$. However, these events carry information on where likely recombinations occurred in sequence position and time, and on which branch of the tree. The matter of shifting the proposal distribution given this information is complicated by the fact that a recorded recombination will have occurred on one particular tree topology. We want to use the recorded recombinations to alter the proposal on all particles, many of which will not share the topology of the particles that provide support for the recombination.

To harness this information, for each haplotype we record the sequence position of recombination events above that sample, with the associated weight for the event.

The weight of an event is $\sum_{i \in G} w_{k+L}(X^{(i)}_{1:k+L})$ where w_{k+L} are the normalised particle weights when the algorithm has scanned data up to position $k+L$, k is the sequence position of the event, L is the lag of the epoch the event occurs in, and G denotes the set of particles which contain the event. Events observed in one EM iteration are then used to alter the proposal distribution of the next EM iteration. The new proposal of recombinations is a mixture of the previous proposal and a guiding proposal. We give the guiding proposal weight α and the original weight $1-\alpha$. To create the guiding proposal, the genome is divided into segments of 100 bases, and the posterior opportunity and event count are recorded for each segment and each sample. These measurements will be incredibly noisy, so we employ a binary segmentation scheme similar to that described in [24] to determine where we observe changes in the recombination rate. Then for a branch above only three samples with rates r_1, r_2, r_3, recombinations are sampled according to the average rate $\frac{1}{3}\sum_{i=1}^{3} r_i$.

To analyse the benefit of using previous EM steps to guide the sampler, we ran the guided recombination sampler (GRS) method both with and without FSDR. We expected GRS could replace FSDR as the sampler would learn when recombinations are likely to occur in the bottom of the tree space. We chose the zigzag model used in [74] as this model has variable N_e in the recent past, where we expect the GRS to provide the most benefit. Unfortunately, we see no improvement (Figure 2.24). For all versions of SMC^2 tested, our inference is comparable to that of MSMC run with 2 samples (Figure 2.25), suggesting that for this model SMC^2 is unable to sample the necessary coalescent events low in the tree.

We suspect the lack of improvement with GRS is due to the changing inferred demographic model. A recombination event may show strong support under the initial parameters simply because the particles simulated under the initial model do not fit the data well. We tried to supress this effect by testing a couple of values for α. A better approach may be to run SMC^2 for 10 EM iterations to get the parameters



(a) naive, no guiding (b) naive, guiding $\alpha = 0.25$ (c) naive, guiding $\alpha = 0.5$

(d) FSDR, no guiding (e) FSDR, guiding $\alpha = 0.25$ (f) FSDR, guiding $\alpha = 0.5$

Figure 2.24: Zigzag model; 8 samples; 10k particles; the black line is the true N_e.

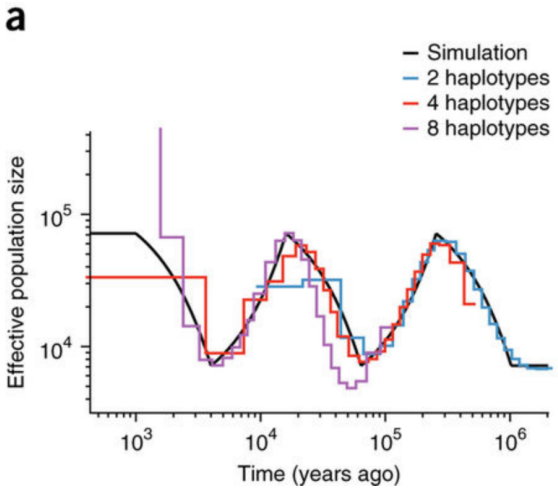


Figure 2.25: MSMC results for zigzag model with 2, 4, and 8 samples from Schiffels and Durbin [74].

closer to the true values, and then implement the recombination guiding to increase precision. This idea is similar to the idea of running early EM iterations with few particles and then increasing the number to reach convergence [9, p. 403], but does not involve the increase in computational resources. For now we have abandoned this sampler to focus instead on a lookahead sampling procedure discussed in Section 3.5.1.

2.7 Data-based model adjustments

We have shown that SMC^2 is able to infer parameters from data simulated under a variety of demographic models. However, all of these experiments used unrealistic data. In this section we explore a few of the complications that may arise from real data.

2.7.1 Phasing

For diploid species, like humans, there is an additional level of uncertainty in the data due to unknown phasing. If the phase of a sample is unknown at a heterozygous site the mutation cannot be attributed to a particular leaf of the tree. For example, if there are two diploid samples one of which is heterozygous and the other homozygous at site j, then we have either y_j =1000 or y_j =0100. We will represent this unphased mutation pattern as y'_j =//00.

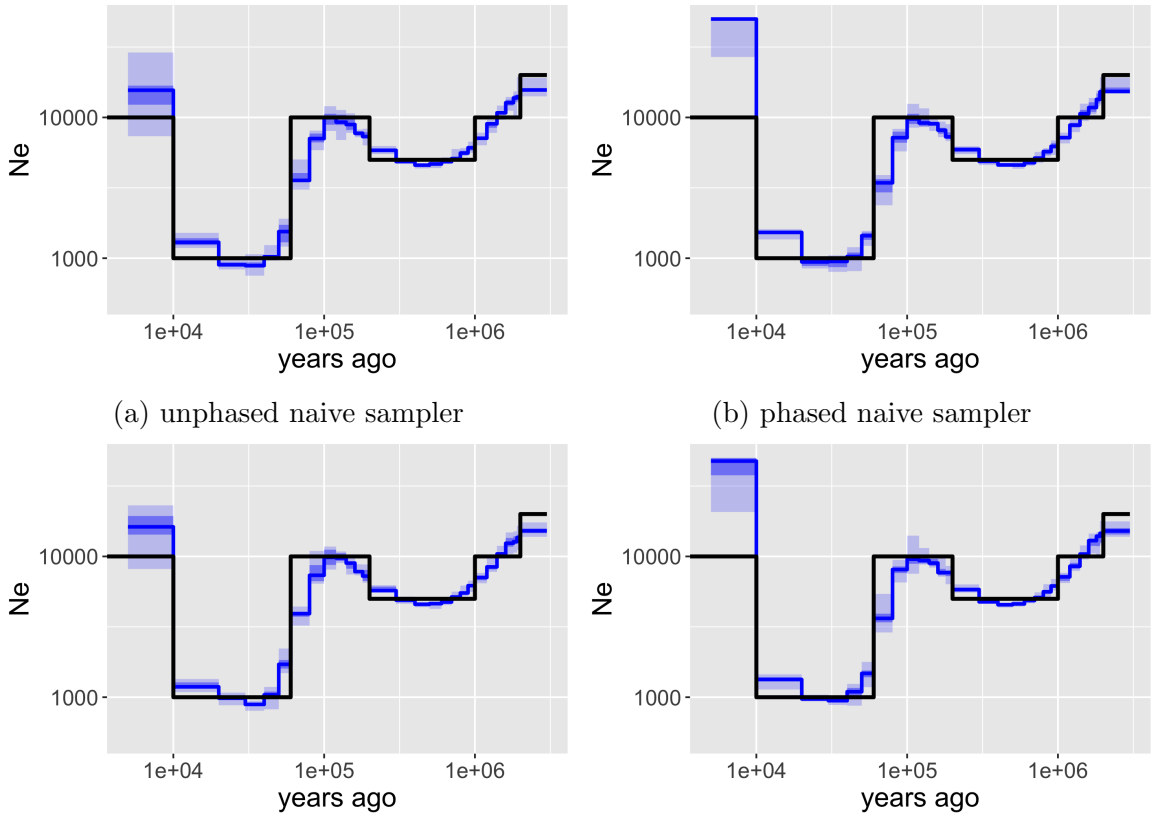
Clearly an adjustment must be made to the emission distribution to account for unphased data y'_j. We chose the naive approach of averaging the likelihoods of all possible phasings so that in the example

$$g(y'_j = //00|x_j) = \frac{1}{2}\Big(g(y_j = 1000|x_j) + g(y_j = 0100|x_j)\Big).$$

While this approach is simple to implement, it is not a perfect fix. By not making use of the phase of heterozygous sites in the model, the data becomes less informative.

To show that demographic parameters can be correctly inferred without phasing information, we simulated data from the bottleneck model (Appendix B) and treated the data as 4 unphased diploid individuals. Running SMC^2 produced the estimates in Figure 2.26, which are close to the true values.

Surprisingly, the estimates from analysing the unphased data are better than the estimates from analysing the phased data (Figure 2.26). We attribute the apparent



(a) unphased naive sampler

(b) phased naive sampler

(c) unphased FSDR

(d) phased FSDR

Figure 2.26: Estimates for the bottleneck model with or without phasing; the black line is the true N_e.

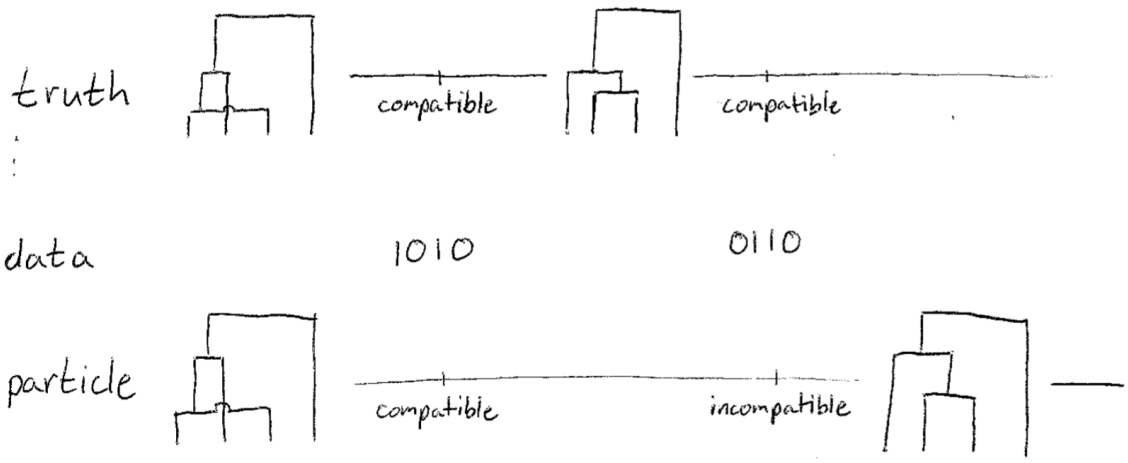
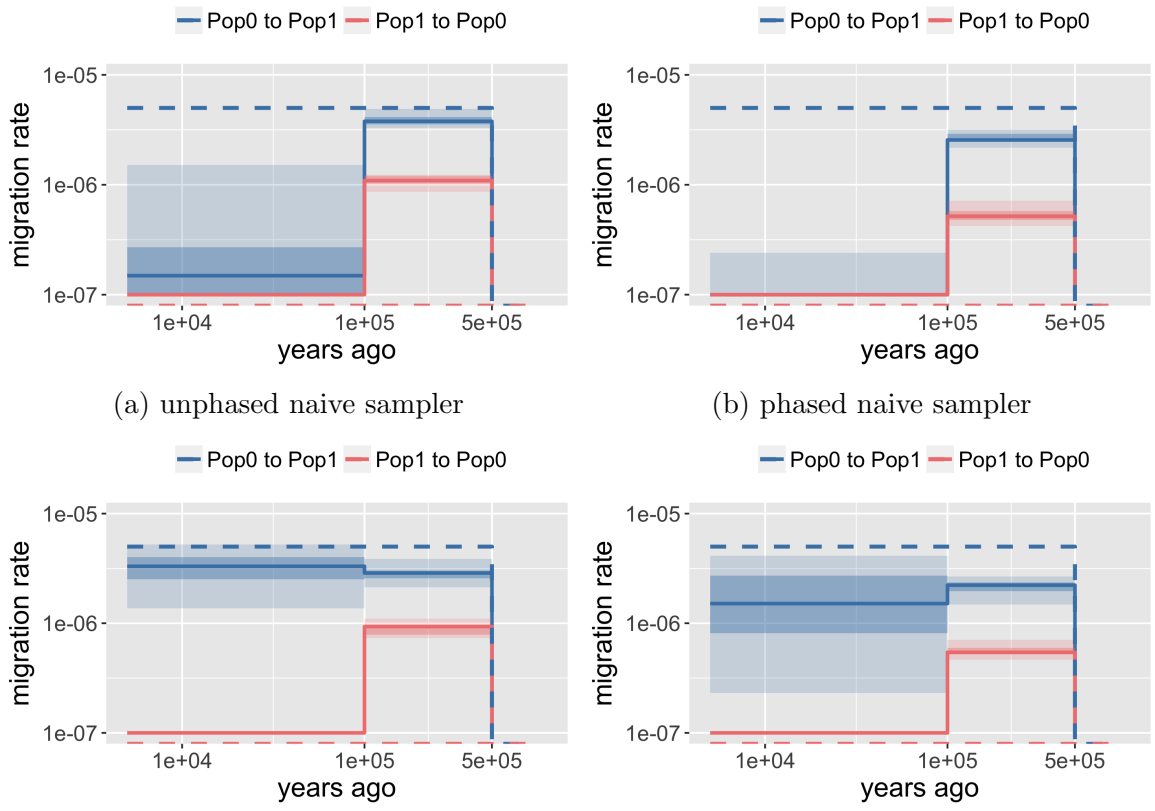


Figure 2.27: A particle has an incompatible tree at one position, but otherwise is similar to the truth.

improvement to the fact that the resampling process will be more forgiving due to averaging over possible phasings. Under the phased emission distribution a particle is more likely to be assigned a very small weight due to the tree topology not being compatible with the data and a single mutation event. However, with the unphased emission distribution the particles are not culled as quickly (the unphased version has 97±1% the resamplings of the phased version). This may benefit particles which experience a high probability transition at a position slightly to the right of the true transition event (see Figure 2.27).

The above finding is both reassuring and worrying. It is reassuring that SMC^2 is capable of correctly inferring parameters without phasing information, which is often not available. However, it is worrying that our sampler struggles to the extent that removing information improves the results. Both the naive and the FSDR sampler benefit from removing phase information.

It seems SMC^2 can correctly infer effective population sizes using unphased data, but we saw in Section 2.3 that inferring migration rates is more difficult than inferring population sizes. As such, we should check that the our estimates of migration rate are not greatly influenced by a lack of phasing information. To this end, we ran SMC^2



(a) unphased naive sampler

(b) phased naive sampler

(c) unphased FSDR

(d) phased FSDR

Figure 2.28: Migration estimates for the uni-directional migration model with or without phasing; the truth is represented by the dashed lines.

on both phased and unphased simulated data of 4 diploid individuals from the uni-directional migration demographic model (Figure 2.28). As in the bottleneck model, inference is improved by removing the phase information, now to an even greater extent (Figure 2.28).

2.7.2 Ancestral allele awareness

The simulated data analyses have so far neglected some informative data that is often available. The likelihood of the sequence data has been computed assuming an unknown root state for the tree. With the state of the root (the ancestral allele) known, the emission density from Felsenstein's algorithm uses a prior of $\pi_0 = 1, \pi_1 = 0$. In many applications, this state can be found by use of an outgroup, if not for the

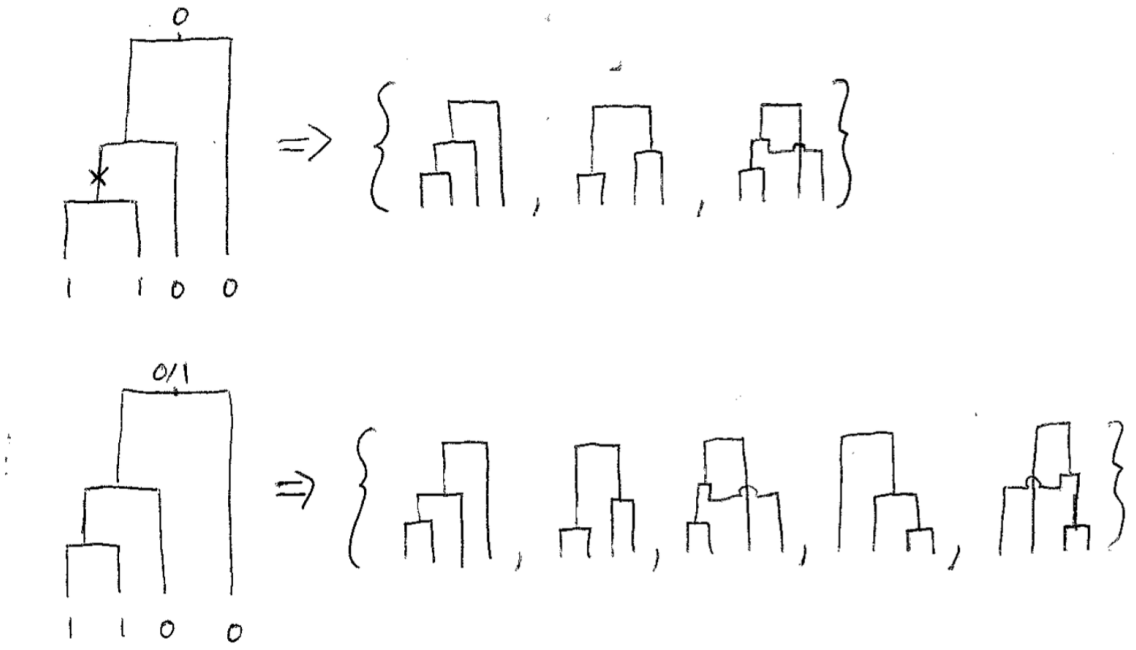


Figure 2.29: A known ancestral allele limits the possible topologies.

whole-genome then for large sections of the genome.

SMC^2 relies on mutation patterns to determine the probable genealogies at a given genome position. The knowledge of which samples are below a mutation should be a powerful indicator of likely trees. Consider the example in Figure 2.29 where the true topology is displayed on the left. If the ancestral allele is known then there are three possible topologies. If the ancestral allele is unknown there are five possible topologies. Of course the likelihood of the particle is not solely determined by the mutation pattern at a single position. The data around this site should be informative as to which topologies are likely and in aggregate the algorithm should favour the true topology. Still, the main challenge is to sample sufficiently from regions of the state-space with high posterior, so it should be more efficient to encourage resampling of the best particles.

Furthermore, ancestry information may boost our ability to infer directional migration, which is the driving aim of this method. In Figure 2.30 an ancestral allele of 0 necessitates either a migration from Pop 0 to Pop 1 or a long period of no coalescence

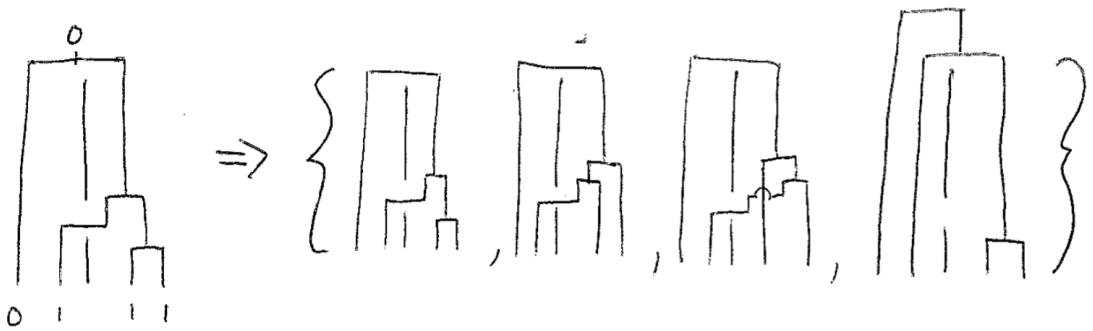


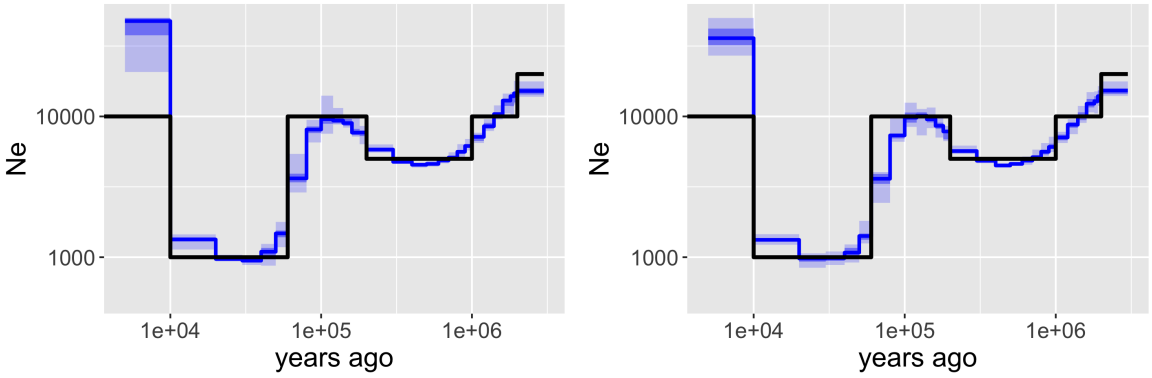
Figure 2.30: A known ancestral allele indicates probable migration.

in Pop 0. Particles with the latter topology will get a small weight as the surrounding data will not have many singletons for sample 2, and so an emission distribution which uses the ancestry information will shape particles to the correct topology. Without the known ancestral allele, many other topologies will be considered likely as they fit with the possibility that the mutation is a singleton on sample 1. Considering the results of Section 2.5 (which showed that obtaining accurate migration estimation requires sophisticated sampling), we should use every available option to drive our particles toward regions of the state space with high posterior density.

Running SMC^2 with or without known ancestry information on 8 samples simulated from the bottleneck model, we see little difference in the estimates (Figure 2.31). The 4 sample case and the naive sampler all see similar results. Ancestral information does not seem to improve population size estimation.

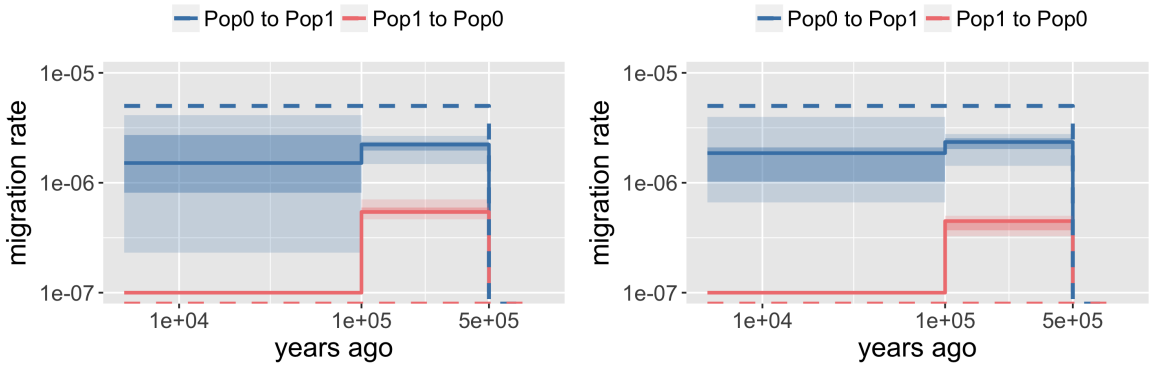
The effect of known ancestry on migration inference is slightly more promising (Figure 2.32). The ancestry information seems to increase the precision of migration estimates for the most recent epoch, and help distinguish directionality in the post-split epoch. However, this effect is modest and no drastic improvement is observed.

The lack of dramatic improvement is an initially surprising result. One explanation could be that the posterior distribution is not significantly altered by including knowledge of the ancestral alleles because the sequence data itself is sufficiently informative of the ancestry. Figures 2.29 and 2.30 detail the benefit ancestral information



(a) unknown ancestral allele (b) known ancestral allele

Figure 2.31: Bottleneck model; 8 samples; with or without a known ancestral allele; the black line is the true N_e.



(a) unknown ancestral allele (b) known ancestral allele

Figure 2.32: Uni-directional migration model; 8 samples; with or without a known ancestral allele; the truth is represented by the dashed lines.

can provide for inferring a genealogy at a single site. Perhaps when the context around the genealogy is considered, the ancestral allele becomes superfluous information.

Another possibility is that the benefits described in Figures 2.29 and 2.30 are offset by the difficulty of sampling from a peakier posterior distribution. This was our reasoning for obtaining better results from unphased data than phased data in Section 2.7.1. This theory is also compatible with the improvements we see due to FSDR. At face value these results are not comparable as focused sampling concerns transitions and ancestry information concerns emissions. However, we employ focused sampling to broaden the proposal distribution in order to oversample rare transitions and delay the transition penalty to maintain a variety of trees. The known ancestral allele does not alter the proposal distribution, but it does impose a peakier likelihood which will reduce the variety of the particles through resampling. When it comes to importance weights, the FSDR provides a buffer to increase particle diversity whereas known ancestral alleles reduce particle diversity.

It is still difficult to imagine that maintaining a collection of particles that are incompatible with the data is helpful. I suspect the reason we would want to maintain a completely incompatible particle is the possibility that the particle will become compatible. The parameter inference is based on the tree transitions that are deemed likely by the data. The size of the ARG state-space precludes the possibility of considering all possible transitions at each position of the genome. We must rely on a small sample of ARGs. If one particle makes a branch movement very similar to the true ARG but at a position shifted slightly to the left or the right, the majority of the genome data would support this, but a small region may doom the particle. It is best for parameter estimation that this particle is maintained until a long stretch of data to the right of the sampled transition has influenced the particle's weight.

2.7.3 Other

There are various other situations that should be considered. A few of these we explore further in the next chapter, but some remain open questions.

Sequencing the genome of an individual is much cheaper now than 10 years ago. Unfortunately, getting an error-free genome sequence today is incredibly costly, if not impossible. Most sequence data will have false positive or true negative variant calls. We have not tested how robust SMC^2 is to errors in the sequence data. In principle, the emission distribution could utilise the quality of each variant call. We have chosen not to pursue this as the field is moving towards more confident calls.

We have allowed for samples to have uncalled regions of the genome. If the called sequence has low quality scores in a region, that region is marked as masked in the input file, and that region of the sample does not affect the importance weights of the particles.

The transition distribution makes many simplifying assumptions about the biology of recombination. One of these assumptions is that the genome is neutrally evolving at all positions. In Chapter 3 we consider the effect of masking out genes which are certainly evolving under selective pressure. When aiming to infer population histories, non-neutral regions are a nuisance often ignored by demographic inference methods. However, there is much scientific interest in identifying non-neutral regions. When demography is known, SMC^2 could be reframed to identify regions evolving under selection (more details on this in Section 3.5.1).

Additional simplifications of the recombination process are the constant recombination rate and our restriction to crossover recombination events. Crossover recombination is a biological process whereby proteins create double stranded breaks in the genome and cross the chromosomes when repairing the break. These breaks do not occur uniformly along the sequence. The break sites have been associated with several sequence motifs and for many species (including humans) a recombination map

has been created. A known map of recombination hotspots could be incorporated into the transition distribution. Alternatively the proposal distribution could depend on the recombination map, without any change to the transition distribution. This should make the sampler more efficient by focusing on higher density regions of the state space. We ignore non-crossover recombination where a small segment of the chromosome will be derived from a different lineage. Non-crossover recombination causes strong dependencies in a small range of the sequence. This type of recombination could be modelled by using a closer approximation to the ARG than SMC'. This would be implemented in SMC^2 using SCRM's window size feature. However, the better approximation would have a computational cost.

The emission distribution, as we model it, has a large simplification of its own. We assume a constant mutation rate, independent of sequence context and type of mutation. These are known to have profound effects on the mutation rate. There is even evidence that mutation rates vary through time and between populations [35].

Genetic inheritance is a complicated process and mathematical models will always involve some simplifications. I expect most of the ideas above to be unneccessary complications for analysing human demography. Still, it would be useful to demonstrate this to be true. Moreover, we have developed this method with the hope that others will extend it to their own applications. Non-human data may have significant complications we have yet to consider, and so we hope this simulation-based method is flexible enough to be adapted to any problem of interest.

2.8 Runtime of SMC^2

The previous sections have demonstrated the capability of SMC^2 to infer demographic parameters, even when using a fairly modest number of particles. Of course, the Monte Carlo approximations would be improved by increasing the number of particles.

This could be particularly beneficial to estimation in multi-population models where the state space is larger. We have thus far restricted ourselves to 10,000 particles due to demands on compute time; memory requirements have not been a limiting factor. Here we briefly detail the computational burden and scalability of SMC^2.

For reference, the analysis in Figure 2.7d consisted of 10 replicates. Each of these replicates used 10,000 particles and 30 EM steps to analyse 200Mb of data from 4 haploid samples simulated under the bottleneck model. Each replicate took $\sim$ 18 CPU hours, and so $\sim$ 180 CPU hours for the entire analysis. The compute time scales linearly with the sequence length, number of samples, number of particles, and number of EM steps. Moreover, the compute time is dependent on the underlying demographic model. In a similar analysis the data was generated under a constant population size, but the setup was otherwise identical. This constant population size experiment took $\sim$ 40 CPU hours per replicate. The increase in runtime is a consequence of larger genealogies and the accompanying increase in recombinations and mutations. Multi-population models take even longer as the majority of genealogies have a TMRCA before the split time.

When several CPUs are available, real time can be saved by running the replicates independently. Runtimes can be further reduced by parallelising each replicate. This is done by dividing the data into chunks. Each EM step then consists of running sequential Monte Carlo on each chunk of data independently. Once all of the chunks have been processed, the summary statistics are averaged and then used to produce the new parameter estimates. This averaging step relies on the additive functional form of the statistics. In practice, we do not use chunks smaller than 20Mb, in order to retain long distance dependencies in the data.

2.9 Conclusion

We have developed an algorithm to estimate demographic parameters using sequential Monte Carlo methods to model the SMC′. We have shown that SMC^2 is capable of inferring population sizes and directional-migration rates.

Inferring parameters from sequence data is difficult due to the high-dimensional state space of ARGs. We have developed two sophisticated samplers to boost state space exploration, and concluded that the FSDR is best able to produce a high-posterior set of particles. In many cases, the algorithm infers an approximately correct demography, although some limitations remain (e.g. post-split epoch migration estimation).

With SMC^2 built, we now turn our attention to real data. The next chapter explores what SMC^2 can and cannot tell us about the demographic history of Neanderthals and their relation to anatomically modern human populations.

Chapter 3

Neanderthal history

3.1 Introduction

A new age of self-discovery has started for humans. Recent advances in sequencing technology have brought opportunities to investigate the molecular history of our species. Genomic data is yielding insights unattainable through previous anthropological methods alone [73, 37, 45]. The evidence teased out of genomes has been, and will continue to enhance our understanding of human history and evolution.

One particular area of interest is the fate of the sister species of modern humans. The recent discovery of Neanderthal and Denisovan remains harbouring DNA has opened the door to genomic comparisons of these species. This has not been straightforward, due to the sparsity of surviving endogenous DNA, the accumulation of DNA damage (particularly deamination), and overwhelming contamination from foreign bacterial and modern-day human sources. Nevertheless, scientists in the field have tackled these problems head-on and a boom in ancient DNA studies has begun [26, 51, 27].

As of 2017, researchers have access to three high quality archaic whole-genomes. Two of these are Neanderthal genomes. The first high-quality Neanderthal genome

was published in 2014 [68]. The sample was extracted by Prüfer et al. from a toe phalanx discovered in Denisova Cave in the Altai Mountains in 2010. The second high-quality Neanderthal genome is currently in preparation, but has been made available to researchers by the Max Planck Institute for Evolutionary Anthropology [54]. This sample came from the Vindija cave in Croatia. The third fully sequenced archaic genome is that of a Denisovan. This sample was found in 2008 in the same cave as the Altai Neanderthal, indicating co-localisation of the sister species, although the Denisovan sample is believed to be older than the Neanderthal [68].

The full genomes of archaic hominins are advancing knowledge beyond that previously obtained from mitochondrial DNA (mtDNA). Before the technological advances that enabled nuclear DNA extraction and sequencing, the field relied on mtDNA. Each cell carries hundreds of copies of mtDNA, making it far more abundant than nuclear DNA, and increasing the chance of DNA preservation. An early analysis of a 360bp region of mtDNA found no evidence of Neanderthal introgression into modern humans [47]. Further studies of additional Neanderthal samples substantiated this claim [75]. Sequencing of the complete Neanderthal mitochondrial genome furthered this theory, as it found the variation of mtDNA of a Vindija Neanderthal to be outside the variation between modern humans, indicating the common mtDNA ancestor to modern humans post-dates the split from the Neanderthal lineage [29].

Two landmark studies changed our understanding of our shared history with these archaic hominins by analysing nuclear DNA [72, 68] (summarised in Figure 3.1). The mitochondrial genome of the Denisovan sample is deeply diverged from those of Neanderthals and anatomically-modern humans (AMH) [68]. However, a phylogenetic reconstruction based on autosomal DNA revealed the Denisovans and Neanderthals are sister groups with respect to modern humans [72, 68]. Prüfer et al. used the complete genomes of the Altai Neanderthal and Denisovan to place their divergence time at 380-470kya. They placed the split time between AMH and the shared archaic

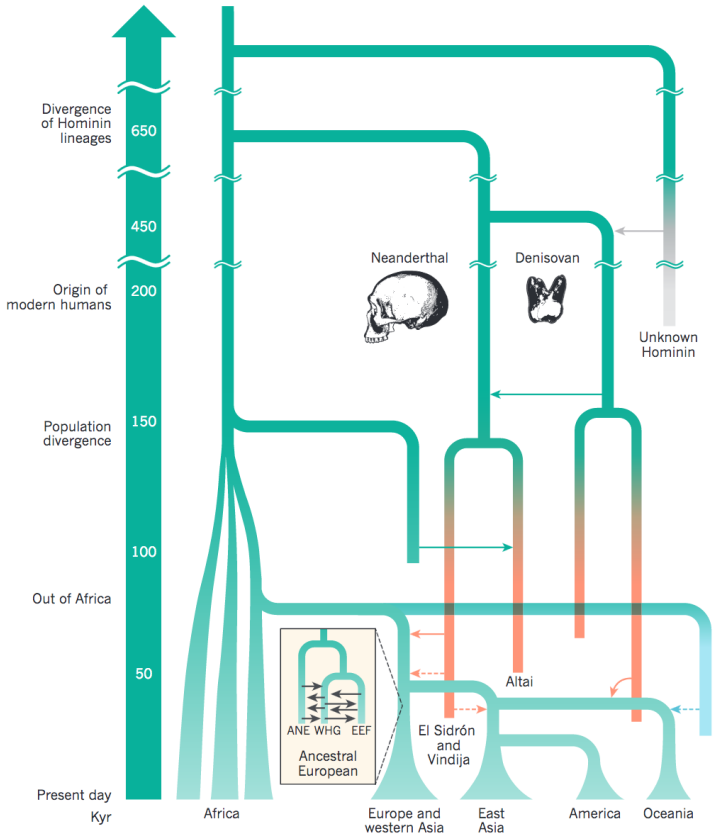


Figure 3.1: Summary of human evolutionary history borrowed from Nielsen et al. [61].

lineage at the earlier date of 550-765kya.

Both Neanderthal nuclear genomes have shown evidence of introgression into the ancestors of modern non-African populations. On average, ∼2% of the genome of a present-day non-African individual is descended from a Neanderthal lineage [68]. The first study of the nuclear genome of the Vindija Neanderthal, which analysed only chromosome 21, concluded the population of Neanderthals which introgressed into AMH is more closely related to the Vindija individual than the Altai individual [48]. The same study additionally detected a signal of migration from a diverged AMH population into the Altai Neanderthal ∼100kya (note that for this chapter we refer to forwards-in-time migration). Although there was no evidence of AMH-Neanderthal admixture in the mitochondrial genomes, analysis of the nuclear genome in multiple

samples and multiple studies have shown interbreeding has resulted in a signifcant genetic contribution of Neanderthals to modern humans.

Several studies found the Denisovans have also contributed to modern human genomes, although in a less global manner. They found Denisovan lineages contributed on average 5% of the genomes of modern day Oceanians [72, 68, 52]. However, only 0.2% of the genomes of mainland Asians was found to derive from Denisovans, suggesting the admixture occurred after the divergence of these populations.

Regions of extreme divergence have been identified in the Denisovan genome, which may be evidence of a more deeply diverged hominin group that introgressed into the Denisovan. This theory is consistent with the finding that the mitochondrial genomes of Neanderthals and AMH are more closely related to each other than the Denisovan mtDNA, despite the nuclear phylogeny. The Denisovan genome has provided evidence for a potential fourth "unknown archaic hominin" sister species which diverged from other hominins 0.9-1.4Mya [68].

While the contribution of archaic hominins to modern human genomes is fascinating, it is not the only subject of interest. Many studies investigate the traits of the archaics, identifying adaptive alleles, some of which were introduced into human lineages [57, 15, 40, 69]. However, we as a species are generally fascinated by these archaic species, not just because they influence our own genetics, but also because the Neanderthals and Denisovans were not so different from our ancestors, their contemporaries. The most notable difference is that AMH societies survived and theirs did not. PSMC has been used to infer the population sizes and found that while the Altai Neanderthal and Denisovan populations seemed to dwindle out (possibly from competition with our ancestors), the Vindija Neanderthal population grew [68, 48]. If the Vindija population did indeed experience an expansion after 100kya, to the point that it was substantially larger than the contemporary non-African population, then what caused its extinction? Wide ranging theories from volcanic activity to a

lack of trade practices have been proposed to explain the extinction of Neanderthals and survival of AMH [3, 38], but these theories largely rely on an already dwindling Neanderthal population.

In this chapter we investigate the recently sequenced Vindija individual and its relation to modern human populations using our newly developed method SMC^2. We have chosen to focus on the Vindija individual as phylogenetic analyses have found the Vindija to be more closely related to the introgressing population than the Altai Neanderthal [68, 48]. The approximate time of Neanderthal introgression has been determined by comparisons of ancient AMH and Neanderthal genomes [26]. We endeavour to infer this time directly from a single ancient sample and two modern Europeans. The Kuhlwilm et al. paper, which analysed chromosome 21 of the Vindija sample using G-PhoCS, provides a useful comparison for our results.

3.2 Proof of principle models

The previous chapter demonstrated the capability of SMC^2 to infer demographic parameters in a variety of demographic models. However, the analysis of real ancient samples introduces a number of complications not yet discussed. One of these complications is that the samples are no longer contemporaneous. In a model with non-contemporaneous samples, some of the leaves of the genealogies will not be placed at present day. This is handled in SMC^2 by specifying the number of generations before the present when the samples lived, and only modelling the associated lineages back from that time. For Neanderthal populations, we additionally assume the population ceased to exist around the time of the sample, and do not allow for migration between the populations in the years since the sample time. In this section we will consider data simulated from models with non-contemporaneous samples, in order to evaluate the performance of SMC^2 in these cases.

We use the findings of the Prüfer et al. and Kuhlwilm et al. papers to motivate our models. One of these models includes a period of uni-directional migration from the archaic population to the extant population (forward-in-time migration). The other model has bi-directional migration in the same period. We chose the migration rate to be approximately consistent with 2% of the AMH genome originating in the archaic population. Both models have N_e curves similar to the putative population histories of Europeans and the Vindija Neanderthals, and we simulate four haploid samples from the AMH population and two from the archaic population. In addition to testing the performance when using non-contemporaneous samples, these models should demonstrate that neither the skewed sample number nor the differing histories confound our ability to infer migration. See Appendix B for details on the human-archaic models.

Figure 3.2 shows the inference from data generated under the uni-directional migration model. As in the previous chapter, we display the results of 10 independent runs of SMC^2 to convey uncertainty in the estimates. The light band spans the 0th to 100th quantile, the dark band spans the 25th to 75th quantile, and the solid line is the median estimate. The estimates of migration are greatly improved by increasing the focus strength. Both focus strength 3 and 5 are better able to infer migration in the period 44-100kya than the naive sampler. They do drastically underestimate the migration rate for 100-150kya, indicating the precise placement of migration events in time has not yet been achieved. It is clear SMC^2 with FSDR is capable of inferring the true demography in this model, despite the asymmetry of the model and data.

Ideally, we would like to be able to infer the true values in any demographic scenario. However, it is intractable to test every case, and there are certainly scenarios where our default of 10,000 particles will not be sufficient (for example models with very low levels of migration). One open question for Neanderthal-AMH interaction is to what extent modern human ancestors contributed to the Neanderthal population.

We need to check that the differing N_e histories, sample counts, and sample dates do not impair our ability to infer migration in either direction. Analysis of the bi-directional Neanderthal-AMH model (Figure 3.3) shows promising results, but also raises some concerns. As in the uni-directional case, the focused sampler is needed to infer both migration and N_e. This is reassuring, as we cannot base our choice of sampler on an unknown truth. Overall, SMC^2 succeeds in inferring the rate and directionality of migration in both the uni-directional and bi-directional cases.

The concerning matter is the somewhat skewed inference of the directional migration rates, which is most easily seen in the $s^{0-56kya} = 5$ case in Figure 3.3. This skew becomes more obvious for the $s^{0-56kya} = 3$ case if we examine the evolution of the parameter estimates over the EM iterations (Figure 3.4). The evolution of estimates for each run show consistent upward movement towards the truth in Neanderthal-to-AMH estimates. However, the AMH-to-Neanderthal estimates initially move down away from the truth before starting to converge. This highlights the problem that migration estimates can easily be absorbed at 0. We will discuss potential solutions in Section 3.5.1, but for now this remains a possibility. It seems in these models absorption at a false value was avoided, but if the true rate were lower or if the N_e curves were different, false absorption could occur, leading to a false conclusion of uni-directional migration.

The AMH-to-Neanderthal migration parameters appear to be more easily underestimated in the 44-100kya range. This directionality does relatively well in the period of no true migration 100-500kya. For this time period the other direction (Neanderthal-to-AMH) overestimates migration. It seems there is a consistent underestimation in one direction, and consistent overestimation in the other direction. At least this is true for 40 EM iterations; we are unable to test longer term behaviour of estimates due to computational limitations. For practical purposes, SMC^2 can correctly infer migration when the signal is strong, but may miss periods of weak

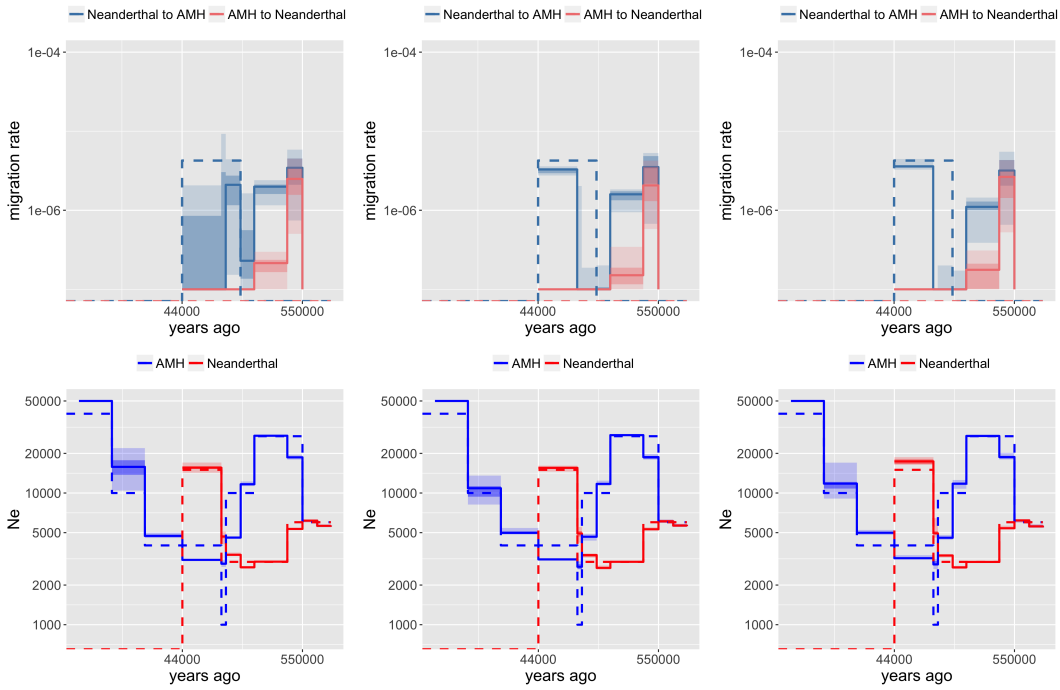


Figure 3.2: Compare focus strength of FSDR; uni-directional migration model; from left to right $s^{0-56kya} = 1$ (naive sampler), $s^{0-56kya} = 3$, $s^{0-56kya} = 5$; 6 samples; 10k particles; the truth is represented by the dashed lines.

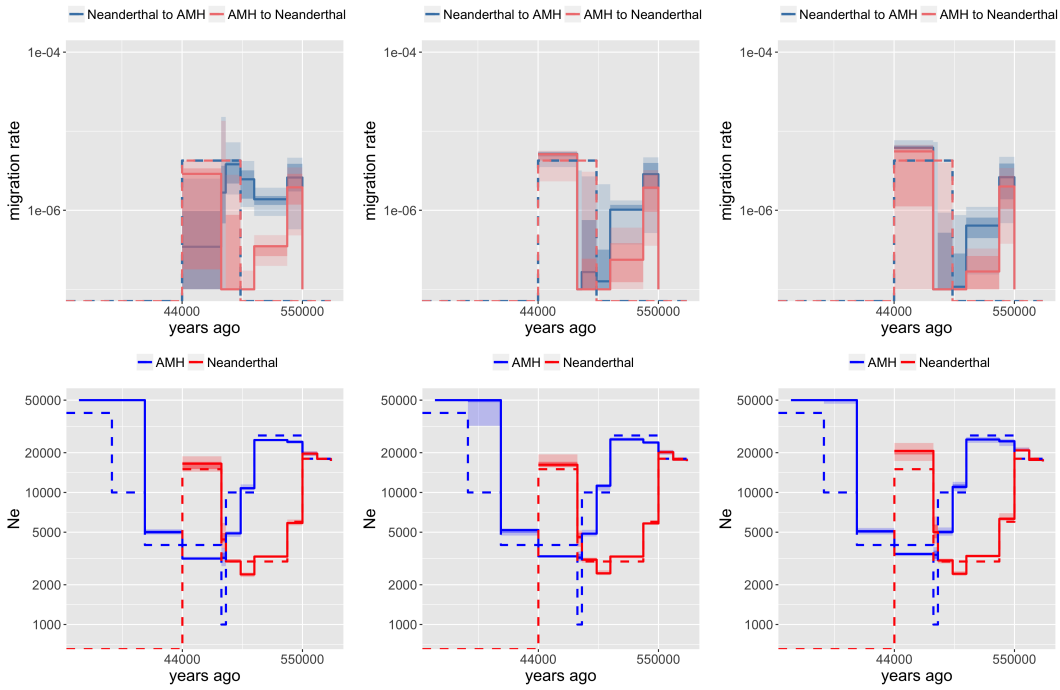


Figure 3.3: Compare focus strength; bi-directional migration model; from left to right $s^{0-56kya} = 1$ (naive sampler), $s^{0-56kya} = 3$, $s^{0-56kya} = 5$; 6 samples; 10k particles; the truth is represented by the dashed lines.

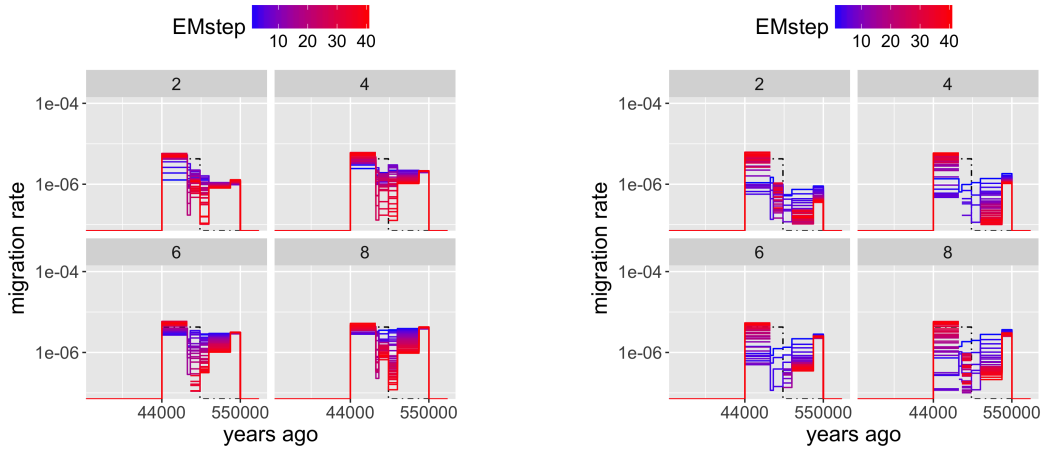


Figure 3.4: Comparison of directional migration rate estimate evolution for 4 of our replicate analyses (*Left*: Neanderthal-to-AMH; *Right*: AMH-to-Neanderthal); bi-directional migration model; 6 samples; 10k particles; the truth is represented by the dashed lines.

migration and will overestimate migration in the post-split epoch.

To interrogate the potential problem of incorrectly inferring no migration, we simulated data under the same bi-directional migration model, but with a weaker migration rate (2.125e-6). SMC^2 incorrectly infers uni-directional migration in this case (Figure 3.5). It seems SMC^2 is only sensitive to bi-directional migration above rate 2.125e-6, at least for 10,000 particles and 100Mb of data. When applying to real data, we should qualify that inferring a migration rate of zero does not preclude the possibility of low levels of migration.

We pushed the method even further by trying to infer migration in a model with a substantially lower level of migration. For this model we used a rate of 4.25e-7, which is one-tenth of our standard value. This would correspond to about 0.2% of the genome being derived from Neanderthals according to our crude calculation in Appendix B.2.4. With this low level of migration, SMC^2 is not able to detect any migration for either the bi-directional or uni-directional case (Figure 3.6).

The proof of principle analyses above have benefited from our knowledge of the true model. In SMC^2, at least in its current implementation, the epoch boundaries

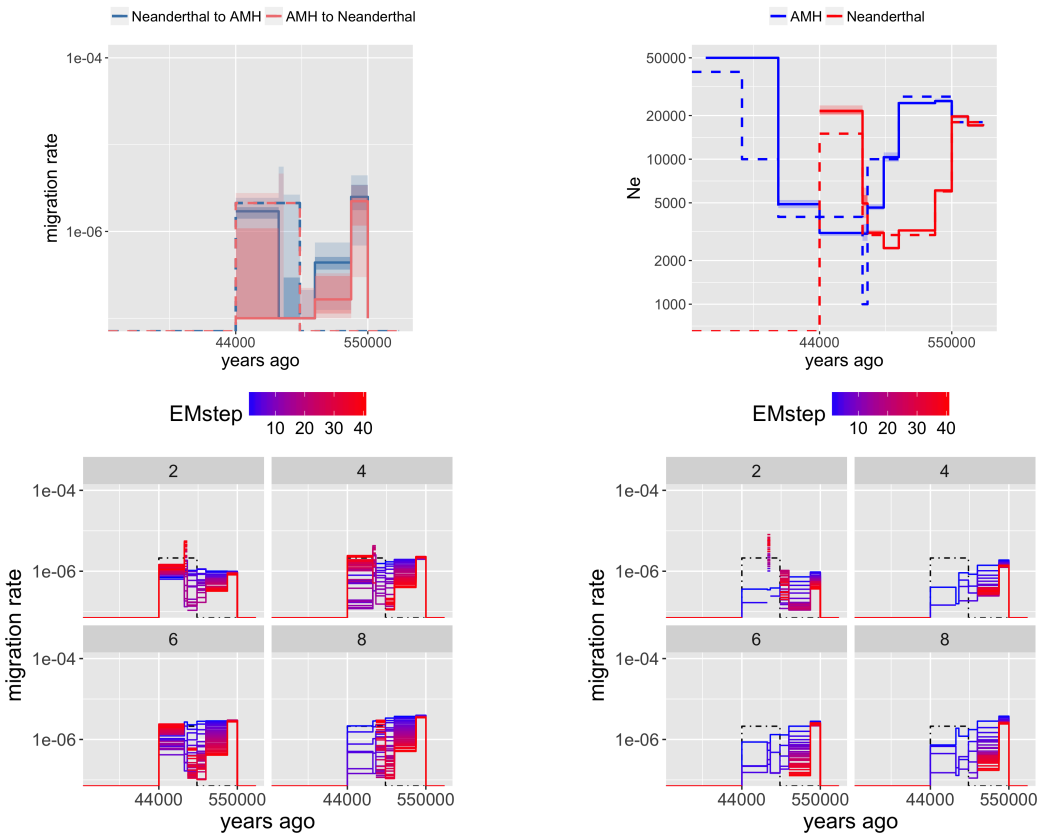


Figure 3.5: Bi-directional migration model with weaker migration; *Bottom left*: Neanderthal-to-AMH; *Bottom right*: AMH-to-Neanderthal; 6 samples; 10k particles; the truth is represented by the dashed lines.

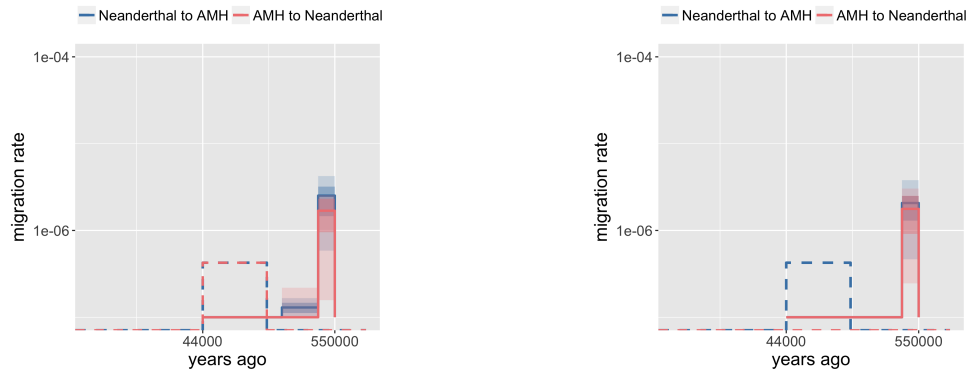


Figure 3.6: Bi-directional and uni-directional migration models with very weak migration; 6 samples; 10k particles; the truth is represented by the dashed lines.

must be set manually. In real-data applications the times of major demographic changes will not be known, and so typically many small epochs are used. We now evaluate the performance of SMC^2 when using many small epochs.

The first thing to notice in the small epoch analyses is the increase in estimate variation (Figure 3.7). With a larger parameter space, it is difficult to distinguish between all of the possible curves. This effect is most pronounced in the migration estimates, but is true of the N_e estimates as well (compare Figure 3.7 to Figure 3.2). For the bi-directional migration model, we see a failure to infer migration in one direction (Figure 3.8). The reduced opportunity sufficient statistic explains the large Monte Carlo variance in these estimates (Figure 3.9). This issue could be lessened by analysing more genomic data than the 100Mb used here. However, in its current implementation, SMC^2 is inhibited by its runtime. In light of these findings, we will use fairly large epochs when analysing real data.

For sufficiently large epochs and a high rate of migration, SMC^2 is a able to detect the existence of migration in both directions. These findings motivate our use of 10,000 particles and focus strength of 3 in the following Neanderthal analysis.

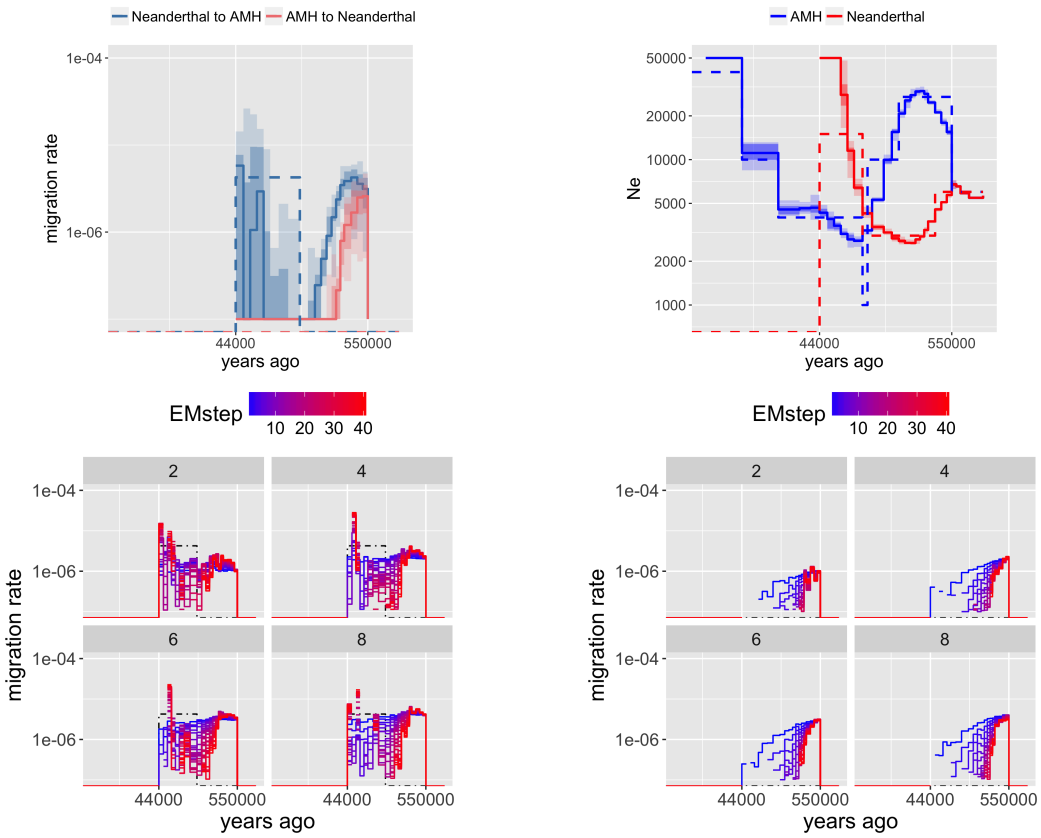


Figure 3.7: Uni-directional migration model; small epochs for inference; *Bottom left*: Neanderthal-to-AMH; *Bottom right*: AMH-to-Neanderthal; 6 samples; 10k particles; the truth is represented by the dashed lines.

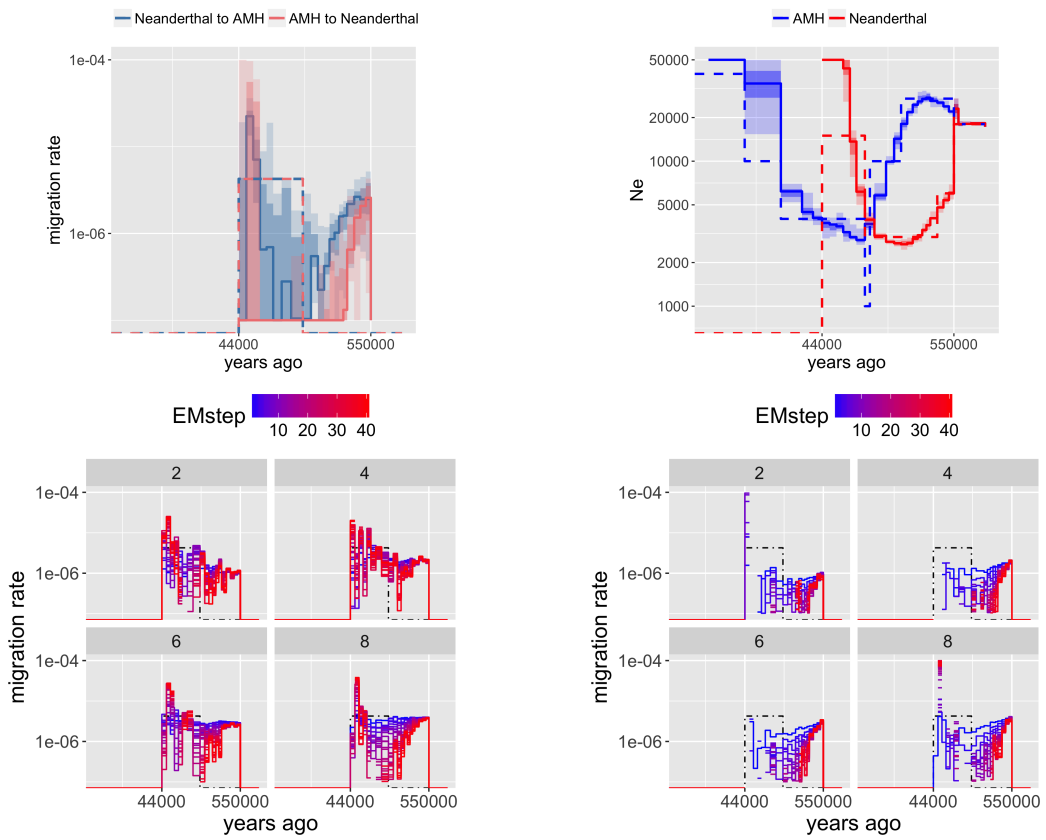


Figure 3.8: Bi-directional migration model; small epochs for inference; *Bottom left*: Neanderthal-to-AMH; *Bottom right*: AMH-to-Neanderthal; 6 samples; 10k particles; the truth is represented by the dashed lines.

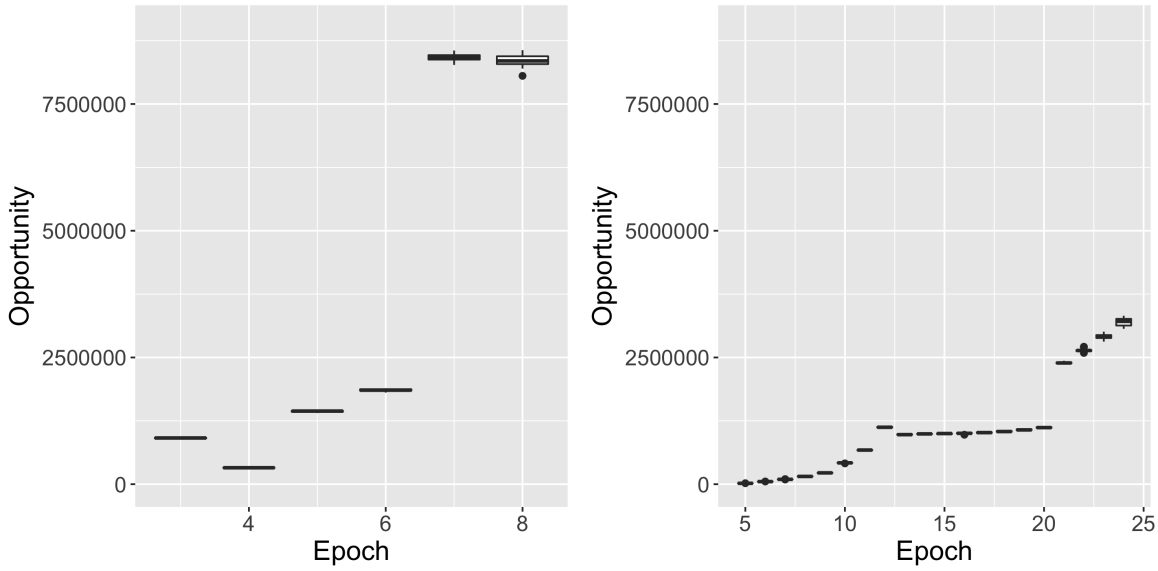


Figure 3.9: AMH-to-Neanderthal migration opportunity summary statistics for the large epoch and small epoch analyses.

3.3 Neanderthal analysis

We have shown an ability to infer migration rates when using non-contemporaneous samples. However, in the case of simulated data we know how many generations separate the samples. Unfortunately, for real data we must rely on imprecise dating of the Neanderthal sample (in years) and an imprecise estimate of generation time. We use a naive method of fixing the time of the Neanderthal lineage leaves based on point estimates. In the process of writing this chapter, the favoured estimate of the Vindija sample age changed from ∼44kya to ∼55kya [67]. In light of this, I first present the findings when using the original estimate of 44kya, and then run the analyses with a sample time of 55kya to evaluate the impact on estimates.

An additional parameter that needs to be set in SMC^2 is the split time between populations. Previous publications have determined the human-Neanderthal split to be 550-765kya. We choose to use the upper bound of 765kya as SMC^2 tends to overestimate migration following the split, and so an older choice for split is more conservative.

In the previous chapter we discussed several complications which may arise when applying SMC^2 to real data. Here we use 1,000 Genomes data of individuals from the CEU and ESN populations [13], and the Vindija Neanderthal genome available from the Max Planck Institute for Evolutionary Anthropology [54]. The Vindija genome is unphased, and so we will rely on our positive results in Section 2.7.1. In addition, some regions of the Vindija genome are uncalled due to the following filtering conditions provided by [54]:

- Coverage filter stratified by GC content
- minimum coverage 10
- Heng Li's Mapability 35 (requiring all 35mers to be unique)

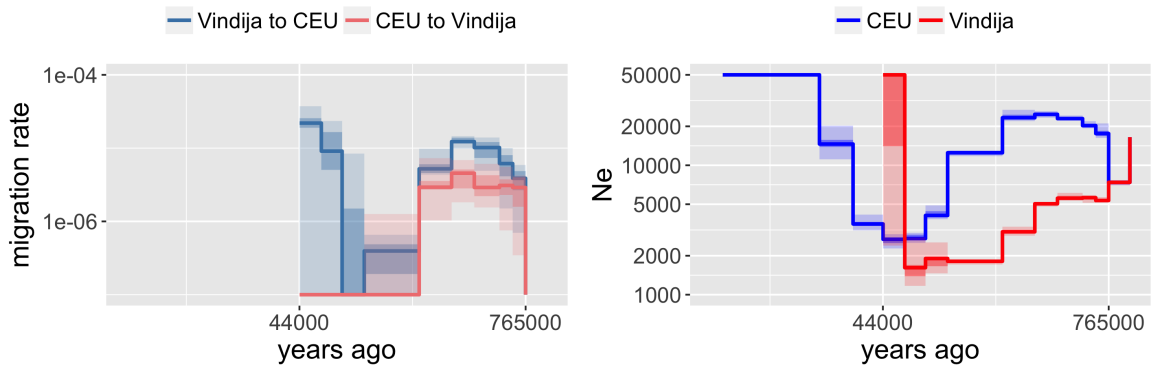
- MQ25
- no tandem repeats (from UCSC tandem repeat finder track)
- no indels (removes indels called with GATK)

These uncalled regions are common throughout the genome. To avoid these regions contributing to particle weights, we will explicitly mask them.

3.3.1 Vindija sample date of 44kya

We start by running SMC^2 over 2 diploid individuals from an AMH population and the diploid Vindija sample. Consistent with our proof of principle analyses, we use data from 160Mb of chromosome 1 (of which 62Mb is masked), and initialise the N_e at 10,000 for all times and each population. Again, the migration rate is initialised at a constant bi-directional rate as a function of time, but each replicate has a different rate $m_0 \in \{.02, .04, .06, .08, .10, .12, .14, .16, .18, .20\}$ in coalescent units. For all real Neanderthal data analysis, we assume a mutation rate of 1.45e-8 mutations per base per generation (based on a mutation rate of .5e-9 mutations per base per year and a generation time of 29 years).

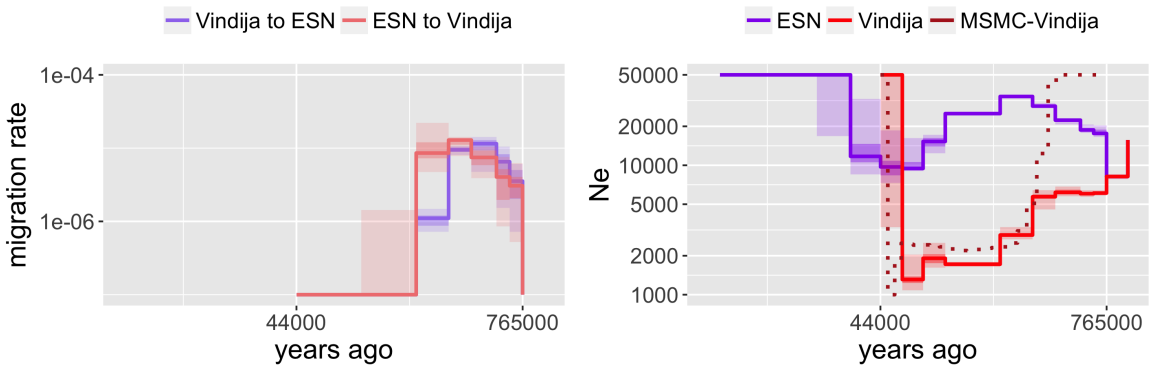
We infer forward-in-time migration from the Vindija to the CEU population in



(a) CEU-Vi migration

(b) CEU-Vi N_e

Figure 3.10: CEU-Vindija inference after 40 EM iterations; 160Mb; 10k particles.



(a) ESN-Vi migration

(b) ESN-Vi N_e

Figure 3.11: ESN-Vindija inference after 40 EM iterations; 160Mb; 10k particles.

44-75kya at a rate of about 1.5e-5 migrations per lineage per generation (Figure 3.10). This is roughly consistent with 2% of bases in modern CEU being inherited from Neanderthals (see Appendix B.2.4 for similar calculations, with a different time span and generation time). There was no migration inferred in the other direction during the same time period which is consistent with the findings of Kuhlwilm et al. [48]. We infer very low levels of migration in 100-200kya. Due to the known issues with inferring migration close to the split time (see Sections 2.5 and 3.2), we cannot determine if the apparent migration before 200kya is a true signal.

For the ESN-Vindija population analysis, we do not infer any recent migration (Figure 3.11). In contrast to the CEU-Vindija case, there is no sign of migration in 44-200kya.

The N_e estimates are largely consistent with those obtained from MSMC (Figure 3.11b). We see large variance in the estimates of Vindija N_e for 44-58kya, and so cannot confidently reaffirm or dispute the claim that the Vindija population grew in that period. The collection of more samples from the Vindija population will increase the power to infer this parameter. It is also worth noting that this parameter may be especially sensitive to false variant calls in the ancient sample (see Section 3.5.1).

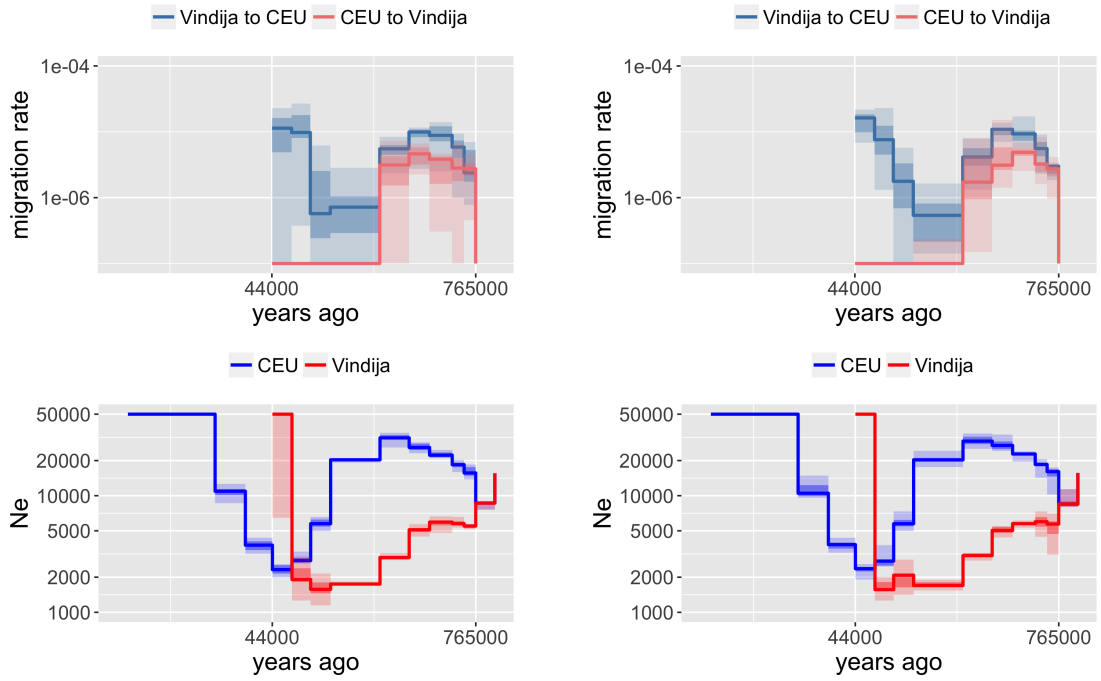
3.3.2 Vindija parameter initialisation

We observe much larger uncertainty in the migration estimates than in the effective population size estimates. This has been true throughout our simulated examples as well. One major factor of our EM procedure which we have so far not discussed in detail is parameter initialisation. In all multi-population models we have initialised all N_e parameters at 10,000. For a single replicate all migration parameters are initialised at the same value, so at a constant level of bi-directional migration. The rate for each replicate varies between 0.02 and 0.20 in coalescent units, which is equivalent to 5e-7 and 5e-6 migrations per lineage per generation. In this section we address if the uncertainty in migration estimates is an artifact of our chosen initialisation.

We ran SMC^2 over the CEU and Vindija samples with two different choices of intialisation. First, the same procedure as before where N_0 is consistent between the replicates and m_0 is inconsistent. Second, we ran 10 replicates where N_0 varied between 2,000 and 20,000 and m_0 was consistently 0.1. We conducted this experiment with an older version of the FSDR, so the results are not directly comparable to those in the other sections. Figure 3.12 shows the greater uncertainty in migration estimates relative to population size estimates is robust to which initialisation procedure is employed. We do see a slight reduction in migration uncertainty and a slight increase in effective population size uncertainty due to the new intialisation.

3.3.3 Vindija split time comparison

A major limiting factor in the application of SMC^2 is the need to impose a split time. This is a concern for this analysis as the time of the split between AMHs and Neanderthals is highly uncertain, with estimates ranging from 550-765kya. We ran the same experiment with a split time of 550kya, 650kya, and 765kya to check for sensitivity to this factor. This analysis was performed with an older version of SMC^2 with a different implementation of FSDR, so the results are not directly comparable



(a) Consistent N_0, inconsistent m_0 (b) Inconsistent N_0, Consistent m_0

Figure 3.12: Comparison of estimates from different parameter initialisations.

to those in other sections, but the split time comparison does provide insight.

We find the evidence of recent Vindija to CEU migration and lack of evidence for Vindija to ESN are largely robust to the split time (Figure 3.13). The most recent split time of 550kya infers high levels of migration in 100kya+ whereas the alternative models do not. The inferred effective population size curves are similar, except for around the split time.

A point estimate of the likelihood for a demographic model can be obtained by running SMC^2 for one EM iteration and recording the emission density $g(y_t|x_t)$ [41]. The likelihood is approximated by

$$\log L(\theta|y_{1:T}) = \sum_{t=1}^{T} \log\Big(\sum_{i=1}^{N} g(y_t|x_t^i, \theta) \Big) + T \log N. \tag{3.1}$$

To produce a decent Monte Carlo estimate we need to use a large number of particles, here we chose $N = 40,000$, and we ran over the 160Mb of data. For each of the

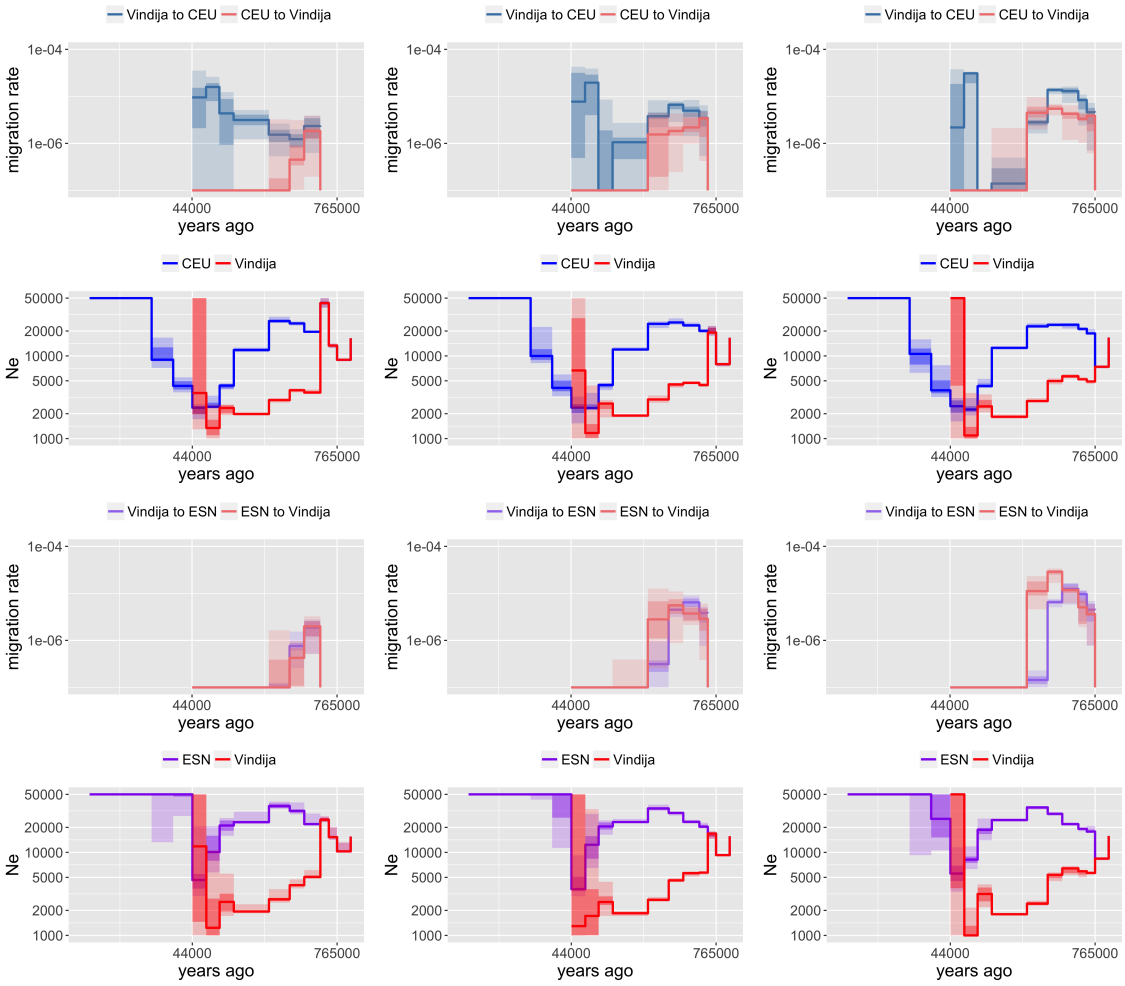
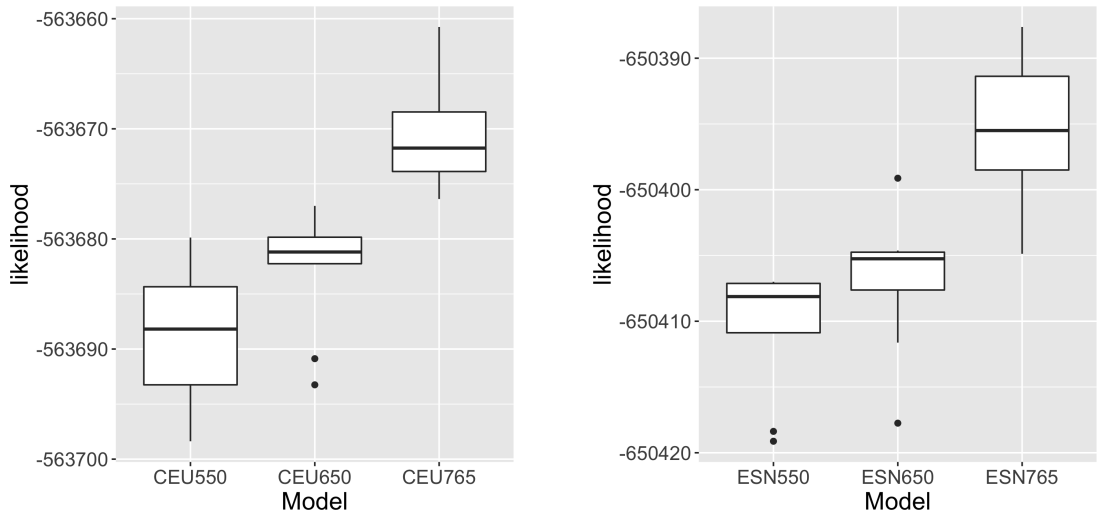


Figure 3.13: Results by split time; from left to right 550kya, 650kya, 765kya.

split time analyses, we considered the median parameter estimates. For each median demography we ran 10 replicates to obtain 10 point estimates for the likelihood. Figure 3.14 shows there is high Monte Carlo variance in the likelihood estimates for each model.



(a) CEU-Vindija (b) ESN-Vindija

Figure 3.14: Estimated likelihood of median demography after 57 EM iterations.

With estimates of the likelihoods, the models can be compared using the Akaike information criterion (AIC). The AIC involves calculating $AIC_i = 2k - 2\log(L)$ for each model i. Then $\exp\{(AIC_{min} - AIC_i)/2\}$ is proportional to the probability that the i-th model minimises the estimated information loss, i.e. the relative likelihood of model i. We chose the median likelihood from the high particle count replicates to represent each model. Our estimates of the likelihood for each case have high uncertainty, which is not accounted for in this analysis. However, the difference between cases is large compared to the uncertainty.

CEU

(the likelihood is taken from the median of the 10 replicates)

$AIC_{550} = 2 \cdot 39 - 2 \cdot (-563688.2) = 1127454.4$

$AIC_{650} = 2 \cdot 42 - 2 \cdot (-563681.2) = 1127446.4$

$AIC_{765} = 2 \cdot 45 - 2 \cdot (-563671.8) = 1127433.6$

$\Longrightarrow$

$p_{550} = \exp\{(AIC_{min} - AIC_{550})/2\} = 0.000030$

$p_{650} = \exp\{(AIC_{min} - AIC_{650})/2\} = 0.0017$

$p_{765} = \exp\{(AIC_{min} - AIC_{765})/2\} = 1.0$

ESN

(the likelihood is taken from the median of the 9 replicates as one replicate died)

$AIC_{550} = 2 \cdot 39 - 2 \cdot (-650408.1) = 1300894.2$

$AIC_{650} = 2 \cdot 42 - 2 \cdot (-650405.2) = 1300894.4$

$AIC_{765} = 2 \cdot 45 - 2 \cdot (-650395.5) = 1300881.0$

$\Longrightarrow$

$p_{550} = \exp\{(AIC_{min} - AIC_{550})/2\} = 0.0014$

$p_{650} = \exp\{(AIC_{min} - AIC_{650})/2\} = 0.0012$

$p_{765} = \exp\{(AIC_{min} - AIC_{765})/2\} = 1.0$

The AIC strongly favours a split time of 765kya. We suspect this is primarily influenced by the inferred demography rather than the split time itself. As mentioned previously, SMC^2 struggles to reduce migration estimates in the post-split epoch. We would expect a model inferred using the correct split time to have a relatively low likelihood if the true model has no migration in the post-split epoch. Still, it seems our choice of 765kya is justified.

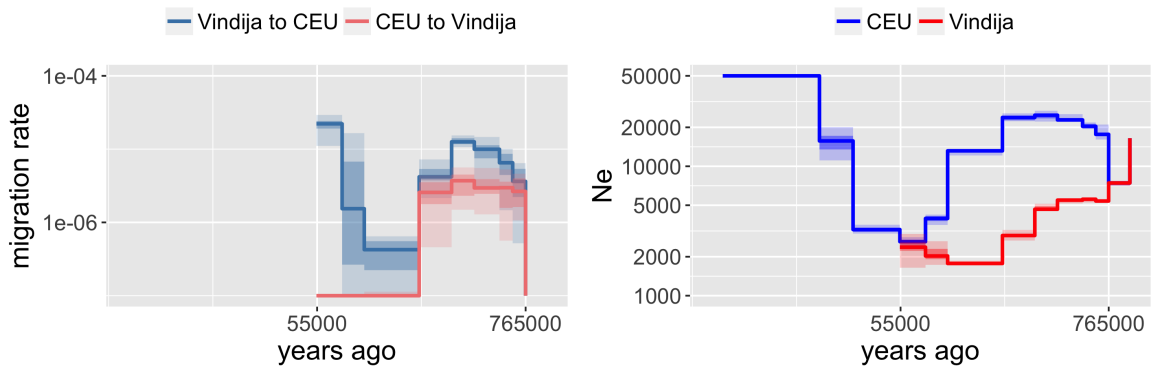
An alternate method to imposing a split time in the model would be to allow the migration inference to model the joining of two populations. When a single population is being falsely modelled as two populations, the inferred migration rates between the populations should be high. If the inferred migration is balanced, the

inferred N_e for each population will be half of that of the true joint population (we have observed this in our own simulated data analyses, although I do not present the evidence here). This may be a better way to model the dynamics of splitting populations as it allows for more subtle changes in time. Our conservative choice of a split at 765kya is similar, in that it allows inferred migration rates to provide the detail on the severity of the split.

3.3.4 Vindija sample date of 55kya

The choice of Vindija sample age in the analyses above was primarily based on the values used in the Kuhlwilm et al. paper, which was based on carbon dating [48]. However, since the publication of that paper, the estimated age of the Vindija sample age has been modified from ∼44kya to ∼55kya using branch shortening inference [67]. The branch shortening method counts the number of derived changes since the chimp-human ancestor for the samples, and then scales the sample dates based on the proportions. We are reluctant to overly rely on branch shortening estimates, as they are sensitive to differences in error rates between modern and ancient samples [68]. We reran some of the experiments with the altered sample age.

The updated sample age affects the N_e inference for the Vindija population, but does not seem to alter the migration estimates in a significant way (Figure 3.15). The main difference is the inference of N_e in the most recent Vindija epoch (44-58kya for the original analysis, 55-75kya in this analysis). The original analysis inferred a large N_e, which was largely consistent with the findings of G-PhoCS in [48]. With the updated sample date, the $\hat{N}_e$ is ∼2,000. This estimate is more consistent with a population shortly before its extinction.



(a) CEU-Vi migration

(b) CEU-Vi N_e

(c) ESN-Vi migration

(d) ESN-Vi N_e

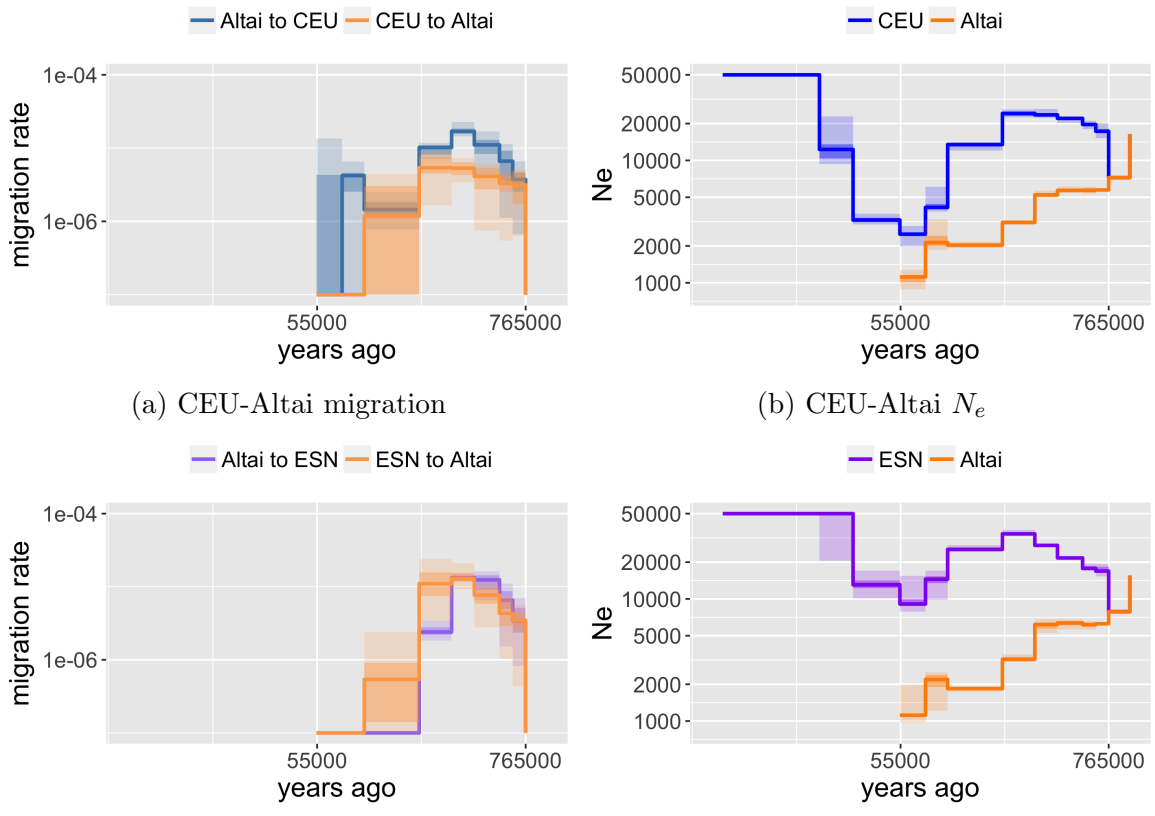
Figure 3.15: Results when imposing a Vindija sample time of 55kya; 40 EM iterations.

3.3.5 Altai analysis

This chapter has focused on the Vindija individual primarily because the Altai sample has been shown to have high levels of recent inbreeding [68]. However, there is growing interest in distinguishing the Neanderthal populations and their interactions with AMHs. As such, we conducted the same analyses on the Altai Neanderthal.

Recently inbred regions in the Altai could cause problems for SMC^2 as they have a true coalescence just two or three generations back. We introduced a pre-processing step to remove one of the haplotypes in these regions, which is equivalent to forcing an extremely recent coalescence. To identify the regions, we followed the methodology outlined in SI10 of Prüfer et al. [68]. Prüfer et al. described a high false positive rate in detecting homozygous by descent (HBD) regions smaller than 3.85Mb, so we retained only those larger than 3.85Mb. This identified only one HBD on chromosome 1, which happens to be outside the 160Mb we have been analysing. There were 23 segments smaller than 3.85Mb, which amount to 27.6Mb in the 160Mb region. Due to the high false positive rate for the detection of small HBDs, we did not alter the data in these regions. As such, we did not need to impose the single haplotype, although these steps should be followed if other chromosomes are analysed.

Running SMC^2 as in Section 3.3.4, but now with data from the Altai Neanderthal, we see a few noteworthy differences (Figure 3.16). We infer a lower level of migration in the period 55-75kya for the CEU-Altai than we did for the CEU-Vindija. This is consistent with the theory that the introgressing Neanderthals were more closely related to the Vindija than the Altai [68, 48]. The split between the Altai and the introgressing Neanderthal is estimated to be 77-114kya [68], which may explain the increase in migration before 75kya for the Altai (relative to the Vindija migration rate). The Altai analysis also has higher levels of migration from AMH to Neanderthal in the period 100-200kya. This is consistent with the finding in Kuhlwilm et al. of a highly diverged AMH population migrating into the Altai population after the split



(a) CEU-Altai migration

(b) CEU-Altai N_e

(c) ESN-Altai migration

(d) ESN-Altai N_e

Figure 3.16: Altai inference after 40 EM iterations.

with Vindija.

Our analysis of the Altai Neaderthal is largely in line with previous findings. These previous findings were based on methods (D-statistics, G-PhoCS) which summarise directional migration with a single statistic. SMC^2 has the advantage of inferring rates of migration as a function of time, where other methods must analyse samples from multiple populations in order to deduce the time of migration events.

3.4 Effect of genes in CEU-YRI

The preceding analyses treated the genome as a neutrally evolving sequence. This was the case for all simulated data, but of course is not true of actual genomes. There are regions of the genome under strong selective pressure. In this section, I investigate the effect of excluding some of these non-neutral regions in analyses.

The CEU and YRI populations are both widely studied [36, 76, 74, 80], and so provide a good case to test the effect of masking selected genomic regions. I selected two unrelated individuals from each population from the 1,000 Genomes database [13]. The split time, generation time, and mutation rate were set consistent with the values used in Terhorst et al. [80] (110kya, 29y, 1.25e-8 mutations per base per generation), although the accuracy of the split time remains controversial [25]. Running SMC^2 over 160Mb of chromosome 1, leads to high estimates of forward-in-time migration from CEU to YRI (Figure 3.17), indicating consistent high levels of European to Yoruban introgression.

If we mask gene regions with a buffer of 0.1cM on either side (according to the NRE [1]), and then run SMC^2, we see some differences in the inference (Figure 3.18). The criterion of masking all genes is quite severe, masking all UCSC known genes, including both protein-coding genes and non-coding RNA genes, with a large buffer on either side. This mask covers 103Mb of the 160Mb. With these gene regions masked, the inferred migration rate in 10-50kya is an order of magnitude less than when the gene regions were included. Moreover, the rate of migration in 0-10kya is now inferred to be zero. There are two possible explanations for these differences. Either the inclusion of gene regions biases the estimates away from their true values, or the presense of masked regions alters the convergence of estimates.

The elevation of inferred migration rates due to the presence of genes is certainly possible. Genes will be highly conserved, and so haplotypes will be more similar than expected in a neutral region with the same underlying demography. In conserved re-

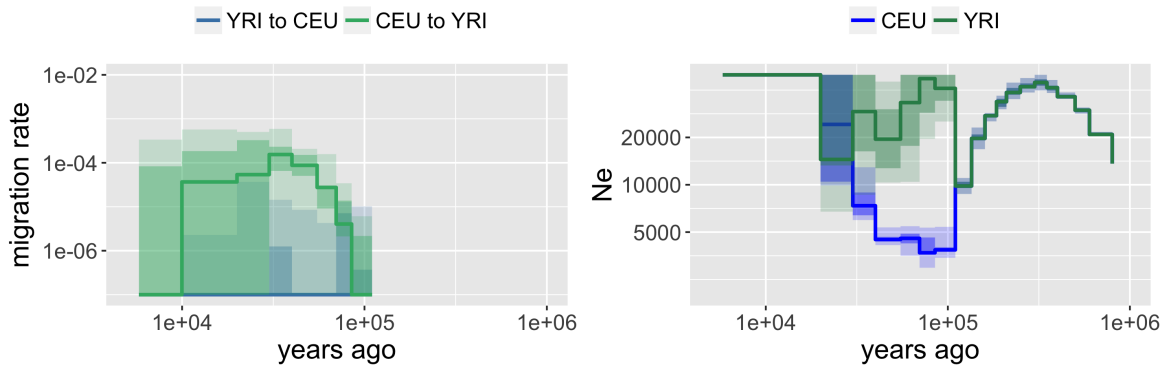


Figure 3.17: CEU-YRI inference without masking.

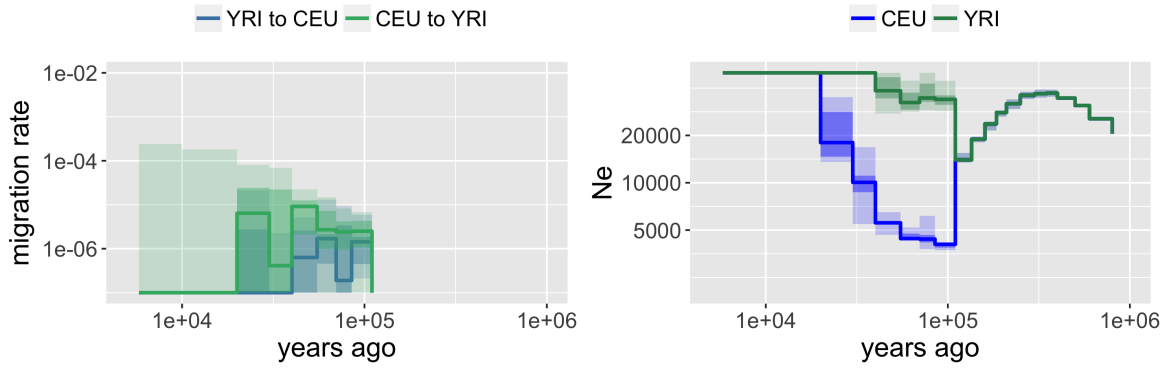
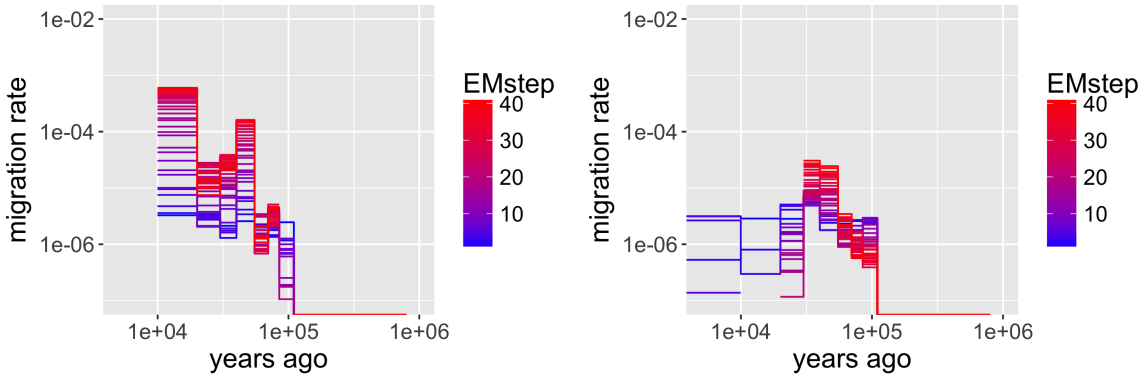


Figure 3.18: CEU-YRI inference with genes masked.

gions, the emission distribution will put high weight on particles with small TMRCA. In a two population model, the only way to have a small TMRCA is to have lineages migrate in the recent past. And so high migration rates may be a symptom of including non-neutral regions. If this is indeed the case, then the directionality of the falsely inferred migration is likely a consequence of the population demographies. In the CEU-YRI case, the small TMRCAs are most likely to arise from a coalescence in the smaller CEU population. This favours overestimation of CEU to YRI migration, which is what we observe in our comparison.

The alternative possibility is that masking regions of the genome affects the convergence of the estimates. SMC^2 currently treats masked regions as alleles with missing data. Particles are propogated according to the current parameter estimates, and the ratio of the transition to proposal density contribute to the particle weights, but



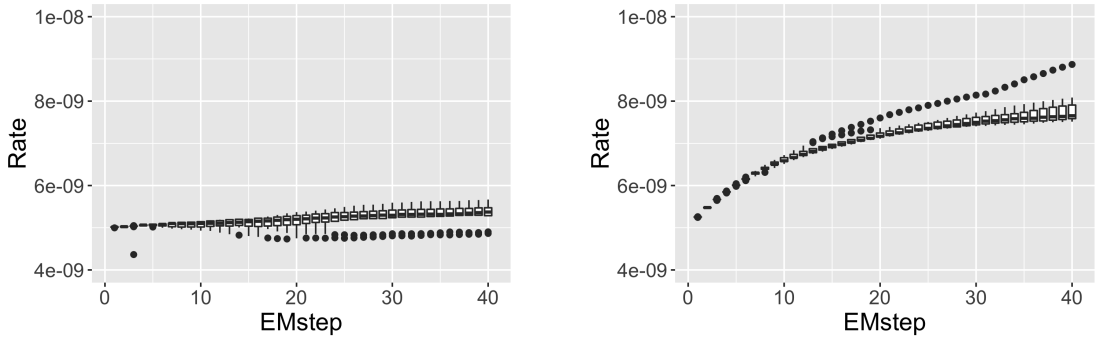
(a) no masking (b) genes masked

Figure 3.19: CEU-to-YRI migration estimate evolution.

the emission density does not. The events which occur in these regions do contribute to the summary statistics. This means long sections of missing data will contribute opportunities and event counts which are consistent with the current parameter estimates. This will lead to slower evolution of parameter estimates. Comparing the estimate evolution in the non-masked and gene-masked cases (Figure 3.19), we do not see signs of slower evolution in the gene-masked case. This suggests the presence of masked regions is not the cause of the differences.

There is another noteworthy difference in the inference from the non-masked and the gene-masked analyses. The final inferred recombination rate differs by a significant margin (Figure 3.20). Here we see $\hat{\rho}$ seem to be converging to about 8e-9 for the gene-masked case, whereas the no masking case has a steady $\hat{\rho}$ around 5e-9. The overall difference is not too surprising as genes are known to have a lower overall recombination rate than the rest of the genome [58], and so their removal will raise the estimate. The fact that the recombination evolves in the gene-masked case provides further evidence that it is not the extent of genome masking which is causing the difference, as large volumes of missing data would discourage estimate movement.

To interrogate this further, we took the gene mask and shifted it 2Mb. This retains the same fraction of the data masked (103/160), but lessens the proportion



(a) no masking (b) genes masked

Figure 3.20: Recombination rate estimate evolution.

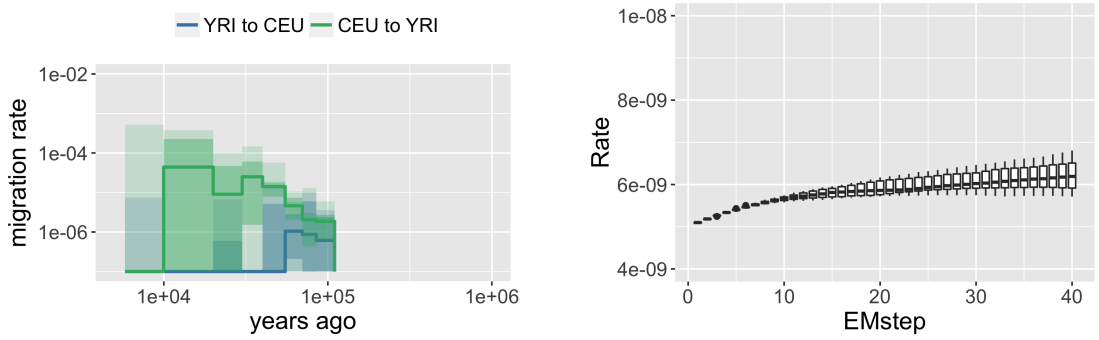


Figure 3.21: Inference from data with a shifted mask.

of genes masked. The shifted mask covers 69Mb of the genes, leaving the unmasked regions to be 59.6% gene regions. This is slightly lower than the unmasked data which is 64.4% gene region. With the new mask, the estimates are inbetween that of the gene-masked and that of the non-masked analysis (Figure 3.21). This result suggests the difference is not the extent of masking, but which regions are masked.

The inclusion of genes affects the inferred demography. In light of these results, we would advise masking regions known to be under selective pressure. In its current implementation, SMC^2 is inefficient when large regions of the genome are masked. The software should be updated to ignore events that occur in large masked regions, to avoid hindering estimate evolution. Alternatively, SMC^2 could be modified to identify regions putatively under selective pressures, either with the intent of removing the confounding effect of these on demographic inference or for the identification itself.

3.5 Conclusion

In this chapter we identified similar signals in the Neanderthal data as G-PhoCS. We were able to infer not only the extent of migration, but to place it reasonably well in time. Other methods often place migration events in time by comparing many populations and deducing when migration occurred; SMC^2 infers the time directly from the data of the two populations involved. We see stronger evidence of Vindija-to-CEU than Altai-to-CEU migration. We suspect the large inferred population size for Vindija population is an artifact of a mis-specified sample age.

The field of palaeogenomics is rapidly expanding, due in part to the discovery that the petrous bone of the inner ear [64, 51, 34] can harbour DNA for long periods of time, even in warm climates. As the availability of ancient genomes grows, it is worth remembering the limitations these samples impose on demographic inference methods. We showed in Section 2.7.1 that SMC^2 is largely robust to phasing, a common problem with ancient samples. However, the problem of an uncertain sample age does have a rather profound effect on the N_e estimates.

3.5.1 Future works

This chapter explores many limitations of the current version of SMC^2. Each of these limitations is an opportunity for future development. Some of these are unique to our algorithm, but many should be considered for any analysis of this type.

One factor that should be explored by the field in general is the affect of false variant calls on these analyses. I believe this has not attracted much interest as the methods for calling variants are improving, particularly for humans where large reference panels are available. With the adoption of these methods for highly diverged archaics and completely distinct species, the sensitivity to imperfect data needs to be better understood. While we now suspect the apparent large N_e for the Vindija

in 45-58kya is an artifact of imposing too recent a sample age, this could also be explained by false variant calls in that sample.

Our gene masking analyses in Section 3.4 demonstrated the potential influence of non-neutral regions in these analyses. We chose quite a simple criterion for classifying regions as neutral or non-neutral. Unfortunately, the selective pressures acting on genomes are not so straightforward. SMC^2 could be engineered to identify regions that are likely under strong selection, by monitoring atypical sufficient statistics. We did not pursue this advancement as the current runtimes prevent SMC^2 from practically scanning sufficient volumes of data to exclude large regions. If runtimes can be significantly shortened by some of our proposed modifications, then the effect of selective-region exclusion should be explored further.

Another limitation to address is the pre-selection of epoch boundaries. Many similar methods use dynamic epochs which move based on the estimated event counts for the epochs. This could be done for SMC^2 as well, as we do track the opportunity for events in each epoch. Section 3.2 showed the danger of imposing a large number of small epochs. Dynamic epochs based on the opportunity sufficient statistics could be allowed to combine epochs with insufficient opportunity.

We have shown SMC^2 is capable of inferring migration at the putative level between AMHs and Neanderthals. However, Section 3.2 revealed an alarming skew in the directionality of inferred migration when a lower level of bi-directional migration is simulated. Some of the migration rates can easily be absorbed at zero in this case. It may benefit the algorithm to somehow reintroduce the possibility of migration at later EM steps, particularly if the N_e curve has changed substantially since the iteration of migration absorption.

Many of the proposed extensions would benefit from faster runtimes. We explore one avenue for increasing parameter convergence in the next chapter. Another method we are developing is a more sophisticated sampler than our FSDR. The FSDR first

broadens the tails of the proposal distribution, then takes inspiration from lookahead methods by altering the resampling procedure. The FSDR is better described as a delay than a lookahead as the particles are simulated up to the resampling point and a leniency is applied to some events that came before due to the broad proposal. Alternatively, lookahead methods resample at the current position, with a consideration of some of the data to the right of the position [65, 18, 50]. A broadened proposal coupled with a lookahead resampler would produce a similar effect as the FSDR, but with more control over the considered transitions for the extra data to the right.

Chapter 4

Adaptive learning in Online Expectation Maximisation

4.1 Introduction

In Chapter 2 we discussed using sophisticated samplers as an alternative to simulating a large number of particles, N. The motivation for keeping N small is to save on compute time. It is worth considering what other methods could be used to reduce the computation time. This chapter discusses alternative EM procedures which could increase the speed of the algorithm.

There are many factors affecting the runtime of SMC^2. These include:

- The number of particles N
- Sequence length
- Genealogy size
- The parallelisation implementation
- BEM batch length

We observed in Section 2.3 that the number of particles and length of sequence data greatly affect the accuracy of the inference. This has motivated us to maintain at least 3,000 particles and 100Mb of genomic data in all analyses. However in many scenarios, particularly ones with large genealogies (due to many samples or multiple populations with ancient coalescent events), the runtime is impractical. To combat this problem, we introduced a crude parallelization by chopping the data into chunks. For each EM iteration we run the particle filter over each of these chunks in parallel, and the summary statistics are combined for the M-step. This allows us to save on runtime at the cost of more CPUs. However, even without constraints on the number of CPUs, there are limits to the gains from parallelisation as segmenting the data removes dependencies. In all analyses we maintain chunks of length at least 20Mb to avoid losing too much information from long distance dependencies. With these considerations the Vindija analysis in Section 3.3 took $\sim$4.5 days on 80 CPUs for 40 EM iterations. More parameter updates would be better as the results had clearly not yet converged. To make the algorithm more useful in practice, we now consider modifying the EM procedure.

Expectation Maximisation (EM) is a widely used and general technique for estimating maximum likelihood parameters of a latent variable model [14]. In the context of models which do not admit analytic solutions, EM methods are favoured in practice over gradient-based approaches due to their relative stability and computational efficiency when estimating high dimensional parameters [12, 43].

One of the most popular implementations of recursive EM is batch EM (BEM). BEM is an adaptation of the standard offline EM to infer parameters from either a stream of incoming data or a large volume of data where it is impractical to analyse the entire dataset for each iteration. In BEM, a batch size b is set by the user. The parameter estimate is updated every b observations by maximising the expected likelihood over the previous batch of data. The performance of the algorithm depends

on the user-specified value for the tuning parameter b. For instance a large choice for b can lead to inaccurate estimates because of slow convergence. A large batch size enforces a small learning rate for the algorithm. However, a small choice for b, and so a large learning rate, can lead to imprecise estimates due to the inherent stochasticity of the model within a small batch of observations. The optimal learning rate in BEM depends on the particularities of the model.

Another popular implementation of recursive EM is online EM (OEM), sometimes referred to as adaptive EM [8]. OEM addresses the problem of slow convergence in BEM by updating the parameter estimate after every observation. Whereas BEM considered only a single batch of data in each iteration, OEM maximises the expected likelihood of all preceding data. The contribution of the early data to the expected likelihood is not as reliable as the more recent data because the estimated parameter has moved towards the true value since initialisation. For this reason, OEM requires a user-specified weight sequence to define the relative weight of each observation towards the running expected likelihood. Assigning relatively small weights to the newest data enforces slow convergence. Assigning relatively large weights to the newest data allows for quicker evolution in the parameter estimates, but decreases precision due to large stochastic effects. As in BEM, the performance of OEM is largely dependent on the learning rate of the algorithm which is determined by a user-specified parameter.

The EM procedure in SMC^2 uses BEM with a batch size equivalent to the length of the sequence data provided. This implementation was chosen for its simplicity, but efficiency in estimate convergence could be improved by a more sophisticated choice. Unfortunately, there is little intuition for what the optimal batch size or learning rate may be in the context of inferring demographic parameters from ARGs. The intuition we do have is that the optimal learning rate is likely to depend on two unknown factors, how close our initial estimates are to the true demography and how

informative the data is of the demography. The former idea is straightforward as the algorithm should not waste compute time simulating particles from a demography that is far from the truth when the estimates could quickly be moved to a higher likelihood region of the parameter space. The latter idea is best thought of as the problem of inferring a large N_e or a weak migration rate. When the rate parameter we are trying to estimate is small, too large a learning rate can easily lead to absorption of the estimate at zero (recall our proof of principles analysis in the previous chapter Figure 3.5). As problems arise from either too large or too small a batch size, there is no cautious choice.

Here we introduce a novel algorithm, termed Introspective Online EM (IOEM), which removes the need for setting these tuning parameters by estimating the optimal parameter-specific learning rate along with the parameters of interest. This is particularly helpful when inferring parameters in a high dimensional model, since the optimal learning rate may differ between parameters. Broadly, IOEM works by estimating both the precision and the accuracy of parameters in an online manner through weighted linear regression, and uses these estimates to tune the learning rate so as to improve both simultaneously.

This chapter explores the relative performance of different EM procedures applied to a variety of models. Section 4.2 uses a one-unknown-parameter autoregressive state-space model to introduce BEM, OEM, and a simplified version of IOEM. Section 4.3 considers the full autogressive model with all parameters unknown, which requires the complete IOEM algorithm. Section 4.4 compares the parameter estimation methods in a 2-dimensional autoregressive model to show the benefit of the proposed algorithm when inferring many parameters. Finally, Section 4.5 demonstrates desirable performance in the stochastic volatility model, an important case as it is non-linear and hence more similar to models where stochastic approximation EM (SAEM) will be favoured over analytic solutions.

4.2 EM for a simplified autoregressive model

In this section we will review sequential Monte Carlo (SMC), BEM, OEM, and present the IOEM algorithm with a simple model. This allows us to illustrate the main concepts behind IOEM before delving into details in Section 4.3.

We consider a simple autoregressive model with one unknown parameter. We observe the sequence of random variables $Y_{1:t} := \{Y_k\}_{k=1,\dots,t}$ which depends on the unobserved sequence $X_{1:t} := \{X_k\}_{k=1,\dots,t}$, as follows:

$$\begin{aligned} X_t &= aX_{t-1} + \sigma_w W_t, \\ Y_t &= X_t + \sigma_v V_t, \end{aligned} \tag{4.1}$$

where W_t and V_t are i.i.d. standard normal variates, $a = 0.95$ and $\sigma_w^2 = 1$ are known parameters, and σ_v^2 is unknown. Under this model, we have the following transition density $f(x_t|x_{t-1})$ and emission density $g(y_t|x_t)$:

$$\begin{aligned} f(x_t|x_{t-1}) &= (2\pi\sigma_w^2)^{-1/2} \exp\Big\{ -\frac{(x_t - ax_{t-1})^2}{2\sigma_w^2} \Big\}, \\ g(y_t|x_t) &= (2\pi\sigma_v^2)^{-1/2} \exp\Big\{ -\frac{(y_t - x_t)^2}{2\sigma_v^2} \Big\}. \end{aligned}$$

We have chosen σ_v^2 as the unknown parameter as it is the most straightforward to estimate, allowing us to introduce the idea of IOEM without certain complications which we address in Section 4.3. As f and g are members of the exponential family of distributions, the M step of EM can be done using sufficient statistics, and so the E step amounts to the expectation of the sufficient statistics. In this model, the parameter σ_v^2 has the sufficient statistic

$$S_t = \mathbb{E}_{X_{1:t}|Y_{1:t},\theta} \left[\frac{1}{t} \sum_{k=1}^{t} (Y_k - X_k)^2 \right]. \tag{4.2}$$

The estimate of σ_v^2 is obtained by setting $\hat{\sigma}_{v,t}^2 = \hat{S}_t$. More generally, for an unknown parameter θ, $\hat{\theta}_t = \Lambda(\hat{S}_t)$ where Λ is a known function mapping sufficient statistics to parameter estimates.

We will use SMC to estimate S_t, despite the availability of analytic estimates for this model, as our primary interest is in non-linear non-Gaussian models. We simulate particles $X_{1:t}^{(i)}$ and calculate their associated weights $w(X_{1:t}^{(i)})$, $i = 1, \ldots, N$, so that

$$\sum_{i=1}^{N} w(X_{1:t}^{(i)}) \delta_{X_{1:t}^{(i)}}$$

approximates the distribution $p(X_{1:t}|Y_{1:t}, \theta)$. The standard Monte Carlo EM (MCEM) approximation of $p(X_{1:t}|Y_{1:t}, \hat{\theta})$ would require storage of all observations $Y_{1:t}$, the simulation of $X_{1:t}^{(i)}$ each time $\hat{\theta}$ is updated, and ideally an increasing Monte Carlo sample size as the parameter estimates near convergence. To avoid this, we employ SAEM which effectively averages over previous parameter estimates as an alternative to generating a new Monte Carlo sample every time an estimate is updated, and hence is more suitable to online inference. This method as proposed in [8] approximates the expectation in equation (4.2) recursively.

The outline of the SMC with EM algorithm we consider is described in Algorithm 4.1. Here $\mu(\cdot|\hat{\theta}_0)$ is the initial distribution for X_1, ESS is the effective sample

Algorithm 4.1 Sequential Importance Resampling (bootstrap filter)

for time $t \geq 1$ **do**
  for $i = 1, \ldots, N$ **do**
    Sample $X_t^{(i)} \sim \begin{cases} \mu(\cdot|\hat{\theta}_0), & \text{if } t = 1 \\ f(\cdot|X_{t-1}^{(i)}, \hat{\theta}_{t-1}), & \text{if } t \geq 2 \end{cases}$
    Compute normalized weights satisfying
    $w_t(X_{1:t}^{(i)}) \propto w_{t-1}(X_{1:t-1}^{(i)}) \cdot g(Y_t|X_t^{(i)}, \hat{\theta}_{t-1})$
  end for
  Update $\hat{\theta}_{t-1}$ to $\hat{\theta}_t$ using chosen EM method
  Resample particles if $ESS < \frac{N}{2}$
end for

size defined as $[\sum_{i=1}^{N} w_t(X_{1:t}^{(i)})^2]^{-1}$, $w_0(\cdot) = 1/N$, and $X_t^{(i)}$ is shorthand for the t^{th}

coordinate of $X_{1:t}^{(i)}$. In models with multiple unknown parameters, each parameter is updated in step 3 of Algorithm 4.1, however we will refer only to a single parameter θ to keep the notation simple.

Throughout this paper we follow common practice in using the fixed-lag technique in order to reduce the mean square error between S_t and $\hat{S}_t$ [7, 10]. In particular, we choose a lag $\Delta > 0$ and then at time t, using particles $X_{1:t}^{(i)}$ shaped by data $Y_{1:t}$, estimate the $t-\Delta^{\text{th}}$ term of the summation in equation (4.2). We will use $X_{1:t}^{(i)}(t-\Delta)$ to denote the $t-\Delta^{\text{th}}$ coordinate of the particle $X_{1:t}^{(i)}$, but we will continue to write $X_t^{(i)}$ as a shorthand for $X_{1:t}^{(i)}(t)$.

The fixed-lag technique involves making the approximation

$$S_t \approx \mathbb{E}_{X_{1:t}|Y_{1:t},\theta}\left[\frac{1}{t-\Delta}\sum_{j=1}^{t-\Delta} s(Y_j, X_j)\right] \tag{4.3}$$

$$\approx \frac{1}{t-\Delta}\sum_{j=1}^{t-\Delta} \mathbb{E}_{X_{1:j+\Delta}|Y_{1:j+\Delta},\hat{\theta}_{0:j+\Delta}}\left[s(Y_j, X_j)\right], \tag{4.4}$$

where we assume that S_t is an additive functional and so can be written as

$$S_t = \mathbb{E}_{X_{1:t}|Y_{1:t},\theta}\sum_{j=1}^{t} s(Y_j, X_j).$$

This allows S_t to be updated in an online manner by computing the componentwise sufficient statistics

$$\begin{aligned}\tilde{s}_t :&= \mathbb{E}_{X_{1:t}|Y_{1:t},\hat{\theta}_{0:t}}\left[s(Y_{t-\Delta}, X_{1:t}(t-\Delta))\right] \\ &\approx \sum_i w_k(X_{1:t}^{(i)}) s(Y_{t-\Delta}, X_{1:t}^{(i)}(t-\Delta)),\end{aligned}$$

allowing $\hat{S}_t$ to be updated by

$$\hat{S}_t = \gamma_t \cdot \tilde{s}_t + (1-\gamma_t)\cdot \hat{S}_{t-1}, \tag{4.5}$$

with some weight γ_t; in equation (4.3) $\gamma_t = 1/(t - \Delta)$. This approach is slightly different from that of [7]; see Section C.2 for a discussion.

Choosing a large value of Δ allows SMC to use many observations to improve the posterior distribution of $X_{t-\Delta}$. However the cost of a large Δ is a loss in particle independence due to the resampling procedure which increases the sample variance. The optimal choice for Δ balances the opposing influences of the forgetting rate of the model and the collapsing rate of the resampling process due to the divergence between the proposal distribution and the posterior distribution. For the examples in this paper we chose $\Delta = 20$ as recommended by [7], which seems to be a reasonable choice for our models.

There are various other techniques to improve on this basic SMC method, including improved resampling schemes [17, 62, 18, 10], and choosing better sampling distributions through lookahead strategies or resample-move procedures [65, 50, 18], which are not discussed further here. Instead, in the remainder of this chapter, we focus on the process of updating the parameter estimates $\hat{\theta}_t$. The remainder of this section describes the options for step 3 of Algorithm 4.1.

4.2.1 Batch Expectation Maximisation

Batch Expectation Maximisation (BEM) processes the data in batches. Within a batch of size b, the parameter estimate stays constant ($\hat{\theta}_t = \hat{\theta}_{t-1}$) and the update to the sufficient statistic

$$\tilde{s}_t := \sum_i w_t(X_{1:t}^{(i)}) \cdot (Y_{t-\Delta} - X_{1:t}^{(i)}(t - \Delta))^2,$$

is collected at each iteration t. At the end of the mth batch we have $t = mb$, at which time

$$\hat{S}_t^{BEM} := \frac{1}{b} \sum_{k=(m-1)b+1}^{mb} \tilde{s}_k,$$

is our approximation of S, and $\hat{\sigma}^2_{v,t} := \hat{S}^{BEM}_t$.

The batch size determines the convergence behavior of the estimates. For a fixed computational cost, choosing b too small will result in noise-dominated estimates and low precision, whereas choosing b too large will result in relatively precise but inaccurate estimates due to slow convergence.

4.2.2 Online Expectation Maximisation

BEM only makes use of the collected evidence at the end of each batch, missing potential early opportunities for improving parameter estimates. OEM addresses this issue by updating the parameter estimate at every iteration (Algorithm 4.2). The approximation of S at time t is a running average of $\{\tilde{s}_k\}_{k=\Delta+1,\dots,t}$, weighted by a pre-specified weighting sequence as in equation (4.5). The choice of weighting sequence determines how quickly the algorithm "forgets" the earlier parameter estimates. In OEM at time t,

$$\hat{S}^{OEM}_t = \gamma_t \cdot \tilde{s}_t + (1 - \gamma_t) \cdot \hat{S}^{OEM}_{t-1}, \tag{4.6}$$

where $\{\gamma_k\}_{k=1,2,\dots}$ is the chosen weighting sequence, typically of the form $\gamma_t = t^{-c}$ for a chosen $c \in (0.5, 1]$ [6]. Note that when using lag Δ, $\gamma_t = (t - \Delta)^{-c}$ for $t \geq \Delta$, and $\hat{\theta}_t = \theta_0$, $\tilde{s}_t = 0$ for $t \leq \Delta$. This update rule ensures that at time t, $\hat{S}^{OEM}$ is a weighted sum of $\{\tilde{s}_k\}_{k=\Delta+1,\dots,t}$ where the term $\tilde{s}_k$ has weight

$$\eta^t_k := \gamma_k(1 - \gamma_{k+1}) \cdots (1 - \gamma_{t-1})(1 - \gamma_t). \tag{4.7}$$

Although this method can outperform BEM, its performance remains strongly dependent on the parameter c determining the weighting sequence, and a suboptimal choice can reduce performance by orders of magnitude. At one extreme, the estimates will depend strongly only on the most recent data, resulting in noisy parameter estimates and low precision. At the other extreme, the estimates will average out

Algorithm 4.2 Online Expectation Maximisation for a simplified autoregressive model

for time $t \geq 1$ **do**
  Simulate and calculate weights of new particles as outlined in Algorithm 4.1
  Collect sufficient statistic $\tilde{s}_t = \sum_{i=1}^{N} w_t(X_{1:t}^{(i)}) \cdot (Y_{t-\Delta} - X_{1:t}^{(i)}(t-\Delta))^2$
  Update running average of sufficient statistics $\hat{S}_t^{OEM} = \gamma_t \tilde{s}_t + (1-\gamma_t)\hat{S}_{t-1}^{OEM}$
  Maximise expected likelihood by setting $\hat{\theta}_t := \hat{S}_t^{OEM}$
end for

stochastic effects but be severely affected by false initial estimates, resulting in more precise but less accurate estimates. Again, the best choice depends on the model.

A pragmatic approach to the problem of choosing a tuning parameter in OEM is discussed in [9] and takes inspiration from [66]. In this method, a weight sequence that emphasizes incoming data is used to ensure quick initial convergence, while imprecise estimates are avoided at later iterations by averaging all OEM estimates beyond a threshold t_0

$$\hat{\theta}_t^{AVG} = \begin{cases} \hat{\theta}_t^{OEM} & \text{for } t < t_0 \\ \frac{1}{t-t_0+1} \sum_{k=t_0}^{t} \hat{\theta}_k^{OEM} & \text{for } t \geq t_0. \end{cases}$$

Choosing an appropriate threshold t_0 can be more straightforward than choosing c for $\gamma_t = t^{-c}$, but it still requires the user to have an intuition for how the estimates for each parameter will behave. We will refer to this method as AVG, use $c = 0.6$, and set $t_0 = 50,000$ which is half the total observations for our examples.

4.2.3 Introspective Online Expectation Maximisation

We now introduce IOEM to address the issue of having to pre-specify a weighting sequence $\{\gamma_k\}_{k=1,\ldots}$. The algorithm is similar to OEM, but instead of pre-specifying γ_t, we estimate the precision and accuracy in the sufficient statistic updates $\{\tilde{s}_k\}_{k=\Delta+1,\ldots,t}$ and use these to determine the next weight γ_{t+1}. More precisely, we keep online estimates of a weighted regression on the dependent variables $\{\tilde{s}_k\}_{k=\Delta+1,\ldots,t}$ where the index k serves as the explanatory variable and the data point $(k, \tilde{s}_k)$ has regression

weight equal to its weighted sum weight η_k^t in equation (4.7). This weighted regression provides intercept and slope estimates $\hat{\beta}_0$, $\hat{\beta}_1$, and estimates of their variance $\hat{\sigma}_0^2$, $\hat{\sigma}_1^2$. We then use these estimates to define a proposed weight as follows:

$$\gamma_{t+1}^{reg} = \frac{|\hat{\beta}_1| + \hat{\sigma}_1}{\hat{\sigma}_0},$$

This definition of γ_{t+1}^{reg} ensures that a substantial slope estimate $\hat{\beta}_1$ indicating low accuracy in our previous parameter estimates will put a large weight on the incoming statistic, improving accuracy. A large $\hat{\sigma}_0$ reflecting low precision in the estimates will result in a small weight, so that successive estimates are smoothed out, improving precision.

We do not use standard weighted regression, where the weights are assumed to be inversely proportional to the variance of the observation; as this assumption is not justified here, the standard prodecure would lead to biased estimates of $\hat{\sigma}_{0,1}^2$ and would impact the performance of IOEM. Instead we assume that observations share an unknown variance, and we use the weights to modulate the influence of each observation to the estimates of both $\hat{\beta}_{0,1}$ and $\hat{\sigma}_{0,1}^2$. See Appendix C.3 for details.

We impose restrictions on γ_t which keep it between the most extreme choices for OEM. Taken together, the update step for γ becomes

$$\gamma_{t+1} = \min\left((t+1)^{-(0.5+\epsilon)}, \max\left(\gamma_{t+1}^{reg}, (t+1)^{-1}\right)\right)$$

where $0 < \epsilon \ll 1$.

These restrictions ensure that our algorithm satisfies the assumptions of Theorem 1 of [8], namely that $0 < \gamma_t < 1$, $\sum_{t=1}^{\infty} \gamma_t = \infty$, and $\sum_{t=1}^{\infty} \gamma_t^2 < \infty$. Hence for any model for which f and g satisfy the assumptions guaranteeing convergence of the standard OEM estimator, the IOEM algorithm (Algorithm 4.3) is also guaranteed to converge. The precise assumptions are detailed in Assumption 1, Assumption 2, and

Theorem 1 of [8].

Algorithm 4.3 Introspective Online Expectation Maximisation for a simplified autoregressive model

for For time $t \geq 1$ **do**
  Simulate and calculate weights of new particles using SMC with parameter $\hat{\theta}_{t-1}$
  Collect sufficient statistic $\tilde{s}_t = \sum_{i=1}^{N} w_t(X_{1:t}^{(i)}) \cdot (Y_{t-\Delta} - X_{1:t}^{(i)}(t-\Delta))^2$
  Maximise expected likelihood by setting $\hat{\theta}_t = \hat{S}_t^{IOEM} := \gamma_t \cdot \tilde{s}_t + (1-\gamma_t) \cdot \hat{S}_{t-1}^{IOEM}$
  Perform weighted regression on $\{(k, \tilde{s}_k)\}_{k=1,\ldots,t}$ to calculate γ_{t+1}
end for

The results of using BEM, OEM, and IOEM to perform parameter inference on the simplified autoregressive model (4.1) with a wide range of tuning parameters b from 100 to 10,000, and c from 0.6 to 0.9, are presented in Figure 4.1. Each EM method was run on 100 replicates to produce the box-plots and each run included a burn-in period of 500 observations. The choice of tuning parameter in BEM and OEM makes a significant difference to the precision of the estimate even after 100,000 observations. IOEM was able to recognize that behavior similar to BEM with $b = 10,000$ or OEM with $c = 0.9$ was optimal. The accuracy and precision of IOEM are comparable with those of the post-OEM averaging technique (AVG).

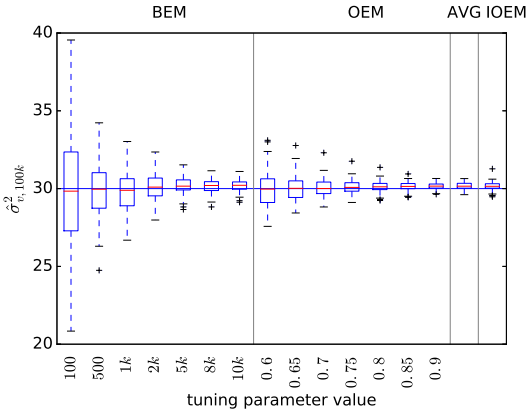


Figure 4.1: Comparison of EM methods on simplified AR model with known true parameters $a = .95$, $\sigma_w = 1$, and unknown true $\sigma_v^2 = 30$, and initial parameter estimate $\sigma_{v,0}^2 = 20$. $\hat{\sigma}_{v,100k}^2$ is plotted for 100 replicates, $N = 100$.

The adapting weight sequence $\{\gamma_k\}_{k=1,\ldots}$ sets IOEM apart from OEM. This for-

mulation of IOEM only works in the setting where θ has a linear relationship with a single sufficient statistic (here $\hat{\sigma}_{v,t}^2 = \hat{S}_t$) and is meant as an introduction to some of the ideas involved in IOEM. The method outlined in Algorithm 4.3 will not suffice when the function Λ mapping the sufficient statistics to θ does not have this simple form. We introduce the general IOEM algorithm in Section 4.3 below.

4.3 EM Simulations in the full autoregressive model

The model of Section 4.2 is special in that the sufficient statistic and the parameter of interest coincide. Generally this is not the case, leading to a more involved setup that we explore here. To this end, we now consider the full noisily-observed autoregressive model AR(1) with master equations as in (4.1), but now with unknown parameters a, σ_w, and σ_v. We define four summary sufficient statistics,

$$
\begin{aligned}
S_{1,t} &= \mathbb{E}_{X_{1:t}|Y_{1:t},\theta}\left[\frac{1}{t-1}\sum_{k=1}^{t-1} X_k^2\right], \\
S_{2,t} &= \mathbb{E}_{X_{1:t}|Y_{1:t},\theta}\left[\frac{1}{t-1}\sum_{k=1}^{t-1} X_k \cdot X_{k+1}\right], \\
S_{3,t} &= \mathbb{E}_{X_{1:t}|Y_{1:t},\theta}\left[\frac{1}{t-1}\sum_{k=2}^{t} X_k^2\right], \\
S_{4,t} &= \mathbb{E}_{X_{1:t}|Y_{1:t},\theta}\left[\frac{1}{t}\sum_{k=1}^{t} (Y_k - X_k)^2\right].
\end{aligned}
$$

Then, in BEM and OEM, we update the parameter estimates to

$$\hat{a}_t = \hat{S}_{2,t}/\hat{S}_{1,t}, \tag{4.8}$$

$$\hat{\sigma}_{w,t} = (\hat{S}_{3,t} - (\hat{S}_{2,t})^2/\hat{S}_{1,t})^{1/2}, \tag{4.9}$$

$$\hat{\sigma}_{v,t} = (\hat{S}_{4,t})^{1/2}, \tag{4.10}$$

where $\hat{S}_t$ is the appropriate approximation of S_t. These estimators are derived from the exponential family forms of the transition and emission distributions as in [62].

In most cases, as above, the function Λ mapping $\hat{S}_t$ to $\hat{\theta}_t$ is non-linear, and requires multiple sufficient statistics as input. To avoid potential bias, we want all sufficient statistics that inform one parameter estimate to share a weight sequence $\{\gamma_k\}_{k=1,2,\ldots}$. We therefore estimate an adapting weight sequence for each parameter independently, by performing the regression on the level of the parameter estimates (Algorithm 4.4), rather than on the level of the sufficient statistics. We will calculate $\hat{S}_t$ as in OEM equation (4.6) using our adapting weight sequence instead of a user specified weighting sequence. Because the adapting weight sequence is specific to each parameter, we will have multiple estimates of certain summary sufficient statistics. In this case $S_{1,t}$ and $S_{2,t}$ are estimated by $\hat{S}^a_{1,t}$ and $\hat{S}^a_{2,t}$ for (4.8) and by $\hat{S}^{\sigma_w}_{1,t}$ and $\hat{S}^{\sigma_w}_{2,t}$ for (4.9).

Simply regressing on $\hat{\theta}_{1:t}$ with respect to t would correspond to regression on $\hat{S}_{1:t}$, not $\tilde{s}_{1:t}$. As $\hat{S}$ is a running average, there is a strong correlation between $\hat{S}_{t-1}$ and $\hat{S}_t$ which largely depends on γ_t, and hence also a strong dependence between $\hat{\theta}_{t-1}$ and $\hat{\theta}_t$. In order to perform the regression on the parameters we must "unsmooth" $\hat{\theta}_{1:t}$ to create pseudo-independent parameter updates $\tilde{\theta}_t$ (see Algorithm 4.4). This is accomplished by taking linear combinations,

$$\tilde{\theta}_t := \frac{1}{\gamma_t} \cdot \hat{\theta}_t + \left(1 - \frac{1}{\gamma_t}\right) \cdot \hat{\theta}_{t-1},$$

where the coefficients $\frac{1}{\gamma_t}$ and $(1 - \frac{1}{\gamma_t})$ are chosen so as to minimise the covariance between successive updates, justifying the term pseudo-independent. The resulting updates correspond with the unsmoothed sufficient statistics updates $\tilde{s}_t$ used in Section 4.2.3. See Appendix C.4 for further details on this step.

Estimates for the parameters under different EM methods are presented in Figure 4.2. In the AR(1) model, IOEM outperforms most other EM methods when

Algorithm 4.4 Introspective Online Expectation Maximisation in the general model

for For time $t \geq 1$ **do**
  Simulate and calculate weights of new particles using SMC
  with parameter $\hat{\theta}_{t-1}^{IOEM}$
  Collect sufficient statistics $\tilde{s}_t$
  Update running average of sufficient statistics $\hat{S}_t = \gamma_t \tilde{s}_t + (1 - \gamma_t)\hat{S}_{t-1}$
  Maximise expected likelihood $\hat{\theta}_t = \Lambda(\hat{S}_t)$
  Create pseudo-independent parameter updates $\tilde{\theta}_t = \frac{1}{\gamma_t} \cdot \hat{\theta}_t + (1 - \frac{1}{\gamma_t}) \cdot \hat{\theta}_{t-1}$
  Perform weighted regression on $\{(k, \tilde{\theta}_k)\}_{k=1,\ldots,t}$ to calculate γ_{t+1}
end for

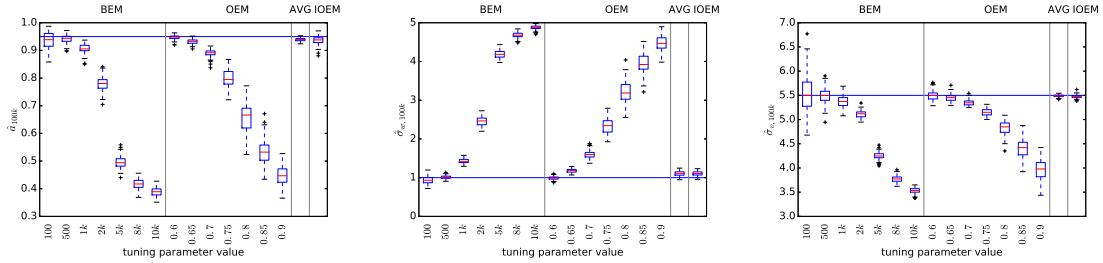


Figure 4.2: Comparison of EM methods on full autoregressive model with unknown true parameters $a = 0.95$, $\sigma_w = 1$, $\sigma_v = 5.5$ and inital parameters $a_0 = 0.8$, $\sigma_{w,0} = 3$, $\sigma_{v,0} = 1$. Parameter estimates at $t = 100,000$ are plotted for 100 replicates, $N = 100$.

estimating the parameters. In this case, OEM with $c = 0.6$ substantially outperforms OEM with $c = 0.9$. This is a result of the bad initial estimates. OEM with $c = 0.6$ forgets the earlier simulations much faster than OEM with $c = 0.9$ and hence is able to move its estimates of a, σ_w, and σ_v much more quickly. Here IOEM recognizes that it should have similar behavior to OEM with $c = 0.6$, whereas in the inference displayed in Figure 4.1 IOEM chose behavior similar to OEM with $c = 0.9$. IOEM can indeed adapt to the model.

4.4 EM simulations in a two-dimensional AR model

Now we investigate a model with a larger number of parameters and varying accuracy of initial parameter estimates. The main advantage of IOEM over OEM is its ability to adapt to each parameter independently. To highlight this, we applied IOEM to a

simple 2-dimensional autoregressive model. For this model we consider the sequences $\{Y^A, Y^B\}_{1:t}$ as observed, while $\{X^A, X^B\}_{1:t}$ are unobserved, where

$$X_t^A = a^A X_{t-1}^A + \sigma_w^A W_t^A, \qquad X_t^B = a^B X_{t-1}^B + \sigma_w^B W_t^B,$$
$$Y_t^A = X_t^A + \sigma_v V_t^A, \qquad Y_t^B = X_t^B + \sigma_v V_t^B,$$

where W_t^A, W_t^B, V_t^A, V_t^B are i.i.d. standard normal variates. Note that Y^A and Y^B are uncoupled, and that their master equations have independent parameters except for a shared parameter σ_v. By giving component A good initial estimates and B bad initial estimates, we can see how the different EM methods cope with a combination of accurate and inaccurate initialisations. Our expectation is that IOEM will be able to identify the set with good initial estimates (a^A, σ_w^A) and start smoothing out noise. While the other parameter estimates will be free to move quickly as they show no sign of nearing convergence (σ_w^B and σ_v because they are at the wrong value, a^B because it will be changing to compensate for σ_w^B and σ_v).

OEM with $c = 0.6$ and OEM with $c = 0.9$ both suffer in this model as they are both well suited to parameter estimation in one of the components, but not the other (Figure 4.3). IOEM on the other hand is able to capture the best of both worlds, striving for precision in component A and initially foregoing precision in favour of accuracy in component B. This leads to particularly favourable IOEM estimation of σ_v, which due to its dependence on components A and B, suffers the most from a blanket choice of tuning parameter in BEM or OEM.

A closer look at the estimate evoultion and weight sequence $\{\gamma_k\}_{k=1,\ldots,t}$ for a single run of a given EM technique shows IOEM broadly behaves like the optimal OEM or BEM choice (Figure 4.4). For the parameters which clearly benefit from a high learning rate (a^B, σ_w^B), the curve of IOEM updates resembles that of OEM and BEM with a high learning rate. For the parameter σ_v which benefits from a small

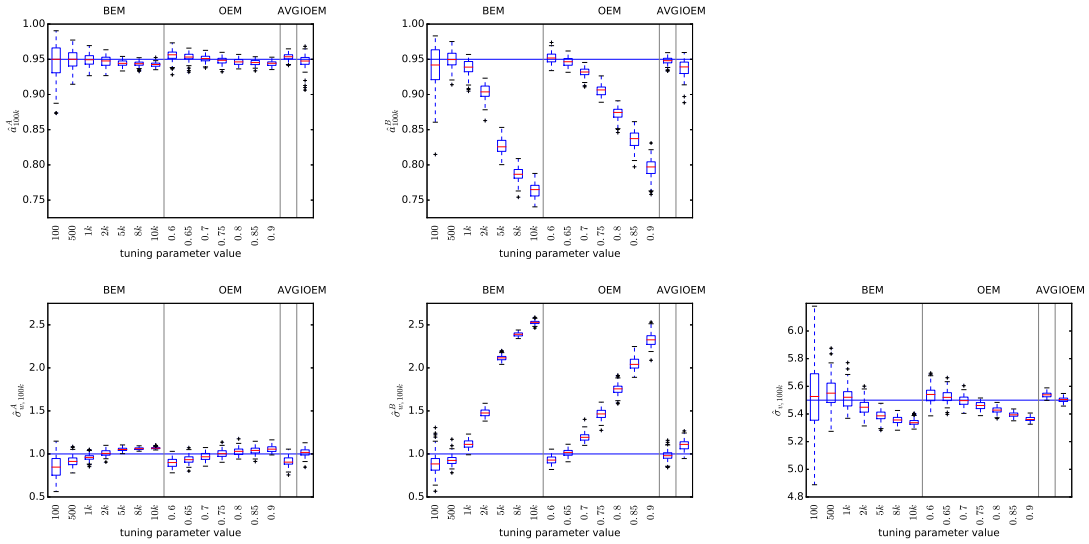


Figure 4.3: Comparison of EM methods on 2-dimensional autoregressive model with true parameters $a^A = 0.95$, $\sigma_w^A = 1$, $\sigma_v = 5.5$, $a^B = 0.95$, $\sigma_w^B = 1$ and inital parameters $a_0^A = 0.95$, $\sigma_{w,0}^A = 1$, $\sigma_{v,0} = 3$, $a_0^B = 0.95$, $\sigma_{w,0}^B = 3$. Parameter estimates at $t = 100,000$ are plotted for 100 replicates, $N = 100$.

learning rate, IOEM again finds the optimal weight sequence. The only exception appears to be a^A where IOEM adopts a large learning rate, when a small learning rate would increase precision.

In all cases the AVG OEM technique also produces accurate and precise estimates. However, if we set too low a threshold the estimates suffer. Figure 4.5 shows the cost of setting $t_0 = 10,000$ (as opposed to our standard $t_0 = 50,000$).

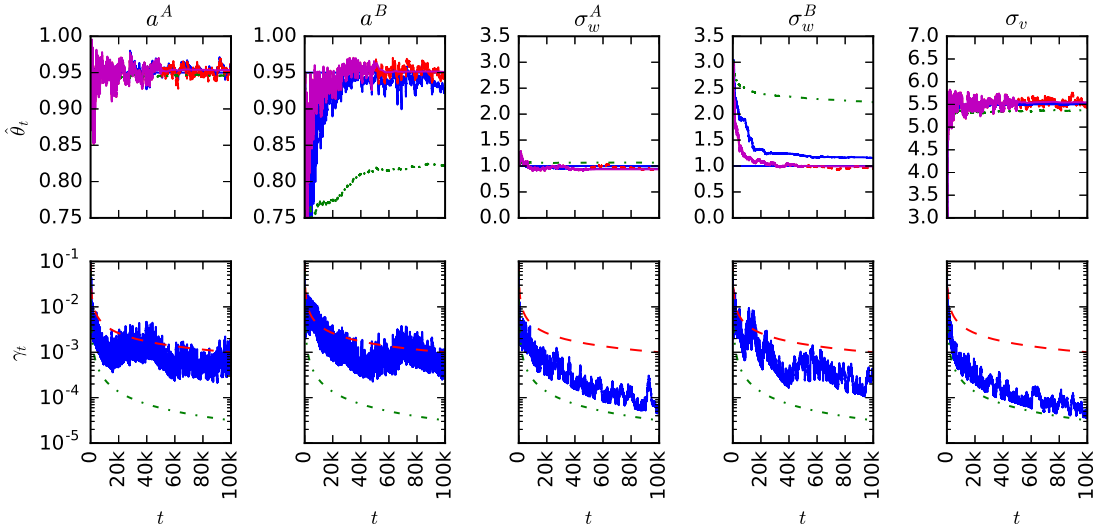


Figure 4.4: Parameter-specific convergence in the 2-dimensional autoregressive model over 100,000 observations. Each column displays information for a single parameter. The top row shows the sequence of parameter estimates for three EM methods. The bottom row shows the sequence of weights γ_t for the three EM methods. Blue solid line: IOEM; red dashed line: OEM with $c = 0.6$; green dash-dot line: OEM with $c = 0.9$; magenta solid line: averaged OEM technique with a threshold $t_0 = 50,000$.

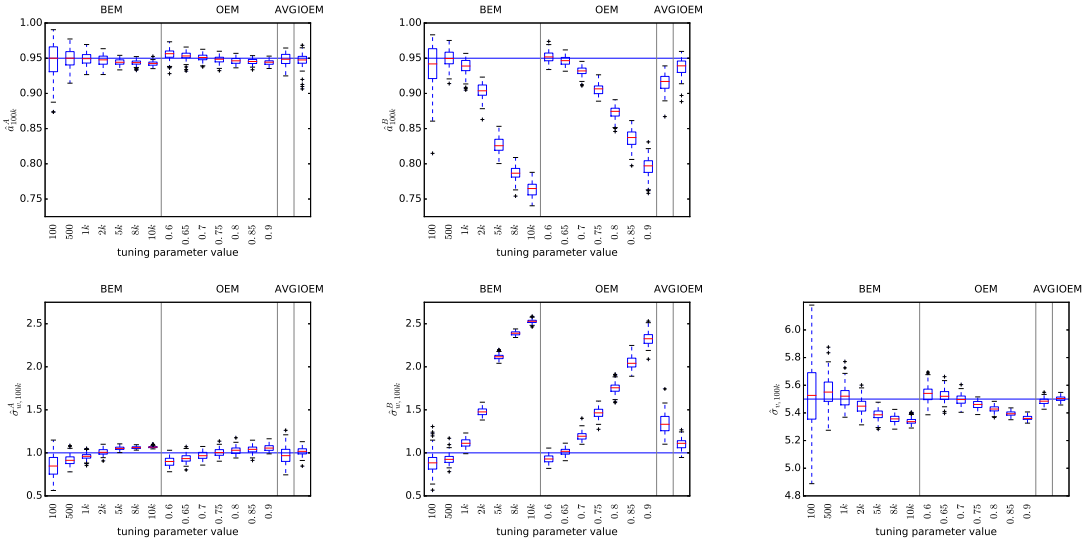


Figure 4.5: Comparison of EM methods on 2-dimensional autoregressive model with true parameters $a^A = 0.95$, $\sigma_w^A = 1$, $\sigma_v = 5.5$, $a^B = 0.95$, $\sigma_w^B = 1$ and inital parameters $a_0^A = 0.95$, $\sigma_{w,0}^A = 1$, $\sigma_{v,0} = 3$, $a_0^B = 0.95$, $\sigma_{w,0}^B = 3$. Parameter estimates at $t = 100,000$ are plotted for 100 replicates, $N = 100$. The averaged OEM technique has a threshold $t_0 = 10,000$.

4.5 The non-linear non-Gaussian stochastic volatility model

The previous sections have demonstrated IOEM is comparable to choosing the optimal tuning parameter in OEM or BEM in certain models. However, the models shown have all been based on the noisily observed autoregressive model, which is a linear Gaussian case where in practice analytic techniques would be prefered over SAEM. We now examine the behaviour of these algorithms when inferring the parameters of the stochastic volatility model defined by transition and emission densities

$$f(x_t|x_{t-1}) = (2\pi\sigma^2)^{-1/2}\exp\Big\{-\frac{(x_t-\phi x_{t-1})^2}{2\sigma^2}\Big\},$$
$$g(y_t|x_t) = (2\pi\beta^2 e^{x_t})^{-1/2}\exp\Big\{-\frac{1}{2\beta^2 e^{x_t}}y_t^2\Big\}.$$

We define four summary sufficient statistics,

$$S_{1,t} = \mathbb{E}_{X_{1:t}|Y_{1:t},\theta}\left[\frac{1}{t-1}\sum_{k=1}^{t-1} X_k \cdot X_{k+1}\right],$$
$$S_{2,t} = \mathbb{E}_{X_{1:t}|Y_{1:t},\theta}\left[\frac{1}{t-1}\sum_{k=1}^{t-1} X_k^2\right],$$
$$S_{3,t} = \mathbb{E}_{X_{1:t}|Y_{1:t},\theta}\left[\frac{1}{t-1}\sum_{k=2}^{t} X_k^2\right],$$
$$S_{4,t} = \mathbb{E}_{X_{1:t}|Y_{1:t},\theta}\left[\frac{1}{t}\sum_{k=1}^{t} e^{-X_k}\cdot Y_k^2\right].$$

Then the set of parameters that maximises the likelihood at step t are

$$\hat{\phi}_t = \hat{S}_{1,t}/\hat{S}_{2,t},$$
$$\hat{\sigma}_t = (\hat{S}_{3,t} - (\hat{S}_{1,t})^2/\hat{S}_{2,t})^{1/2},$$
$$\hat{\beta}_t = (\hat{S}_{4,t})^{1/2},$$

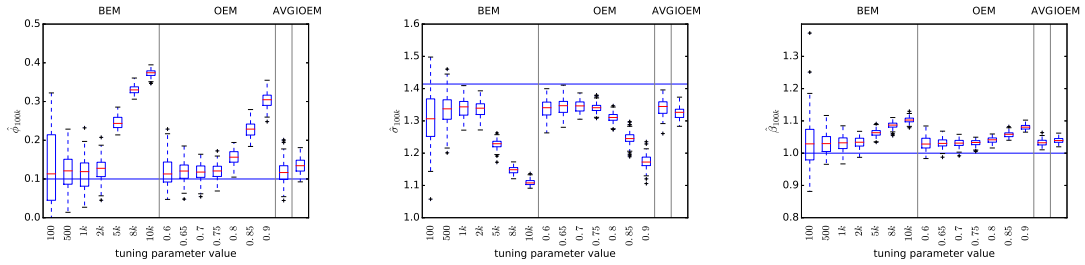


Figure 4.6: Comparison of EM methods on stochastic volatility model with unknown true parameters $\phi = 0.1$, $\sigma = \sqrt{2}$, $\beta = 1$ and inital parameters $\phi_0 = 0.5$, $\sigma_0 = 1$, $\beta_0 = \sqrt{2}$. Parameter estimates at $t = 100,000$ are plotted for 100 replicates, $N = 100$.

as in [62].

Again IOEM results in similar estimates to the optimal BEM/OEM and the online averaging technique with a well-chosen threshold (Figure 4.6). Although, for this model there is a notable bias in the estimates from all EM procedures. This is likely due to the use of the fixed-lag method which is known to introduce bias [62, 11, 60].

4.6 Discussion

We have shown that IOEM produces accurate and precise parameter estimates when applied to continuous state-space models. Across models with varying accuracy of the initial estimates, the efficiency of IOEM matches that of BEM/OEM with the optimal choice of tuning parameter. Moreover, when estimating multiple parameters within a single model, it is difficult to predict the optimal choice of $\{\gamma_{(k)}\}_{k=1,2,\ldots}$ for each. IOEM performs this task for the user resulting in better performance than BEM/OEM with a single learning rate parameter.

IOEM is able to optimize the EM procedure, with minimal prior knowledge of the model's behavior. Within a model IOEM will adapt to each individual parameter without the user guidance required in other EM methods. This provides an efficient, practical approach to parameter estimation in SMC methods.

The IOEM procedure does introduce its own time burden, but this should be

minimal compared to the gains from quick convergence. In a comparison of the time taken for 100 runs of the various methods IOEM had the 5th shortest runtime of the 15 (1 IOEM, 7 choices of c for OEM, 7 choices of b for BEM).

There is no systemic difference in the estimates obtained from IOEM and the OEM AVG technique. They maintain similar accuracy and the more precise technique varies between models. The OEM AVG method has the advantage of simplicity, but does require setting the threshold t_0 for each parameter.

For the purpose of decreasing computation time in SMC^2, IOEM seems to be the best choice. OEM AVG would be a better choice than BEM or OEM as its tuning parameter t_0 is more robust to the particularities of the model. However, it still requires some knowledge of the number of observations needed to guide each estimate within range of the true value. For complex demographic models with interdependencies between the parameters, this is infeasible. Moreover, the initially large learning rate in OEM AVG could lead to absorption of rate estimates at zero, particularly in cases with large N_e or low levels of migration. IOEM would give SMC^2 the ability to tune the learning rate of each parameter without the same risk of over-shooting the true value.

The benefits of employing either IOEM or OEM AVG are clear when processing the data in sequence. If CPUs are a limiting resource, then these methods would speed up convergence. However, OEM as described here is incompatible with parallelisation over multiple CPUs. If multiple CPUs are available for each run of SMC^2, then time is easiest saved by splitting the data into chunks. There is potential to combine the two speed ups in a variety of ways. One idea is to employ IOEM or OEM AVG on each parallel process and to average the estimates at the end. However, this is still an inefficient use of compute time, as all the processes will spend time around the initial bad estimates; it would be better to design a system where the estimates of the processes routinely synchronise. This type of synchronisation is more straightforward

in BEM. We could design a technique simialr to IOEM based around BEM instead of OEM. In the case of BEM, tracking the parameter estimates would inform the next batch size instead of the γ_t.

In its current form, SMC^2 is capable of inferring demography. However, its computational cost leads to imprecise estimates for many demographic scenarios. There are multiple avenues that could be persude in order to speed up estimate convergence. Any computational gains could then be redirected towards obtaining more precise estimates, either through more particles or more data.

Chapter 5

Conclusion

Modelling the ancestral recombination graph from sequence data remains a challenge to the field of population genetics. Genome sequence data contains a lot of information about population histories, but teasing out the signal is not an easy task. Much progress has been made in the last decade on inferring effective population sizes, but the complexity of multi-population models has made inference of directional migration rates difficult.

In this thesis, we tackled the problem of inferring demographic parameters by employing sequential Monte Carlo methods. This is an ambitious endeavour as the ARG space is high-dimensional and simulation based techniques typically struggle in these scenarios. Nevertheless, simulations show our method is capable of inferring approximately correct parameters under a variety of demographic models. In particular, SMC^2 is able to distinguish the directionality of migration in many cases.

We tested our method on data from the 1000 Genomes Project and Neanderthal data, to get a sense of the challenges arising from real data. Despite a lack of phasing information, missing sequence, and uncertainty in sample time, our inference was largely as expected from other methods.

The major limitation of SMC^2 is its long runtime. We have explored alternative

EM procedures which we hope will provide a speed boost to the algorithm. The first step to take in changing the update procedure would be to allow different learning rates for the different parameters. It is clear the N_e parameters are able to converge far more quickly than the migration rates. A more sophisticated method would recognise this and allow the N_e curve to converge as quickly as possible before wasting simulations under a clearly incorrect demography. Ideally, SMC^2 would run until estimate convergence according to a stopping condition, but with the current rate of estimate evolution this is infeasible.

Beyond the EM procedure, there are several avenues to explore to increase efficiency. SMC^2 may be improved by more sophisticated sampling strategies. We are currently considering a lookahead strategy as an alternative to the FSDR, which will likely be more efficient. The shortcoming of imprecise estimates may be remedied by using dynamic epochs which move the boundaries depending on the number of events in the epoch. In this case, the epochs for N_e and migration should be uncoupled, and potentially allow for changes in the number of epochs driven by the data. Although, in the case of migration, this may lead to particularly large epochs with little resolution for changes in the migration rate.

We have developed a framework for demographic inference, which I consider merely a starting point. The algorithm can easily be adapted for biological complexities and data imperfections. In particular, incorporating recombination hotspots should be a priority. There is even potential to infer hotspots, possibly even time-dependent hotspot maps.

SMC^2 is unique in explicitly inferring non-constant directional migration rates using both site-specific patterns and positional dependencies. The robustness to unphased data makes the method ideal for studying population histories of less studied species, although the fixing of population splits may then be an issue. We chose to focus on ancient samples as they suffer from the problems of imperfect data, but in the

case of Neanderthals and Denisovans there are relatively precise estimates for split times. A logical next application would be to investigate the Oceanian-Denisovan history.

Bibliography

[1] Leonardo Arbiza, Elaine Zhong, and Alon Keinan. Nre: a tool for exploring neutral loci in the human genome. *Bmc Bioinformatics*, 13(1):301, 2012.

[2] Anand Bhaskar and Yun S Song. Descartesrule of signs and the identifiability of population demographic models from genomic variation data. *Annals of statistics*, 42(6):2469, 2014.

[3] Benjamin A Black, Ryan R Neely, and Michael Manga. Campanian ignimbrite volcanism, climate, and the final decline of the neanderthals. *Geology*, 43(5): 411–414, 2015.

[4] Brian L Browning and Sharon R Browning. Detecting identity by descent and estimating genotype error rates in sequence data. *The American Journal of Human Genetics*, 93(5):840–851, 2013.

[5] Ralph Burgess and Ziheng Yang. Estimation of hominoid ancestral population sizes under bayesian coalescent models incorporating mutation rate variation and sequencing errors. *Molecular biology and evolution*, 25(9):1979–1994, 2008.

[6] Olivier Cappé. Online sequential monte carlo em algorithm. In *Statistical Signal Processing, 2009. SSP'09. IEEE/SP 15th Workshop on*, pages 37–40. IEEE, 2009.

[7] Olivier Cappé and Eric Moulines. On the use of particle filtering for maximum likelihood parameter estimation. In *Signal Processing Conference, 2005 13th European*, pages 1–4. IEEE, 2005.

[8] Olivier Cappé and Eric Moulines. On-line expectation–maximization algorithm for latent data models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(3):593–613, 2009.

[9] Olivier Cappé, Eric Moulines, and Tobias Rydén. *Inference in hidden Markov models*, volume 6. Springer, 2005.

[10] Olivier Cappé, Simon J Godsill, and Eric Moulines. An overview of existing methods and recent advances in sequential monte carlo. *Proceedings of the IEEE*, 95(5):899–924, 2007.

[11] Olivier Cappe, Pierre Del Moral, Arnaud Doucet, and Sumeetpal S. Singh. Particle implementations of the online expectation-maximization algorithm for state-space models. February 2012.

[12] Saneej B Chitralekha, J Prakash, H Raghavan, RB Gopaluni, and Sirish L Shah. A comparison of simultaneous state and parameter estimation schemes for a continuous fermentor reactor. *Journal of Process Control*, 20(8):934–943, 2010.

[13] 1000 Genomes Project Consortium et al. A global reference for human genetic variation. *Nature*, 526(7571):68, 2015.

[14] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[15] Qiliang Ding, Ya Hu, Shuhua Xu, Jiucun Wang, and Li Jin. Neanderthal introgression at chromosome 3p21. 31 was under positive natural selection in east asians. *Molecular biology and evolution*, 31(3):683–695, 2013.

[16] Peter Donnelly and Simon Tavare. Coalescents and genealogical structure under neutrality. *Annual review of genetics*, 29(1):401–421, 1995.

[17] Randal Douc and Olivier Cappé. Comparison of resampling schemes for particle filtering. In *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*, pages 64–69. IEEE, 2005.

[18] Arnaud Doucet and Adam M Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, 12:656–704, 2009.

[19] Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10(3): 197–208, 2000.

[20] Alison Etheridge. *Some Mathematical Models from Population Genetics: École D'Été de Probabilités de Saint-Flour XXXIX-2009*, volume 2012. Springer Science & Business Media, 2011.

[21] Laurent Excoffier, Isabelle Dupanloup, Emilia Huerta-Sánchez, Vitor C Sousa, and Matthieu Foll. Robust demographic inference from genomic and snp data. *PLoS Genet*, 9(10):e1003905, 2013.

[22] Joseph Felsenstein. Evolutionary trees from dna sequences: a maximum likelihood approach. *Journal of molecular evolution*, 17(6):368–376, 1981.

[23] Joseph Felsenstein. Statistical inference of phylogenies. *Journal of the Royal Statistical Society. Series A (General)*, pages 246–272, 1983.

[24] Piotr Fryzlewicz et al. Wild binary segmentation for multiple change-point detection. *The Annals of Statistics*, 42(6):2243–2281, 2014.

[25] Qiaomei Fu, Alissa Mittnik, Philip LF Johnson, Kirsten Bos, Martina Lari, Ruth Bollongino, Chengkai Sun, Liane Giemsch, Ralf Schmitz, Joachim Burger, et al. A revised timescale for human evolution based on ancient mitochondrial genomes. *Current Biology*, 23(7):553–559, 2013.

[26] Qiaomei Fu, Heng Li, Priya Moorjani, Flora Jay, Sergey M Slepchenko, Aleksei A Bondarev, Philip LF Johnson, Ayinuer Aximu-Petri, Kay Prüfer, Cesare de Filippo, et al. Genome sequence of a 45,000-year-old modern human from western siberia. *Nature*, 514(7523):445–449, 2014.

[27] Qiaomei Fu, Cosimo Posth, Mateja Hajdinjak, Martin Petr, Swapan Mallick, Daniel Fernandes, Anja Furtwängler, Wolfgang Haak, Matthias Meyer, Alissa Mittnik, et al. The genetic history of ice age europe. *Nature*, 2016.

[28] Neil J Gordon, David J Salmond, and Adrian FM Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 107–113. IET, 1993.

[29] Richard E Green, Anna-Sapfo Malaspinas, Johannes Krause, Adrian W Briggs, Philip LF Johnson, Caroline Uhler, Matthias Meyer, Jeffrey M Good, Tomislav Maricic, Udo Stenzel, et al. A complete neandertal mitochondrial genome sequence determined by high-throughput sequencing. *Cell*, 134(3):416–426, 2008.

[30] Robert C Griffiths and Paul Marjoram. Ancestral inference from samples of dna sequences with recombination. *Journal of Computational Biology*, 3(4):479–502, 1996.

[31] Robert C Griffiths and Paul Marjoram. An ancestral recombination graph. *Institute for Mathematics and its Applications*, 87:257, 1997.

[32] Ilan Gronau, Melissa J Hubisz, Brad Gulko, Charles G Danko, and Adam Siepel. Bayesian inference of ancient human demography from individual genome sequences. *Nature genetics*, 43(10):1031–1034, 2011.

[33] Ryan N Gutenkunst, Ryan D Hernandez, Scott H Williamson, and Carlos D Bustamante. Inferring the joint demographic history of multiple populations from multidimensional snp frequency data. *PLoS Genet*, 5(10):e1000695, 2009.

[34] Henrik B Hansen, Peter B Damgaard, Ashot Margaryan, Jesper Stenderup, Niels Lynnerup, Eske Willerslev, and Morten E Allentoft. Comparing ancient dna preservation in petrous bone and tooth cementum. *PloS one*, 12(1):e0170940, 2017.

[35] Kelley Harris. Evidence for recent, population-specific evolution of the human mutation rate. *Proceedings of the National Academy of Sciences*, 112(11):3439–3444, 2015.

[36] Kelley Harris and Rasmus Nielsen. Inferring demographic history from a spectrum of shared haplotype lengths. *PLoS Genet*, 9(6):e1003521, 2013.

[37] John Hawks. Significance of neandertal and denisovan genomes in human evolution. *Annual Review of Anthropology*, 42:433–449, 2013.

[38] Richard D Horan, Erwin Bulte, and Jason F Shogren. How trade saved humanity from biological exclusion: An economic theory of neanderthal extinction. *Journal of Economic Behavior & Organization*, 58(1):1–29, 2005.

[39] Richard R Hudson. Properties of a neutral allele model with intragenic recombination. *Theoretical population biology*, 23(2):183–201, 1983.

[40] Emilia Huerta-Sánchez, Xin Jin, Zhuoma Bianba, Benjamin M Peter, Nicolas Vinckenbosch, Yu Liang, Xin Yi, Mingze He, Mehmet Somel, Peixiang Ni, et al. Altitude adaptation in tibetans caused by introgression of denisovan-like dna. *Nature*, 512(7513):194–197, 2014.

[41] Markus Hürzeler and Hans R Künsch. Approximating and maximising the likelihood for a general state-space model. *Sequential Monte Carlo methods in practice*, pages 159–175, 2001.

[42] John A Kamm, Jonathan Terhorst, and Yun S Song. Efficient computation of the joint sample frequency spectra for multiple populations. *Journal of Computational and Graphical Statistics*, 26(1):182–194, 2017.

[43] Nicholas Kantas, Arnaud Doucet, Sumeetpal Sindhu Singh, and Jan Marian Maciejowski. An overview of sequential monte carlo methods for parameter estimation in general state-space models. In *15th IFAC Symposium on System Identification (SYSID), Saint-Malo, France.(invited paper)*, volume 102, page 117, 2009.

[44] P.J. Kaufman. *Smarter Trading: Improving Performance in Changing Markets*. McGraw-Hill, 1995. ISBN 9780070340022. URL `https://books.google.co.uk/books?id=ndq\_21wRJjEC`.

[45] Janet Kelso and Kay Prüfer. Ancient humans and the origin of modern humans. *Current opinion in genetics & development*, 29:133–138, 2014.

[46] John Frank Charles Kingman. The coalescent. *Stochastic processes and their applications*, 13(3):235–248, 1982.

[47] Matthias Krings, Anne Stone, Ralf W Schmitz, Heike Krainitzki, Mark Stoneking, and Svante Pääbo. Neandertal dna sequences and the origin of modern humans. *cell*, 90(1):19–30, 1997.

[48] Martin Kuhlwilm, Ilan Gronau, Melissa J Hubisz, Cesare de Filippo, Javier Prado-Martinez, Martin Kircher, Qiaomei Fu, Hernán A Burbano, Carles Lalueza-Fox, Marco de La Rasilla, et al. Ancient gene flow from early modern humans into eastern neanderthals. *Nature*, 530(7591):429–433, 2016.

[49] Heng Li and Richard Durbin. Inference of human population history from individual whole-genome sequences. *Nature*, 475(7357):493–496, 2011.

[50] Ming Lin, Rong Chen, Jun S Liu, et al. Lookahead strategies for sequential monte carlo. *Statistical Science*, 28(1):69–94, 2013.

[51] M Gallego Llorente, ER Jones, A Eriksson, V Siska, KW Arthur, JW Arthur, MC Curtis, JT Stock, M Coltorti, P Pieruccini, et al. Ancient ethiopian genome reveals extensive eurasian admixture in eastern africa. *Science*, 350(6262):820–822, 2015.

[52] Anna-Sapfo Malaspinas, Michael C Westaway, Craig Muller, Vitor C Sousa, Oscar Lao, Isabel Alves, Anders Bergström, Georgios Athanasiadis, Jade Y Cheng, Jacob E Crawford, et al. A genomic history of aboriginal australia. *Nature*, 2016.

[53] Paul Marjoram and Jeff D Wall. Fast" coalescent" simulation. *BMC genetics*, 7 (1):16, 2006.

[54] The Max Planck Institute for Evolutionary Anthropology. Vindija genomic data, 2016. `http://cdna.eva.mpg.de/neandertal/Vindija/`.

[55] Olivier Mazet, Willy Rodríguez, and Lounès Chikhi. Demographic inference using genetic data from a single individual: Separating population size variation from population structure. *Theoretical population biology*, 104:46–58, 2015.

[56] Gilean AT McVean and Niall J Cardin. Approximating the coalescent with recombination. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 360(1459):1387–1393, 2005.

[57] Fernando L Mendez, Joseph C Watkins, and Michael F Hammer. A haplotype at stat2 introgressed from neanderthals and serves as a candidate of positive selection in papua new guinea. *The American Journal of Human Genetics*, 91 (2):265–274, 2012.

[58] Simon Myers, Leonardo Bottolo, Colin Freeman, Gil McVean, and Peter Donnelly. A fine-scale map of recombination rates and hotspots across the human genome. *Science*, 310(5746):321–324, 2005.

[59] Simon Myers, Charles Fefferman, and Nick Patterson. Can one learn history from the allelic spectrum? *Theoretical population biology*, 73(3):342–348, 2008.

[60] Christopher Nemeth, Paul Fearnhead, and Lyudmila Mihaylova. Particle approximations of the score and observed information matrix for parameter estimation in state–space models with linear computational cost. *Journal of Computational and Graphical Statistics*, 25(4):1138–1157, 2016.

[61] Rasmus Nielsen, Joshua M Akey, Mattias Jakobsson, Jonathan K Pritchard, Sarah Tishkoff, and Eske Willerslev. Tracing the peopling of the world through genomics. *Nature*, 541(7637):302–310, 2017.

[62] Jimmy Olsson, Olivier Cappé, Randal Douc, Eric Moulines, et al. Sequential monte carlo smoothing with application to parameter estimation in nonlinear state space models. *Bernoulli*, 14(1):155–179, 2008.

[63] Pier Francesco Palamara, Todd Lencz, Ariel Darvasi, and Itsik Peer. Length distributions of identity by descent reveal fine-scale demographic history. *The American Journal of Human Genetics*, 91(5):809–822, 2012.

[64] Ron Pinhasi, Daniel Fernandes, Kendra Sirak, Mario Novak, Sarah Connell, Songül Alpaslan-Roodenberg, Fokke Gerritsen, Vyacheslav Moiseyev, Andrey Gromov, Pál Raczky, et al. Optimal ancient dna yields from the inner ear part of the human petrous bone. *PloS one*, 10(6):e0129102, 2015.

[65] Michael K Pitt and Neil Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American statistical association*, 94(446):590–599, 1999.

[66] Boris Teodorovich Polyak. A new method of stochastic approximation type. *Avtomatika i telemekhanika*, (7):98–107, 1990.

[67] Kay Prüfer. personal communication.

[68] Kay Prüfer, Fernando Racimo, Nick Patterson, Flora Jay, Sriram Sankararaman, Susanna Sawyer, Anja Heinze, Gabriel Renaud, Peter H Sudmant, Cesare De Filippo, et al. The complete genome sequence of a neanderthal from the altai mountains. *Nature*, 505(7481):43–49, 2014.

[69] Fernando Racimo, Sriram Sankararaman, Rasmus Nielsen, and Emilia Huerta-Sánchez. Evidence for archaic adaptive introgression in humans. *Nature Reviews Genetics*, 16(6):359–371, 2015.

[70] Bruce Rannala and Ziheng Yang. Bayes estimation of species divergence times and ancestral population sizes using dna sequences from multiple loci. *Genetics*, 164(4):1645–1656, 2003.

[71] Matthew D Rasmussen, Melissa J Hubisz, Ilan Gronau, and Adam Siepel. Genome-wide inference of ancestral recombination graphs. *PLoS Genet*, 10(5): e1004342, 2014.

[72] David Reich, Richard E Green, Martin Kircher, Johannes Krause, Nick Patterson, Eric Y Durand, Bence Viola, Adrian W Briggs, Udo Stenzel, Philip LF Johnson, et al. Genetic history of an archaic hominin group from denisova cave in siberia. *Nature*, 468(7327):1053, 2010.

[73] John H Relethford. Genetic evidence and the modern human origins debate. *Heredity*, 100(6):555, 2008.

[74] Stephan Schiffels and Richard Durbin. Inferring human population size and separation history from multiple genome sequences. *Nature genetics*, 46(8):919–925, 2014.

[75] David Serre, André Langaney, Mario Chech, Maria Teschler-Nicola, Maja Paunovic, Philippe Mennecier, Michael Hofreiter, Göran Possnert, and Svante Pääbo. No evidence of neandertal mtdna contribution to early modern humans. *PLoS biology*, 2(3):e57, 2004.

[76] Sara Sheehan, Kelley Harris, and Yun S Song. Estimating variable effective population sizes from multiple genomes: a sequentially markov conditional sampling distribution approach. *Genetics*, 194(3):647–662, 2013.

[77] Paul R Staab, Sha Zhu, Dirk Metzler, and Gerton Lunter. Scrm: efficiently simulating long sequences using the approximated coalescent with recombination. *Bioinformatics*, 31(10):1680–1682, 2015.

[78] Matthias Steinrücken, Joshua S Paul, and Yun S Song. A sequentially markov conditional sampling distribution for structured populations with migration and recombination. *Theoretical population biology*, 87:51–61, 2013.

[79] Matthias Steinrücken, John A Kamm, and Yun S Song. Inference of complex population histories using whole-genome sequences from multiple populations. *bioRxiv*, page 026591, 2015.

[80] Jonathan Terhorst, John A Kamm, and Yun S Song. Robust and scalable inference of population history from hundreds of unphased whole genomes. Technical report, Nature Research, 2016.

[81] Peter R Wilton, Shai Carmi, and Asger Hobolth. The smc is a highly accurate approximation to the ancestral recombination graph. *Genetics*, 200(1):343–355, 2015.

Appendix A

Calculation of sufficient statistics

A.1 Recombination rate

Recall our aim is to maximise the Q-function

$$\begin{aligned} Q(\theta|\hat{\theta}_j) &= \int \log p_\theta(x_{0:t}, y_{0:t}) p_{\hat{\theta}_j}(x_{0:t}|y_{0:t}) dx_{0:t}, \\ &= \mathbb{E}_{\hat{\theta}_j}[\log \mu_\theta | y_{0:t}] + \sum_{k=2}^{t} \mathbb{E}_{\hat{\theta}_j}[\log f_\theta(X_k|X_{k-1})|y_{0:t}] + \sum_{k=1}^{t} \mathbb{E}_{\hat{\theta}_j}[\log g_\theta(y_k|X_k)|y_{0:t}], \end{aligned}$$

with respect the ρ,

$$\begin{aligned} \frac{\partial Q(\theta|\hat{\theta}_j)}{\partial \rho} &= \frac{\partial}{\partial \rho} \sum_{k=2}^{t} \mathbb{E}_{\hat{\theta}_j}[\log f_\theta(X_k|X_{k-1})|y_{0:t}], \\ &= \frac{\partial}{\partial \rho} \mathbb{E}_{\hat{\theta}_j}[\log p_\theta(X_{1:k}|X_0)|y_{0:t}], \\ &= \frac{\partial}{\partial \rho} \sum_{i=1}^{N} w_i \log p_\theta(X_{1:k}^{(i)}|X_0). \end{aligned}$$

Recall equation (2.2)

$$p(X_{k:l}|X_k = x_k) = \Big(\prod_{j=1}^{|x|} \rho e^{-\rho B(x_{k_j})(k_{j+1}-k_j)}\Big) \cdot \Big(\prod_{j=1}^{|x|} C_{\tau_j} b_{\tau_j}(x_{k_{j-1}}) e^{-\sum_{t=1}^{T-1} b_{h_t^j}(x_{k_{j-1}}) C_{h_t^j}(h_{t+1}^j - h_t^j)}\Big).$$

where τ_j is the height of the j-th coalescence and h_1^j is the height of the j-th recombination. Hence

$$\frac{\partial Q(\theta|\hat{\theta}_j)}{\partial \rho} = \sum_{i=1}^{N} w_i \Big(\frac{|x|^{(i)}}{\rho} - \sum_{j=1}^{|x|^{(i)}} B(x_{k_j}^{(i)})(k_{j+1}^{(i)} - k_j^{(i)})\Big)$$

$$\implies 0 = \sum_{i=1}^{N} w_i \Big(\frac{|x|^{(i)}}{\hat{\rho}} - \sum_{j=1}^{|x|^{(i)}} B(x_{k_j}^{(i)})(k_{j+1}^{(i)} - k_j^{(i)})\Big)$$

$$\implies \hat{\rho} = \frac{\sum_{i=1}^{N} w_i |x|^{(i)}}{\sum_{i=1}^{N} w_i \sum_{j=1}^{|x|^{(i)}} B(x_{k_j}^{(i)})(k_{j+1}^{(i)} - k_j^{(i)})}.$$

And so our sufficient statistics are additive functionals. $S^{\text{rec}}_{\text{count}}$ is the count of recombination events $|x|$ and $S^{\text{rec}}_{\text{opp}}$ is the opportunity for recombination. The Q-function is maximised by setting

$$\hat{\rho} = \frac{S_1^{rec}}{S_2^{rec}}.$$

In practice, we use the further approximations detailed in Appendix C.2.

A.2 Effective population size

The derivation for the sufficient statistics needed to estimate N_e are similar. Recall that we define the coalescent rate $C_u = 1/2N_e(u)$. Let U be one of our specifed epochs, i.e. a time interval with a closed lower bound and open upper bound, with constant N_e. We will now denote the coalescent rate of epoch U as C_U. Then our estimate of the coalescent rate for this epoch is obtained by maximising the Q-function with respect to C_U.

$$\frac{\partial Q(\theta|\hat{\theta}_j)}{\partial C_U} = \frac{\partial}{\partial C_U}\Big(\mathbb{E}_{\hat{\theta}_j}[\log\mu_\theta(X_0)|y_{0:t}] + \mathbb{E}_{\hat{\theta}_j}[\log p_\theta(X_{1:k}|X_0)|y_{0:t}]\Big). \tag{A.1}$$

The first term of the sum in equation (A.1) is the probability density of the genealogy simulated at position 0. Define λ to be one less than the number of haploid samples, so that there are λ coalescence in the genealogy. Let m index the coalescences from the bottom of the tree up. For particle i, $T^{(i)}$ is the number of epoch boundaries plus the number of coalescences, so that $h_1, \ldots, h_{T^{(i)}}$ are the heights at which either the number of lineages changes or the N_e changes.

$$\begin{aligned}
&\frac{\partial}{\partial C_U}\mathbb{E}_{\hat{\theta}_j}[\log\mu_\theta(X_0)|y_{0:t}] \\
&\quad= \frac{\partial}{\partial C_U}\sum_{i=1}^{N} w_i \log\Big(\prod_{m=1}^{\lambda} C_{\tau_m^{(i)}}(\lambda+2-m)e^{-\sum_{t=1}^{T^{(i)}-1}(\lambda+2-m)C_{h_t^{(i)}}(h_{t+1}^{(i)}-h_t^{(i)})}\Big), \\
&\quad= \frac{\partial}{\partial C_U}\sum_{i=1}^{N} w_i\Big(\sum_{m=1}^{\lambda}[\log(C_{\tau_m^{(i)}}(\lambda+2-m)) - \sum_{t=1}^{T^{(i)}-1}(\lambda+2-m)C_{h_t^{(i)}}(h_{t+1}^{(i)}-h_t^{(i)})\Big), \\
&\quad= \sum_{i=1}^{N} w_i\Big(\sum_{m=1}^{\lambda}\frac{1}{C_U}\mathbb{1}(\tau_m^{(i)}\in U) - \sum_{t=1}^{T^{(i)}-1}(\lambda+2-m)(h_{t+1}^{(i)}-h_t^{(i)})\mathbb{1}(h_t^{(i)}\in U)\Big).
\end{aligned}$$

The second term of the sum in equation (A.1) is the probability density of the

coalescences caused by recombination events along the sequence. Here T is dependent on the particle i and the position of the recomination k_j, but for simplicity we will simply write T. We also drop the particle index superscripts for $\tau^{(i)}$, $h^{(i)}$, $|x|^{(i)}$, and $x^{(i)}$.

$$\begin{aligned}
&\frac{\partial}{\partial C_U}\mathbb{E}_{\hat{\theta}_j}[\log p_\theta(X_{1:k}|X_0)|y_{0:t}] \\
&\qquad = \frac{\partial}{\partial C_U}\sum_{i=1}^{N} w_i \Bigg(\sum_{j=1}^{|x|} \Big(\log(C_{\tau_j} b_{\tau_j}(x_{k_{j-1}})) - \sum_{t=1}^{T-1} b_{h_t}(x_{k_{j-1}}) C_{h_t^j}(h_{t+1}^j - h_t^j) \Big) \Bigg), \\
&\qquad = \sum_{i=1}^{N} w_i \Bigg(\sum_{j=1}^{|x|} \Big(\frac{1}{C_{\tau_j}} \mathbb{1}(\tau_j \in U) - \sum_{t=1}^{T-1} b_{h_t}(x_{k_{j-1}})(h_{t+1}^j - h_t^j)\mathbb{1}(h_t^j \in U) \Big) \Bigg).
\end{aligned}$$

Setting the derivative equal to zero and solving for $\hat{C}_U$ yields

$$\hat{C}_U = \frac{S'_{\text{count}} + S_{\text{count}}}{S'_{\text{opp}} + S_{\text{opp}}},$$

where

$$\begin{aligned}
S'_{\text{count}} &= \sum_{i=1}^{N} w_i \sum_{m=1}^{\lambda} \mathbb{1}(\tau_m \in U), \\
S_{\text{count}} &= \sum_{i=1}^{N} w_i \sum_{j=1}^{|x|} \mathbb{1}(\tau_j \in U), \\
S'_{\text{opp}} &= \sum_{i=1}^{N} w_i \Big(\sum_{m=1}^{\lambda} \sum_{t=1}^{T-1} (\lambda + 2 - m)(h_{t+1} - h_t)\mathbb{1}(h_t \in U) \Big), \\
S_{\text{opp}} &= \sum_{i=1}^{N} w_i \Big(\sum_{j=1}^{|x|} \sum_{t=1}^{T-1} b_{h_t}(x_{k_{j-1}})(h_{t+1}^j - h_t^j)\mathbb{1}(h_t^j \in U) \Big).
\end{aligned}$$

Again we have a count statistic in the numerator and an opportunity statistic in the

denominator.

The sufficient statistics for the migration rate are almost identical. The migration of a lineage does not depend on the number of contemporary branches, and so the migration sufficient statistics do not involve the $\lambda + 2 - m$ and $b_{h_t}(x_{k_{j-1}})$ terms.

Appendix B

Demographic models

B.1 Single population models

B.1.1 Constant

scrm command:

```
./scrm 8 1 -N0 10000 -t 100000.0 -r 40000.0 100000000.0 -eN 0 1 -eN 0.02
1 -eN 0.1 1 -eN 0.5 1 -seed 104 -T -L -p 10 -l 300000
```

this model is used in Figures:

Figure 2.5 on pg. 44; Figure 2.8 on pg. 48.

B.1.2 Bottleneck

scrm command:

```
./scrm 8 1 -N0 10000 -t 100000.0 -r 40000.0 100000000.0 -eN 0 1 -eN 0.01
0.1 -eN 0.06 1 -eN 0.2 0.5 -eN 1 1 -eN 2 2 -seed 103 -T -L -p 10 -l 300000
```

this model is used in Figures:

Figure 2.7 on pg. 47; Figure 2.9 on pg. 49; Figure 2.10 on pg. 50; Figure 2.11 on pg. 50;

Figure 2.12 on pg. 51; Figure 2.13 on pg. 51; Figure 2.14 on pg. 52; Figure 2.17 on pg. 57; Figure 2.19 on pg. 61; Figure 2.26 on pg. 69; Figure 2.31 on pg. 74.

B.1.3 Zigzag

scrm command:

```
./scrm 8 1 -N0 14312 -t 71560.0 -r 20036.8 100000000.0 -eN 0 5 -eG 0.000582262
1318.18 -eG 0.00232905 -329.546 -eG 0.00931619 82.3865 -eG 0.0372648 -20.5966
-eG 0.149059 5.14916 -eN 0.596236 0.5 -seed 108 -T -L -p 10 -l 300000
```

this model is used in Figures:

Figure 2.24 on pg. 66; Figure 2.25 on pg. 66.

B.2 Two population models

B.2.1 Uni-directional migration

scrm command:

```
./scrm 8 1 -N0 10000 -t 100000.0 -r 40000.0 100000000.0 -I 2 4 4 -eN 0 1
-ema 0 0 0.2 0 0 -eN 0.1 1 -ema 0.1 0 0.2 0 0 -eN 0.5 1 -eM 0.5 0 -ej .5
2 1 -seed 107 -T -L -p 10 -l 300000
```

this model is used in Figures:

Figure 2.15 on pg. 53; Figure 2.20 on pg. 62; Figure 2.28 on pg. 71; Figure 2.32 on pg. 74.

B.2.2 Split no migration

scrm command:

```
./scrm 8 1 -N0 10000 -t 100000.0 -r 40000.0 100000000.0 -I 2 4 4 -eN 0 1
-eM 0 0 -eN 0.1 1 -eM 0.1 0 -eN 0.5 1 -eM 0.5 0 -ej .5 2 1 -seed 103 -T
-L -p 10 -l 300000
```

this model is used in Figures:

Figure 2.16 on pg. 53; Figure 2.22 on pg. 62.

B.2.3 Period of migration

scrm command:

```
./scrm 8 1 -N0 10000 -t 100000.0 -r 40000.0 100000000.0 -I 2 4 4 -en 0 1
0.4 -en 0 2 1.5 -ema 0 0 0.17 0 0 -en 0.056 1 0.1 -en 0.056 2 0.3 -ema 0.056
0 0.17 0 0 -en 0.066 1 1 -en 0.066 2 0.3 -ema 0.066 0 0.17 0 0 -en 0.106
1 1 -en 0.106 2 0.3 -eM 0.106 0 -en 0.156 1 2.7 -en 0.156 2 0.3 -eM 0.156
0 -en 0.356 1 2.7 -en 0.356 2 0.6 -eM 0.356 0 -eN 0.506 1.8 -eM 0.506 0
-ej .506 1 2 -seed 107 -T -L -p 10 -l 300000
```

this model is used in Figures:

Figure 2.21 on pg. 62; Figure 2.23 on pg. 63.

B.2.4 Archaic Uni-directional migration

scrm command:

```
./scrm 6 1 -t 300000 -r 120000 300000000 -T -p 10 -I 2 4 0 -eI .044 0 2
-ej 0.55 1 2 -eM 0 0 -em 0.044 1 2 .17 -eM .15 0 -en 0 1 4 -en .01 1 1 -en
.02 1 .4 -en .1 1 .1 -en .11 1 1 -en .2 1 2.7 -en .55 1 1.8 -en .044 2 1.5
-en .1 2 .3 -en .4 2 .6 -seed 1
```

this model is used in Figures:

Figure 3.2 on pg. 88; Figure 3.7 on pg. 92.

motivation for migration rate:

In this model we impose a time span for a constant non-zero migration rate, and want the rate to correspond to approximately 2% of the genome deriving from the other population. Given the length of time for possible migration and the extent of the sequence inherited from the other population, we can approximate the rate of migration. In this model we allow for 106ky of migration. This is equivalent to 4,240 generations of possible migration, assuming generations of 25 years. In a single haplotype, 2% of bases should have their ancestral lineage migrate. If we assume a recombination occurs at the bottom of the time period, the lineage then has the option of migrating or remaining in its population for the 4,240 generations. We impose that 2% of such lineages migrate in the 4,240 generations, so they migrate at a rate of $\frac{.02}{4240}$ =4.72e-6 migrations per base per generation, which is 0.189 in coalescent time. This is an overestimate of the migration rate as it ensures 2% of such lineages migrate instead of 2% of bases. To ensure 2% of bases migrate, we need to consider the dependencies along the sequence, and so consider recombination events that occur during the migration time period. When considering a recombination during the period, the chance of a non-migrating branch transitioning to a migrating branch is larger than the chance of a migrating branch transitioning to a non-migrating branch, due to the relative opportunity for recombination. This suggests the proportion of trees with a migration among those that experience a recombination below the time period is less than 2%. We do not attempt to calculate this factor, but instead lower our estimate to 0.17 (4.25e-6 migrations per base per generation).

B.2.5 Archaic Uni-directional very weak migration

scrm command:

```
./scrm 6 1 -t 300000 -r 120000 300000000 -T -p 10 -I 2 4 0 -eI .044 0 2
-ej 0.55 2 1 -eM 0 0 -em 0.044 1 2 .017 -eM .15 0 -en 0 1 4 -en .01 1 1
-en .02 1 .4 -en .1 1 .1 -en .11 1 1 -en .2 1 2.7 -en .55 1 1.8 -en .044
2 1.5 -en .1 2 .3 -en .4 2 .6 -seed 1
```

this model is used in Figures:

Figure 3.6 on pg. 90.

B.2.6 Archaic Bi-directional migration

scrm command:

```
./scrm 6 1 -t 300000 -r 120000 300000000 -T -p 10 -I 2 4 0 -eI .044 0 2
-ej 0.55 2 1 -eM 0 0 -eM 0.044 .17 -eM .15 0 -en 0 1 4 -en .01 1 1 -en .02
1 .4 -en .1 1 .1 -en .11 1 1 -en .2 1 2.7 -en .55 1 1.8 -en .044 2 1.5 -en
.1 2 .3 -en .4 2 .6 -seed 1
```

this model is used in Figures:

Figure 3.3 on pg. 88; Figure 3.4 on pg. 89; Figure 3.8 on pg. 93; Figure 3.9 on pg. 93.

B.2.7 Archaic Bi-directional weak migration

scrm command:

```
./scrm 6 1 -t 300000 -r 120000 300000000 -T -p 10 -I 2 4 0 -eI .044 0 2
-ej 0.55 2 1 -eM 0 0 -eM 0.044 .085 -eM .15 0 -en 0 1 4 -en .01 1 1 -en
.02 1 .4 -en .1 1 .1 -en .11 1 1 -en .2 1 2.7 -en .55 1 1.8 -en .044 2 1.5
-en .1 2 .3 -en .4 2 .6 -seed 1
```

this model is used in Figures:

Figure 3.5 on pg. 90.

B.2.8 Archaic Bi-directional very weak migration

scrm command:

```
./scrm 6 1 -t 300000 -r 120000 300000000 -T -p 10 -I 2 4 0 -eI .044 0 2
-ej 0.55 2 1 -eM 0 0 -eM 0.044 .017 -eM .15 0 -en 0 1 4 -en .01 1 1 -en
.02 1 .4 -en .1 1 .1 -en .11 1 1 -en .2 1 2.7 -en .55 1 1.8 -en .044 2 1.5
-en .1 2 .3 -en .4 2 .6 -seed 1
```

this model is used in Figures:

Figure 3.6 on pg. 90.

Appendix C

IOEM supplement

C.1 Notation reference

Table C.1: This table serves as a reference for notation used throughout the paper

notation	meaning	associated EM methods
θ	true parameter	all
$\hat{\theta}_t$	parameter estimate at time t	all
$\tilde{\theta}_t$	pseudo-independent parameter update	IOEM
$\tilde{s}_t$	sufficient statistic update at time t	all
$\hat{S}_t$	summary sufficient statistic from averaging $\tilde{s}$	all
N	number of particles	all
Δ	lag of fixed-lag technique	all
$\hat{\beta}_0$	regression intercept ML estimate	IOEM
$\hat{\beta}_1$	regression slope ML estimate	IOEM
$\hat{\sigma}_0^2$	variance of regression intercept ML estimate	IOEM
$\hat{\sigma}_1^2$	variance of regression slope ML estimate	IOEM

C.2 Fixed-lag technique

Our fixed-lag technique is slightly different than that proposed in the literature [7, 62]. Compared to the existing approach it uses less intermediate storage. Recall that the

approximation we aim to evaluate is

$$\hat{S}_t = \sum_i w_t(X_{1:t}^{(i)}) \cdot \sum_{u=1}^{t} s_u(X_{1:t}^{(i)}(u), Y(u)),$$

where the sufficient statistic is written explicitly as a sum over the path traced out by the particle $X_{1:t}^{(i)}$. The drawback is that for $u \ll t$ the paths will have collapsed due to resampling, increasing the variance for those contributions to S. The solution proposed in [7] is to use instead the approximation

$$\hat{S}_t \approx \sum_i \Bigg(\sum_{u=1}^{t-\Delta} w_{u+\Delta}(X_{1:u+\Delta}^{(i)}) s_u(X_{1:u+\Delta}^{(i)}(u), Y(u)) + w_t(X_{1:t}^{(i)}) \sum_{u=t-\Delta+1}^{t} s_u(X_{1:t}^{(i)}(u), Y(u)) \Bigg).$$

This requires storing the quantities

$$\{s_u(X_{1:u+\Delta}^{(i)}(u)), Y(u)\}_{u=t-\Delta,\ldots,t}$$

for each sufficient statistic and each particle. This storage can be expensive if large numbers of sufficient statistics are tracked. Instead, at iteration t we use the approximation

$$\hat{S}_t \approx \sum_{u=1}^{t-\Delta} \sum_i w_{u+\Delta}(X_{1:u+\Delta}^{(i)}) s_u(X_{1:u+\Delta}^{(i)}(u), Y(u)).$$

By disregarding terms involving s_u for $u > t-\Delta$ and switching the summation in this way, we can now update $\hat{S}$ at each iteration by adding the contribution of the current particles to a single summary statistic at a distance Δ, without requiring per-particle storage other than each particle's recent history.

C.3 Weighted regression

The term "weighted regression" usually refers to regression where the errors are independent and normally distributed with zero mean and known variance (up to a multiplicative constant), and the data is weighted inversely proportionally to its variance. In our case, the data is assumed to drift, contributing an additional, non-independent term to the error. Weights are used to only focus on recent data where the drift contributes an error of the same order of magnitude as the normally distributed noise, while discounting the impact of data points further away. In this setup we are interested both in estimating the regression coefficients, and the error in these estimates.

Perry Kaufman's adaptive moving average (AMA) [44] is a similar averaging technique which reacts to the trends and volatility (jointly referred to as the behaviour) of the sequence. The difference lies in the measure of the behaviour. AMA relies on a user specified window length n. The n most recent data points are used to measure the behaviour. This would be equivalent to using equally-weighted linear regression over the last n points. By using weighted regression, the contribution of points to the behaviour measures is also influenced by the previously observed behaviour. For example, a sharp trend will effectively employ a smaller n value as we have lost interest in the behaviour before that trend.

Let X be the $2 \times n$ matrix consisting of a column of 1s and a column with the dependent variable, let y be the vector of observations, let β be the two coefficients, and ϵ the vector of errors, with $\epsilon_k \sim N(0, \sigma^2)$. Finally let w be a vector of weights. We estimate β by minimising

$$\begin{aligned} s^2 &= \sum_k w_k^2 (y_k - \beta_0 - \beta_1 x_{k,2})^2 \\ &= (X_w \beta - y_w)^\top (X_w \beta - y_w), \end{aligned}$$

where X_w and y_w are defined as

$$X_w := \begin{bmatrix} w_1 & w_1 \cdot (-n+1) \\ \vdots & \vdots \\ w_n & w_n \cdot 0 \end{bmatrix}; \qquad y_w := \begin{bmatrix} w_1 \cdot y_1 \\ \vdots \\ w_n \cdot y_n \end{bmatrix}.$$

The derivative is $\partial s^2/\partial\beta = 2(X_w\beta - y_w)^\top X_w$; equating to zero and solving for β results in the standard estimator for weighted regression

$$\hat{\beta} = (X_w^\top X_w)^{-1} X^\top y_w,$$

or more explicitly

$$\begin{aligned} \hat{\beta}_1 &= \frac{(\sum w_k^2 x_{2k} y_k) - (\sum w_k^2 x_{2k})(\sum w_k^2 y_k)}{(\sum w_k^2 x_{2k}^2) - (\sum w_k^2 x_{2k})^2}, \\ \hat{\beta}_0 &= \frac{(\sum w_k^2 x_{2k}^2)(\sum w_k^2 y_k) - (\sum w_k^2 x_{2k} y_k)(\sum w_k^2 x_{2k})}{(\sum w_k^2 x_{2k}^2) - (\sum w_k^2 x_{2k})^2}. \end{aligned}$$

Because the estimators of the coefficients can be written in this way, we can quickly update the estimates in an online manner as k increases simply by storing and updating the above summations. The variance in the estimate $\hat{\beta}$ can be estimated as follows

$$\begin{aligned} \operatorname{var}\hat{\beta} &= \operatorname{var}(X_w^\top X_w)^{-1} X_w^\top y_w \\ &= \operatorname{var}(X_w^\top X_w)^{-1} X_w^\top \epsilon_w \\ &= E\left[(X_w^\top X_w)^{-1} X_w^\top \epsilon_w \epsilon_w^\top X_w (X_w^\top X_w)^{-1}\right] \\ &= (X_w^\top X_w)^{-1} X_w^\top \operatorname{diag}(w_k^2 \sigma^2) X_w (X_w^\top X_w)^{-1}. \end{aligned}$$

If $w_k^2 = 1$ this simplifies to the usual $\operatorname{var}\hat{\beta} = \sigma^2(X^\top X)^{-1}$. Writing out the expression for $\operatorname{var}\hat{\beta}$ explicitly shows that it is again possible to find online updates for the relevant

terms.

C.4 Pseudo-independent parameter updates

In order to perform our regression on the level of the parameters, we need to map from $\tilde{s}^{(t)}$ to $\hat{S}^{(t)}$ and then to $\hat{\theta}^{(t)}$. We do not wish to regress on $\hat{\theta}^{(1:t)}$, as $\hat{\theta}^{(t-1)}$ and $\hat{\theta}^{(t)}$ are highly correlated. Instead we want a sequence defined in the parameter space where the correlations resemble those in $\tilde{s}^{(1:t)}$. We define this sequence as

$$\tilde{\theta}_t := \frac{1}{\gamma_t}\hat{\theta}_t + \left(\frac{\gamma_t - 1}{\gamma_t}\right)\hat{\theta}_{t-1}.$$

Here we show that $\tilde{\theta}_i$ and $\tilde{\theta}_j$ are uncorrelated for all $i \neq j$, under the assumption that $\tilde{s}_i$ and $\tilde{s}_j$ are uncorrelated ($i \neq j$). Define $\{\eta_k^t\}_{k=0,\ldots,t}$ to be the sequence that satisfies $\hat{S}_t = \sum_{k=0}^t \eta_k^t \tilde{s}_k$ and $\sum_{k=0}^t \eta_k^t = 1$. Note that $\eta_t^t = \gamma_t$, $\eta_{t-1}^t = \gamma_{t-1}(1-\gamma_t)$, and so on. Now,

$$\begin{aligned}
&\operatorname{cov}(\tilde{\theta}_i, \tilde{\theta}_j) \\
&= \operatorname{cov}\left(\frac{1}{\gamma_i}\hat{\theta}_i + \frac{\gamma_i - 1}{\gamma_i}\hat{\theta}_{i-1}, \frac{1}{\gamma_j}\hat{\theta}_j + \frac{\gamma_j - 1}{\gamma_j}\hat{\theta}_{j-1}\right) \\
&= \frac{1}{\gamma_i\gamma_j}\operatorname{cov}(\hat{\theta}_i, \hat{\theta}_j) \\
&\quad + \frac{1}{\gamma_j}\left(1 - \frac{1}{\gamma_i}\right)\operatorname{cov}(\hat{\theta}_{i-1}, \hat{\theta}_j) \\
&\quad + \frac{1}{\gamma_i}\left(1 - \frac{1}{\gamma_j}\right)\operatorname{cov}(\hat{\theta}_i, \hat{\theta}_{j-1}) \\
&\quad + \left(1 - \frac{1}{\gamma_i}\right)\left(1 - \frac{1}{\gamma_j}\right)\operatorname{cov}(\hat{\theta}_{i-1}, \hat{\theta}_{j-1}).
\end{aligned} \tag{C.1}$$

Writing $\hat{\theta}_i = f_0 + f_1 \sum_{k=0}^{i} \eta_k^i \tilde{s}_k$ and recalling that

$$\operatorname{cov}(\tilde{s}_i, \tilde{s}_j) = \begin{cases} 0, & \text{if } i \neq j \\ \sigma_i^2, & \text{if } i = j, \end{cases}$$

it follows that

$$\begin{aligned} \operatorname{cov}(\hat{\theta}_i, \hat{\theta}_j) &= \operatorname{cov}\left(f_1 \sum_{k=0}^{i} \eta_k^i \tilde{s}_k, f_1 \sum_{k=0}^{j} \eta_k^j \tilde{s}_k \right) \\ &= \sum_{k=0}^{i} f_1^2 \eta_k^i \eta_k^j \sigma_i^2, \end{aligned}$$

for $i < j$. Substituting into the four terms of (C.1) yields

$$\begin{aligned} \operatorname{cov}(\tilde{\theta}_i, \tilde{\theta}_j) =& \frac{1}{\gamma_i \gamma_j} \sum_{k=0}^{i} f_1^2 \eta_k^i \eta_k^j \sigma_k^2 \\ &+ \frac{1}{\gamma_j} \left(\frac{\gamma_i - 1}{\gamma_i} \right) \sum_{k=0}^{i-1} f_1^2 \eta_k^{i-1} \eta_k^j \sigma_k^2 \\ &+ \frac{1}{\gamma_i} \left(\frac{\gamma_j - 1}{\gamma_j} \right) \sum_{k=0}^{i} f_1^2 \eta_k^i \eta_k^{j-1} \sigma_k^2 \\ &+ \left(\frac{\gamma_i - 1}{\gamma_i} \right) \left(\frac{\gamma_j - 1}{\gamma_j} \right) \sum_{k=0}^{i-1} f_1^2 \eta_k^{i-1} \eta_k^{j-1} \sigma_k^2. \end{aligned}$$

If we define $a = f_1^2 \eta_i^i \eta_i^{j-1} \sigma_i^2$, $b = \sum_{k=0}^{i-1} f_1^2 \eta_k^{i-1} \eta_k^{j-1} \sigma_k^2$ and note that $\eta_k^j = (1 - \gamma_j) \eta_k^{j-1}$

for all $k < j$, then

$$\begin{aligned}
\operatorname{cov}(\tilde{\theta}_i, \tilde{\theta}_j) &= \frac{1}{\gamma_i \gamma_j}(1-\gamma_j)a + \frac{1}{\gamma_i \gamma_j}(1-\gamma_i)(1-\gamma_j)b \\
&+ \frac{1}{\gamma_j}\left(\frac{\gamma_i - 1}{\gamma_i}\right)(1-\gamma_j)b \\
&+ \frac{1}{\gamma_i}\left(\frac{\gamma_j - 1}{\gamma_j}\right)a + \frac{1}{\gamma_i}\left(\frac{\gamma_j - 1}{\gamma_j}\right)(1-\gamma_i)b \\
&+ \left(\frac{\gamma_i - 1}{\gamma_i}\right)\left(\frac{\gamma_j - 1}{\gamma_j}\right)b \\
&= 0.
\end{aligned}$$

Hence, if $\tilde{s}_i$ and $\tilde{s}_j$ are independent for all $i \neq j$, then $\tilde{\theta}_i$ and $\tilde{\theta}_j$ are uncorrelated $(i \neq j)$, justifying the term "pseudo-independent updates" for $\tilde{\theta}_i$.