Inferring gene flow between populations with statistical methods

Svend V. Nielsen
Bioinformatics Research Centre, Aarhus University
PhD dissertation

July 31, 2018
Contents

1 Introduction 5
  1.1 Acknowledgements ........................................... 5
  1.2 Description of PhD studies ................................... 6
  1.3 Abstract in English ......................................... 8
  1.4 Resumé på Dansk ........................................... 9

2 Data 11

3 CoalHMM 13
  3.1 HMM methods .................................................. 13
  3.2 CoalHMM’s ...................................................... 18
    3.2.1 Variable Migration Model .................................. 21
    3.2.2 Isolation Migration in the Elephant Study ................. 27
    3.2.3 Migration Direction Analysis ............................... 28
    3.2.4 Other CoalHMM’s .......................................... 33
  3.3 Discussion .................................................... 34

4 Particle Filtering 37
  4.1 Simulation of coalescence histories .......................... 37
    4.1.1 ARGweaver .............................................. 39
  4.2 Particle Filtering ............................................. 40
    4.2.1 An importance sampling technique ......................... 40
    4.2.2 Particle Filtering Implementation ........................ 41
    4.2.3 Results .................................................. 43
  4.3 Discussion .................................................... 46

5 AdmixtureBayes and admixturegraph 49
  5.1 Admixture graphs ............................................. 49
  5.2 Summaries ..................................................... 50
    5.2.1 Definitions ............................................... 51
    5.2.2 Equivalences ............................................ 52
    5.2.3 Estimation ............................................... 53
  5.3 Model ........................................................ 55
  5.4 Admixture graph space ....................................... 57
    5.4.1 TreeMix ................................................. 57
    5.4.2 MixMapper ............................................... 57
    5.4.3 AdmixtureBayes .......................................... 59
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.5</td>
<td>Bayesian approach</td>
<td>59</td>
</tr>
<tr>
<td>5.6</td>
<td>AdmixtureBayes method</td>
<td>60</td>
</tr>
<tr>
<td>5.6.1</td>
<td>Prior</td>
<td>60</td>
</tr>
<tr>
<td>5.6.2</td>
<td>MCMC</td>
<td>61</td>
</tr>
<tr>
<td>5.7</td>
<td>AdmixtureBayes Results</td>
<td>63</td>
</tr>
<tr>
<td>5.7.1</td>
<td>Simulations</td>
<td>63</td>
</tr>
<tr>
<td>5.7.2</td>
<td>Analysis on real data</td>
<td>68</td>
</tr>
<tr>
<td>5.8</td>
<td>Discussion</td>
<td>70</td>
</tr>
<tr>
<td>6</td>
<td>ImmediateAncestry</td>
<td>73</td>
</tr>
<tr>
<td>6.1</td>
<td>Chimpanzee dataset</td>
<td>73</td>
</tr>
<tr>
<td>6.2</td>
<td>Model</td>
<td>74</td>
</tr>
<tr>
<td>6.3</td>
<td>Results</td>
<td>76</td>
</tr>
<tr>
<td>6.3.1</td>
<td>Robustness issue</td>
<td>77</td>
</tr>
<tr>
<td>6.4</td>
<td>Discussion</td>
<td>79</td>
</tr>
<tr>
<td></td>
<td>Bibliography</td>
<td>81</td>
</tr>
<tr>
<td>A</td>
<td>Elephant paper</td>
<td>87</td>
</tr>
<tr>
<td>B</td>
<td>Particle Filtering paper</td>
<td>97</td>
</tr>
<tr>
<td>C</td>
<td>admixturegraph paper</td>
<td>111</td>
</tr>
<tr>
<td>D</td>
<td>AdmixtureBayes draft</td>
<td>115</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Acknowledgements

I would like to thank the people at Bioinformatics Research Centre for every thing I have learned and the fun times. A big thank to my supervisor, Thomas Mailund, for all the meetings, feedback and advice. His approach to research is very learning and inspiring. Unfortunately, there has been periods where Thomas could not work. In those periods, I am grateful that Kasper Munch and Asger Hobolth stepped in and supervised. It meant a lot to me. I am happy to have had Maria Simonsen and Dan Søndergaard as office mates throughout most of my PhD studies. Thanks to Dan for making me a much better programmer.

On my stay abroad in Berkeley I visited Rasmus Nielsen. He is a fantastic supervisor and I am grateful that he has been so reachable throughout our project. In the lab I also met Peter Wilton who was a great support helping me both academically and to feel welcome.

During all four years my family has been an amazing support. Thanks to Julie, Anders and Signe for correcting my thesis. In particular I want to thank my mother for countless great advice and helping me so many different things. Tak!
1.2 Description of PhD studies

I started my PhD at Bioinformatics Research Centre at Aarhus University in August 2014 with Thomas Mailund as supervisor. I was accepted on the 4+4 program which meant that I received my Master’s Degree in Statistics on the Qualifying Exam in 2016. I have now finished the last two years.

In the beginning of my PhD, I developed new versions of CoalHMM to model to infer gene flow. Due to either difficulties on simulated or real data, neither of the new CoalHMM’s turned into publications. I did get involved in analyzing an elephant dataset using the Isolation and Migration CoalHMM. It resulted in the publication


which is also attached in Appendix A. I have described the work on the models and the elephant dataset in Chapter 3.

Later I examined the potential for using Particle Filtering instead of HMMs in the crucial computation of CoalHMM or other HMM methods. We modeled less complicated histories not involving gene flow. However, it can be expanded to estimate gene flow. It resulted in the publication


Figure 1.1: Visualization of my schedule during my PhD studies. The area of each project is approximately how much time I have spent on it.
1.2. DESCRIPTION OF PHD STUDIES

which is also attached in Appendix B and described in Chapter 4.

Through discussions with Dan Søndergaard, I got involved in a project where we classified origins of metastases based on gene expressions. It resulted in the publication


I chose not to describe this research in this thesis because it is very different. Therefore, I grayed out the text.

After my qualification exam, Thomas and I started a project to classify hybrid chimpanzees using an HMM as part of a collaboration with Copenhagen Zoo and others. I implemented the program ImmediateAncestry, which is described in Chapter 6.

I went on a stay abroad to work with Rasmus Nielsen in UC Berkeley from February to August 2017. We started a new project of estimating admixture graphs using MCMC. I named the program AdmixtureBayes and it is described extensively in Chapter 5. A draft of a paper is attached in Appendix D.

Prior to my stay, I had implemented parts of the program admixturegraph published as


admixturegraph is described in Chapter 5 and the paper is attached in Appendix C.
1.3 Abstract in English

Gene flow is the transfer of genetic material from one population to another. It is very common and important in the description of the genetic history of a population. Gene flow is hidden in the genome as migrated segments of different genetic material. Inferring the gene flow based on sequenced genomes is challenging. I present my work on estimating gene flow using three different statistical models.

With two sequences from different populations, the model Isolation Migration CoalHMM can determine the amount of gene flow between the populations after their initial split. It takes all possible migrated segments into account using the powerful HMM algorithms. In collaboration I have build new CoalHMM’s incorporating several pairs of sequences to infer direction and variation in the gene flow. I show that estimating direction and variation jointly is too hard. I show that estimating direction is possible with some uncertainty. I apply the methods to a dataset of extinct and extant elephants and show that there is extensive gene flow.

Instead of considering all possible segments with an HMM, I examine the potential of only considering the most likely segments with particle filtering. I present challenges and advantageous choices when implementing a particle filter for this problem.

The second statistical model infers gene flow from a covariance matrix between several populations. Some relations between entries in the covariance matrix can only be explained by gene flow. In collaboration I have developed a method that fits the best phylogeny with gene flow events for an observed covariance matrix. The method uses MCMC. A phylogeny with gene flow events is called an admixture graph and the method is called AdmixtureBayes. I show that AdmixtureBayes has a smaller error than the most popular admixture graph estimators on simulated data. AdmixtureBayes produces a posterior sample of admixture graphs and I demonstrate the possibilities with such a sample on a real dataset of Native American genomes.

The last statistical model infers very recent gene flow by classifying hybrid individuals. The genome of a hybrid individual has big segments of alleles originating from different populations. Using the allele frequencies from those populations, it is possible to infer the segments. In collaboration, I implemented ImmediateAncestry which infers the segments and the most likely hybrid type with an HMM. I show that the classifier has good accuracy on simulated data. The classifier is not robust for a real dataset of chimpanzees, so I discuss reasons and remedies.
1.4 Resumé på Dansk

Genflow er overførslen af DNA imellem populationer. Genflow er meget udbredt og vigtigt for beskrivelsen af en populations genetiske historie. Signaler fra genflow er gent i genomet som segmenter af anderledes DNA. Jeg præsenterer min forskning i at inferere genflow med tre forskellige statistiske modeller.


Chapter 2

Data

Assume that a chromosome of a man starts with the bases AACGTGTT. Assume a corresponding chromosome in a child of the man starts with AATCGTGTT.

<table>
<thead>
<tr>
<th>Man</th>
<th>AA CGTGTT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Child</td>
<td>AATCGTGTT</td>
</tr>
</tbody>
</table>

When the child’s DNA was initially created the base T may have been inserted by random. Say we want to compute the percentages of identical bases between the man and the child. Simply comparing the \( i \)'th base pairs for all \( i \) is not sensible because the DNA of the child is shifted. First, we must apply a method to reverse the shifts. This procedure is called mapping. The mutation process is random and can make arbitrarily complicated disorganization, so any mapping method have some error.

Often, there is prior knowledge available about the locations of interesting regions. A sequencer targeted at the locations is cheaper than sequencing everything. For these targeted sequencers, called SNP arrays, mapping is only a minor issue. However, choosing the interesting regions becomes a new issue. This process is called SNP ascertainment.

Furthermore, most DNA sequencing procedures read both copies in a chromosome pair. As a consequence any reconstructed DNA sequence for a single individual will be an unknown mix of its chromosome pair. The stochastic process of detangling this pattern is called phasing. The phasing methods have a high degree of error, so they are often avoided.

Yet another complication is the uncertainty about the observed bases. Many sequencing machines cannot measure every base with the same accuracy. Therefore, so-called next generation sequencers provide a quality score to each base. Another factor is the number of times each position has been sequenced, that is the depth. Subsetting data based on the depth and quality score is called filtering.

A normal goal for mapping, filtering and phasing is to construct a reliable data matrix,

\[
X = \begin{pmatrix}
X_{11} & X_{12} & \cdots & X_{1N} \\
\vdots & \vdots & \ddots & \vdots \\
X_{M1} & X_{M2} & \cdots & X_{MN}
\end{pmatrix},
\]

(2.1)
where $X_{ij}$ is the *allele* in individual $i$ at locus $j$. In this thesis a locus is always a single position in the DNA, but elsewhere it can refer to a set of positions. On a single position in the DNA, one can read a single base. Often one will not observe more than two unique bases on a position across all sequences individuals. Therefore, bases are often transformed into zeros and ones. An allele can refer to either the base or the transformed base.

The typical workflow of a population genetic analysis starts by creating the data matrix. Using the data matrix as the observed data, inference can be done without adjusting for the uncertainty in the data preparation process (Figure 2.1). There are exceptions to this scheme (e.g. use of genotype likelihoods [46]). However, I will stick with the scheme in Figure 2.1 and instead aim for models that are robust towards wrongly estimated data matrices.
Chapter 3
CoalHMM

In this chapter I present my work on estimating population genetic parameters using the Coalescence Hidden Markov Model (CoalHMM) framework. The genetic parameters include recombination rate, coalescence rates, migration rates and divergence times between populations. In a specific CoalHMM we denote the parameters $\theta_M$ where $M$ is the overall demographical model. The goal is to calculate and maximize the likelihood

$$P(X|M, \theta_M).$$ (3.1)

Many other methods in population genetics maximize probabilities of the form (3.1) because it can answer questions about what happened in the past based on DNA from today. In other methods correlation structure between loci is often ignored or treated as independent blocks (TreeMix [53], G-PhoCS [22], $\partial\partial i$ [15]). CoalHMM belongs to a group, which I will call HMM methods, that models the correlation structure with Hidden Markov Models. Most famous is the PSMC method by Li and Durbin [36], because it produced precise estimates of ancestral human population sizes.

3.1 HMM methods

Imagine we could trace back the ancestry at a single locus of $n$ haploid sequences. They would merge into each other until there were only one lineage left forming a tree structure (Figure 3.1). This is the coalescence history for one locus. The coalescence history at two adjacent loci are often identical. The histories may be different when a meiosis has broken up an ancestral sequences right between the two loci. In this case we say that a recombination occurred between the two loci. The joint coalescence history for all loci is embedded in the slightly larger Ancestral Recombination Graph (ARG) [21]. An ARG is larger because it also contains information about when and in which lineages the recombination events occurred. (There are two common definitions of an ARG; the Marjoram-ARG which keeps track of lineages without ancestry created by the recombination process and the Hudson-ARG which does not track lineages without ancestry [69]. In this thesis I assume the Hudson-ARG).

The observed data for a locus, $j$, is informative about the coalescent history. Felsenstein’s pruning algorithm describes how to calculate the probability of
the observed data given the coalescence history and a substitution model [17].

\[ P(X_j|\text{coalescence history } j), \]  

For two haploid sequences the coalescence history can be written as a single coalescence time, \( T \in \mathbb{R}_+ \). If we use the Jukes-Cantor substitutions model, we obtain

\[ P(X_{1j} = X_{2j}|T) = \frac{1}{4} + \frac{3}{4}e^{-4/3\mu T} \]
\[ P(X_{1j} \neq X_{2j}|T) = \frac{3}{4} - \frac{3}{4}e^{-4/3\mu T} \]  

(3.3)

where \( \mu \) is the factor that converts \( T \) into number of expected substitutions. It requires alleles from many loci to narrowly estimate \( T \).

A general distribution of coalescence histories was described by Hudson [28] and implemented in ms [29]. The program ms simultaneously simulates all coalescence histories from the present day and backwards in time. When this process simulates a recombination between two loci on a sequence, it splits up the sequence in two at the locus. Henceforth, the process follows one more sequence back in time (Figure 3.2). The procedure stops when all sequences have merged. The product is the ARG. The ms program can simulate ARGs from various population models by adjusting the rates of coalescence and recombination accordingly.
Let $C$ be the sequence of coalescence histories. The effectivity, with which ms simulates $C$, makes the following calculation seem feasible:

$$P(X|M, \theta_M) = \int P(X|C, M, \theta_M)P(C|\theta_M, M) \, d(C) \quad (3.4)$$

$$\approx \frac{1}{R} \sum_{r=1}^{R} P(X|C_r, \theta_M, M) \quad (3.5)$$

for $C_1, \ldots, C_R$ simulated from $P(C|\theta_M, M)$. The integrand

$$P(X|C, \theta_M, M) \quad (3.6)$$

can be calculated as the product of the locus-wise probabilities in (3.2). Unfortunately, the convergence is extremely slow because the data, $X$, is very unlikely under most simulated coalescence history sequences. Only a small fraction of the histories would contribute significantly to the sum in (3.5). It is important to ‘remember’ those histories. HMM tools have been used both to simulate likely coalescence histories, but also to calculate the integral in (3.4) by summing out all possible histories.

In HMM methods the data matrix, $X = \{X_{ij}\}$, are observed states and the coalescence histories, $\{C_j\}$ are the hidden states. The factor graph of this
model is
\[ C_1 \rightarrow C_2 \rightarrow \cdots \rightarrow C_N \]
\[
\begin{pmatrix}
X_{11} \\
\vdots \\
X_{n1}
\end{pmatrix} \quad \begin{pmatrix}
X_{12} \\
\vdots \\
X_{n2}
\end{pmatrix} \quad \cdots \quad \begin{pmatrix}
X_{1N} \\
\vdots \\
X_{nN}
\end{pmatrix}
\]
In other words, the observed data is conditionally independent on the coalescence histories and the coalescence histories constitute a Markov chain. Conditional independence between alleles is a common assumption, which also reflects how ms and other programs simulate observed data. The coalescence histories are not Markovian [69] and assuming it has been shown to cause a bias [40]. Furthermore, to apply the very efficient HMM algorithms, the number of hidden and observed states should be finite. Fortunately, the data matrix is naturally finite. The coalescence histories are continuous by construction. Therefore, the HMM methods discretize the coalescence histories by dividing the time axis into a finite number of intervals and representing the coalescence times with the intervals. Replacing the coalescence histories with these discretized histories, perturbs the conditional independence amongst the observed data. The accompanying bias is, according to my experience, minuscule. Furthermore, the Felsenstein pruning algorithm can not be applied directly on the discretized coalescence histories. All HMM methods use approximations and the choice of approximation is not always insignificant [47].

Denote the space of possible hidden states \( T = \{T_1, \ldots, T_K\} \) and the space of possible observed states \( O = \{O_1, \ldots, O_L\} \). Most HMM methods explicitly calculates the transition probabilities
\[ T_{a,b} = P(C_i = a | C_{i-1} = b), \quad a, b \in T \quad (3.8) \]
and emission probabilities
\[ E_{o,a} = P\left( \begin{pmatrix}
X_{1i} \\
\vdots \\
X_{ni}
\end{pmatrix} = o | C_i = a \right) \quad a \in T, o \in O \quad . \quad (3.9) \]
Combined with the initial probability, \( P(C_1 = a) \), (3.8) and (3.9) fully specify the distribution of an HMM.

I divide the HMM methods into 3 categories based on how they obtain parameter estimates.

1. Some methods evaluate the likelihood in (3.4) with the forward algorithm. The likelihood is treated as a black-box function and optimized with respect to the parameter vector \( \theta_M \) using any optimization method. I will call this forward optimization.

2. Other methods maximize the likelihood in (3.4) with the Baum-Welch algorithm. This produces estimates of the transition and emission probabilities. The methods subsequently transform the transition and emission matrices into an estimate of \( \hat{\theta}_M \). I will call this the Baum-Welch optimization.
3. Lastly, there is at least one HMM method (ARGweaver [57]) that simulates hidden states conditioned on the observed states and parameters $\theta_0$. Denote such a hidden states sequence $Z_r = (Z_{r1}, \ldots, Z_{rN})$. Applying importance sampling, the likelihood can be calculated with

$$P(X|\theta) \propto \frac{1}{R} \sum_{r=1}^{R} \frac{P(Z_r|\theta)}{P(Z_r|\theta_0)}.$$  \hspace{1cm} (3.10)$$

However, it is more common to do inference using the simulated hidden states as observed values [49], [57]. Thus, maximizing

$$\tilde{P}(Z_1|\theta) \text{ or } \frac{1}{R} \sum_{r=1}^{R} \tilde{P}(Z_r|\theta),$$  \hspace{1cm} (3.12)$$

where $\tilde{P}$ could be an alternative probability distribution. I will call this approach simulation optimization.

If we calculate the density of the joint distribution of the coalescent histories of two adjacent loci, we can also calculate the transition probabilities in (3.8). Ideally one would use the joint distribution derived from the ms distribution (Figure 3.2). Simonsen and Churchill described that distribution [62], so I will name it SC. Initially, a simplification was introduced as SMC [44]. It allows maximum one recombination between each pair of loci. Furthermore, the two sequences created by a recombination are not allowed to coalesce with each other (Figure 3.3). The SMC is faster than the SC distribution because it has a smaller state space. However, its approximations make it less precise [43]. Therefore, Marjoram and Wall created SMC' [43], which does allow coalescence between the two sequences created from the same recombination event. Because SMC' was formulated for a continuous sequence of loci, there now exists at least three interpretations of SMC' for two loci (Figure 3.4). I name them

- SMC'a: only one recombination event allowed between any adjacent pair of loci [61].
- SMC'b: only one of the sequences can be unlinked at a time. It is a good approximation to the SC model [27].
- SMC'c: any number of recombinations allowed between any adjacent pair of loci. If there are more than one recombination point between the two loci, a sequence can be split up into more than two small segments. In contrast to the SC model, these small segments can only coalesce with adjacent segments. If there is only one recombination point, the SMC'c is equivalent to the SC model [68].

This chapter only considers the three models SMC, SMC'a and SC.

Summing out all coalescence histories from (3.4) is computationally expensive, especially when analysing more than two sequences. Therefore, some HMM methods only integrate out a summary of the coalescence histories. Like a discretized coalescence history, a summary will not contain all the information necessary to calculate the likelihood using Felsenstein’s pruning algorithm. For
discretized coalescence histories, the likelihood is often calculated with approximations. For more summarized coalescence histories, the observed data is often summarized as well. Most famous is the MSMC model [61] where the hidden states are the times to the first coalescence event in the coalescence histories and the observed states are dummy variables for presence of singletons.

### 3.2 CoalHMM’s

At Bioinformatics Research Centre at Aarhus University, we have implemented several population models in the CoalHMM framework (Table 3.2). From 2014 all new CoalHMM’s used the two tools:

1. The library ZipHMM [59]. It calculates the likelihood given transition and emission probabilities using the forward algorithm. ZipHMM reduced the evaluation time of a single likelihood value from minutes to seconds. Without it, forward optimization would be at least an order of magnitude more time consuming.

2. Augmented derivation of the transition probabilities. Initially, the transition probabilities in the HMM methods were derived through a set of complicated equations handling different cases often involving approximations (PSMC [36], MSMC [61], CoalHMM 2007 [26]). Several practical developments from Mailund et al. ([40], [42] and [41]) introduced methods for calculating them using only basic information about the population model.

The implementation time of a CoalHMM is relatively short due to these tools. However, the statistical accuracy between the models can vary a lot, so they have to be verified separately.

All recent CoalHMM’s examine gene flow and migration. Often gene flow is estimated treating SNPs independently (Chapter 5). However, there is a lot of information about gene flow events through the linkage pattern. Roll-off [51] and ALDER [39] are methods that estimate the time of instantaneous gene
3.2. COALHMM’S

Figure 3.4: The 3 alternative SMC’ models for two adjacent loci and two sequences illustrated as in Figure 3.3. In the SMC’a model, a state is blue to indicate that it is different from the starting state.
flow using the linkage between migrated alleles. CoalHMM’s use both SNP information and linkage patterns, which will potentially give it more statistical power.

**Augmented derivation of transition probabilities**

Looking backwards in time, the joint coalescence history at two adjacent loci can be described as a continuous time Markov Chain, \( (W(t))_{t \geq 0} \), where \( W(t) \) is the state of \( n \) sequences at time \( t \). For two sequences within a single unstructured population with constant population size, Figure 3.4(SMC’c) shows their transition rates. Let \( Q \) denote the matrix of these transition rates. The matrix exponential \( e^{Qt} \) gives the probabilities

\[
P(W(t) = b \mid W(0) = a), \quad a, b \in \mathcal{S}, t > 0
\]

(3.13) where \( \mathcal{S} \) is the set of possible states. The transition probabilities in the HMM are sums of probabilities of the kind (3.13). For example, to calculate the probability that two sequences coalesced in time interval \( [t_1, t_2] \) at one locus and in time interval \( [t_3, t_4] \) at the next locus, one sums over the paths of \( W(t) \) that fulfills this. Let \( S_0, S_1, S_2, S_B \subseteq \mathcal{S} \) be the disjoint sets of states, where the sequences in none, the first, the second, and both loci have coalesced, respectively. Let \( \pi \) be the initial probabilities of \( W(t) \) and assume \( t_3 \geq t_2 \). The transition probability is

\[
P(C_1 = [t_3, t_4] \mid C_{1-1} = [t_1, t_2]) = \\
\sum_{\text{start} \in \mathcal{S}} \sum_{a \in S_0} \sum_{b \in S_1} \sum_{c \in S_1} \sum_{d \in S_B} \left( P(W(t_1) = a \mid W(0) = \text{start}) \pi_{\text{start}} \right. \\
\cdot P(W(t_2) = b \mid W(t_1) = a) \\
\cdot P(W(t_3) = c \mid W(t_2) = b) \\
\cdot P(W(t_4) = d \mid W(t_3) = c).
\]

(3.14)

The method can, in theory, be generalized to any population model for which is given

1. The transition rate matrix, \( Q \).
2. The initial probability, \( \pi \).
3. A plan for how to calculate the transition probabilities from the probabilities in (3.13). For all CoalHMM’s of two sequences, the formula (3.14) holds, but the sets \( S_0, S_1, S_2, S_B, \mathcal{S} \) need to be specified separately.

Many CoalHMM’s model the history as distinct epochs. The CoalHMM Isolation Model \([40]\) consists of two epochs with their own transition rate matrices and state spaces; one in which the two populations are isolated and one where they have merged. A procedure for how to combine the epochs was presented in Mailund et al. \([41]\). The main idea is to sum across the transitional states. If the two CTMCs are denoted \( (W^1(t))_{t \geq 0} \) and \( (W^2(t))_{t \geq \tau} \) with the state spaces \( S^1 \) and \( S^2 \) and transition function, \( f \), the probability in (3.13) is

\[
P(W^2(t) = b \mid W^1(0) = a) = \sum_{c \in S^1} \left( P(W^1(\tau) = c \mid W^1(0) = a) \\
\cdot P(W^2(t) = b \mid W^2(\tau) = f(c)) \right)
\]
3.2. COALHMM’S

for \( t > \tau \). For all cases and details, see [41].

For complicated population models the transition rate matrix is potentially very big making it cumbersome and error-prone to specify it. Therefore Mailund et. al [41] created a new procedure for generating the state space, \( S \) and \( Q \) automatically from a set of possible events.

Parameters

The unit of time in the CoalHMM’s is expected number of substitutions because it makes calculations simpler. For example, the emission probabilities in (3.3) simplifies to \( \mu = 1 \). The parameters in the models are also expressed in terms of expected number of substitutions. To convert to more interpretable units (e.g. effective population size and split time in years), one needs the constants

1. Years per number of expected substitutions.
2. Generations per number of expected substitutions.

Choosing the constants is left to the user of CoalHMM.

3.2.1 Variable Migration Model

In the Variable Migration Model two populations have continuous gene flow between them at varying rates from the present day and back in time (Figure 3.5(VMM)). Time is divided into different epochs, where each epoch contributes with four parameters; two population sizes and a migration rate for both directions. Including recombination rate the number of parameters in the model is

\[
4 \times (\text{number of epochs}) + 1.
\]

In this thesis I present results obtained with four epochs and hence 17 parameters.

To gain enough statistical power to estimate all the parameters, we combine two sequences from each population in a composite likelihood. Two components are likelihoods of pairs taken from the same population. The third and last component is the likelihood of an inter-population pair.

Initial results of the model indicated dependencies and symmetries between the parameters. For example, an epoch with an underestimated migration rate was often followed by an epoch with an overestimated migration rate. To overcome this, we implemented a Markov Chain Monte Carlo (MCMC) sampler. A sample of parameters could reveal several modes and produce a better description of the uncertainty area.

We chose the prior relatively uniformly; all parameters have exponential distributions with large means.

The proposal distribution of the MCMC is a random walk on all parameters simultaneously.

\[
\theta_{\text{new}}^M = \theta_{\text{old}}^M + \varepsilon, \quad \varepsilon \sim N(0, \Sigma s), \Sigma = I_{17} \quad (3.15)
\]

We adjust the step size, \( s \), with an adaptive scheme that increases or decreases the step size depending on the acceptance probability [4]. I implemented the MCMC as a Metropolis Coupled MCMC (MC³) running several chains in parallel [19]. In MC³ there is one target chain and several chains with increasing
‘temperatures’. Raising the temperature, improves mixing but changes the stationary distribution of the chain. The chains switch their states probabilistically such that the stationary distributions of the chains are preserved. The choice of temperatures matter for the convergence speed [5]. I implemented an adaptive scheme to adjust the temperatures to achieve an optimal distance between chains [5].

Despite many iterations of diagnostics and improvements, the MCMC produces biased results. On simulated data, the MCMC fails to recover the parameters, and there are variations between runs (Figure 3.6 and Figure 3.7). The alternation between overestimation and underestimation is not solved by the MCMC as seen from the overconfidence of the marginal posterior densities. Other, smaller CoalHMM models with migration periods do not have the same problems which suggests that the Variable Migration Model is too complex. The complexity could make it less robust.

**My contribution:** The model was initially coded by Thomas Mailund and I wrote the MCMC optimization code. Diagnostics and remedies of the problems were executed by me, and planned in collaboration with Thomas Mailund. I wrote the remedial changes to the model-specific code.
Table 3.1: Table of different HMM methods. The observed states of PSMC and CoalHMM can be reduced to the heterozygosity pattern. It is a sequence of zeros and ones corresponding to the two genome sequences being identical or different. In its HMM, MSMC only integrates out the time to most recent coalescence using singletons. However, MSMC exploits information about total branch length along the sequence from an earlier analysis (using more than singletons) and includes it as known information in the MSMC.

<table>
<thead>
<tr>
<th>Method</th>
<th>Year</th>
<th>Sequences</th>
<th>Hidden states (Discretization intervals)</th>
<th>Observed states</th>
<th>Optimization</th>
<th>SMC model</th>
<th>Population model</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSMC [36]</td>
<td>2011</td>
<td>2</td>
<td>Coalescence history (64)</td>
<td>Heterozygosity pattern</td>
<td>Baum-Welch</td>
<td>SMC</td>
<td>Piecewise constant population size</td>
</tr>
<tr>
<td>PSMC’ [61]</td>
<td>2011</td>
<td>2</td>
<td>Coalescence history</td>
<td>Heterozygosity pattern</td>
<td>Baum-Welch</td>
<td>SMC’a</td>
<td>Piecewise constant population size</td>
</tr>
<tr>
<td>MSMC [61]</td>
<td>2014</td>
<td>≥ 2 (in practice ≤ 8)</td>
<td>Time to most recent coalescence (40)</td>
<td>Singletons*</td>
<td>Baum-Welch</td>
<td>SMC’a</td>
<td>Piecewise constant population size</td>
</tr>
<tr>
<td>ARGweaver [57]</td>
<td>2014</td>
<td>≥ 2 (in practice ≤≈54)</td>
<td>Coalescence history (20)</td>
<td>Full data</td>
<td>Simulation</td>
<td>SMC’a</td>
<td>Constant populations size, 1 population</td>
</tr>
<tr>
<td>coal-HMM [26]</td>
<td>2007</td>
<td>4</td>
<td>Order of coalescences</td>
<td>ILS pattern</td>
<td>Baum-Welch</td>
<td></td>
<td>4 populations, 1 outgroup, clean splits.</td>
</tr>
<tr>
<td>coalHMM Isolation Model [40]</td>
<td>2011</td>
<td>2</td>
<td>Coalescence history (10)</td>
<td>Heterozygosity pattern</td>
<td>forward</td>
<td>SC</td>
<td>2 populations, clean split (Figure 3.8(I))</td>
</tr>
<tr>
<td>coalHMM Isolation Migration Model [42]</td>
<td>2012</td>
<td>2</td>
<td>Coalescence history (10)</td>
<td>Heterozygosity pattern</td>
<td>forward</td>
<td>SC</td>
<td>2 populations, migration period after split (Figure 3.8(IM))</td>
</tr>
<tr>
<td>Method</td>
<td>Year</td>
<td>Status</td>
<td>Sequences</td>
<td>Optimization algorithm (Parameters)</td>
<td>Performance</td>
<td>Population model</td>
<td></td>
</tr>
<tr>
<td>--------------------------------</td>
<td>------</td>
<td>----------</td>
<td>------------------------------------</td>
<td>-------------------------------------</td>
<td>-------------------------------------------------</td>
<td>--------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Variable Migration</td>
<td>2014</td>
<td>Discarded</td>
<td>4 in a composite likelihood $2 \times 2 \times 2$</td>
<td>MCMC (17)</td>
<td>Convergence issues and bias</td>
<td>2 populations with gene flow between them (Figure 3.5 (VMM))</td>
<td></td>
</tr>
<tr>
<td>Isolation Migration Epochs [9]</td>
<td>2015</td>
<td>Published</td>
<td>4 in composite likelihood $2 \times 2 \times 2$</td>
<td>Particle-Swarm (7,11,14)</td>
<td>Convergence, but some parameters still uncertain</td>
<td>An Isolation-Migration model where the migration period is divided into epochs (Figure 3.15 (E1), (E2), and (E3))</td>
<td></td>
</tr>
<tr>
<td>Isolation Migration Direction</td>
<td>2016</td>
<td>On hold</td>
<td>4 in a composite likelihood $2 \times 2 \times 2 \times 2 \times 2 \times 2$</td>
<td>Particle-Swarm (5,6)</td>
<td>Convergence, but low statistical power on real data</td>
<td>4 different Isolation-Migration models with different migration directions (Figure 3.11 (A), (B), (E), and (T))</td>
<td></td>
</tr>
<tr>
<td>3 population Admixture [8]</td>
<td>2014</td>
<td>In print</td>
<td>2-6 in a composite likelihood</td>
<td>Particle-Swarm (7)</td>
<td>Convergence, but not robust</td>
<td>3 populations joined with 1 admixture events and clean splits (Figure 3.15 (ADM))</td>
<td></td>
</tr>
<tr>
<td>3 population ILS</td>
<td>2015</td>
<td>Discarded</td>
<td>3</td>
<td>Nelder-Mead (4)</td>
<td>Too slow</td>
<td>3 populations with clean splits (Figure 3.15 (ILS))</td>
<td></td>
</tr>
<tr>
<td>Variable Admixture</td>
<td>2015</td>
<td>Discarded</td>
<td>4 in a composite likelihood $2 \times 2 \times 2$</td>
<td>MCMC (13)</td>
<td>Convergence issues and bias</td>
<td>2 populations with admixtures between them (Figure 3.5 (VAM))</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.2: Table of different CoalHMM methods developed since 2012.
Figure 3.6: We simulated 5 datasets with ms and analyzed each dataset five times. A curve estimates the scaled posterior density of a parameter. The rows represent datasets, the colors represent the repetition of the analysis and the columns represent the different parameters. The header theta_{X, Y} refers to the unscaled population size for the X’th population in the Y’th epoch. The plot is continued in Figure 3.7 on the next page.
CHAPTER 3. COALHMM

Figure 3.7: Continuation of Figure 3.6. The header mig_{XZ}_{Y} refers to the migration rate from population X to population Z in epoch Y. The parameter $\rho$ is the recombination rate.
3.2. COALHMM’S

3.2.2 Isolation Migration in the Elephant Study

As part of a consortium organized by Palkopoulou et al., we had access to 14 elephant genomes. Two genomes come from the extinct mastodon, which is a known outgroup of the other elephant species. One genome comes from the Columbian mammoth and two from the Eurasian mammoth. The rest of the genomes are present-day elephant samples from Asia and Africa. We were particularly interested in the speciation events of their shared history; when did the speciation events occur? Was it clean splits or was the initial split followed by a period of gene flow? These questions can be answered by the Isolation and Isolation Migration CoalHMM’s [40], [42].

The population model for the Isolation Model includes two populations, initially, without gene flow looking back in time. At time $\tau$ the populations merge into a homogenous constant-sized population (Figure 3.8(I)). The parameters are the recombination rate, the speciation time, $\tau_{\text{split}}$, and the population size in the ancestral population. Nelder-Mead optimization recovers the population size and speciation time on data simulated with ms. The estimate for the recombination rate is biased downwards because of the Markov assumption [42].

The Isolation Migration Model is also initialized with two isolated populations looking back in time. Before (after looking forward in time) the split there is a migration period with constant gene flow between the two populations (Figure 3.8(IM)). There are five parameters in the model. 1) the recombination rate. 2) a single population size for both the ancestral population and for the populations during the migration period. 3) the split time, $\tau_1$. 4) the time at which the migration period begins, $\tau_2$. 5) The migration rate, $M$. A simulation study found that all parameters except the recombination rate are recovered [42].

We select one of the two models with the Akaike Information Criterion (AIC). The AIC is (-2 times) an estimate of the relative Kullback-Leibner distance between the true model and the fitted model [2]. The relative Kullback-Leibner can be estimated with the maximum log likelihood value of the model. Unfortunately, that estimator is biased but, luckily, the bias can be estimated with the number of parameters in the model. The AIC for a model with likelihood, $L$ and maximum likelihood parameters, $\hat{\theta} \in \mathbb{R}^k$ is

$$-2 \cdot \left( \log L(\hat{\theta}) - k \right).$$
Given several models, the model with the lowest AIC value is estimated to have the shortest distance to the true model.

The AIC value of a model depends on the discretization process of the coalescence times. In simulations, dividing time into many intervals produces a better fit than a few intervals (Figure 3.9). It influenced AIC inference. Therefore, we increased the number of time intervals to 140 for the Isolation Model and 150 in the Isolation Migration Model (where the 10 extra states constitutes the migration period). This, however, increased model fitting time from \(\sim 5\) to \(\sim 250\) CPU hours.

For every pair of genomes we fitted the models and selected the model with the lowest AIC value. On the autosomes, there was overwhelming support for the Isolation Migration Model (Figure 3.10). All splits between the species indicated the existence of a migration period following the initial population split. The migration period between Asian Elephants and Mammoths was short and weak, yet it was found. Full results are attached in the paper in Appendix A. We calculated bootstrap intervals of all parameters.

In conclusion, we have found widespread migration following the initial splits amongst all elephant species. We used the CoalHMM Isolation and Isolation Migration Models that previously have been used to do meaningful inference.

My contribution: I ran all analyses and the diagnostics and tests to determine the number of hidden states. The planning was done in collaboration with Thomas Mailund, who had written the code behind the model and the optimization. I also made theoretical improvements in the model code.

### 3.2.3 Migration Direction Analysis

In the Elephant Study, we estimated many splits followed by a migration period. The migration was modeled as a constant, symmetric flow between the
Figure 3.10: Sketch of the CoalHMM results on the elephant data. I constructed the phylogeny heuristically based on the pairwise analyses. The red area below a speciation event represents a migration period and the stronger color, the higher the migration rate. The letter codes are: AF - African Forest Elephant, IX - Mastodon, BC - African Savannah Elephant, N - Columbian Mammoth, DE - Asian Elephant, PQ - Mammoth. The Columbian Mammoth is added twice because it exhibits strong migration with both African subspecies.
two populations. However, the migration could have occurred only in one direction or been asymmetric.

We implemented four new CoalHMM’s that model the migration period as either symmetric(E), asymmetric(T), going only in one direction(A) and going only in the other direction(B) (Figure 3.11). Data consists of 2 sequences from each population and the model uses a composite likelihood of all possible pairs between 4 sequences.

The CoalHMM produces a likelihood function which we maximize with Particle-Swarm Optimization, which has worked better than Nelder-Mead on other CoalHMM likelihoods [9]. The big composite likelihood and the costly Particle-Swarm puts the computation cost of one maximization at $\sim$1000 (parallelizable) CPU hours. On data simulated with ms, the optimized parameters recover the truth well (Figure 3.12).

We use AIC values to determine which model describes the data best. The AIC values still depend on the discretization process as for the Isolation and Isolation Migration Model. To mitigate the discretization effect we fixed the discretization time intervals in all analyses. In simulations, the accuracy was good for all models except for data simulated with the (E)-model which was often classified as the T-model (Figure 3.13).

The most interesting speciations in the elephant data is the Forest-Savannah split, the African-Asian split and the Columbian-African split. For all those speciations, our previous analyses inferred that their initial splits were followed by migration periods. Unfortunately, the Migration Direction analysis was not possible for the Columbian Mammoth because we only have one Columbian genome. The African-Asian split preferred the B-model yet the fit was troublesome; The length of the migration period went to 0 and the migration rate diverged towards infinity. The Savannah-Forest split converged to reasonable values inside the parameter area. It preferred the B-model, which means that
3.2. **COALHMM’S**

![Figure 3.12](image1.png)

**Figure 3.12:** On ms simulated data from the (A), (B), (E), and (T) models, we fitted the four models to the data. This plot shows the estimated split times, $\tau_1$ and $\tau_2$.

![Figure 3.13](image2.png)

**Figure 3.13:** On ms simulated data from the (A), (B), (E), and (T) models, we fitted the four models to the data. We classified each dataset into the model which produces the fit with the lowest AIC value. The datasets were 1 gb linked bases.
CHAPTER 3. COALHMM

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Lower - Upper</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta$</td>
<td>0.00248</td>
<td>[0.00244 - 0.00254]</td>
</tr>
<tr>
<td>$m_{12}$</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$m_{21}$</td>
<td>569</td>
<td>[487 - 641]</td>
</tr>
<tr>
<td>$\tau_1$</td>
<td>0.0044</td>
<td>[0.0037 - 0.0049]</td>
</tr>
<tr>
<td>$\tau_2$</td>
<td>0.00238</td>
<td>[0.00235-0.00243]</td>
</tr>
</tbody>
</table>

Table 3.3: Table of the B-model estimates and bootstrapped parameters of the Forest-Savannah split. The $\theta$ parameter is proportional to the population size and $m_{21}$ is the number of migration events per base per expected substitution. $\tau_i$ is measured in number of expected substitutions.

Figure 3.14: Bootstrapped preferred models of Migration Direction Models on the Forest-Savannah split.

hybrids of Savannah and Forest Elephants were absorbed into the Forest population (Table 3.3).

To determine the certainty of the choice of the B-model we bootstrapped the results. Unfortunately, there was no clear signal in the bootstrapped results (Figure 3.14). Therefore, we decided to exclude the analysis from the Elephant Study.

In conclusion, the Direction Migration is a computation-heavy model to detect direction of migration following a population split. On ms-simulated data the model estimates the parameters well and it has some power to distinguish the different directions of migration. On real data, the models can diverge showing that the model lacks robustness. We have also observed that the power to distinguish the models decreases on real data. This could be due to noise in the data or optimization problems.

My contribution: I planned the model, its diagnostics and remedies in collaboration with Thomas Mailund. All model-specific code was written by me. All analyses were run by me.
3.2. **COALHMM’S**

3.2.4 **Other CoalHMM’s**

Several other CoalHMM’s have been implemented with different population models.

1. In the Isolation Migration Epochs [9] Model, the migration period in the Isolation Migration Model is divided into different epochs with each their own migration rate and population size (Figure 3.15 (E1), (E2), and (E3)). The model estimates many parameters correctly. However, for (E2) and (E3) the older population sizes and migration rates could not be estimated accurately. Particle-Swarm makes significantly better estimates than Nelder-Mead.

2. In the 3 population admixture model [8], the population model is three populations where one population is a mixture of the ancestral populations of the other populations (Figure 3.15(ADM)). The likelihood is a composite likelihood of possible pairs of up to 6 sequences distributed...
3. In the 3 population ILS model, the population model consists of three populations that merge forming in a binary tree (Figure 3.15(ILS)). The hidden states in the HMM are the discretized coalescence histories of 3 sequences and not 2 sequences as in the other models. We only expect small improvements compared to the ILS CoalHMM from 2007 [26]. However, extending the model to more complicated admixture models could give it more power. Unfortunately, the evaluation time of a single likelihood amounted to 5 minutes as opposed to 10 seconds in the conventional models. Although not impossible, we decided to spend time on more computationally tractible models.

4. The Variable Admixture model is a modification of the Variable Migration model where the migration periods are replaced by admixture events (Figure 3.5(VAM)). It was motivated by the success of the 3 populations admixture model. Unfortunately the bias and convergence problem persisted.

My Contribution: I took part in the implementation of the ILS model and ran the tests. I implemented the Variable Admixture model and ran its simulations.

3.3 Discussion

The CoalHMM framework is a flexible framework for estimating parameters a big suite of different population models. We can take several sequences into account by constructing a likelihood for each pair of sequences and combine them in a composite likelihood. The models consider both the similarity between species and the distribution of similarity. This can potentially give them a lot of statistical power. We have implemented, optimized and tested several applications of this framework.

Unfortunately, we have not achieved accurate estimates for more than around 8 parameters at a time. My results on the Variable Migration and Migration Direction model support this upper limit. Because the CoalHMM’s model the data closely, lack of robustness could be a problem that gets exacerbated in the big CoalHMM’s. Lack of statistical power to estimate all parameters is another likely problem. The PSMC and MSMC successfully estimates many more parameters. However, these parameters are population sizes and the recombination rate and not migration rates.

Optimization of CoalHMM’s is surprisingly challenging given the number of parameters. A linear profile in a likelihood surface reveals that ridges do occur (Figure 3.16). In addition, several optimization procedures have diverged and we have observed dependencies between parameters. As I have shown, these issues can be partly handled with Particle-Swarm and Markov Chain Monte Carlo, yet they are computationally costly.
Figure 3.16: On an elephant dataset, two independent runs of the Isolation Migration model found two different maxima. Here I have plotted log likelihood values (with the formula \( \log L_T(x/190 \cdot \theta_1 + (1 - x/190)\theta_2) \)). It shows that there are at least 3 ‘ridges’ in this likelihood surface.

Although it is technically possible to increase the number of sequences in the CoalHMM, it is very computationally costly. This limits the applications of multi-sequence CoalHMM’s to simple models that does not require many optimization iterations. Alternatively, we could integrate out summaries of the coalescence histories instead of the full coalescence histories. The CoalHMM CTMC probabilities in (3.13) can be used to calculate the pair-loci distribution of any summary of the (discretized) coalescence histories. When the hidden states are summarized, it is more convenient to compute the likelihood of the observed states by summarizing the data. The resulting CoalHMM would have less statistical power than a CoalHMM build on the full coalescence history, but it could be faster and more robust. ARGweaver is another candidate for increasing the number sequences. Unfortunately, there is no good way of maximizing the likelihood function in (3.4) using ARGweaver yet.

I have shown that the Elephantoid speciation events are almost universally followed by a migration period. The Isolation and Isolation Migration CoalHMM’s are valuable and tested tools to determine the speciation types and times. They analyze 2 sequences and only have three and five parameters, respectively.
Chapter 4

Particle Filtering

Particle Filtering is a sampling technique [12] that we explored in its ability to compute the integral in (3.4)

\[ P(X|\theta) = \int P(X|C, \theta)P(C|\theta) \, d(C) \]  

(4.1)

where \( X \) is the data matrix, \( \theta \) is the parameters, and \( C \) is the coalescence histories for the data matrix (I have omitted the \( M \) notation for simplicity). In Chapter 3 I described how HMM methods compute the integral by discretizing the coalescence histories which turns the integral in (4.1) into a finite sum. By assuming the HMM probability structure, efficient algorithms can calculate the sum fast. In this chapter, we calculate the integral using simulation methods without discretizing.

4.1 Simulation of coalescence histories

As mentioned in Chapter 3, the crude Monte Carlo estimator converges extremely slowly.

\[ P(X|\theta) \approx \frac{1}{R} \sum_{r=1}^{R} P(X|C_r, \theta), \quad C_r \sim P(C|\theta), r = 1, \ldots, R. \]

(4.2)

It is slow because \( P(X|C_r, \theta) \approx 0 \) for almost all simulated \( C_r \)'s, yet a tiny fraction is significantly higher than rest. Even if the coalescence histories are simulated with the ‘true’ \( \theta \), it would take extremely many samples to converge. That is unfortunate because there exists many methods to simulate from the prior (ms [29], fastsimcoal [15]).

Assuming that the sequence of coalescence histories, \( C \), is constant, just one coalescence history describes the entire data matrix \( X \). Consequently, (4.1) simplifies to Felsenstein’s Equation [18]. Crude Monte Carlo sampling is still not feasible in this case, but there has been a lot of progress applying Markov Chain Monte Carlo (MCMC) and importance sampling [65] [25]. I will denote these methods the traditional coalescent sampling methods. The MCMC methods apply MCMC to the posterior distribution

\[ P(C|X, \theta_0) \]

(4.3)
for a 'driver' value of $\theta_0$ to obtain a sample of coalescence histories. The Importance Sampling methods produce a sample by simulating a proposal sample from a distribution $q(C|X, \theta_0)$ and subsequently reweighing to obtain a sample from (4.3) (See Subsection 4.2.1). Whether importance sampling or MCMC is used, any likelihood function value can be computed by applying importance sampling again

$$P(X|\theta) \approx \frac{1}{R} \sum_{r=1}^{R} P(C_r|\theta) \cdot P(C_r|\theta_0). \quad (4.4)$$

In practice, the approximation should only be used for $\theta$ values close to the driver value $\theta_0$ [33]. Nevertheless, modeling coalescence histories as constant across the data matrix is a large assumption. If there are recombinations making the actual coalescence histories heterogeneous across the data matrix, most maximum likelihood results will be biased [60].

The traditional coalescent sampling methods have been extended to include blocks of independent data matrices [23] and recombinations within blocks [16]. Seemingly, this could be used to model a heterogenous collection of coalescence histories. Indeed, every single locus can be put in its own block. However, that would mean that every loci is independent and a lot of information would be lost from the data. Alternatively, all loci could be put in one big block while allowing any number of recombinations within. Unfortunately, that is only feasible for the traditional coalescent sampling methods if the data matrix is in the order of thousands of base pairs [16]. The described methods have had most success on small datasets with dozens of unlinked blocks with hundreds of basepairs. They can, however, analyze dozens of sequences at once.

As the biological datasets became bigger and the HMM methods (CoalHMM [26], PSMC [36], MSMC [61] and more) appeared, the original sampling methods lost popularity [57]. Instead of considering blocks of loci with identical coalescence histories, the HMM methods assume that the coalescence histories constitute a markov chain. In this way, information from the information from the linkage pattern is still exploited. In addition, computational costs are relatively low because it is only necessary to consider two loci at a time. This has allowed powerful, whole genome analyses as described in Chapter 3. However, there are some downsides to the HMM methods

(i) The Markov assumption introduces biases. As mentioned in Chapter 3, estimates of the recombination rate are biased, yet many other parameters are still accurately estimated.

(ii) The HMM requires that coalescence histories are discretized. I described the discretization process in Chapter 3. In theory, the process can introduce biases because it breaks down the dependency structure of the state space model. I have not observed those biases in the parameters of the CoalHMM. However, I did show in Figure 3.9 that it has a significant impact on the maximum likelihood value of the model, which is important for model selection. We mitigated the problem by increasing the number of discretization intervals which will also increase running time further.

(iii) The HMM methods do not easily analyze more than two sequences. The 3 population ILS CoalHMM (Subsection 3.2.4) integrates out the full coalescence histories for 3 sequences, but besides there are very few examples.
MSMC [61] integrates out only a summary of the full coalescent histories. On the other hand, ARGweaver is a method that can simulate the full coalescent histories for dozens of sequences.

### 4.1.1 ARGweaver

ARGweaver [57] applies MCMC to simulate a sample of ARGs conditioned a data matrix.

\[
\text{ARG}_1, \ldots, \text{ARG}_R \sim \text{ARG} | X \tag{4.5}
\]

where \( R \) is the length of the MCMC. Recall that an ARG is the information of all coalescence histories plus the recombination positions and branches. In other words, ARGweaver also simulates the coalescence histories, \( C \), necessary to calculate (4.1). ARGweaver assumes the HMM structure in (3.7) between \( C \) and \( X \). After having initialized \( \text{ARG}_1 \), the MCMC algorithm proceeds by iteratively

1. Choosing a branch in each coalescence history to simultaneously regraft. Ideally, the same branch would be chosen in all histories, because it would improve mixing. However, that is not possible for internal branches as the coalescence history changes between loci. Therefore, a heuristic algorithm creates a set of internal branch sequences, \( S(\text{ARG}) \), that are deemed related given the ARG. Then, either an external branch or a uniformly drawn internal branch sequence from the set is chosen.

2. Simulating the (discrete) regraft positions of all the chosen branches conditioned on all other branches. As a consequence of the model, the regraft positions are hidden states of a heterogeneous Hidden Markov Model (where the data matrix is the observed states). That enables simulation with the efficient stochastic traceback algorithm. To calculate the transition probabilities for each locus, all possible recombination events are summed out according to the SMC’s model.

3. Resimulating the recombination events involving the regrafted branch conditioned on the regraft positions. It is done independently for each locus.

4. Calculating the Metropolis Hastings acceptance probability. For internal branches it reduces to

\[
\alpha = 1 \wedge \frac{|S(\text{ARG})|}{|S(\text{ARG}_{\text{new}})|}
\]

and for external branches it reduces to 1.

In contrast to the HMM methods in Chapter 3, ARGweaver does not consider the joint distribution of coalescence histories at two adjacent loci. That distribution is manageable for a small number of sequences but intractable for many sequences as the space of possible states in the CTMC (Figure 3.4) increases super-exponentially in number of sequences. The state space explosion is avoided here by only looking at the distribution of one branch given the other branches and the previous coalescence history. However, the procedure does not offer a way to evaluate the likelihood nor a way to calculate the density.
of a single ARG, \( P(\text{ARG}) \). Availability of the density would allow importance sampling estimation of the likelihood as in (4.4). Without those, inference is often done conditioning on the ARGweaver estimates of the ARG ([57], [49]) with formula (3.11) or (3.12).

### 4.2 Particle Filtering

#### 4.2.1 An importance sampling technique

Particle filtering is an importance sampling technique to sample the hidden states of a model with the HMM structure in (3.7). Simulating coalescence histories with importance sampling requires a proposal function, 

\[
q(C | X)
\]  

that can simulate any possible sequence of coalescence histories. The resulting sample \( Z_1, \ldots, Z_R \) (also called particles) is related to the intended distribution, \( P(C | X) \), through the estimates

\[
E_{P(C|X)}[f(C)] \approx \frac{1}{R} \sum_{r=1}^{R} \frac{f(Z_r) P(Z_r | X)}{q(Z_r | X)} = \frac{1}{R} \sum_{r=1}^{R} f(Z_r) w_r \tag{4.7}
\]

The variance of the estimator in (4.7) depends greatly on the distribution of weights, \( \{w_r\}_{r=1, \ldots, R} \). If one weight is much higher than the other, (4.7) is effectively estimated by just one particle. Even worse, we can not know if the particle is a true sample. A higher-weight particle could be simulated which would alter (4.7) significantly. It is normal to guard against this problem by requiring that the heaviest weighted particles have similar weights. The Effective Sample Size (ESS), which is defined

\[
\text{ESS} = \frac{\left( \sum w_r \right)^2}{\sum w_r^2}, \tag{4.8}
\]

is used to measure this. It can be interpreted as the number of full particles in the sample.

If \( q \) is Markovian and we assume the HMM structure for \( (X, C) \), we can use the Particle Filtering methods. We assume the SMC model for two sequences under one population with constant population size. Hence, a coalescence history is just a single number and the transition probability is

\[
P(C(i) = t | C(i - 1) = s) = \begin{cases} 
\frac{\rho(e^{-\rho t} - e^{-s})}{1 - \rho} & \text{for } t < s \\
\frac{e^{-\rho t}}{1 - \rho} & \text{for } t = s \\
\frac{\rho e^{-(t-s)}(e^{-\rho t} - e^{-s})}{1 - \rho} & \text{for } t > s 
\end{cases} \tag{4.9}
\]

where \( C(i) \) is the coalescence history at the \( i \)th locus. The emission probability for observed alleles \( X_i := (X_{1,i}, X_{2,i}) \) is

\[
P(X_i | C(i) = t) = \begin{cases} 
e^{-\mu t} & \text{for } X_{1,i} = X_{2,i} \\
1 - e^{-\mu t} & \text{for } X_{1,i} \ne X_{2,i} \end{cases} \tag{4.10}
\]
The particle filtering method is similar to the MCMC and Importance Sampling methods outlined in Simulation Methods. They all perform simulation of the same continuous coalescence histories. To move beyond the bad scaling properties of the traditional coalescent sampling methods, we use the very efficient HMM structure formulation. In contrast to the HMM methods, we do not discretize the coalescence histories. Doing inference with the likelihood from (4.4) and continuous coalescence histories will naturally remove the discretization bias on the maximum likelihood value (described in Section 4.1). Instead all likelihood value estimates are stochastic and have a quantifiable variance.

Handling more than two sequences simultaneously is difficult for the HMM methods. Even the multisequence method ARGweaver is structured such that a hidden state never contains more than one coalescence time. The reason is that an HMM requires the probability for each transition to be calculated and the state of possible transitions explodes with number of sequences. In contrast, a particle filter only needs to know the probability of the actually simulated transitions. In some cases even that can be avoided by using the so-called bootstrap filter. However, we have not attempted implementing a particle filter for more than two sequences. Our goal is to examine possibilities with particle filtering in the simpler case with two sequences. If the two sequence particle filtering algorithm is feasible, one could extend it to more sequences. One could also imagine that it replaced the HMM stochastic tracebacks sampling in ARGweaver. With that important step, ARGweaver might be modified to generate continuous ARGs.

4.2.2 Particle Filtering Implementation

Let $Z_{rj}$ denote the $j$’th coalescence history of the $r$’th sample. Our implementation, follows the scheme

1. Forward Sampling. (Start with $\tilde{Z}_{r0}$ for $r = 1, \ldots, R$ and $j = 0$).
   
   (i) Starting in locus $j$ simulate the next $K$ coalescence histories for all $R$ particles. Calculate the partial weights $w^1_j, \ldots, w^R_j$ with the formula
   
   \[ w^r_j = \frac{P(X_{r(j+1)}, \ldots, X_{r(j+K)}, Z_{r(j+1)}, \ldots, Z_{r(j+K)} | Z_{rj}, \theta_0)}{q(Z_{r(j+1)}, \ldots, Z_{r(j+K)} | Z_{rj})} \]  
   
   (4.11)

   (ii) For all $r$, sample $\tilde{Z}_{r(j+K)}$ from the list $(Z_{1(j+K)}, \ldots, Z_{R(j+K)})$ with replacement and with respect to the weights $w^1_j, \ldots, w^R_j$. Set $j = j + K$ and if $j \geq N$ return to step (i).

2. Recreation
   
   (i) For all $r' = 1, \ldots, R'$, sample the vector $(Z_{r'(j-K+1)}, \ldots, Z_{r'(j)})$ from the list of $(Z_{r(j-K+1)}, \ldots, Z_{rj})$ for $r = 1, \ldots, R$ with respect to $w^{j+1}_1, \ldots, w^{j+1}_R$. Set $j = j - K$ and repeat this step until $j \leq 0$.

3. Return \{\hat{Z}_{rk}\}_{r', k=1,1}^{R', N}.

Ideally, step 2 would be the Backward Recursion algorithm.
For all \( r' = 1, \ldots, R' \), sample the vector \((\hat{Z}_{r'(N-K+1)}, \ldots, \hat{Z}_{r'(N)})\) from the list of \((Z_{r(N-K+1)}, \ldots, Z_{r(N)})\) with respect to \(w_1^N, \ldots, w_R^N\). Set \( j = N - K \).

For all \( r' = 1, \ldots, R' \), sample the vector \((\hat{Z}_{r(j-K+1)}, \ldots, \hat{Z}_{r(j+K)})\) from the list of \((Z_{r(j-K+1)}, \ldots, Z_{r(j+K)})\) with respect to \(\tilde{w}_1^{j+1,r'}, \ldots, \tilde{w}_R^{j+1,r'}\). The adjusted weight is

\[
\tilde{w}_1^{j+1,r'} = w_1^{j+1} P(\hat{Z}_{r'(j+K+1)}|Z_{r(j+K)}).
\]

Set \( j = j - K \) and repeat this step until \( j \leq 0 \).

The Backward Recursion algorithm samples all coalescence histories considering all the observed data. Unfortunately, the Backward Recursion is incompatible with our \( P \) because the densities \( P(\hat{Z}_{r'(j+K+1)}|Z_{r(j+K)}) \) put positive mass on the point \( Z_{r(j+K)} \) which is not the same across different values of \( r \).

To our knowledge, there is no workaround. When we settle for the Recreation algorithm, we do not obtain a sample of coalescence histories conditioned on the fully observed data, that is \( \hat{Z}_{r'j} \sim P(C(j)|X) \). Instead, each coalescence history is only conditioned on data up until the next multiplum of \( K \)

\[
\hat{Z}_{r'j} \sim P(C(j)|X),
\]

(4.13)

In addition, the Recreation algorithm breaks up links between segments. Therefore we write a sample as \( R' \times N/K \) segments

\[
(\hat{Z}_{1,1}, \ldots, \hat{Z}_{1,K}), \ldots, (\hat{Z}_{R',1}, \ldots, \hat{Z}_{R',K})
\]

\[
(\hat{Z}_{1,N-K+1}, \ldots, \hat{Z}_{1,N}), \ldots, (\hat{Z}_{R',N-K+1}, \ldots, \hat{Z}_{R',K}).
\]

Because the samples are simulated under a driver parameter value \( \theta_0 \), we calculate the likelihood as

\[
L(\theta) = \sum_{r'=1}^{R'} \prod_{0 \leq i \leq N/K-1} \frac{P(\hat{Z}_{r',i|K+1}|\theta)}{P(\hat{Z}_{r',i,K+1}|\theta_0)} \prod_{2 \leq j \leq K} \frac{P(\hat{Z}_{r',i|K+1}|\hat{Z}_{r',i|K+1-j+1}, \theta)}{P(\hat{Z}_{r',i|K+1}|\hat{Z}_{r',i|K+1-j+1}, \theta_0)}.
\]

(4.14)

Using the Recreation algorithm instead of the Backward Recursion algorithm is a source of bias. Fortunately, we choose \( K \) high enough to cover dozens of expected recombination events such that the bias is insignificant.

The performance of a particle filter depends on the proposal function, \( q \). The immediate choice is the prior

\[
q^{\text{prior}}(t|s) = P(C(i) = t|C(i - 1) = s).
\]

The optimal proposal (in terms of estimating the likelihood) is the posterior, \( P(C(i) = t|C(i - 1) = s, X) \). We compared the prior proposal function with our proposal recJumper that we made in the spirit of the optimal posterior. Here, I will explain the idea of recJumper. For full technical definition, I refer to [63].
4.2. PARTICLE FILTERING

1. Given tree height, \( s \), at locus \( i - 1 \) and the alleles of the following \( h \) loci, we calculate a pseudo probability that the next recombination is right before position \( i + k \) for \( k = 0, \ldots, h \). Ideally it is calculated as the product of the terms

\[
P(\text{no recombination for height } s)^{k-1} = e^{-\rho s (k-1)} \quad (4.15)
\]

\[
P(\text{recombination for height } s) = 1 - e^{-\rho s} \quad (4.16)
\]

\[
\int P(C(i+h), \ldots C(i+k)|C(i+k) \neq C(i+k-1) = s, X_{i+k}, \ldots X_{i+h}) \quad (4.17)
\]

where the integration in (4.17) is with respect to the tree heights for loci \( i + k, \ldots, i + h \). Computation of (4.17) is costly so we replace it with

\[
\int P(C(i+h), \ldots C(i+k)|C(i+k) \neq C(i+k-1), X_{i+k}, \ldots X_{i+h}) \quad (4.18)
\]

which does not depend on the previous tree height and does not need to be computed separately for each particle. We calculate (4.18) with a Monte Carlo estimate. Finally we calculate the probability that no recombination occurs before \( i + h \).

2. We simulate the distance to the next recombination, \( L \), with the probabilities in step 1. If no recombination occurred we jump to position \( i + h \) and return to step 1.

3. We simulate a new tree height valid until the simulated recombination position from the density

\[
P(C(i) = t|C(i + L) = \cdots = C(i) \neq C(i-1) = s, X_i, \ldots X_{i+L}). \quad (4.19)
\]

By approximating the number of mutations in the window \( i, \ldots, i + L \) with a poisson distribution, (4.19) becomes a linear combination of gamma densities which we simulate from with rejection sampling.

Our particle filtering method resamples every \( K \)'th locus and resampling requires weights (4.11). However, recJumper particles, as described above, might not ‘stop’ at the \( K \)'th locus. To resolve this, we adjust \( h \), such that \( i + h \leq \lceil i/K \rceil K \). Therefore, a small value of \( K \) would impair the benefit from using recJumper. Due to our compromise using the recreation algorithm, a small value will also increase the bias from this choice. For big values of \( K \) the weights disperse such that the Effective Sample Size (ESS) decreases. \( K = 1000 \) seems to be a good compromise for \( \rho = \mu = 0.1 \).

4.2.3 Results

In Figure 4.1, we compare the prior proposal to the recJumper proposal. We simulated data under the SMC model with \( \mu = \rho = 0.1 \). We generated hidden state sequence samples with either the prior or recJumper at various settings for the forward-looking parameter, \( h \). For each sample, we calculated the ESS and divided with the running time to get number of samples per second. recJumper skips many loci at once when it simulates a recombination further ahead in the
sequence. This feature is easy to implement for the prior proposal without
changing its distribution. Therefore, the prior in Figure 4.1 jumps to the next
recombination event, but still without information from the observed data. In
our implementation, recJumper at $h = 20$ is more effective than the prior for
sequences longer than 400 loci. Disregarding time, recJumper at $h = 2$ is better
than the prior. However the time consumption makes it less effective than the
prior. Increasing $h$ to 100 does not produce apparent advantages.

To test the particle filtering likelihood function in (4.14), we simulated
datasets with $\mu = \rho = 0.1$ and 20,000 loci from the SMC model. We imple-
mented an HMM for the same simple model. For both likelihoods, we optimized
with respect to the parameter $\rho$. Initial results showed that the particle filter-
ing likelihood estimates were biased towards the driver value, $\rho_0$. To combat
this, we generated several samples in a grid of driver values. We pooled all
samples from which we constructed a new likelihood function. It was, however,
still biased towards the driver values (Figure 4.2).

The bias towards the driver value is expected in importance sampling and it
should disappear when the number of particles is sufficiently high (Figure 4.4).
To examine the bias, we generated 100 datasets with $\mu = 1, \rho = 0.05$ and only
250 loci. We analyzed the datasets with particle filtering using driver value
$\rho_0 = 0.1$ and various number of particles (Figure 4.3). Unbiased estimation
was not achieved until we sampled 500,000 particles. At 20,000 particles, the
mean estimate had moved 0.02 from the driving value. Having more than one
parameter would also increase distance between driver values and demand more
particles.

My contribution: I created recJumper in collaboration with Master’s stu-
dent Simon Simonsen. I implemented the likelihood and the other particle
filtering techniques. All analyses was run by me but designed in collaboration
with Professor Asger Hobolth.
4.2. PARTICLE FILTERING

Figure 4.2: Comparison between (amongst others) the HMM and our particle filtering method. For 100 datasets with $\rho = \mu = 0.1$, the methods produced an estimate of $\rho$. They are sorted by their value in the plot.

Figure 4.3: Mean of the estimates of the recombination rate when simulating under the driver values $\rho = 0.1, \mu = 0.1$ (red). The real values are $\rho = 0.05, \mu = 0.1$ (green). The datasets are simulated with 100 loci.
Figure 4.4: Likelihood functions for $\rho$ plotted for a dataset with $\rho = \mu = 0.1$ and driver value $\rho_0 = 0.1$. As the number of particles increases, the particle filtering likelihood approaches the HMM likelihood. Far from the driver value, the variance of the likelihood function increases.

4.3 Discussion

We have shown that likelihood inference with particle filtering is possible. The plan was to do likelihood inference using importance sampling on the resampled particle filter sample. Unfortunately, the time consumption of this is enormous. After we finished the project, a PhD thesis describing another attempt at particle filtering, SMC$^2$, was published [24]. They avoided the importance sampling likelihood with an EM-similar procedure. First, a sample was generated for a driver value. Next, based on the sample, a likelihood of summary statistics of the sample was maximized to generate the next driver values. In this way, they did not need the enormous amount of particles needed to make their sample converge for many parameter values. Convergence was announced after around 40 iterations. Unfortunately, the running time in their method was (at time of publishing) detrimentally long.

We have shown that the effectiveness of the particle filter is improved when using an informed proposal function over the prior proposal function. The informed proposal function, recJumper, achieves this by suggesting high coalescence times for segments with many mutations and vice versa. The jumping
process, where recJumper simulates the next recombination event and jumps to that position, speeds up the process immensely. In SMC\(^2\) the prior proposal was used, but for a model with up to eight sequences. Their method might be faster if recJumper ideas were applied.

We have found that it is not feasible to simulate full conditional probabilities. The backward recursion algorithm, which normally would transform a forward sample into a full sample, is not meaningfully applicable. To see this, imagine we are tracing back a particle and stand in locus \(i\) with coalescence time \(t\). Picking a coalescence time for locus \(i-1\) requires us to pick between the available particles in locus \(i-1\). Let us denote them \((t_1, w_{i-1}^{(1)}), \ldots, (t_R, w_{i-1}^{(R)})\) and assume \(t_j = t\) and \(t_r \neq t\) for \(r \neq j\) (which occurs for one \(j\) with a probability approaching \(1 - \rho\) as \(K \to \infty\)). Given that one of these particles is the predecessor of our coalescence time \(t\), particle \(j\) is the predecessor with probability 1. In other words, we would recreate the forward sample except in those few places where the coalescence time is not in the previous sample. Instead of using the backward recursion algorithm, our solution was to split up segments and only condition on the observed data up until the next multiple of \(K\). Arguably a better solution was presented in SMC\(^2\). There, the sample was approximated with the fixed lag(\(\Delta\)) sample; for all \(i\), \(Z_{ri}\) was simulated conditioned on data up until \(i + \Delta\).

The problems with simulating from the full conditional distribution, makes particle filtering unattractive for incorporating in ARGweaver. ARGweaver is based on MCMC whose interpretability is vulnerable to a slight deviation in the simulation.

The bias of the recombination rate estimates towards the driver values and the computational efforts to resolve them should be a warning to other methods. Our simulated coalescence histories resemble coalescence histories from the driver model and not the true model. Inference with ARGweaver is normally done by using the ARG samples from the neutral model as observed values [49], [32]. Therefore, parameter estimates are biased towards the driver model which is also confirmed in the simulation study of ARGweaver [57]. To use ARGweaver one should simulate ARGs under different parameter values to check if one’s analysis is robust.
Chapter 5

AdmixtureBayes and admixturegraph

In this chapter I present my work on the AdmixtureBayes method. It is an extension of the method, admixturegraph, which I have also been involved in.

AdmixtureBayes and admixturegraph are programs that infer admixture graphs. An admixture graph can be divided into its graph structure, $G$, and its continuous parameters, $c_G$. The program admixturegraph uses Markov Chain Monte Carlo (MCMC) to sample from the distribution

$$P(c_G|X,G), \quad (5.1)$$

which it uses to calculate

$$P(X|G). \quad (5.2)$$

AdmixtureBayes samples with MCMC from the distribution

$$P(c_G,G|X). \quad (5.3)$$

Sampling from the joint distribution of parameters and graph structure is a hard computational problem because the discrete space of possible graph structures grows dramatically when the number of sequences increases. On the other hand, such a posterior sample will be of great use as it can be used to compute posterior probabilities of any admixture graph summary. Both admixturegraph and AdmixtureBayes calculate the likelihood with summary statistics whose distribution is predicted assuming brownian, unbounded, genetic drift.

5.1 Admixture graphs

In Figure 3.1 we saw the coalescence tree. It represents the history of some sequences at a single locus. A phylogeny is different because it represents the history of entire populations for all loci with a tree. We expect all coalescence trees to obey the phylogeny and only coalesce sequences that are in the same population. However, that requirement can lead to strange situations. Let us consider three populations where one is an admixture of the ancestral populations of the other two (Figure 3.15). Under this model, many coalescence histories will coalesce the admixed population with the first population and many will coalesce the admixed population with the third population.
Figure 5.1: The top two phylogenies do not agree on the position of $P_2$. In the bottom there is a phylogeny with an admixture event. When the $P_2$-population is an admixed population as depicted in the bottom, a phylogeny-estimating program may report either of the top trees.

However, we do not expect coalescence histories where the first and third population coalesce first (for long enough branches). This harmonizes poorly with the phylogeny description.

There are many popular programs for estimating phylogenies (e.g. Mr. Bayes [58], phylogeny.fr [11]). Admixture events can cause uncertainty in the estimates of these methods. Imagine a bootstrap sample of estimated phylogenies (or a bayesian sample). If there is an admixture event, the bootstrapped phylogenies could be divided into phylogenies supporting different paths through the admixture event (e.g. the blue and red phylogeny in Figure 5.1). This problem instigated research in constructing phylogenetic networks or consensus trees for visually summarizing the bootstrap samples (e.g. SplitTree [31]). However, one could easily imagine an admixture event going unnoticed. For example if 20% of loci in Figure 5.1 originate from the red tree and 80% from the blue, it is likely that all bootstrap replicates prefer the blue tree. To infer the admixture events it is therefore better to model them directly. There are several methods that estimate phylogenies containing admixture events (TreeMix [53], MixMapper [37], qpGraph [51]). I will denote phylogenies with admixture events as admixture graphs.

5.2 Summaries

Because phylogenies with many populations are more interesting, phylogenetic methods often summarize data heavily to reduce the computational burden. For admixture graphs, especially second moments of the allele frequencies ($f$-statistics and covariances) are popular (Table 5.1). In the following section I will define the most popular of those summaries and show equivalences between
5.2. SUMMARIES

5.2.1 Definitions

Consider the extended data matrix

\[
X = \begin{pmatrix}
x_{1,1,1} & x_{1,1,2} & \ldots & x_{1,1,N} \\
x_{1,2,1} & x_{1,2,2} & \ldots & x_{1,2,N} \\
\vdots & \vdots & \ddots & \vdots \\
x_{1,n_{1},1} & x_{1,n_{1},2} & \ldots & x_{1,n_{1},N} \\
x_{2,1,1} & x_{2,1,2} & \ldots & x_{2,1,N} \\
\vdots & \vdots & \ddots & \vdots \\
x_{m,n_{m},1} & x_{m,n_{m},2} & \ldots & x_{m,n_{m},N}
\end{pmatrix}
\]

where \(x_{i,j,k} \in \{0,1\}\) is the allele of locus \(k\) in individual \(j\) of population \(i\). The number of individuals in population \(i\) is \(n_{i}\). In this chapter, the data matrix is summarized using the sample allele frequencies

\[
\bar{x}_{i,k} = \frac{1}{n_{i}} \sum_{j=1}^{n_{i}} x_{i,j,k}.
\]

The sample allele frequencies are closely related to the population allele frequencies, \(p_{i,k}\). The relation is

\[
\bar{x}_{i,k} \sim \frac{1}{n_{i}} \text{bi}(n_{i}, p_{i,k})
\]

under the approximation that the population size is infinite. We regard all vectors \((p_{1,k}, \ldots, p_{m,k})\) for \(k = 1, \ldots, N\) as originating from one probability distribution. Let \((p_{1}, \ldots, p_{m})\) be a random vector from that probability distribution. The theoretical \(f\)-statistics are (up to a sign) the expected values

\[
E[(p_{i} - p_{j})(p_{k} - p_{l})], \quad i, j, k, l \in \{1, \ldots, m\}, i \neq j, k \neq l.
\]

They are further subdivided into 3 categories depending on how many of the populations, \(i, j, k, l\), are different

\[
F_{2}(i, j) = E[(p_{i} - p_{j})^{2}], \quad i \neq j
\]

\[
F_{3}(j, k; i) = E[(p_{j} - p_{i})(p_{k} - p_{i})], \quad \#\{i, j, k\} = 3
\]

\[
F_{4}(i, j; k, l) = E[(p_{i} - p_{j})(p_{k} - p_{l})], \quad \#\{i, j, k, l\} = 4.
\]

I denote the theoretical \(f\)-statistics with an uppercase \(F\) and the estimated \(f\)-statistics with a lowercase \(f\). The \(f\)-statistics are used by many methods including admixture graph (Table 5.1).

The expected value of \(p_{i}\) is the allele frequency in the root, \(p_{0}\). Ideally we could calculate the covariance matrix, \(\Sigma\), conditioned on \(p_{0}\) as

\[
\Sigma_{ij} = E[(p_{i} - p_{0})(p_{j} - p_{0})]
\]

However, that is not possible because \(p_{0}\) is unknown. Therefore, TreeMix uses \(\bar{p} = \frac{1}{m} \sum_{i=1}^{m} p_{i}\). In our method, AdmixtureBayes, we choose an outgroup population among the \(m\) population, say population \(s\). Put differently,

\[
\Sigma_{ij}^{\text{AdmB}} = E[(p_{i} - p_{s})(p_{j} - p_{s})], \quad \text{for } i, j \in \{1, \ldots, m\} \setminus \{s\}
\]

\[
\Sigma_{ij}^{\text{TreeMix}} = E[(p_{i} - \bar{p})(p_{j} - \bar{p})], \quad \text{for } i, j \in \{1, \ldots, m\}.
\]
5.2.2 Equivalences

In this section I will show that choosing between $F$-statistics and covariance matrix is not crucial.

Following the calculations in [53], we get

$$
\Sigma_{ij}^{\text{TreeMix}} = E[(p_i - \bar{p})(p_j - \bar{p})]
$$

$$
= E[(p_i - p_0 + p_0 - \bar{p})(p_j - p_0 + p_0 - \bar{p})]
$$

$$
= E[(p_i - p_0)(p_j - p_0)] - E[(p_i - p_0)(\bar{p} - p_0)]
$$

$$
- E[(p_j - p_0)(\bar{p} - p_0)] + E[(\bar{p} - p_0)^2]
$$

$$
= \Sigma_{ij} - \frac{1}{m} \sum_{k=1}^{m} E[(p_i - p_0)(p_k - p_0)]
$$

$$
- \frac{1}{m} \sum_{k=1}^{m} E[(p_j - p_0)(p_k - p_0)] + \frac{1}{m^2} \sum_{k=1}^{m} \sum_{l=1}^{m} E[(p_k - p_0)(p_l - p_0)]
$$

$$
= \Sigma_{ij} - \frac{1}{m} \sum_{k=1}^{m} \Sigma_{ik} - \frac{1}{m} \sum_{k=1}^{m} \Sigma_{jk} + \frac{1}{m^2} \sum_{k=1}^{m} \sum_{l=1}^{m} \Sigma_{kl}.
$$

It can be seen that

$$
\Sigma_{ij}^{\text{TreeMix}} = A \Sigma A^*, \quad A = \begin{pmatrix}
1 - \frac{1}{m} & -\frac{1}{m} & \cdots & -\frac{1}{m} \\
-\frac{1}{m} & 1 - \frac{1}{m} & \cdots & -\frac{1}{m} \\
\vdots & \vdots & \ddots & \vdots \\
-\frac{1}{m} & -\frac{1}{m} & \cdots & 1 - \frac{1}{m}
\end{pmatrix}.
$$

Notice that the rank of $A$ is $m - 1$ and $A \in \mathbb{R}^{m \times m}$. Hence, there will be linear dependencies in $\Sigma_{ij}^{\text{TreeMix}}$.

It can be calculated that

$$
\Sigma^{\text{AdmB}} = R \Sigma R^*, \quad R = \begin{pmatrix}
I_{s-1} & 0 \\
\vdots & & \vdots \\
0 & \vdots & \vdots \\
-1 & 0 & I_{m-s-1}
\end{pmatrix}.
$$

The rank of $R$ is also $m - 1$, but it has dimensions $(m - 1) \times m$ and there will not (always) be linear dependencies in $\Sigma^{\text{AdmB}}$. From the relation

$$
RA = R,
$$

it is evident that $\Sigma^{\text{AdmB}}$ is a (linear) function of $\Sigma_{ij}^{\text{TreeMix}}$. Furthermore for any $F$-statistics

$$
F = E[(p_i - p_j)(p_k - p_l)]
$$

$$
= E[(p_i - p_s)(p_k - p_s)] - E[(p_j - p_s)(p_k - p_s)]
$$

$$
- E[(p_i - p_s)(p_l - p_s)] + E[(p_j - p_s)(p_l - p_s)]
$$

$$
= \Sigma^{\text{AdmB}}_{i,k}1(i \neq s, k \neq s) - \Sigma^{\text{AdmB}}_{j,k}1(j \neq s, k \neq s)
$$

$$
- \Sigma^{\text{AdmB}}_{i,l}1(i \neq s, l \neq s) + \Sigma^{\text{AdmB}}_{j,l}1(j \neq s, l \neq s)
5.2. SUMMARIES

In other words, the theoretical $F$-statistics are linear transformation of the AdmixtureBayes covariance matrix. An entry in the TreeMix matrix can be written as

$$
\Sigma_{i,j}^{\text{TreeMix}} = E[(p_i - \bar{p})(p_j - \bar{p})] = \frac{1}{m^2} \sum_{k=1}^{m} \sum_{l=1}^{m} E[(p_i - p_k)(p_j - p_l)].
$$

All the terms in the sum are either a $F$-statistic or trivially 0. Hence, the TreeMix matrix is a linear transformation of the $F$-statistics.

In conclusion, all the described summaries are linear transformations of each other. Consequently, they contain the same information.

5.2.3 Estimation

Like the theoretical summaries, the observed summaries of the methods in Table 5.1 are linear transformations of the estimates

$$
\frac{1}{N} \sum_{j=1}^{N} (\bar{x}_{ij} - \bar{x}_{kj})(\bar{x}_{lj} - \bar{x}_{mj}) \tag{5.8}
$$

for suitable values on $j, k, l, m$ (if there is no missing data). Therefore, the observed summaries carry the same linear equivalences as their true values. The quantity in (5.8) is, however, biased from $E[(p_i - p_k)(p_l - p_m)]$ if $|\{i, k, l, m\}| < 4$. The bias is a consequence of the binomial distribution in (5.4), yet it can easily be computed and subtracted from (5.8). Linearity of the mean makes all bias-corrected estimates of the summaries linearly equivalent. In the rest of the thesis ‘observed summary’ will be any of these equivalent estimates and ‘summary’ will be their theoretical value.
<table>
<thead>
<tr>
<th>Method</th>
<th>Summary</th>
<th>Error Model</th>
<th>Goal</th>
<th>Maximization strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>TreeMix [53], 2012</td>
<td>Covariance matrix ((\Sigma_{\text{TreeMix}}))</td>
<td>Normally distributed with different variances, ((5.12))</td>
<td>(\arg\max_{G,c_G} \text{ of } P(X</td>
<td>G,c_G))</td>
</tr>
<tr>
<td>MixMapper [37], 2013</td>
<td>(F)-statistics</td>
<td>Normally distributed with the same variance ((5.11))</td>
<td>(\arg\max_{G,c_G} \text{ of } P(X</td>
<td>G,c_G))</td>
</tr>
<tr>
<td>qpGraph [51], 2012</td>
<td>(F)-statistics</td>
<td>Normally distributed with different variances, ((5.12))</td>
<td>(\arg\max_{c_G} \text{ of } P(X</td>
<td>G,c_G)) and compute (P(X</td>
</tr>
<tr>
<td>admixturegraph [35], Appendix C, 2017</td>
<td>(F)-statistics</td>
<td>Normally distributed with different variances, ((5.12))</td>
<td>Simulate from (P(c_G</td>
<td>G,X)) and compute (P(X</td>
</tr>
<tr>
<td>AdmixtureBayes, Appendix D, 2018</td>
<td>Covariance matrix ((\Sigma_{\text{AdmBayes}}))</td>
<td>Wishart distributed ((5.14))</td>
<td>Simulate (P(G,c_G</td>
<td>X))</td>
</tr>
</tbody>
</table>

Table 5.1: Table of models that estimates admixture graphs using second moments of allele frequencies. *) MixMapper needs user input to decide the order in which admixed populations are added.
5.3. Model

All methods in Table 5.1 calculate the summary, $S(X)$, and its expected value under the admixture graph, $S(G, c_G)$. To calculate the expected values it is assumed that allele frequencies develop from the root with random drift. Assume that $p_0$ is the allele frequency at the root. The drift of a branch with length $c_L$ is

$$t \sim (0, c_L p_0 (1 - p_0))$$  \hspace{1cm} (5.9)

where $(\mu, \sigma^2)$ is short notation for a distribution with mean $\mu$ and variance $\sigma^2$. Drift on one branch is independent of drift on the other branches. In an admixture event two branches merge (looking forward in time) and the allele frequency in the new population is a weighted average of the two incoming allele frequencies. The weight is the admixture proportion. The allele frequencies at the leaves of the graph are linear combinations of independent drifts. Therefore, we can easily calculate the covariance matrix, which will be of the form $\Sigma p_0 (1 - p_0)$ where $\Sigma$ depends on the graph structure, branch lengths and admixture proportions of the admixture graph. An example can be seen in Figure 5.2.

The methods of Table 5.1 compare the expected summary with the observed summary using a density function

$$h(S(X)|S(G, c_G)).$$  \hspace{1cm} (5.10)

In the previous section I showed that the methods have equivalent summaries. In contrast, the choices of $h$ contribute to different inferences. The choices of
CHAPTER 5. ADMIXTUREBAYES AND ADMIXTUREGRAPH

$h$ come from the options

\begin{align}
S(X) & \sim N(S(G, c_G), I_m \sigma^2) \quad \text{(5.11)} \\
S(X) & \sim N(S(G, c_G), \text{diag}(\sigma_1^2, \ldots, \sigma_m^2)) \quad \text{(5.12)} \\
S(X) & \sim N(S(G, c_G), \Psi) \quad \text{(5.13)} \\
S(X) & \sim W(S(G, c_G), \text{df}) \quad \text{(5.14)}
\end{align}

where $W(\Psi, \text{df})$ is the wishart distribution with mean $\Psi$ and degrees of freedom $\text{df}$. In (5.11) all entries of the observed summary are treated identically and independently. MixMapper uses this strategy. In (5.12) entries of the summary are treated independently but their individual variance is taken into account. Both admixturegraph, TreeMix and qpGraph use this strategy and estimate the constants $\sigma_1^2, \ldots, \sigma_m^2$ by resampling methods (bootstrap, jackknife). In the qpGraph paper [51], the model in (5.12) was considered. There, the error is freely, normally distributed, but the covariance $\Psi$ could not be estimated well using resampling.

Under the assumption that allele frequency drift is unbounded and normally distributed, any one locus $j$ fulfills

\[
\begin{pmatrix}
p_{1,j} \\
\vdots \\
p_{m,j}
\end{pmatrix} \sim N(p_{0j}, \Sigma p_{0j}(1 - p_{0j})), \quad \text{(5.15)}
\]

where $p_{0j}$ is the allele frequency at the root. If one had $N$ independent realizations of (5.15), one could calculate the empirical covariance, $\hat{\Sigma}$, which would be distributed

\[
\hat{\Sigma} \sim W(\Sigma p_{0j}(1 - p_{0j}), N - 1). \quad \text{(5.16)}
\]

However, we only have the observed allele frequencies $\bar{x}_{ij}$ and they are sampled from loci where the value of $p_{0j}$ varies. Nevertheless, in AdmixtureBayes we treat $p_{0j}(1 - p_{0j})$ as an unknown constant estimated by

\[
\beta = \frac{1}{N} \sum_{j=1}^{N} \bar{x}_j(1 - \bar{x}_j), \quad \text{where } \bar{x}_j = \frac{1}{m} \sum_{k=1}^{m} \bar{x}_{k,j}. \quad \text{(5.17)}
\]

In our data matrices, the loci are also neither independent as in (5.16) nor are the alleles normally distributed. This further decreases the suitability of the wishart distribution. To account for the accompanying inaccuracies, we allow the degrees of freedom, $\text{df}$, to vary freely. The AdmixtureBayes model is then

\[
\frac{1}{\beta} \hat{\Sigma}_{\text{AdmBayes}} \sim W(\Sigma_{\text{AdmBayes}}, \text{df}). \quad \text{(5.18)}
\]

The parameter $\text{df}$ is estimated by resampling the data. Instead of normalizing with $\beta$, it is possible to normalize all observations

\[
\frac{\bar{x}_{ij}}{\sqrt{\bar{x}_j(1 - \bar{x}_j)}}
\]

Calculating the empirical covariance for the normalized observed data yields

\[
\hat{\Sigma}_{\text{normalized}} \sim W(\Sigma, N - 1) \quad \text{(5.19)}
\]
which does not depend on $j$. This method is used by Ohana [10] and SpaceMix [7]. However, in simulations, we found that $\hat{\Sigma}_{\text{normalized}}$ was further from the truth than $\hat{\Sigma}_{\text{AdmBayes}}$.

In contrast to (5.11) and (5.12), the wishart distribution takes into account the dependencies within the observed summary. The normal distribution models in (5.11) and (5.12) ignores the dependencies and thereby loses power. On the other hand, the wishart distribution is computationally slightly more costly and builds on the dubious normal assumption about the observed allele frequencies.

5.4 Admixture graph space

Many imaginable admixture graph structures are not identifiable with the summaries (Figure 5.3). For example it is not possible to distinguish two subsequent admixture events from a single event (Figure 5.3B). If a branch has several incoming admixtures, it is not possible to determine their order (Figure 5.3C). TreeMix and MixMapper never visit unidentifiable graph structures in their search. In constrast, AdmixtureBayes searches through many graphs with unidentifiable structures. Including the unidentifiable structures makes the graph space more connected which improves mixing. In addition, there are also unidentifiable branch lengths; every admixture node has two parent branches and one child branch, yet only one parameter is identifiable (Figure 5.3 A).

Despite the smaller graph space of TreeMix and MixMapper, they also search a huge graph space where it is impossible to visit all admixture graph structures. In the following I will describe how the three search algorithms in Table 5.1 searches their graph space.

5.4.1 TreeMix

At first, TreeMix initializes a tree by fitting leaves to a base tree in a random order. Next, to add admixtures it locates pairs of populations with large residuals. For these populations, it finds non-admixture branches that will be candidates for start and end points of admixture branches. For every pair of candidates, an admixture branch is fitted to the tree by optimizing the branch lengths. The admixture branch producing the highest likelihood is then permanently added to tree. This procedure is repeated for a fixed number of admixture events. One could imagine fitting TreeMix for different number of admixture events and selecting the one with the lowest AIC value. Unfortunately, that is not possible because the admixture graph structure, $G$, is not a continuous parameter. Instead heuristic methods must be applied to determine this number.

5.4.2 MixMapper

At first, MixMapper will need a subset of populations that are not admixed and will constitute the scaffold tree. The program makes a ranked list of candidate scaffold trees from which the user can choose. Next, the non-scaffold populations will be added to the scaffold tree. The rest of the populations are added one by one or two by two as candidate populations.
Figure 5.3: Different graph structures. In A) there is one identifiable admixture event, but the blue branches are linearly dependent and constitute only one parameter (See Figure 5.2). In B) there are two admixture branches between the same two branches. Only one is identifiable and all the blue branches constitute just one parameter. In C) there are two identifiable admixtures and 5 blue branches that constitute just one parameter. However, the order of the admixture events is not identifiable. Therefore, TreeMix and MixMapper models them with the structure in D). In E) an admixture branch creates an 'eye'. That is not allowed in either program. In F) there is an identifiable admixture event. In contrast to the admixture event in A) this admixture event would not be present in the minimal topology (Figure 5.8).
(i) A candidate population is fitted to the tree as an admixture of two branches in the scaffold.

(ii) A previously fitted population, which I denote target population, is used to add the candidate population. The candidate population is added as an admixture of the target population and a scaffold branch.

No target populations is allowed to be a type (ii). Given a target branch and a candidate admixed population, MixMapper 2.0 [38] can decide between (i) and (ii). Version 2.0 also fits the target population and candidate population jointly. However, the program does not offer a way to choose suitable target populations and candidate populations. Furthermore, the procedure limits the graph space significantly; All admixed populations are leaf nodes and no population can be a direct admixture of more than 2 branches.

5.4.3 AdmixtureBayes

AdmixtureBayes starts in any admixture graph and samples other admixture graphs through a number of random MCMC transitions. We calculate the prior and likelihood for all graphs proposed by the proposal function. The graphs are accepted based on those values. This is a non-greedy approach as any graphs are visitable at any step in algorithm. It comes at a significantly increased computational cost. AdmixtureBayes uses an outgroup and does not allow any admixture branches going into the outgroup.

5.5 Bayesian approach

When estimating admixture graphs with a maximum likelihood approach it is common to replicate the analysis with resampled data. To assess uncertainty, MixMapper and TreeMix uses this strategy. In AdmixtureBayes, a sample of graphs is simulated from a posterior distribution and no bootstrap is needed. Both strategies can produce uncertainty estimates of any parameter. A natural question is which strategy is best?

There are already several bayesian estimators of phylogenies without admixture events (Mr. Bayes [58], BEAST [13]). The differences between the bayesian and maximum likelihood estimation of these phylogenies is a complicated question that has instigated a lot of research [30], [14]. The main differences between bayesian posterior and bootstrap uncertainty estimates are

1. The interpretability of the posterior uncertainty estimates is easier. The approach estimates the probability of \( f(G, c_G) \in A \) for some function \( f \) and some set \( A \) given that the overall model is correct. Bootstrap uncertainty estimates answer the fraction of repetitions of the experiment where the maximum likelihood estimate fulfill \( f(G, c_G) \in A \).

2. Bootstrapping is slow because it requires many repetitions of the full analysis.

3. The posterior uncertainty estimates are less robust.

4. The bootstrap method is less certain about the correct model.
Table 5.2: Number of possible admixture graph structures for $n$ leaves and $K$ admixture events. The growth in number of admixture events seems to be slightly faster than exponential. The growth in number of leaves is super-exponential. We calculate the numbers with a recursion.

![Table 5.2](data:image/svg+xml)

Neither approach is clearly better than the other. Our choice of approach stems from accommodating the great computational cost of searching the admixture graph space non-greedily. In AdmixtureBayes, we have chosen a thorough, but expensive search strategy. Luckily, it is only necessary to repeat the analysis once because it is bayesian. In contrast, TreeMix and MixMapper have chosen fast, greedy search procedures where bootstrap is manageable.

### 5.6 AdmixtureBayes method

AdmixtureBayes is a method for sampling from the distribution

$$P(G, c_G | X) \propto P(X | G, c_G) P(G, c_G)$$

(5.20)

using MCMC over the space of all the admixture graphs that we allow (Outlined in Section 5.4 and described in detail in attached paper D). The likelihood, $P(X | G, c_G)$, is the likelihood based on the wishart distribution of the observed AdmixtureBayes covariance (5.18). I define the prior, $P(G, c_G)$, below.

#### 5.6.1 Prior

Assume that the number of populations in graph is fixed to $n$. The number of different admixture graph structures with a given number of admixture events, $K$, is finite. However, it increases very fast in $K$ (Table 5.2). Besides, for any admixture graph $(G, c_G)$ with $K = k$ admixture events, it is possible to find an admixture graph with $K = k + 1$ leaves with the same or higher likelihood. Therefore, the prior is crucial in preventing the MCMC from diverging to $K = \infty$ because it assign low probability to high values of $K$.

Let $k(G)$ be the number of admixture events and let $b(G)$ be the number of branches for an admixture graph structure $G$. In addition, we write

$$c_G = (c_1, \ldots, c_{k(G)}, w_1, \ldots, w_{k(G)})$$

(5.21)
5.6. ADMIXTUREBAYES METHOD

Our prior is

\[ P(\mathcal{G}, c_{\mathcal{G}}) = P(\mathcal{G}|K=k(\mathcal{G}))P(K=k(\mathcal{G}))P(c_{\mathcal{G}}|\mathcal{G}). \] (5.22)

The first term in the product is the prior on the admixture graph structure given the number of admixture events. We chose the uniform prior because we can calculate the numbers (Table 5.2). We wanted the prior the prior on the number of admixture events to be easily interpretable. Therefore, we chose the geometrical prior with parameter 0.5. The last term is the prior on the continuous parameters. First we chose an exponential prior on the branch lengths and uniform priors on the admixture proportions. Surprisingly, setting the mean in the exponential prior was not straightforward.

**Mean of the exponential prior on branch lengths**

Let us assume that the exponential mean on branch lengths is always the same. The length of a branch is a measure of how much the alleles change on the branch. In datasets with bigger differences between allele frequencies, the total branch length should be higher. A high total branch length can be achieved through long branches or many branches. An admixture graph can get more branches by having more admixture events. Therefore, the more diverged the allele frequencies, the more admixture events would we infer. This is unsatisfactory because the allele divergence depends more on locus ascertainment than the actual population history. As a solution we would use the mean

\[ c_i \sim e(\lambda_{\mathcal{G}}), \quad \lambda_{\mathcal{G}} = \sum_{i=1}^{m} \frac{\hat{\Sigma}_{\text{AdmBayes}}}{B(\mathcal{G})} \frac{1}{\log_2(n)n} \] (5.23)

where \( \hat{\Sigma}_{\text{AdmBayes}} \) is the observed AdmixtureBayes covariance matrix. The quantity \( \sum_{i=1}^{m} \hat{\Sigma}_{\text{AdmBayes}} \) can be seen as the empirical value of the sum of all distances from a root to the node. In a tree without admixtures that corresponds to approximately \( \log_2(n)n \) branches. Hence, \( \sum_{i=1}^{m} \hat{\Sigma}_{\text{AdmBayes}} \log_2(n)n \) is the approximately the length of each branch in an admixture graph with no admixture events. Because \( 2n-2 \) is the number of branches in a tree and \( B(\mathcal{G}) \) is the number of branches in our admixture graph, \( \lambda_{\mathcal{G}} \) is approximately the average branch length in \( \mathcal{G} \). However, it is not allowed for the prior to depend on the data as in (5.23). Fortunately, we get the same effect by normalizing \( \hat{\Sigma}_{\text{AdmBayes}} \) and replacing the numerator in (5.23) with 1.

In conclusion, the prior only depends on the average branch length and the number of admixture events.

5.6.2 MCMC

The number of continuous parameters in an admixture graph depends on the number of branches and admixture events in the graph. To calculate the MCMC Metropolis Hastings correctly, we therefore have to use the Reversible Jump generalization [20]. To increase convergence speed, we implemented the MCMC as an MC³ algorithm (See Section 3.2.1).
Proposals

To move around in the admixture graph space, our proposal is a mix of seven proposals

1. **Adding admixture event.** Two random branches are chosen and an admixture branch added between them.

2. **Removing admixture event.** This is the reversal of the above proposal. A random admixture branch is chosen and removed from the graph.

3. **Node sliding.** This is a Subtree Pruning and Regrafting (SPR) move where the regraft position is chosen close to the pruning position. The proposal adapts the distance to the regraft position with adaptive techniques (Section 3.2.1 and [4])

4. **Drift all continuous parameters.** A random walk increment is added to the vector of continuous parameters. The proposal is also adaptive.

5. **Drift distance to outgroup.** A random walk increment is added to the branch length of the outgroup to the root. The proposal is also adaptive.

6. **Drift admixture proportions.** A random walk increment is added to the admixture proportions. The proposal is also adaptive.

7. **Drift on the likelihood contour.** Due to the nonidentifiable structures, many choices of branch lengths, \((c_1, \ldots, c_B(G))\) will produce the same \(\Sigma_{\text{AdmBayes}}\) and thereby the same likelihood. For a current branch length vector, we identify a subspace of branch length vectors producing the same covariance matrix, \(\Sigma_{\text{AdmBayes}}\). We then add random walk increment staying inside the subspace. The proposal is also adaptive.

Convergence criteria

It is important to ensure convergence of an MCMC because accepting premature samples can lead to incorrect inference. For phylogenetic MCMC’s determining convergence with measures of Effective Sample Size (ESS) is used [34]. Denote the sample of a phylogenetic MCMC \(x_1, \ldots, x_R\) and assume the sample is independent. For any real-numbered graph summary, \(f\), the calculation

\[
\frac{1}{R} \sum_{r=1}^{R} f(x_r) \sim \left( E[f(X)], \frac{\text{Var}[f(X)]}{R} \right)
\]

would hold. If the sample is not independent, the variance would be \(\frac{\text{Var}[f(X)]}{R'}\) where \(R' < R\). This \(R'\) is the ESS. However, the ESS depends on the summary, \(f\). Therefore, we choose an array of different summaries and calculate the ESS of each of them. If all ESS’s are above the arbitrary number 200, we announce convergence [34]. For phylogenies without admixture events there is already software implementing several reasonable choices of \(f\) (RWTY [66]). To use the software, we implemented functions transforming an admixture graph into a phylogeny by removing a parent branch of each admixture node.
5.7 AdmixtureBayes Results

5.7.1 Simulations

To test AdmixtureBayes, we ran simulations. For this we needed a process to simulate admixture graphs and datasets. To simulate an admixture graph we simulated it from a distribution similar to our prior

\[
(G_{\text{sim}}^0, c_{\text{sim}}^0) \sim \tilde{P}(G, c|K = k),
\]

which we normally condition on a number of admixture events, \(k\). The difference between our prior in (5.22) and (5.25) lies in \(P(G|K = k)\). It is not apparent how to simulate an admixture graph structure uniformly because it is not feasible to list all possible admixture graphs. Instead we deviced another prior that simulates admixture graph structures from the leaves and back to the root using a markov chain. All admixture graph structures without admixture events have identical probability. For admixture graph structures with admixture events, the admixture events are ‘closer’ to the root in the (5.25)-distribution compared to the (5.22)-distribution. It may be possible to simulate from the prior in (5.22) by applying rejection sampling, but we have not pursued this strategy.

To simulate a dataset from an admixture graph, we used ms [29].

Consistency

The MCMC in AdmixtureBayes is rather complicated. That makes it vulnerable to coding errors. Small calculation errors in the densities or simulation of any of the seven proposals may perturb convergence. To prevent this we made sure that the prior was recovered when setting the likelihood equal to 1 (Figure 5.4). Nothing suggests that the proposal distribution, acceptance probability, or prior have errors.

Convergence

The space of admixture graphs is huge and hard to search. It grows dramatically with the number of admixture events and number of leaves (Table 5.2). There will naturally be a limit for how big graphs we can analyze. To determine the limit, we generated an admixture graph, computed its theoretical covariance matrix, \(\Sigma_{\text{AdmBayes}}\), and used it as the observed covariance matrix. That is

\[
\hat{\Sigma}_{\text{AdmBayes}} = \Sigma_{\text{AdmBayes}}.
\]

For each graph, we ran AdmixtureBayes twice. One run started in the true (perfect) graph and the other started in a random graph. We repeated the whole process 10 times for each combination in a grid of number of populations 5, 10, 20, number of admixture events, 0, 1, 2 and df, 10^3, 10^4, 10^5, 10^6. In our real data analysis, we estimated df \(\approx 60,000\). We compared the perfect
run with the random run using the averaged posterior probability of the true graph structure. If the two averaged sampled probabilities are identical, it indicates that the MCMC would converge for any starting value. All runs were terminated after running 6 hours on 15 cores whether or not their convergence criteria were fulfilled. In conclusion, during $6 \times 15$ hours AdmixtureBayes converges for up to 10 populations (assuming $df \approx 60,000$) (Figure 5.5). It is also worth noting that convergence takes more time when the true admixture graph contains admixture events.

**Performance compared with existing software**

TreeMix and MixMapper both searches an admixture graph space. MixMapper is not designed for user-free interaction and its implementation does not contain a function that returns a full admixture graph. Designing such a function would require us to make decision rules for the user choices. In the origi-
Figure 5.5: Comparison of the average posterior probability of the true admixture graph structure. The red bars are runs starting in the true/perfect graph and the blue bars are runs started in a random tree. The rows show the number of populations in the graph (not counting the outgroup) and the columns show the df parameter. Each subplot is stratified on the number of admixture events in the true graph.
nal MixMapper paper [37], the authors outline a decision scheme to compare against TreeMix; the scaffold tree is the highest scoring scaffold tree and all admixed populations are added as type (i) (Subsection 5.4.2). This limits the populations to be either completely unadmixed or an admixture between two scaffold populations. The corresponding space of possible admixture graphs constitutes only a small fraction of the graph space of TreeMix and AdmixtureBayes. In conclusion, we do not compare AdmixtureBayes to MixMapper. TreeMix, on the other hand, does have a function that produces a full admixture graph. It takes two arguments; the data matrix and the number of admixture events in the graph. To ensure a fair comparison we did the following.

1. We simulated the true admixture graph from the overlap of the graph space of TreeMix and AdmixtureBayes. TreeMix does not allow unidentifiable admixture events like AdmixtureBayes. Conversely, AdmixtureBayes does not allow admixture branches going into the dedicated outgroup. We used the simulator in (5.25) but rejected graphs not in the intersection.

2. We ran TreeMix with the true number of admixture events. Repeating the TreeMix runs with different seeds gave identical results, so we used one repetition. AdmixtureBayes is not told the number of admixture events a priori, yet the MCMC chain can be constrained to a fixed number of admixture events. We include both the unconstrained and constrained AdmixtureBayes runs in the results.

3. We defined 3 comparison measures between admixture two graphs.

   a) The Fröbenius distance between their corresponding AdmixtureBayes covariance matrices. It will be denoted \textit{Covariance Distance}.

   b) A distance between their graph structures. By coding each node as a list of descendants (See the middle of Figure 5.8), we calculated the distance between two graph structures as the symmetric distance of their sets of nodes. It will be denoted \textit{Set Distance}.

   c) A dummy-variable for whether the two graph structures are identical (1) or not (0). It will be denoted \textit{Topology Equality}.

4. We measured the quality of the two methods with the comparison measures. TreeMix only infers a single graph, so the quality is the comparison measure between the true and inferred graph. AdmixtureBayes simulates a sample of graphs making the quality measure more complicated. First, we computed the average comparison measure across all sampled graphs. For the Set Distance and Topology Equality, we also computed the comparison measure between the true graph structure and the most sampled graph structure.

5. We repeated step 1.-4- 20 times for different combinations of sample sizes ($n_i = 2, 10, 50$), genome sizes giving $N \approx 4 \cdot 10^5, 1 \cdot 10^6$, and 0,1 and 2 admixture events. All graphs had 10 leaves plus one outgroup.
5.7. **ADMIXTUREBAYES RESULTS**

We found that TreeMix estimates admixture graphs whose covariance matrix is closer to the true covariance matrix for small sample sizes (Figure 5.6, Covariance Distance). The AdmixtureBayes posterior probability of the true graph structure is slightly lower than the TreeMix accuracy (Figure 5.6 Mean Topology Equality). However, the probability that the highest posterior admixture structure is equal to the true admixture structure is noticeably higher (Figure 5.6, Mode Topology Equality). This suggests that AdmixtureBayes is slightly too conservative and spreads out the probability mass too much. Measured in Set Distance, AdmixtureBayes infers admixture graph structures significantly closer to the true graph structure (Figure 5.6, Set Distances). Namely, when neither method finds the true graph, AdmixtureBayes infers graphs closer to the true graph.

Consider a subgraph for a subset of the populations in a bigger admixture graph. I will denote it a subgraph. With a posterior sample of admixture graphs, it is possible to obtain a posterior sample of any subgraph by marginalizing. Alternatively, a subgraph can be estimated from a subset of the data. We would expect that the latter is a worse option because it ignores information from the other populations. To test this we inferred subgraphs from the data in Figure 5.6. The 3, 4 or 5 populations in the subgraph were chosen randomly. For AdmixtureBayes, we find that estimated subgraphs from the full data are more accurate than subgraphs estimated from a data subset (Figure 5.7). For TreeMix, there is hardly any benefit from estimating the graph in the full distribution.

---

**Figure 5.6**: Inferred graphs are compared with the true graph using different quality measures. The graphs are inferred with TreeMix (green), AdmixtureBayes (uAdmBayes, blue) and AdmixtureBayes constrained to the true number of admixture events (CAdmBayes, red).
Figure 5.7: We inferred marginal admixture graphs from the full data (Big) and the same graphs from the marginal data (Small). The plot shows the Set Distance between the inferred subgraphs and the true sub graph.

5.7.2 Analysis on real data

We applied AdmixtureBayes to a dataset of full genome sequences humans from North America and Asia [45]. Several of the genomes are ancient DNA. Previous studies found significant gene flow between the populations, yet they disagree on the position [45] [56]. We ran AdmixtureBayes on 11 populations and used Yoruba as an outgroup. There are only one or two diploid sequences for each population. The degrees of freedom in (5.14) was estimated to \( \approx 60,000 \). The estimated posterior sample was very diverse and the 95% credibility interval on the number of admixture events was [7, 16]. The high number of admixture events is supported by Moreno-Mayer et al. [45]. The highest posterior probability for any specific admixture graph structure was < 0.5%. Due to the diversity of the sample, it is not meaningful to report just one admixture graph.

To summarize the admixture graphs, we introduced the minimal topology (Figure 5.8). In this mindset we think of an admixture graph structure as a set of nodes coded by their descendants, that is the topology set. When summarizing many graphs we take all nodes which are present in more than \( X \% \) of the topology sets. From such a set of nodes, we construct a graph by joining the nodes. It potentially creates nodes of a high degrees which represents many events (population splits, admixtures). We interpret a high-degree node as a lack of certainty about the order or existence of these events. For the minimal graph with \( X = 50\% \) we see that Koryak, Ket, USR1, Native Americans and Inuits all arose from a population admixed between Han and Malta (Figure 5.9). From this ancestral population arose 5 lineages without strong evidence of interbreeding; The USR1, the Ket, the Saqqaq, the South
American (Anzick and TA6) and Koryak. We found evidence that Athabascans is a mix of the Koryak and South American lineage. There is also strong evidence that Greenlander is a mix of Saqqaq, South American and Koryak.

We looked specifically into the subgraph of Saqqaq, Koryak, Athabascans and Greenlandic Inuit. Raghaven et al. found that the Athabascans were the outgroup but with significant gene flow with the Greenlanders [56]. The other three belonged in a clade where Koryak was the outgroup. However, we find that Koryak is closer related to Athabascans than Saqqaq. Looking at the marginal graphs, the tree ((Koryak, Athabascans),Saqqaq) has posterior probability 59% (Figure 5.10) and the tree ((Koryak, Saqqaq), Athabascans) has probability 11%.

**My contribution to AdmixtureBayes** The model and its choices were designed in collaboration with Professor Rasmus Nielsen. I implemented the code and has run all experiments and analyses. Kalle Leppälä derived the recursion for the number of admixture events. Diagnostics, simulation study
and analysis on real data were planned in collaboration with Rasmus Nielsen and Thomas Mailund.

My contribution to admixturegraph I implemented and tested the code for the MCMC.

5.8 Discussion

AdmixtureBayes is a program capable of producing a sample of admixture graphs for around 10 populations at a time consumption around 100 parallelizable CPU hours. The admixture graph space is both enormous and not a subset of $\mathbb{R}^k$, hence many iterations are needed. In contrast to previous methods, AdmixtureBayes neither uses a greedy nor an exhaustive search.

For simulated datasets, the inferred graphs are closer (in terms of Set Distance) to the true graph than TreeMix’ inferred graphs are. The two methods are different in their model, search algorithm and prior. When conditioning on the number of admixture events, the prior is relatively uninformative, so it is unlikely that the prior can explain the differences. Both TreeMix and Admixture have passed convergence tests, so it is also unlikely that the search algorithm can explain the difference. The model may be a more likely source of explanation. TreeMix models each entry in the covariance matrix as an
### Figure 5.10: The top minimal topology in two different subgraphs.

<table>
<thead>
<tr>
<th>Sub minimal topology: Athabascan, Koryak and Saqqaq.</th>
<th>Sub minimal topology: Greenlander, Koryak, Athabascan and Saqqaq.</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="#" alt="Diagram 1" /> Posterior: 59%, BF=1.9</td>
<td><img src="#" alt="Diagram 2" /> Posterior: 43%, BF=241</td>
</tr>
<tr>
<td><img src="#" alt="Diagram 3" /> Posterior: 13%, BF=8.1</td>
<td><img src="#" alt="Diagram 4" /> Posterior: 10%, BF=56</td>
</tr>
<tr>
<td><img src="#" alt="Diagram 5" /> Posterior: 11%, BF=6.9</td>
<td><img src="#" alt="Diagram 6" /> Posterior: 6%, BF=957</td>
</tr>
</tbody>
</table>


independent normal distribution whereas AdmixtureBayes models the whole
covariance matrix with a Wishart distribution. Normal random deviations
from the true covariance matrix is supposedly more likely with the Wishart
distribution. This could explain why AdmixtureBayes seems to be closer at
the truth for the wrongly inferred graphs.

AdmixtureBayes is based on a summary of allele frequency second moments.
The summary makes likelihood evaluations very fast, yet it discards a lot of
valuable information. For this reason many graph features are unidentifiable.
In Subsection 3.2.4 I described the 3 population admixture model that can
estimate the time of admixture. It exploits the linkage information ignored
by AdmixtureBayes. Interestingly, AdmixtureBayes has a modular structure
where the likelihood could easily be replaced. However, the replacement like-
lihood should be quickly computable because the MCMC requires millions
of iterations to converge.

We have analyzed a full genome dataset of Native Americans and confirmed
the main picture that previous studies found. We also contributed with new
knowledge because our analysis showed that some of the originally inferred
features have low posterior probability. The history of the Native Americans
contains many admixture events, yet AdmixtureBayes still converged. The
many admixture events meant that we estimated an extremely diverse sample
of admixture graphs. To summarize them, we created a new graph structure
where only the nodes with highest posterior probability occur.

AdmixtureBayes only uses the covariance matrix and not any a priori knowl-
edge when searching the graph space. Therefore, the posterior sample of-
ten contains many admixture graphs that are obviously wrong. Contrarily,
MixMapper gives the user some control over which features to include in the
graph. There are some obvious dangers to this approach, but it also gives more
correct graphs. Mathematically, prior knowledge is easy to impose on an Ad-
mixtureBayes posterior sample through thinning. One would simply discard
every admixture graph that violates the prior knowledge. The effect of the
prior knowledge would also be more transparent than in MixMapper.
Chapter 6

ImmediateAncestry

ImmediateAncestry is a program we created to solve a problem specific to animal parks. Many animals in zoos are hybrids of different subpopulations, but it is often unknown what type of hybrid. ImmediateAncestry makes an estimate by calculating and maximizing a likelihood

\[ P(X_i | g) \]  

(6.1)

where \( X_i \) is one row of a data matrix and \( g \) is a configuration assigning the ancestors to subpopulations. To increase accuracy, we exploit linkage between loci by modelling the ancestors along a sequence as a Hidden Markov Model. With the information from linkage we can go beyond the simple classification of individuals into first-generation (F1), second-generation (F2) and third-generation (F3) hybrids. ImmediateAncestry estimates all hybrids in the pedigree up to 2, 3 and 4 generations.

There exists other tools to classify hybrids (NewHybrids [3], STRUCTURE [55], ADMIXTURE [1], Bayesass [67]). However, they model loci independently and thereby has no power to detect all hybrids in the pedigrees.

Applying the method on real data has revealed robustness issues. Adjusting the model is ongoing work.

6.1 Chimpanzee dataset

The ImmediateAncestry model is motivated by a dataset of 267 chimpanzee genomes sequenced with a SNP array. Of the 267 individuals, 61 are wild-caught of known subspecies. There are four subspecies; Pan troglodytes verus (v), Pan troglodytes ellioti (e), Pan troglodytes troglodytes (t), and Pan troglodytes schweinfurthii (s) (Figure 6.1). The phylogeny forms two clades in the structure \(((v,e),(s,t))\) and gene flow has been found between the clades [54].

The remaining 206 chimpanzees have lived in parks for up to 4 generations. After being imported to the zoo, there has been ample interbreeding amongst them forming hybrids. Unfortunately, the records of their pedigrees are incomplete. Hence, we build ImmediateAncestry to recreate this very recent ancestry.

We use a chimpanzee recombination map [6] to model the linkage sensibly. The recombination map is unscaled, so we normalize to one recombination per 100 mb per meiosis.
Figure 6.1: PCA of all individuals in the chimpanzee dataset. There are four subspecies (s, v, e, and t). The Zoo individuals are treated as if their origin is unknown.

6.2 Model

Because the hybridization is only a few generations old, a chromosome of a hybrid individual will have large chunks of DNA coming from the same subspecies. Therefore, the correlation between the ancestry of two nearby loci is substantial. To model this, we assume that the ancestors \((A_j)_{1 \leq j \leq N}\) is Markovian along the sequence. We define \(A_j = (A_{j,1}, A_{j,2})\) where \(A_{j,k}\) is the ancestor to locus \(j\) in the \(k\)'th half of the pedigree (Figure 6.2). The ancestor sequence varies because of recombinations. The probability of a recombination on a branch in a pedigree is provided by the recombination map. When a recombination occur, the ancestor sequence change according to the recombinating branch (Figure 6.2). The ancestor sequence constitute the hidden states in the Hidden Markov Model (HMM) of ImmediateAncestry.

The parameters of the model are the subspecies of the ancestors. We call the set vector of parameters the configuration and denote them with \(g\). When
6.2. MODEL

The observed states in the HMM are a row in the unphased data matrix. The row is coded with 0, 1 or 2 for the number of major alleles at each locus. Let $f(s, j)$ be the allele frequency of the major allele in population $s$ at locus $j$. The emission probabilities of the HMM are

$$P(X_{ij} = x_{ij} | A_j, g) = \begin{cases} 
[1 - f(g(A_{j,1}, j)) \cdot [1 - f(g(A_{j,2}, j))] & \text{if } x_{ij} = 0 \\
\frac{1}{2} \cdot f(g(A_{j,1}, j) \cdot [1 - f(g(A_{j,2}, j)] & \text{if } x_{ij} = 1 \\
+ \frac{1}{2} \cdot [1 - f(g(A_{j,1}, j)] \cdot f(g(A_{j,2}, j)) & \text{if } x_{ij} = 2 \\
f(g(A_{j,1}, j)) \cdot f(g(A_{j,2}, j)) & \text{if } x_{ij} = 2.
\end{cases}$$

In the case $x_{ij} = 1$, one of the two ancestors must have possessed a 1 and the other a 0. However, we do not which ancestor provided the 1. Therefore, (6.3) integrates out the two possibilities.

As in Chapter 3 and 4, we know that the recombination process does not produce a Markovian ancestor sequence. However, the Markov assumption bias is manageable in the previous models, so it is probably not detrimental for ImmediateAncestry.

We use the HMM structure to calculate the likelihood $P(X_i | g)$ with the forward algorithm. Many configurations produce identical likelihoods as there...
are symmetries in the emission and transition probabilities. For 2 generations,
\[ L(abcd) = L(abdc) = L(cdab), \quad \text{for all } a, b, c, d \in \{e, s, t, v\} \] (6.4)
yet still
\[ L(abcd) \neq L(acbd), \quad \text{for } a, b, c, d \in \{e, s, t, v\}, b \neq c. \] (6.5)

In other words, every node in the pedigree can switch the order of its branches
without altering the likelihood. It is still possible to estimate ancestry of all
individuals in all generations in the pedigree. As a consequence we only search
a reduced parameter space. For 2 and 3 generations brute-force is feasible. For
4 generations we have implemented a simulated annealing optimization.

6.3 Results

First, we conducted a simulation study. To examine the performance of ImmediateAncestry 3 generations back in time, we chose \(2^3\) individuals amongst the
wild-caught chimpanzees. We ‘bred’ them according to the ImmediateAncestry model for 3 generations to create one hybrid individual. We ran ImmediateAncestry to estimate the best configuration for generation 3. After repeating
the procedure 371 times, the accuracy was close to 45 % and 96% of the inferred
configurations was wrong with one letter or less (Figure 6.3). 89% of the
wrong letters in the inferred configurations was from the \(t\) subspecies. It is not
peculiar considering their central position (Figure 6.1).

Next, we applied the method to the real chimpanzee data. There are two
versions of the dataset giving very different results (Table 6.1). The first dataset
contains 34,062 SNPs due to smaller a SNP array, lower quality reads and
thereby more filtering. The second dataset contains 481,941 SNPs. The first
dataset also had less than the 267 individuals. Surprisingly, no individual was
classified as a pure individual on the big dataset.

<table>
<thead>
<tr>
<th>34062 SNP dataset</th>
<th>481941 SNP dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\theta)</td>
<td>count</td>
</tr>
<tr>
<td>vvvvvvvv</td>
<td>42</td>
</tr>
<tr>
<td>sssssss</td>
<td>31</td>
</tr>
<tr>
<td>tttttttt</td>
<td>22</td>
</tr>
<tr>
<td>ssssssst</td>
<td>13</td>
</tr>
<tr>
<td>eeeeeee</td>
<td>8</td>
</tr>
<tr>
<td>sttttttt</td>
<td>5</td>
</tr>
<tr>
<td>tttvtttvv</td>
<td>5</td>
</tr>
<tr>
<td>sssttstt</td>
<td>4</td>
</tr>
<tr>
<td>eecteect</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 6.1: Tables of the most commonly estimated configurations for the two
datasets.

In an HMM it is easy to simulate probable hidden states conditioned on
the observed data (and parameters). We simulated such hidden states for the
individual Ptt-Clara. We know is a pure individual from subspecies \(t\), yet
according to the analysis on the new data its configuration is \(evtttttv\). A vast
6.3. RESULTS

Figure 6.3: 371 individuals were simulated and analyzed with ImmediateAncestry. We compared the true configurations with the inferred configurations using a measure that maps into 9 categories. The categories are: identical (id), the configurations are identical if two letters from the same half are permuted (perm-), identical if two letters from different halves are permuted (perm), identical after a larger permutation between the inferred letters (perm+). '1sub' means that one letter should be substituted for them to be identical. The combination of 'sub' and 'perm-' means that the inferred configuration is off by one substitution and one small permutation and so forth.

majority of the loci had ancestors from t (Figure 6.4). However, many small chunks scattered across the genome prefer ancestors e and v. The size of all the small chunks combined is almost negligible and does not justify the inclusion of e and v in the inferred configuration. Nevertheless, the likelihood of the configuration with e and v is higher.

6.3.1 Robustness issue

A chunk in Figure 6.4 consists of 10-500 alleles where the majority of them are more likely under the e or v population. Analyzing the small dataset, the chunks were presumably still present but too small to be detectable (Figure 6.5). We observe similar patterns for other individuals except the deviant chunks are in other positions. Furthermore, we ran the ImmediateAncestry model on thinned versions of the new dataset and estimated more pure individuals.

In other words, there are many small segments of DNA pointing systematically against the population from which they were sampled. This might be explained by the gene flow that was inferred by Prado-Martinez et al [54]. Regardless of reason, ImmediateAncestry accommodates the deviating seg-
Figure 6.4: Simulated hidden states based on the maximum likelihood estimate \textit{evtttttv} conditioned on the observed data for the individual Ptt-Clara. The rows are the different chromosomes, the x-coordinate indicates the locus. There are many small segments.

Figure 6.5: The same as Figure 6.4 based on the configuration \textit{evtttttv}, but for the small dataset. On the small dataset, the maximum likelihood configuration was correctly \textit{tttttttt}. 
ments by including other subspecies in the inferred configuration. The ImmediateAncestry model is not robust to the deviations.

My contribution: I implemented the initial ImmediateAncestry from the idea of Thomas Mailund. I executed all simulations and analyses and planned them in collaboration with Thomas Mailund. I implemented the hidden states plots in collaboration with visiting PhD student Natalí Fernández.

6.4 Discussion

On simulated chimpanzee data I have shown that ImmediateAncestry is capable of approximately estimating the subspecies of the third generation ancestors. On real chimpanzee data ImmediateAncestry did not produce sensible results. The reason is a lack of robustness.

One solution to the robustness issue could be to thin the data. If the data shrank to a size where the chunks are negligibly, ImmediateAncestry would infer better results as mentioned in the previous section. The necessary degree of shrinkage would depend on the data. A major downside to this approach is the loss of statistical power. ImmediateAncestry has an estimated 45% accuracy for 3 generations on simulated data, so decreasing power is unappealing.

From the plots of conditionally simulated hidden states, one can tell that Ptt-Clara is a pure $t$-individual. The small segments of $e$ and $v$ are too small for the configuration $evtttttv$. Therefore, an alternative solution to the robustness issue could be a heuristic algorithm based on simulated hidden states and the biased maximum likelihood estimate.

Towards the end of the project a similar method, PedMix, was published [52]. PedMix assumes that an individual is a admixture of two populations. It estimates the admixture proportions of the ancestors of an individual. In ImmediateAncestry a hidden state of the HMM is an individual’s two ancestors. In PedMix, a hidden state is expanded to the phases of all nodes in the pedigree. The PedMix hidden state contains more information, but is computationally costly. Therefore, they only analyze the first and second generation back in time. The robustness issue is avoided, because they do not require that the ancestors are pure individuals.

Modeling the ancestors as admixed populations as Pedmix might be an ideal solution to the robustness issue. It reflects our perception of the reality because we believe the lack of robustness stems from unmodeled gene flow. The complexity of the problem would increase and the results may become less interpretable. To counter these effects, it could be beneficial to regularize the admixture of the ancestors by, for example, making them at least 95% pure.
Bibliography


Appendix A

Elephant paper

In the attached paper I executed the IM CoalHMM analysis. The analysis design and model improvements were made in collaboration with Thomas Mailund.
A comprehensive genomic history of extinct and living elephants

Eleftheria Palkopoulou,a,b,1 Mark Lipsona, Swapan Mallicka,b, Svend Nielsenb, Nadin Rohlandd, Sina Balekd, Emil Karpinski,b,1,f,g,h Atma M. Ivancevic,1 Thu-Hien To,1 R. Daniel Kortschak,1 Joy M. Raison,1 Zhipeng Qu,1 Tat-Jun Chin,1 Kurt W. Alt1,l,m, Stefan Claessonn, Love Dalén,5,1 Ross D. E. MacPheerc, Harald Mellerf, Alfred L. Rocaf,s, Oliver A. Ryder,1 David Heimana,b, Sarah Youngb, Matthew Breen4, Christina Williams4, Bronwen L. Aken,w,2 Magali Ruffier,w,2 Elinor Karlsson,b,x Jeremy Johnsonb, Federica Di Palma,7,1 Jessica Alfoid, David L. Adelson, Thomas Mailund,7,1 Kasper Munch,2,3 Kerstin Lindblad-Toh,b,2,2 Michael Hofreiterd,e,f,g,h, and David Reicha,b,aa,1,2

1Department of Genetics, Harvard Medical School, Boston, MA 02115; 2Broad Institute of MIT and Harvard, Cambridge, MA 02142; 3Biogenomics Research Centre, Aarhus University, DK-8000 Aarhus, Denmark; 4Unit of General Zoology-Evolutionary Adaptive Genomics, Institute of Biochemistry and Biology, Faculty of Mathematics and Life Sciences, University of Potsdam, 14476 Potsdam, Germany; 5McMaster Ancient DNA Centre, Department of Anthropology, McMaster University, Hamilton, ON L8S 4L9, Canada; 6Department of Biology, McMaster University, Hamilton, ON L8S 4K1, Canada; 7Department of Biochemistry, McMaster University, Hamilton, ON L8S 4L8, Canada; 8The Michael G. DeGroote Institute for Infectious Disease Research, McMaster University, Hamilton, ON L8S 4L8, Canada; 9Department of Genetics and Evolution, School of Biological Sciences, The University of Adelaide, Adelaide, 5005 SA, Australia; 10School of Computer Science, The University of Adelaide, 5005 SA, Australia; 11Center of Natural and Cultural Human History, Danube Private University, A-3500 Krems, Austria; 12Department of Bioinformatic Engineering, University Hospital Basel, University of Basel, CH-4123 Basel, Switzerland; 13Institute of Prehistory and Archaeological Science, University of Basel, CH-4055 Basel, Switzerland; 14Institute of Maritime History, Tall Timbers, CA 20969; 15Department of Bioinformatics and Genetics, Swedish Museum of Natural History, SE-10405 Stockholm, Sweden; 16Division of Vertebrate Zoology/ Mammalogy, American Museum of Natural History, New York, NY 10024; 17State Office for Heritage Management and Archaeology, 06114 Halle (Saale), Germany; 18Department of Animal Sciences, University of Illinois at Urbana-Champaign, Urbana, IL 61801; 19Institute for Genomic Biology, University of Illinois at Urbana-Champaign, Urbana, IL 61801; 20Institute for Conservation Research, San Diego Zoo, Escondido, CA 92027; 21Department of Molecular Biomedical Sciences, College of Veterinary Medicine, North Carolina State University, Raleigh, NC 27670; 22European Molecular Biology Laboratory, European Bioinformatics Institute, Hinxton, CB10 1SD Cambridge, United Kingdom; 23Welcome Sanger Institute, Hinxton, CB10 1SD Cambridge, United Kingdom; 24Program in Bioinformatics and Integrative Biology, University of Massachusetts Medical School, Worcester, MA 01655; 25Earlham Institute, NR4 7UZ Norwich, United Kingdom; 26Science for Life Laboratory, Department of Medical Biochemistry and Microbiology, Uppsala University, 751 23 Uppsala, Sweden; and 27Howard Hughes Medical Institute, Harvard Medical School, Boston, MA 02115

Edited by David M. Hillis, The University of Texas at Austin, Austin, TX, and approved January 24, 2018 (received for review November 24, 2017)

Elephantids are the world’s most iconic mega faunaal family, yet there is no comprehensive genomic assessment of their relationships. We report a total of 14 genomes, including 2 from the American mastodon, which is an extinct elephantid relative, and 12 spanning all three extant and three extinct elephantid species in- cluding an ∼120,000-y-old straight-tusked elephant, a Columbian mammoth, and woolly mammoths. Earlier genetic studies mod- eled elephantid evolution via simple bifurcating trees, but here we show that interspecies hybridization has been a recurrent fea- ture of elephantid evolution. We found that the genetic makeup of the straight-tusked elephant, previously placed as a sister group to African forest elephants based on lower coverage data, in fact comprises three major components. Most of the straight-tusked elephant’s ancestry derives from a lineage related to the ancestor of African elephants while its remaining ancestry consists of a large contribution from a lineage related to forest elephants and another related to mammoths. Columbian and woolly mammoths also showed evidence of interbreeding, likely following a latitudi- nalcline across North America. While hybridization events have shaped elephantid history in profound ways, isolation also ap- pears to have played an important role. Our data reveal nearly complete isolation between the ancestors of the African forest and savanna elephants for ∼500,000 y, providing compelling jus- tification for the conservation of forest and savanna elephants as separate species.

Significance

Elephantids were once among the most widespread mega faunaal families. However, only three species of this family exist today. To reconstruct their evolutionary history, we generated 14 ge- nomes from living and extinct elephantids and from the Ameri- can mastodon. While previous studies examined only simple bifurcating relationships, we found that gene flow between elephantid species was common in the past. Straight-tusked el- ephants descend from a mixture of three ancestral populations related to the ancestor of African elephants, woolly mammoths, and present-day forest elephants. We detected interbreeding between North American woolly and Columbian mammoths but found no evidence of recent gene flow between forest and sa- vanna elephants, demonstrating that both gene flow and iso- lation have been central in the evolution of elephantids.

Members of the family Elephantidae, known as elephantids, first appeared in Africa 5 to 10 Mya and are the only surviving family of the order Proboscidea (1, 2). Although many fossil species have been identified, high levels of within-taxon variation have complicated the delineation of species bound- aries (1–3). Living elephantids include two species of the genus Loxodonta, the forest elephant (Loxodonta cyclotis) and the sa- vanna elephant (Loxodonta africana), which are restricted to Africa, and one of the genus Elephas, which is endemic to Asia (Elephas maximus). Extinct mammoths (genus Mammuthus) comprise several species, of which the once circum polar woolly mammoth (Mammuthus primigenius) survived in small isolated island populations well into the Holocene until ∼4,000 y ago.
(4, 5) while the more temperate North American Columbian mammoth (Mammuthus columbi) disappeared by the end of the last ice age ~11,000 y ago (6, 7). Straight-tusked elephants (genus Palaeoloxodon) potentially survived as late as ~50,000 to 35,000 y ago (8) and have been conventionally grouped within Elephas (3, 9), but recent genomic evidence from European straight-tusked elephants (Palaeoloxodon antiquus) over 100,000 y old showed that they were on average more closely related to forest elephants than to any other extant species and led to the suggestion that they were an ancient sister group of modern African forest elephants (10).

Results and Discussion

A High-Quality Elephant Reference Genome. This study formally reports the high-quality reference genome of the African savanna elephant, which first became available online in May 2005 (LoxAfr1) and has since been iteratively updated with the latest release available online in May 2014 (LoxAfr4). We used classic Sanger-sequencing methods to generate a de novo genome assembly from a savanna elephant at 6.8-fold coverage. Specifically, we performed paired-end Sanger sequencing using multiple insert sizes [4 kilobases (kb), 10 kb, 40 kb, and BAC clones]. We then used FISH mapping of BAC clones to place scaffolds containing 85% of the assembly onto chromosomes. The assembly has a median (N50) contig length of 60 kb and a median scaffold length of 48 megabases, with a total assembly length of 3.2 gigabases (SI Appendix, Table S1.1). The assembly contains 47.8% easily recognized repeat-derived sequences (28.9% long interspersed nuclear elements, 8.7% short interspersed nuclear elements, 6.7% long terminal repeats, 0.5% simple repeats, and 3.0% “other”) and 20,333 protein coding genes.

Proboscidean Dataset and Genome-Wide Phylogeny. In addition to the African savanna elephant reference genome, we generated genome-wide data from 14 proboscidean specimens, of which was from the same savanna elephant individual from which the reference genome was sequenced (SI Appendix, Note 3). Using Illumina paired-end reads, we performed deep shotgun sequencing of the genomes of seven elephants: two forest, two savanna, and two Asian elephants ranging in coverage from 28- to 39-fold (Table 1), and an ~120,000-y-old straight-tusked elephant whose coverage we increased from the previously reported (10) 0.65-fold to ~15-fold. We also generated low- to medium-coverage genomes (0.5-fold to ~sixfold) from four woolly mammoths, one Columbian mammoth, and two American mastodons (Mammuthus americanum). The mastodon diverged from elephantids ~20 to 30 Mya (11) and hence represents an appropriate outgroup for studying Elephantidae evolution. We analyzed these data together with previously published genomes from two woolly mammoths (12) and four Asian elephants (13, 14), as well as low-coverage genomic data from a second straight-tusked elephant (10).

To obtain an overview of the relationships among the genomes, we built phylogenetic trees based on different features of the data. Neighbor-joining trees using pairwise divergence per nucleotide recapitulated previously reported relationships (10, 15) (Fig. 1 and SI Appendix, Fig. S8.1), as did trees based on the presence or absence of interspersed repeats in either a maximum parsimony or maximum likelihood analysis, with the exception of the placement of straight-tusked elephants in the latter (SI Appendix, Fig. S9.8). While straight-tusked elephants were recently found to cluster within the mitochondrial diversity of forest elephants (10) (SI Appendix, Fig. S7.1), we show that the nuclear genomes of these taxa form separate clades in the reconstructed trees (Fig. 1). The two forest elephants in our dataset (one from the Guinean and one from the Congolian forest block, spanning the phylogeographic diversity of L. cyclotis) (Table 1) also comprise a lineage that is distinct from savanna elephants, confirming with complete nuclear genomes that the two African elephants should be classified as distinct taxa. However, our further analyses showed that the average trees do not capture the full complexity of the evolutionary history of elephantid species and in particular obscure major admixture events, which were central features of elephantid evolution.

Table 1. Proboscidean samples analyzed in this study

<table>
<thead>
<tr>
<th>Sample ID</th>
<th>Geographic origin</th>
<th>Date, y before present</th>
<th>Sequencing (source)</th>
<th>No. of mapped reads, million</th>
<th>Average coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>L. cyclotis_A</td>
<td>Central African Republic</td>
<td>Modern</td>
<td>This study (BI)</td>
<td>906</td>
<td>27.78</td>
</tr>
<tr>
<td>L. africana_B</td>
<td>Kenya</td>
<td>Modern</td>
<td>This study (BI)</td>
<td>1,001</td>
<td>30.44</td>
</tr>
<tr>
<td>L. africana_C</td>
<td>South Africa</td>
<td>Modern</td>
<td>This study (BI)</td>
<td>1,114</td>
<td>33.42</td>
</tr>
<tr>
<td>E. maximus_D</td>
<td>Myanmar</td>
<td>Modern</td>
<td>This study (BI)</td>
<td>1,283</td>
<td>38.94</td>
</tr>
<tr>
<td>E. maximus_E</td>
<td>Malaysia (Borneo)</td>
<td>Modern</td>
<td>This study (BI)</td>
<td>1,107</td>
<td>32.20</td>
</tr>
<tr>
<td>L. cyclotis_F</td>
<td>Sierra Leone</td>
<td>Modern</td>
<td>This study (BI)</td>
<td>1,074</td>
<td>32.06</td>
</tr>
<tr>
<td>M. primigenius_G</td>
<td>Taïmyr Peninsula, Russia</td>
<td>~31,500</td>
<td>This study (HMS)</td>
<td>55</td>
<td>0.60</td>
</tr>
<tr>
<td>M. primigenius_H</td>
<td>Alaska, USA</td>
<td>~44,900</td>
<td>This study (HMS)</td>
<td>27</td>
<td>0.49</td>
</tr>
<tr>
<td>M. americanum_J</td>
<td>Alaska, USA</td>
<td>&gt;50,000</td>
<td>This study (IFT, HMS)</td>
<td>399</td>
<td>3.96</td>
</tr>
<tr>
<td>E. maximus_L</td>
<td>India*</td>
<td>Modern</td>
<td>(13)</td>
<td>889</td>
<td>27.02</td>
</tr>
<tr>
<td>E. maximus_M</td>
<td>India*</td>
<td>Modern</td>
<td>(13)</td>
<td>1,014</td>
<td>30.27</td>
</tr>
<tr>
<td>P. antiquus_N</td>
<td>Germany</td>
<td>~120,000</td>
<td>This study (BI, HMS)</td>
<td>1,399</td>
<td>14.64</td>
</tr>
<tr>
<td>P. antiquus_O</td>
<td>Germany</td>
<td>~120,000</td>
<td>This study (BI, HMS)</td>
<td>1,399</td>
<td>14.64</td>
</tr>
<tr>
<td>M. primigenius_P</td>
<td>Oimiyakon, Russia</td>
<td>~44,800</td>
<td>(12)</td>
<td>902</td>
<td>12.77</td>
</tr>
<tr>
<td>M. primigenius_Q</td>
<td>Wrangel Island, Russia</td>
<td>~4,300</td>
<td>(12)</td>
<td>959</td>
<td>19.00</td>
</tr>
<tr>
<td>M. primigenius_S</td>
<td>Yamal Peninsula, Russia</td>
<td>~45,300</td>
<td>This study (IFT, HMS)</td>
<td>132</td>
<td>0.91</td>
</tr>
<tr>
<td>M. columbi_U</td>
<td>Wyoming, USA</td>
<td>~13,400</td>
<td>This study (IFT, HMS)</td>
<td>122</td>
<td>1.53</td>
</tr>
<tr>
<td>Mammuthus_V</td>
<td>Wyoming, USA</td>
<td>~42,400</td>
<td>This study (IFT, HMS)</td>
<td>830</td>
<td>5.86</td>
</tr>
<tr>
<td>M. americanum_X</td>
<td>Gulf of Maine, USA</td>
<td>~13,400</td>
<td>This study (HMS)</td>
<td>71</td>
<td>0.79</td>
</tr>
<tr>
<td>E. maximus_Y</td>
<td>Assam, India</td>
<td>Modern</td>
<td>(13)</td>
<td>1,239</td>
<td>35.90</td>
</tr>
<tr>
<td>E. maximus_Z</td>
<td>Karnataka, India</td>
<td>Modern</td>
<td>(14)</td>
<td>447</td>
<td>14.58</td>
</tr>
</tbody>
</table>

BI, Broad Institute; HMS, Harvard Medical School; IFT, Illumina Fast Track Services.

*Exact geographic origin is unknown.
We integrated the observed signals of gene flow into a single historical model using qpGraph (18), which fits parameters of an admixture model (phylogenetic tree augmented with admixture events) by comparing empirical and predicted f-statistics (16). The admixture graph that most parsimoniously fit the data (Fig. 2A and SI Appendix, Figs. S12.2–S12.4) captured all of the patterns in the individual D-statistics and revealed a more complex history than can be captured by a simple tree-like topology (Fig. 1).

A major surprise that emerged from this analysis is the highly reticulated relationship between straight-tusked elephants and the other species. In contrast to previous work that has shown that straight-tusked elephants are on average more closely related to forest elephants than they are to any other species (10), we found that they do not form a simple clade with forest elephants. The fitted admixture graph revealed three major genetic components for straight-tusked elephants, the largest of which derived from a lineage that is basal to the common ancestor of forest and savanna elephants (Fig. 2A). This finding may help to reconcile the genomic data with the fossil record of elephantids in Africa because species of Palaeoloxodon predominated in the fossil record during most of the Pliocene and Pleistocene and are believed to have given rise to the Eurasian straight-tusked elephant (2, 19).

The remaining genetic contribution to straight-tusked elephants derived from two separate lineages, one related to woolly mammoths and the other related to extant forest elephants (Fig. 2A). Specifically, woolly mammoths, as well as Asian elephants, shared more derived alleles with straight-tusked elephants than expected and the signal was significantly stronger for mammoths than for Asian elephants (Z = 9.25) (Table 2). This pattern is most parsimoniously explained by 6 to 10% admixture from a population related to woolly mammoths into the straight-tusked elephant lineage (Fig. 2A), which could help to resolve an apparent discrepancy. While phylogenetic trees based on genomewide nuclear (Fig. 1) and mtDNA data (10) (SI Appendix, Fig. S7.1) place straight-tusked elephants as closest to forest elephants (due to an additional admixture event described below), morphological criteria have traditionally placed straight-tusked elephants within Elephas (3, 9). The morphological similarity to Asian elephants could be accounted for through hybridization from an ancestral population that split off from the mammoth lineage early in its history, close in time to the common ancestor of Asian elephants and mammoths. This would imply that morphological characters shared between straight-tusked and Asian elephants were present in the common ancestor of Asian elephants and mammoths, and thus became lost from the mammoth lineage. Alternatively, the morphological similarities between straight-tusked elephants and Asian elephants could also be due to homoplasies resulting from convergent evolution, for which there is considerable evidence in the elephantid fossil record (1–3).

Secondly, straight-tusked elephants shared significantly more derived alleles with one of our sequenced forest elephants (L. cyclotis F from the Guinean forest block in West Africa) than with the other (7 ≤ |Z| ≤ 9) (Fig. 2B). The fitted admixture graph indicates that the straight-tusked elephant derives 35 to 39% of its ancestry from a lineage related to the West African forest elephant (L. cyclotis F) (Fig. 2A). This admixture proportion explains the apparent placement of straight-tusked elephants as most closely related to forest elephants in the phylogenetic trees in Fig. 1 and ref. 10. Given the geographic separation and deep divergence between our sampled forest elephants (see below), gene flow from a derived forest elephant lineage into the straight-tusked elephant lineage is plausible and likely occurred in Africa. The intraspecies split time between the West and Central African forest elephants (L. cyclotis A and L. cyclotis F; 609,000 to 463,000 y ago subject to mutation rate uncertainty) (see Fig. 4A) and the approximate date of our sequenced straight-tusked elephants (~120,000 y ago) place upper and lower bounds on the date of the inferred gene flow. This interval, however, overlaps several glacial cycles. In Africa, glacial periods involved drier conditions, contraction of rainforest habitats, and expansion of grassland (20) while interglacial periods involved the opposite. Such ecological factors may have had important consequences for the biota, including facilitating or inhibiting hybridization among related taxa.
true evolutionary history of straight-tusked elephants could have been even more complex; the models reported here are based on transversion polymorphisms only. With sympatric North American woolly mammoths than it does with any of the Eurasian woolly mammoths in our dataset (all |Z|-scores > 9.4) (SI Appendix, Table S11.3). We used an f
ratio test (18) to estimate the Columbian mammoth ancestry proportion to 8.8 to 11.7% (95.4% confidence interval) in Mammuthus V from Wyoming and 4.4 to 8.7% in M. primigenius H from Alaska (SI Appendix, Fig. S11.1 and Table S11.7). These data suggest a north-southcline in the proportion of Columbian admixture, with the Alaskan mammoth having less Columbian ancestry, consistent with the fact that the range of the Columbian mammoth was limited to more southern temperate regions within North America.

Lastly, we tested for evidence of admixture between the ancestors of forest and savanna elephants. Despite their high average pairwise nuclear sequence divergence (0.74%); which is higher than that between Asian elephants and mammoths) (SI Appendix, Table S8.1), the mitochondrial phylogeography of the two African elephant species indicates that hybridization between them must have occurred (22, 23). However, according to D-statistics, we found that the pairs of forest and savanna individuals in our study are mutually symmetrically related. This suggests that little, if any, gene flow has occurred subsequent to the splits of the pairs of sampled elephants from each species (609,000 y ago based on the oldest intraspecific split time estimated for the two forest elephants). Alternatively, gene flow from an unknown ancestral forest elephant lineage into both savanna elephant lineages and in equal proportions (or into the common ancestor of savanna elephants), or vice versa, could have occurred more recently. Hybridization in fact still occurs locally where the two species’ ranges overlap (24–26). Recent work by Mondol et al. (27) shows that gene flow is bidirectional and that hybrids are fertile but appears to have not resulted in detectable introgression of nuclear alleles beyond these hybrid regions. The finding of deep population structure between the two subgroups of forest elephants (see Within-Species Analyses: Diversity, Population Size Change, and Population Substructure and Fig. 4A) and of isolation between forest and savanna elephants has implications for elephant conservation biology. While hybridization occurs between forest and savanna elephants along their current contact zone (24–27), which has long hindered their recognition as distinct species (28), our genome-wide analysis shows that this process has not left detectable traces on the genomes of representative members of the two species across their range. Thus, for conservation purposes, forest elephants and savanna elephants are appropriately viewed as reproductively distinct units, meeting the definition of the Biological Species Concept (29).

Table 2. Additional D-statistics supporting the admixture graph in Fig. 2A

<table>
<thead>
<tr>
<th>D-statistic</th>
<th>D</th>
<th>SE</th>
<th>Z</th>
<th>No. of transversions</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Straight-tusked, forest; Asian, mastodon</td>
<td>0.076</td>
<td>0.004</td>
<td>17.94</td>
<td>371,372</td>
<td>Asian elephants share more alleles with straight-tusked elephants than with African elephants</td>
</tr>
<tr>
<td>Straight-tusked, savanna; Asian, mastodon</td>
<td>0.021</td>
<td>0.004</td>
<td>5.04</td>
<td>336,514</td>
<td>Asian elephants share more alleles with straight-tusked elephants than with African elephants</td>
</tr>
<tr>
<td>Straight-tusked, forest; woolly, mastodon</td>
<td>0.135</td>
<td>0.005</td>
<td>29.69</td>
<td>354,235</td>
<td>Mammoths share more alleles with straight-tusked elephants than with African elephants</td>
</tr>
<tr>
<td>Straight-tusked, savanna; woolly, mastodon</td>
<td>0.054</td>
<td>0.004</td>
<td>12.32</td>
<td>335,375</td>
<td>Mammoths share more alleles with straight-tusked elephants than with African elephants</td>
</tr>
<tr>
<td>Woolly, Asian; straight-tusked, mastodon</td>
<td>0.04</td>
<td>0.004</td>
<td>9.25</td>
<td>275,766</td>
<td>Straight-tusked elephants share more alleles with mammoths than with Asian elephants</td>
</tr>
</tbody>
</table>
15). However, we caution that the elephantid mutation rate is highly uncertain (12) and, when more accurate estimates become available in the future, all absolute time estimates should be rescaled (but relative estimates should remain unchanged).

First, we applied approximate Bayesian computation (ABC) to fit demographic models based on a set of summary statistics consisting of the allelic states of pairs of adjacent variable sites (30) in alignments of three elephantid sequences and the mastodon, as well as estimates of pairwise divergence and D-statistics (SI Appendix, Note 16). Consistent with our pairwise sequential Markovian coalescent (PSMC) results (shown below), inferred ancestral effective population sizes (Fig. 3) were largest for the ancestors of forest, savanna, and straight-tusked elephants, followed by the ancestors of Asian elephants and woolly/Columbian mammoths, and smallest for the common ancestral population of all elephants, although all confidence intervals (CIs) were overlapping (CI, respectively: 37,000 to 233,000; 10,000 to 130,000; and 7,000 to 78,000).

Forest and savanna elephants are inferred to have split from each other ~5 to 2 Mya, soon after their common ancestor split from the straight-tusked elephant lineage. The split between Columbian and woolly mammoths is inferred to have occurred 1.5 to 0.7 Mya, consistent with some, but not all, paleontological estimates (7, 31). Asian elephants and mammoths are estimated to have split at about the same time as the split between Loxodonta and straight-tusked elephants while the initial split within the Elephantidae is inferred to have occurred ~10 to 5 Mya, in good agreement with the divergence time of Loxodonta and Asian elephants/mammoths inferred from the fossil record (15) (9 to 4.2 Mya). All elephantids are estimated to have split from the mastodon at ~28 to 10 Mya, with the upper end of this range in line with evidence from the fossil record (19) (28 to 24 Mya).

The highest migration rate is inferred between forest and straight-tusked elephants (CI: 0.49 × 10^{-6} to 1.49 × 10^{-6}; proportion of migrants per generation), consistent with the largest admixture proportion estimated by the admixture graph and f_{x}-ratio tests (Fig. 2A and SI Appendix, Table S11.8). These are followed by the migration rates between straight-tusked elephants and woolly mammoths (1.84 × 10^{-7} to 6.44 × 10^{-7}), and between straight-tusked and Asian elephants (1.32 × 10^{-7} to 5.71 × 10^{-8}), which is again in agreement with the findings from D-statistics and the admixture graph.

Second, we used a coalescent hidden Markov model (32) (CoalHMM) to infer split times and ancestral effective population sizes.
sizes for selected trios of elephantid species based on incomplete lineage sorting (ILS) (SI Appendix, Note 17). ILS is reflected in regions of the genome where taxa that are not most closely related in the species tree cluster together (15, 33, 34). Here, we also incor-porated data from chromosome X to test for evidence of sex-biased demography. These analyses support the evidence from ABC analysis that the autosomal \( N_e \) for the ancestor of forest and savanna elephants (mean: 165,000 individuals) is higher than that for the ancestor of Asian elephants and woolly mammoths (mean: 72,000) (Fig. 3), and for the common ancestor of all elephantids (48,000 to 53,000, range of means obtained from analyses of different elephantid trios). Forest and savanna elephants are inferred to have split at ~2 Mya, Asian elephants and woolly mammoths at 2.5 Mya, and all elephantids at 5.6 to 5 Mya (Fig. 3). These dates overlap with the lower end of the ranges obtained from the ABC analysis, with the younger average dates from the CoalHMM model likely due to the absence of migration in the model (see also below).

For all analyzed species trios, the observed X-to-autosome ratio of \( N_e \) was lower than 3/4 (the baseline value for a simple demography), even though a higher ratio might be expected considering a higher variance in male reproductive success in elephants (35, 36). Potential factors that could explain this discrepancy include linked selection (37) on chromosome X or male-biased gene flow (38).

An examination of the ILS patterns revealed that, in the forest, straight-tusked, and Asian elephant trio, a higher proportion of regions clustered together straight-tusked and Asian elephants (18.8 to 20.5%) rather than forest and Asian elephants (15.3 to 16.0%) (SI Appendix, Figs. S17.15–S17.18), consistent with the gene flow indicated in the best-fit admixture graph (Fig. 2.4 and Table 2). We did not observe a substantial ILS asymmetry in the trio of Asian elephants, woolly mammoths, and straight-tusked elephants (SI Appendix, Figs. S17.13 and S17.14), but we believe this is still compatible with the findings from the admixture graph analysis, given the proportion of woolly mammoth-related ances-try in straight-tusked elephants, and its source splitting off relatively close to the common ancestor of Asian elephants and woolly mammoths (Fig. 2.4).

Finally, we applied CoalHMM for pairs of elephantid species under isolation-and-migration (IM) models, allowing for the possibility of continuing gene flow after initial population separation (39) (SI Appendix, Note 18). Our autosomal IM CoalHMM analysis strongly supports the presence of migration after initial separation for all interspecies pairs (Fig. 3 and SI Appendix, Fig. S18.1). Consistent with our other analyses, the highest gene flow rates were estimated between the forest and straight-tusked elephant lineages (CE: 1.00 \( \times 10^{-6} \) to 1.49 \( \times 10^{-6} \)). Gene flow between the ancestors of forest and savanna elephants is inferred to have occurred from their split ~5.3 Mya (CE: 5.6 to 2.6 Mya) until 1.3 Mya (CE: 3.0 to 1.2 Mya for pairs including \( L. \ cyclotis \_A \) and 1.4 to 0.1 Mya for pairs including \( L. \ cyclotis \_F \) although the \( D \)-statistics and admixture graph analyses did not provide any evidence of recent gene flow between the two species. Overall, split times were quite similar to those estimated via ABC while estimates of ancestral \( N_e \) were mostly lower than those obtained from the ILS CoalHMM analysis but similar (except with tighter confidence intervals) to those from ABC (Fig. 3).

**Within-Species Analyses: Diversity, Population Size Change, and Population Substructure.** Estimates of genetic diversity for the high-coverage genomes \((n = 13)\) indicated, consistent with previous reports, that African forest elephants harbor the highest levels of heterozy-gosity (0.00285 to 0.00364) (Fig. 4B) and sequence divergence (SI Appendix, Table S8.1) among extant and extinct elephantids (15, 40–42). Mammoths, straight-tusked elephants, and Asian elephants displayed intermediate levels of heterozygosity (0.00093 to 0.00167) (Fig. 4B), except for \( E. \ maximus \_E \) from Malaysian Borneo, which had extremely low heterozygosity (0.00032). Savanna elephants exhibited the lowest heterozygosity among all elephantids (0.00085 to 0.00088) (Fig. 4B).

To reconstruct elephantid population size changes over time, we used the PSMC (43) (SI Appendix, Note 14). The two forest elephants had similar population size histories before ~370,000 y ago but very different ones thereafter. Current effective population size \((N_e)\) in \( L. \ cyclotis \_F \) (from the smaller Guinean forest block in West Africa) was ~fourfold lower than in \( L. \ cyclotis \_A \) (from the larger Congolian forest block in Central Africa) (Fig. 4C), in line with the ~21% lower het-erzygosity in the former. The two savanna elephants had lower \( N_e \) relative to forest elephants for hundreds of thousands of years (Fig. 4D), potentially reflecting ecological competition from the African elephant \( Palaeoloxodon recki \) (including \( Palaeoloxodon iolensis \)) that dominated the African savannas until the Late Pleistocene (2, 19), or the high levels of male–male competition documented in this species.

Early in its history (>1 Mya), the straight-tusked elephant had a population size trajectory similar to that of forest and savanna elephants (Fig. 4C), including a period of population expansion ~2 Mya followed by decline. This observation may be explained by evidence that these species share deep ancestry (Fig. 2.4). Asian elephants are inferred to have gone through phases of population growth, succeeded by decline ~120,000 y ago, resulting in a current \( N_e \) estimated to be about half that of savanna elephants (Fig. 4E). The population sizes of the two woolly mammoths are inferred to have been similar before their split, but, subsequently, the ancestors of the Wrangel Island mammoth experienced a severe bottleneck (Fig. 4F), which led to an ~20% drop in heterozygosity, as shown earlier in the study that reported the Wrangel and mainland Siberian mammoth genomes (12).

We estimated split times of elephantids within species using the \( F(A|B) \) statistic (17), which measures the fraction of heterozygous positions discovered in one individual that are de- rived in a randomly sampled chromosome from an individual of a second population of the same species (SI Appendix, Note 15). This fraction is expected to decrease as a function of population separation time (reflecting the fact that, for an older split, a greater proportion of discovered mutations will have occurred after population divergence), with the exact form of the decay depending on the demographic history of the first individual, which we can infer using PSMC. The oldest in-traspecific split within elephantid taxa was estimated between the two forest elephants (\( L. \ cyclotis \_A \) and \( L. \ cyclotis \_F \); 609,000 to 463,000 y ago) (Fig. 4A). This is consistent with a hypothesis of deep population structure with limited gene flow, as well as with the high ancestral \( N_e \) among forest elephants (15). By contrast, the two savanna elephants were estimated to have split from each other only 38,000 to 30,000 y ago, in line with their nearly identical \( N_e \) curves (Fig. 4D), as well as with a previous hypothesis for a relatively recent founder event (40, 41), and with high levels of male dispersal documented in this species (44). Among Asian elephants, split times were oldest between the Bornean \( E. \ maximus \_E \) and other individuals (190,000 to 103,000 y ago) (Fig. 4A), consistent with the uniqueness of the mitochondrial DNA haplogroup of elephants in Malaysian Borneo (45). The Asian elephant from Myanmar (\( E. \ maximus \_D \)) exhibited higher heterozygosity than other Asian elephants and intermediate split times with elephants from India (43,000 to 24,000 y ago), compatible with a hypothesized sec- ondary admixture of diverged populations that may have occurred in this part of Southeast Asia, as suggested by mitochondrial DNA (46). Within \( Mammuthus \), the inferred interspecific split between Columbian mammoths and Eurasian woolly mammoths 712,000 to 423,000 y ago, was overlapping but mostly lower than that obtained from the ABC analysis described above (1.5 to 0.7 Mya), but still far older than that between the two Eurasian
woolly mammoths ($M. \text{primigenius}_P$ and $M. \text{primigenius}_Q$; 225,000 to 112,000 y ago) (Fig. 4A).

**Conclusion**

Our genomic analyses of present-day and extinct elephantids revealed a history of multiple major interspecies admixture events. Evidence for gene flow among closely related mammalian species is not unprecedented. Examples include cases of unidirectional gene flow [e.g., from polar bears into brown bears (47), similar to the Columbian mammoth gene flow into woolly mammoths observed in our study]; emergence of admixed species [e.g., North American wolves with ancestry from coyotes and gray wolves (48), similar to the straight-tusked elephants in our study]; different extents of gene flow [e.g., between gray wolves and Eurasian/African golden jackals (49), and between bonobos and central/eastern chimpanzees (50), as in the case of straight-tusked elephants and west African forest elephants/woolly mammoths in our study]; extended periods of gene flow during the initial diversification of species [e.g., between eastern and western gorillas (39), Sumatran and Bornean orangutans (39), and the ancestors of humans and chimpanzees (39, 51), like those inferred from most pairwise species comparisons in our study]; and adaptive introgression [e.g., in the great cats of the genus *Panthera* (52)], which could have played an important role in the...
evolution of elephantids as well. Our results in elephantids thus add to the growing weight of evidence in favor of the view that capacity for hybridization is the norm rather than the exception in many mammalian species over a time scale of millions of years. Three different outcomes followed interspecies hybridization among elephantids: emergence of a species with three ancestral genetic components (straight-tusked elephants); the continued isolation of species and lack of genome-wide introgression even after recurrent hybridization (forest and savanna elephants); or a modest degree of introgression (Columbian and North American woolly mammoths). An important priority for future work should be to explore whether admixture was not only an important phenomenon in the demographic history of the elephantids, but also played a biologically important role in their evolution, facilitating adaptation after migration into new habitats, or in the face of fluctuating climatic conditions and resulting ecological shifts (53).

Materials and Methods

Detailed information on the samples and methods is provided in SI Appendix, including de novo genome assembly, mitochondrial phylogeny, and analysis of repetitive elements.

Genome Sequencing. Illumina libraries were prepared from genomic DNA of six modern elephants and sequenced at the Broad Institute. Illumina genomic libraries were also prepared for seven ancient proboscideans, following established methods (54, 55), and were sequenced together with previously generated libraries (10) at the Broad Institute, Harvard Medical School, and Illumina Fast Track Services.

Data Processing. Paired-end reads were trimmed and merged (ancient data) or trimmed only (modern data) with SeqPrep v.1.1 (https://github.com/jojohn/SeqPrep), aligned against the African savanna elephant reference genome (Loxodonta) with Burrows–Wheeler Aligner (BWA) (56), using parameters optimized for ancient DNA or default parameters, and converted to bam format with SAMTools (57) v.2.1.19. Duplicate reads were discarded using a custom python script or the SAMTools “rmdup” command. Previously published genomes for two woolly mammoths (12), two straight-tusked elephants (10), and four Asian elephants (13, 14) were also reprocessed and included in the dataset. Applied filters included base quality threshold of 30, mapping quality of ≥30 or 37, and mappability filters as described in SI Appendix, Note 6.

Sequence Divergence. Pseudohaplid sequences of chromosomes 1 to 27 were generated for each elephantid with single randomly sampled alleles per site to eliminate reference alignment biases (as explained in detail in SI Appendix, Note 6). Pairwise sequence divergence was estimated from alignments ranging in size from 45 Mbp to 1,609 Mbp, based on all substitutions or only transversions. A neighbor-joining tree with support values from 100 bootstrap drawings in Figs. 1 and 3. Deep sequencing of the straight-tusked elephant sample (P. antiquus) was funded by National Human Genome Research Institute Grant US4 H003067-08. A.L.R. was supported by the US Fish and Wildlife Service African Elephant Conservation Fund. B.L.A. and M.R. were funded through a Natural Sciences and Engineering Research Council Consolidator Grant 310763 GeneFlow. H.P. was supported by Euro- pean Research Council Consolidator Grant 310763 GeneFlow. H.P. was funded through a Natural Sciences and Engineering Research Council of Canada Discovery Grant R8MAC-10539150 and the Canada Research Chairs program. D.R. was funded by NSF (HOMINID) Grant BCS-1032255 and NIH (National Institute of General Medical Sciences) Grant GM100233 and is an Investigator of the Howard Hughes Medical Institute.

These analyses were based on transversion SNPs (called from randomly sampled alleles per site) to alleviate biases from residual postmortem damage in Ctg pangenomes and recurrent mutations.

Interspecies Demographic Inference. Three modeling approaches were implemented to infer species ancestral effective population sizes, split migration rates, and introgression rates: (i) coalescent simulations with approximate Bayesian computation (ABC), (ii) incomplete lineage sorting (ILS) CoaHMM, and (iii) isolation with migration CoaHMM models (IM CoaHMM). For the first approach, demographic scenarios of three elephantid lineages and the mas- todon (outgroup) were modeled in scm (59), using prior distributions for all demographic parameters (Tapson et al. 2011) in R (R Development Core Team 2011) was used to fit parameters based on the following summary statistics: allelic states of pairs of adjacent variable sites (30), D-statistics, and pairwise divergence per base pair. For the second approach, CoaHMM isolation models (32) (without gene flow) were used to estimate proportions of ILS alone among lineages of three elephantid lineages, and to infer in parallel unbiased estimates of effective population size and split time parameters, as described in ref. 32. For the third approach, the isolation and the isolation-with-initial-migration (39) CoaHMM models were fitted to pairwise interspecies sequence alignments. The Akaike information criterion (AIC) was used to choose the preferred model and maximum likelihood estimates of ancestral effective population sizes, split times, start and end of migration period, and migration rates were obtained. Parameter estimates were con- verted to years, assuming a mutation rate of 0.406 × 10−9 per base per year (as calculated in SI Appendix, Note 16; but we caution there is substantial uncertainty in this estimate) and a generation interval of 31 y (as in ref. 15). For more details, see SI Appendix, Notes 16–18.

Within-Species Demographic Analyses. Individual heterozygosity was estimated for high-coverage genome sequences with mRho (61) v.2.7. The PSMC (43) was used to reconstruct changes in effective population size through time by examining patterns of heterozygosity across the diploid genome of single individuals. Within-species population split times were estimated us- ing the F(A)8 statistic (17) as implemented in the software POPSTATS, using transversion SNPs only, and the reconstructed PSMC to infer the decay of this statistic as a function of population split time. Time estimates were rescaled assuming the mutation rate and generation time described above.

ACKNOWLEDGMENTS. We thank Bruce Upchurch and Teri Hermann (Woodland Park Zoological Gardens), Melissa Dickson (Dickerson Park Zoo), David Shepherdson (Oregon Zoo), and Nicholas J. Georgiades (University of Washing- ton) for providing elephant samples; Bernard Buiges (International Mammoth Committee), Pamela Groves (Institute of Arctic Biology), Daniel Fisher (University of Michigan), Mark Clementz (University of Wyoming), and Paul Matheus and Dale Guthrie (University of Alaska, Fairbanks) for provide- ing proboscidean samples; Jacobs Enk (McMaster University and MyCroarray) for assisting in laboratory analyses; Pontus Skoglund (Harvard Medical School) for providing software for population genetic analyses; Veronika (University of Colorado) for quantifying and sending elephant samples to the Broad Institute; and Karol Schauer for the proboscidean drawings in Figs. 1 and 3. Deep sequencing of the straight-tusked elephant sample (P. antiquus) was funded by National Human Genome Research Institute Grant US4 H003067-08. A.L.R. was supported by the US Fish and Wildlife Service African Elephant Conservation Fund. B.L.A. and M.R. were funded by Wellcome Trust Grants WT089051 and WT108749/2/15/Z2 and by the European Molecular Biology Laboratory. M.H. was supported by Euro- pean Research Council Consolidator Grant 310763 GeneFlow. H.P. was funded through a Natural Sciences and Engineering Research Council of Canada Discovery Grant R8MAC-10539150 and the Canada Research Chairs program. D.R. was funded by NSF (HOMINID) Grant BCS-1032255 and NIH (National Institute of General Medical Sciences) Grant GM100233 and is an Investigator of the Howard Hughes Medical Institute.
Appendix B

Particle Filtering paper
Inferring Population Genetic Parameters: Particle Filtering, HMM, Ripley’s K-Function or Runs of Homozygosity?

Svend V. Nielsen, Simon Simonsen, and Asger Hobolth
Aarhus University, Bioinformatics Research Centre, C.F. Møllers Allé 8, 8000 Aarhus C, Denmark
svn@cs.au.dk

Abstract. The coalescent with recombination is widely accepted as the key model to understand genetic diversity within a species. Many theoretical properties of the model are well understood, but formulating and implementing efficient inference methods remains a challenge. A major breakthrough has been to approximate the coalescent with recombination by a Markov chain along the sequences. Here we describe a new tool, RECJumper, for inference in the Markov approximated coalescent model. Previous methods are often based on a discretisation of the tree space and hidden Markov models. We avoid the discretisation by using particle filtering, and compare several proposal distributions. We also investigate runs of homozygosity, and introduce a new summary statistics from spatial statistics: Ripley’s K-function. We find that (i) choosing an appropriate proposal distribution is crucial to obtain satisfactory behaviour in particle filtering, (ii) tree space discretisation in HMM-methodology is non-trivial and the choice can influence the results, and (iii) Ripley’s K-function is a much more informative statistics than runs of homozygosity for recombination rate estimation.

1 Introduction

Consider the black piecewise constant function in Fig. 1(a). The function provides the coalescence times \( x = \{x_i\}_{1 \leq i \leq N} \) for two genomic sequences (e.g. from sequencing a single diploid individual) in positions indexed by \( i, i = 1, \ldots, N \). The coalescence times are unknown (hidden), and we instead observe the binary mutation pattern \( y = \{y_i\}_{1 \leq i \leq N} \). Sites with large coalescence times are more likely to experience a mutation than sites with short coalescence times; a simple mutation model assigns probability

\[
p(y_i = 1|x_i = t) = 1 - \exp(-\theta t)
\]

for being heterozygote (and \( \exp(-\theta t) \) for being homozygote). Recombination events are responsible for the jumps in the coalescence path, and in general the dependence structure between coalescence times is very complex [8]. Fortunately the Markov assumption is a good approximation [13]. We observe that
the Markov approximated coalescent with recombination and mutation is a state space model where \( \{x_i\}_{1 \leq i \leq N} \) is the latent process and \( \{y_i\}_{1 \leq i \leq N} \) is the measurement data. State space models have been analysed in financial econometrics for more than a decade (e.g. [2]), and an efficient class of algorithms is particle filtering.

Particle filtering is an importance sampling method. Recall that the main idea in importance sampling is to simulate from a distribution \( q(x) \) (possibly depending on \( y \)), assign each simulation a weight \( w(x) = f(x)p(x)/q(x) \), and approximate the mean, \( E[f(X)] \), by the weighted sample average. For example if the quantity of interest is the likelihood \( p(y) \) we have

\[
p(y) = \int x p(x, y) dx = \int x \frac{p(x)p(y|x)}{q(x)} q(x) dx = \int w(x)q(x)dx \approx \frac{1}{n} \sum_{j=1}^{n} w(z_j),
\]

where \( z_j \) is a sample from \( q \). The challenge in particle filtering is to formulate proposal distributions that are easy to simulate and close to the optimal proposal distribution \( p(x|y) \).

Figure 1(a) shows three coalescence paths from a naive but very fast proposal distribution, and Fig. 1(b) shows their corresponding weights \( w(x) \). The distribution of weights in Fig. 1(b) shows that the price for the fast proposal distribution is very high: Almost all coalescence paths have a very low weight and are therefore not useful for subsequent analysis. This unfortunate situation is avoided in Figs. 1(c) and 1(d) where we have used the proposal function from RECJumper. Very many RECJumper coalescent paths have a reasonable high weight.

Particle filtering is far from the only way to infer population genetic parameters. Another popular method is to discretise the state space so that only a discrete and finite number of coalescent times are possible. Most known is perhaps the PSMC [11] that uses the Sequential Markov Coalescent (SMC) model [13] on two DNA sequences. ARGweaver [16] and MSMC [17] are extensions of the PSMC to more than two sequences. The CoalHMM [12] is an implementation with the Simonsen Churchill model [18] for two sequences. The advantage of a discrete and finite, latent state space is that the classical HMM algorithms apply. The challenge is to formulate a reasonable discretisation procedure. How many bins and where to place them?

In Fig. 2 we show the results of a simulation study. We simulated 100 data sets from the model with a recombination rate of \( \rho = 0.1 \), a fixed mutation rate \( \theta = 0.1 \) in a genomic segment of size \( N = 20,000 \) base pairs. We then estimated the recombination rate using particle filtering and an HMM with a state space of size 20. We observe that they yield similar results. This is expected since they are both approximations to the same integral.

In Fig. 2 we have also included two more parameter estimation procedures. The first is based on runs of homozygosity. Harris and Nielsen [9] use this statistics for demographic inference, but we observe that actually a lot of power is lost by summarizing the data this way. The second is based on Ripley’s \( K \)-function. In spatial statistics the runs of homozygosity (often called the nearest neighbour
function) is only seldom used. Ripley’s $K$-function is a much more popular summary of a point pattern. Ripley’s $K$-function gives the mean number of points $K(r)$ within a distance $r$ from a typical point. It is straightforward to determine an empirical estimate of Ripley’s $K$-function. In Fig. 2 we observe that Ripley’s $K$-function is a very useful statistics. We emphasize that this observation also has important consequences for simulation-based procedures such as Approximate Bayesian Computation [1] where the data is summarized in terms of simple summary statistics.

Our paper is organized as follows. In Sect. 2 we describe in detail the state space model. In particular we provide the probability of a new coalescent height,
Fig. 2. The four methods have each produced an estimate of the recombination rate, $\rho$, based on 100 simulations of size 20,000 bases from the state space model. This plot visualises their distributions by sorting all the estimates. The $K$-function produces better estimates of $\rho$ than runs of homozygozities whereas HMM and particle filtering performs even better. The ‘steps’ of the particle filter estimates are due to the grid of driver values of $\rho$.

and the density for a new height conditional on the old height. We work with the most general recombination model for two loci and two sequences. In Sect. 3 we describe particle filtering and RECjumper, and in Sect. 4 we consider the HMM framework. Section 5 is concerned with Ripley’s $K$-function and runs of homozygosity. Our paper ends with a general discussion of the various methods.

2 The State Space Model

A state space model is fully specified by its transition probabilities $p(x_i|x_{i-1})$ and its emission probabilities $p(y_i|x_i)$. Therefore, only the joint distribution of the coalescence times at two adjacent genomic positions is needed in order to specify the model. Below we let $s = x_{i-1}$ and $t = x_i$ denote the left and right coalescence times. The coalescence times are determined by the Simonsen-Churchill model [18] which is a continuous time Markov chain. The states are the ancestry of two pairs of loci from two sequences; see Fig. 3. The model is given a careful treatment in the textbooks [6,19], and we only use the following theorem which provides the transition probabilities.

**Theorem 1.** Let $A$ denote the $8 \times 8$ rate matrix for the states in the Simonsen-Churchill model (see Fig. 3(c)). The conditional probability of no change from the left to the right tree is

$$P(T = s|S = s) = e^{s[e^A s]}_{11}, \tag{2}$$
and the conditional density $\pi(t|s)$ of $T$ given $S = s$ and given $T \neq S$ is

$$
\pi(t|s) = \begin{cases} 
  e^{-(s-t)} \left[ e^{A_{11}} + \frac{e^{A_{12}-e^{A_s}}}{e^{A_s}} \right] & \text{for } t < s, \\
  e^{-(t-s)} \left[ e^{A_{22}} + \frac{e^{A_{23}-e^{A_s}}}{e^{A_s}} \right] & \text{for } t > s.
\end{cases}
$$

(3)

Proof. See Lemma 2 of Hobolth and Jensen [10].

A simplification of the above model, called the SMC model has a simpler structure and will also be used in the particle filter. In this model we have (see Hobolth and Jensen [10], p. 52, bottom right)

$$
P^{SMC}(T = s | S = s) = e^{-\rho s}
$$

(4)

and

$$
\pi^{SMC}(t|s) = \begin{cases} 
  \frac{\rho(e^{-\rho t} - e^{-t})}{(1-\rho)(1-e^{-\rho s})} & \text{for } t < s \\
  \frac{\rho e^{-(t-s)}(e^{-\rho s} - e^{-s})}{(1-\rho)(1-e^{-\rho s})} & \text{for } t > s.
\end{cases}
$$

(5)

For more discussions of sequential Markov chains for two loci, two sequences we refer to Wilton et. al. [21].

3 Particle Filtering

Particle filtering is a statistical method that can be used to improve a specific type of importance sampler [4]. The goal is to simulate from a distribution $p(\cdot)$ such that a low variance estimate of $E_{p}\{f(X)\}$ can be constructed. This is achieved by simulating particles $z_1, \ldots, z_n$ from any distribution, $q$, called the proposal distribution, which satisfies that every possible sample under $p$ is also possible under $q$. The particles are each assigned a weight $w(z) = f(z)p(z)/q(z)$. In this study $f(x) = p(y|x)$ such that the weighted particles

$$(z_1, w(z_1)), \ldots, (z_n, w(z_n))$$

constitute a sample from $p(x|y)$ in the sense that $p(y) = E_{p}\{f(Z)\} \approx \frac{1}{n} \sum_{j=1}^{n} w(z_j)$.

Having $n$ weighted particles is not as powerful as having $n$ regular samples because some particles may be insignificant due to a very low weight. This problem is called sample degeneracy. Furthermore, all simulated particles could, in principle, be made useless by a future particle that has a much higher weight. Under the assumption that no such future particle exists, we will interpret the approximation to the Effective Sample Size

$$
\text{ESS}(w_1, \ldots, w_n) = \frac{\left(\sum_{j=1}^{n} w_j\right)^2}{\sum_{j=1}^{n} w_j^2}
$$
Fig. 3. (a) Realisation of the continuous time Markov Chain where no recombination between the two sites occur. In the present both pairs of loci are linked (—) yet not coalesced with each other (●) meaning that the chain is positioned in state 1. At time point \( s \) the chain jumps directly from state 1 to state 8 where both loci are coalesced (×).

(b) A recombination occurs which unlinks the two loci. They subsequently find different coalescence times.

(c) All possible transitions are shown with their corresponding transition rates. The corresponding rate matrix is denoted \( \Lambda \).

as the actual number of samples from \( p \). It is evident that the choice of \( q(x) \) affects the weights and thereby the ESS. In the case \( q \propto p(x, y) \) all the weights will be constant and ESS will be the highest attainable value, \( n \). This suggests that choosing \( q \) close to \( p(x|y) \) will lead to a high ESS.

An importance sampler is susceptible to particle filtering methods if \( p \) and \( q \) can be decomposed as follows

\[
P(x_{1:N}, y_{1:N}) = p(x_1)p(x_2|x_1) \cdots p(x_N|x_{N-1})p(y_1|x_1) \cdots p(y_N|x_N) \quad (6)
\]

\[
q(x_{1:N}|y_{1:N}) = q(x_1|y_{1:N})q(x_2|x_1,y_{1:N}) \cdots q(x_N|x_{N-1},y_{1:N}). \quad (7)
\]

The natural way to simulate from \( q \) is to simulate \( x_1 \) from \( q(x_1|y_{1:N}) \), then \( x_2 \) from \( q(x_2|x_1,y_{1:N}) \) and so forth. If done this way, conditions (6) and (7) allow calculation of preliminary weights after \( i \) steps

\[
w(x_{1:i}) = \frac{p(x_1)p(x_2|x_1) \cdots p(x_i|x_{i-1})p(y_1|x_1)p(y_2|x_2) \cdots p(y_i|x_i)}{q(x_1|y_{1:N})q(x_2|x_1,y_{1:N}) \cdots q(x_i|x_{i-1},y_{1:N})}.
\]

Preliminary weights make it possible to gauge the final weight of a particle before the particle is fully produced. If a particle turns out to yield low preliminary weights, we would like to discard it so that we do not waste computing power on a particle that will most likely be insignificant. In addition we duplicate the preliminarily high-weighted particles. This is done through resampling which removes the problem of sample degeneracy completely but introduces...
sample impoverishment. Sample impoverishment is the dependencies between the particles and it consists of two issues: (i) A particle having a low preliminarily weight could recover as more coordinates are simulated. Removing such a comeback particle through resampling will give insufficient diversity within the sample, (ii) The first coordinates of the particles will eventually converge to one value if there are enough resamples. The last coordinates of the particles will not have this issue because they are only resampled a few times.

Issue (ii) is normally solved using smoothing and in particular the smoothing algorithm called Forward-Backward Recursions. Issue (i) is a more fundamental problem and its remedies are often costly in terms of computations. The simplest remedies are increasing the number of particles, choosing a good proposal distribution and fine-tuning the positions of resampling. The optimal resampling positions depend both on the target distribution and proposal distribution, but will not be explored in this study.

One of the simplest proposal distributions in particle filtering is the prior as proposed in the early literature \[ q(x_i|x_{i-1}, y_{1:N}) = p(x_i | x_{i-1}). \]

In this paper we use the distribution \[ p^\text{SMC}(x_i|x_{i-1}) \] specified in (4) and (5). The advantage is that it is fast and easy to simulate. On the other hand it does not use the data \( y \) so we expect a lot of particles with low weight. Some realisations from this proposal are shown in the upper panel of Fig. 1.

We formulated a more informed choice of proposal distribution

\[ q(x_i|y_{1:N}, x_{i-1}) = p^\text{SMC}(x_i|x_{i-1}, y_{i:(i+h)}), \]

for some lag \( h \). Our desire was to simulate from the distribution \( p^\text{SMC}(x_i | x_{i-1}, y_{i:(i+h)}) \). Its distribution is fully specified by the emission probabilities in (1), transition probabilities in (4) and (5), and the state space model assumption. It was, however, not computationally feasible without making some approximations. Therefore, we increased the forgetfulness of the latent Markov chain and we substituted a binomial distribution with a Poisson distribution. This reduced model is denoted \( p^\text{SMC} \). The algorithm first simulates the genomic distance \( d \) to the next recombination from \( p^\text{SMC}(d|x_{i-1}, y_{i:(i+h)}) \) and then draws \( x_i \) from \( p^\text{SMC}(x_i|x_{i-1}, y_{i:(i+h)}, d) \). The effect of the forgetfulness assumption seems to be that fewer extreme values are simulated.

When \( x_i \) is simulated, the algorithm forgets the previous \( d \) and simulates a new \( d \) to generate \( x_{i+1} \). A natural extension of this procedure is to use the simulated \( d \) by setting \( x_i = x_{i+1} = \cdots = x_{i+d-1} \) and then continue the algorithm at \( x_{i+d} \). This speeds up the algorithm yet only decreases the accuracy slightly. We call this faster version RECJumper. Similarly the prior can also be made faster by simulating the next recombination event and jumping to its position. This faster version will simply be called the prior proposal.

We compared the two proposal distributions by looking at how the ratio between ESS and time consumption depends on sequence length in Fig. 4. The proposals were informed of the true values \( (\rho, \theta) = (0.1, 0.02) \).

RECJumper is
superior for sequences that are longer than 400 base pairs when \( h \) is big. For large \( h \) RECjumper simulates particles more slowly but closer to the true distribution, \( p(x|y) \).

If the distribution \( p \) is parametrised in terms of a parameter, \( \rho \), it makes sense to talk about the likelihood. Remembering that the Particle Filter produces weighted samples \((z_j, w(z_j))_{i=1,...,n}\) from \( p_{\rho_0} \), enough samples will justify the importance sampling approximation to the likelihood function

\[
L(\rho) \approx \sum_{j=1}^{n} w(z_j) \frac{p(\rho, z_j)}{p_{\rho_0}(z_j)}.
\]

(8)

In Fig. 2 we show 100 estimates using this method.

![Fig. 2. The measure of samples per seconds is ESS divided per time. Proposals from prior and RECJumper with different lags are shown. For sequences of length 450 and above, the prior is inferior to the more informed versions of RECJumper.](image)

4 Hidden Markov Model

When there is only a finite number of states in a state space model, it becomes a Hidden Markov Model. In this framework powerful algorithms exists. The Forward (or Backward) algorithm calculates the likelihood exactly and quickly [5], and hidden paths are also simulated quickly [3].

The established coalescent HMM methods obtain a finite number of states by discretising the state space model. Discretisation is done by dividing the time axis into a number of intervals and letting the hidden states be the intervals in which the coalescence time falls. Unfortunately, discretising a state space model does not preserve the dependence structure of a state space model. It does not follow that the hidden states of the discretised state space model form a Markov chain nor that the observed data are independent conditioned on the hidden states.
Therefore, in addition to the usual loss of power when discretising, discretisation introduces a bias from the intended model. These disadvantages are alleviated by applying a finer discretisation at the cost of more computations. Nevertheless, the choice of number and shape of intervals in the discretisation influences the results as demonstrated in Fig. 5.

![Graph showing the relationship between the maximum likelihood and the number of intervals for a HMM-based coalescence model. The same graph is plotted on both plots, but the axis are different. The HMM converges towards an optimum as the number of intervals increases. However, it requires 200+ states to make the number of intervals insignificant in a comparison of maximum likelihood values.](image)

**Fig. 5.** A typical relationship between the maximum likelihood and the number of intervals for a HMM-based coalescence model. The same graph is plotted on both plots but the axis are different. The HMM converges towards an optimum as the number of intervals increases. However, it requires 200+ states to make the number of intervals insignificant in a comparison of maximum likelihood values.

5 **Ripley's $K$-Function**

A popular summary function in point processes theory is Ripley’s $K$-function. Let $\theta$ be the intensity of events. In our case an event is a mutation, and $\theta$ is the mutation rate per base pair. The mutation pattern can be seen as a point process by letting the points be the indices of mutations. On this point process Ripley’s $K$ function is

$$K(r) = \frac{1}{\theta} E \left[ \# \{ s : |s - t| \leq r, Y_s = 1 \} \mid Y_t = 1 \right]$$

or, equivalently, the relative number of mutations at most distance $r$ away from a position with a mutation. The discreteness of our point process allows us to make sense of the ‘derivative’ of $K(r)$

$$\kappa(r) = \frac{1}{\theta} E \left[ \# \{ s : |t - s| = r, Y_s = 1 \} \mid Y_t = 1 \right]$$

which we will use in the following.

Another descriptive summary in spatial statistics is the nearest neighbour function. The nearest neighbour function is the probability distribution of the distance from a typical point to the nearest neighbouring point. The nearest neighbour function is less popular than Ripley’s $K$-function in spatial statistics because it has
less power to discriminate between point pattern models [20]. However, in population genetics the nearest neighbour function (or ‘runs of homozygosity’ or ‘distance between segregating sites’) is very popular [9], and Ripley’s $K$-function is seldom used. In Fig. 6(a) we show $\kappa(r)$ as a function of $\rho$ and $\theta$. Note that the curves converge to $\theta/(1 + \theta)$ and that the behaviour for small $r$ is determined by the recombination rate. In Fig. 6(b) we show the distribution of runs of homozygosity as a function of $\rho$ and $\theta$.

To assess the power of the descriptive statistics, we make an estimator which minimises the $\chi^2$-distance between the observed and the theoretical statistics. In Fig. 2 we demonstrate that Ripley’s $K$-function is a much more powerful summary for parameter estimation than nearest neighbour.

Fig. 6. On (a) and (b) the summary functions, the unnormalised derivative of Ripley’s $K$-function and runs of homozygosity, are plotted for different parameter values. The starting position of the $K$-statistics depends on the recombination rate $\rho$ and it slowly converges to the mutation rate $\theta$. For runs of homozygosity, the difference between different recombination rates are not as profound as for $K$-functions.

6 Conclusion

The particle filter is an alternative method for inference about population genetic parameters. Asymptotically, all allowed proposal distributions converge, but the time consumption can be detrimental in applications. In our simulation study, the RECJumper particle filter is significantly better than the prior particle filter when the sequence segments are larger than 400 base pairs. The set-up of resampling positions and number of particles determines whether simulating segments of length larger than 400 base pairs is advantageous. It could be enough to sample segments shorter than 400 base pairs and then stitching them together with particle filtering methods. Best practice might also differ from dataset to dataset.
which makes it convenient to have both proposals; prior and RECJumper. Our particle filter is slower than the HMM when it comes to evaluating the likelihood \( p(y) \) at a precision necessary to estimate the recombination rate on two sequences.

Generally, there are two strategies for calculating an integral numerically. One is to discretise the function so that the integral of the discretised function can be calculated exactly while another strategy is to make a Monte Carlo estimate which converges towards the correct integral. In this study we estimated the integral \( L(\rho) = \int p_\rho(y|x)p_\rho(x) \, dx \) using the latter strategy in contrast to the widespread HMM based methods applying the former. The discretisation method normally struggles in higher dimensions like the HMM methods do with many sequences. If time has to be divided into 200 intervals, the speed will be even slower.

Besides estimating a constant recombination rate, the model can also be extended to estimation of varying recombination rate, and varying mutation rate. It could also be used to estimate variability in population sizes as Palacios and Wakeley [15] successfully did with simulated coalescent paths under a constant population size model.

The summary statistics investigated proved to display a big difference in power of estimating the recombination rate. The variance of the estimator based on Ripley’s \( K \)-function was significantly lower than that of runs of homozygosity in the set-up with a constant-sized, panmictic population. In this study the theoretical values of the \( K \)-function and runs of homozygosity were calculated using the Simonsen-Churchill model. These could also be estimated empirically from simulations avoiding the Markov assumption.

The \( K \)-function can also be used for posterior predictive checks where a fitted model is tested by comparing datasets simulated from the fitted model with the actual dataset. The comparison is through a summary statistic on the datasets and a transformation of the \( K \)-function could be such a summary statistic. The global rank envelope test [14] is one way to make the transformation.

References

Appendix C

admixtreegraph paper

In the attached paper I implemented the MCMC for simulating branch lengths and admixture proportions.
Genetics and population analysis

admixturegraph: an R package for admixture graph manipulation and fitting

Kalle Leppälä, Svend V. Nielsen and Thomas Mailund*

Bioinformatics Research Centre, Aarhus University, Aarhus 8000, Denmark

*To whom correspondence should be addressed.
Associate Editor: Jonathan Wren

Received on August 23, 2016; revised on December 14, 2016; editorial decision on January 20, 2017; accepted on January 24, 2017

Abstract

Summary: Admixture graphs generalize phylogenetic trees by allowing genetic lineages to merge as well as split. In this paper we present the R package admixturegraph containing tools for building and visualizing admixture graphs, for fitting graph parameters to genetic data, for visualizing goodness of fit and for evaluating the relative goodness of fit between different graphs.

Availability and Implementation: GitHub: https://github.com/mailund/admixture_graph and CRAN: https://cran.r-project.org/web/packages/admixturegraph.

Contact: mailund@birc.au.dk.

1 Introduction

The relationship between populations is not always a simple tree. In addition to the splitting events, where an ancestral population split into two or more isolated groups, admixture events can merge two or more populations. Admixture graphs are extensions of phylogenetic trees that allow such merging events.

Inference of admixture graphs has not received the same attention as phylogenetic trees, but a number of methods have recently been developed for fitting genetic data to graphs and for using heuristics or brute-force search approaches to finding best-fitting graphs qpgraph (Castelo and Roberato, 2006), TreeMix (Pickrell and Pritchard, 2012), AdmixTools (Patterson et al., 2012; Zhao and Patterson, 2016), MixMapper (Lipson et al., 2013). These methods model the genetic relationship between populations as a graph where observed populations are represented as leaves, inner nodes represent ancestral populations, and edges represent the genetic drift separating an ancestral population from a descendent population. Without admixture events, the structure is simply a tree, but when the ancestry of the populations contain admixture, the graph contains nodes with more than one parent.

The graph describes the genetic drift within populations and the correlation of drift between populations. Data is usually summarized in some form studying patterns of allele frequency correlations across populations. In the TreeMix method by Pickrell and Pritchard (2012) data is thus represented as the covariance matrix of genetic drift while the AdmixTools software by Patterson et al. (2012) summarizes patterns of drift through so-called f-statistics. Given a graph topology together with edge lengths and admixture proportions the expected drift patterns can be computed and a likelihood derived and parameters of the graph can be inferred.

In this paper we describe the R package admixturegraph. This package contains functionality for:

- constructing and visualizing admixture graphs
- fitting graph parameters and visualizing the goodness-of-fit
- computing Bayes factors between graphs for comparing them
- exploring the space of graph topologies to find the best fitting graphs

For comparison, qpgraph and AdmixTools work on a user specified admixture graph, and MixMapper and TreeMix use a sequential heuristic building new admixture events based on the previous ones. The package admixturegraph can be used either for brute-force search on the graph topologies or for a heuristic approach generalizing the sequential one, building more complicated admixture graphs from a selected set of well performing simpler admixture graphs (as the sequential approach is not guaranteed to converge towards the best fitting admixture graphs). The package does not automatically infer an admixture graph from the data.

The R package does not add functionality that cannot be found in existing software, but by providing an R interface to working interactively with admixture graphs, fitting and visualizing the goodness of fit of graphs, we believe that we make admixture graphs more accessible to users.
2 Features

The admixturegraph R package provides a framework for constructing and analysing admixture graphs and evaluating their fit to genetic data summarized as $f$-statistics (Patterson et al., 2012).

2.1 Constructing and visualizing admixture graphs

Admixture graphs are constructed using R functions specifying the nodes and edges and naming the admixture proportion parameters. Figure 1a shows a toy example dataset, adapted from Cahill et al. (2013), consisting of one black bear (BLK), one polar bear (PB) and a number of brown bear samples. Figure 1b shows example code for constructing an admixture graph that models that the ABC-bears (Adm, Bar, Chl) are admixed between polar bears and brown bears as suggested by Cahill et al. (2013). Figure 1e shows an alternative where polar bears are admixed (see Lan et al., 2016, https://dx.doi.org/10.1101/047498).

Unlike visualizing trees, which are always planar graphs, it is not trivial to present graphs in a visually pleasing way. We have therefore implemented heuristics for laying out graphs for plotting while providing a number of options for overruling the heuristics to customize plots. Examples of plotted graphs are shown in c and e.

2.2 Fitting graph parameters and visualizing fits

For fitting graph parameters to data, the data should be collected in an R data frame or equivalent (see package documentation for details on the expected format). The fitting procedure first extracts from the graph topology a set of equations for the expected values of each $f$-statistic in the data. These are linear equations on edge lengths and polynomials on admixture proportions. From the observed data and this set of equations, the likelihood of parameters can be computed and maximized. The likelihood used is

$$\text{det}(2\pi\Sigma)^{-1/2} \exp\left(-\frac{1}{2} (f - \mu)^T \Sigma^{-1} (f - \mu)\right),$$

where $L = \text{det}(2\pi\Sigma)^{-1/2} \exp\left(-\frac{1}{2} (f - \mu)^T \Sigma^{-1} (f - \mu)\right)$. $f$ is the vector of observed statistics, $F$ is the vector of statistics predicted by the graph topology and parameters, and $\Sigma$ is the covariance matrix of the observed statistics $f$, which is either given by the user or replaced by a proxy of the identity or a diagonal matrix constructed from Z-scores given by AdmixTools for instance. The maximizing procedure used alternates between solving the linear problem on edge lengths, which can be optimized analytically, and the polynomial problem on admixture proportions, which is solved using numerical optimization.

Once we have fitted a graph to data we can visualize the goodness-of-fit by plotting the expected statistics against the observed statistics (see Fig. 1d and f where shows the fit of the two graphs in c and e). The genetic data is shown as observations with error bars (black lines) while the expected values are shown as solid dots.

2.3 Posterior distributions and graph comparisons

Fitting graph parameters to data provides a maximum likelihood point-estimate. To obtain confidence intervals for parameters, one can use a blocked jackknife or bootstrap procedure as in (Patterson et al., 2012) but the admixturegraph package also provides an alternative in the form of a Markov Chain Monte Carlo (MCMC) procedure for sampling from the posterior distribution of joint parameters.

Comparing the fit of two different graphs is not straightforward since graphs can have very different numbers of parameters and are usually not nested models. Instead we propose to use Bayes factors—the ratio of the likelihood of one graph over another—to compare models. To do this it is necessary to integrate out the graph parameters and obtain a likelihood for a topology alone. To estimate this integral we use the MCMC to obtain samples from the posterior likelihood and use these in an importance sampler procedure to compute the graph likelihood.

2.4 Exploring the space of graph topologies

While we can compare the fit of different graph topologies to data, there are no known algorithms for inferring the optimal graph topology. Instead the package implements a few functions for brute force exploration of topologies and heuristics for extending topologies.

The set of all possible graphs, even when limited to one or two admixture events, grows super-exponentially in the number of leaves and it is generally not computationally feasible to explore this set exhaustively. Still, we give graph libraries for searching through all possible topologies with not too many leaves and admixture events. For larger graphs we provide functions for exploring all possible graphs that can be reached from a given graph by adding one extra admixture event or by adding one additional leaf. However, the best fitting admixture graphs are not necessarily extensions of best fitting smaller graphs, so we recommend that users not only expand the best smaller graph but a selected few best of them.
3 Conclusion

We have presented an R package for exploring and fitting admixture graphs. The package provides functionality for constructing and visualizing admixture graphs, for fitting graph parameters to genetic data, for visualizing goodness-of-fit, and for comparing the quality of fits between non-nested graph models. While the package does not contain algorithms for automatically inferring optimal graph topologies, it does provide functionality for exploring the space of possible topologies.

Funding

This research was funded by the Danish Council for Independent Research, Sapere Aude grant 12-125062.

Conflict of Interest: none declared.

References


Appendix D

AdmixtureBayes draft

This is a rough draft of a paper I am writing with Rasmus Nielsen and Thomas Mailund about AdmixtureBayes. The conclusions are not polished yet but the draft contains many extra graphs and technical descriptions not found in the Chapter 3.
Bayesian Inference of Admixture Graph on Native American Populations

Svend V Nielsen and Thomas Mailund

Aarhus University, Bioinformatics Research Centre, C.F. Møllers Allé 8, DK-8000 Aarhus C, Denmark

Rasmus Nielsen

University of California, Berkeley, Department of Integrative Biology, Berkeley, CA, USA

July 31, 2018

Abstract

An admixture graph is a phylogeny with admixture events and it is an increasing popular structure to describe a set of populations. Fitting admixture graphs to data is a challenging problem as the space of graphs grows super-exponentially with the number of populations. To fit admixture graphs, we calculate the sample allele frequencies in each population and model their differences with a brownian motion process. We add a prior to the space of admixture graphs and sample the graph space with a Reversible Jump MCMC that allows us to jump between graphs with different number of admixture events. With this sample we can estimate the number of admixture events, the graph topology and branch lengths of the true admixture graph. We apply this method to a dataset of 11 Native American populations and find that their history contains more than 7 admixture events. The method is implemented in the program AdmixtureBayes, which is available on github.

Introduction

The first models of the genetic relationships of human populations assumed that human population diversification could be described by binary trees [2]. However, the past 30 years of population genetic research has demonstrated that human evolution in general is not tree-like, but rather a network of populations merging and splitting, called an 'admixture graph'. A split (forward in time) represents a divergence event between two populations and a merge represents an admixture event between two populations. There has been substantial focus over the past 5-10 years on methods for estimating such graphs [10] [7] [9]. Estimation of these graphs holds the key for understanding historical relationships among populations - both for humans and for other species.
The most popular methods for estimating admixture trees are ‘Treemix’ by Pickrell and Pritchard [10] and ‘qpgraph’ by Patterson et al. [9]. These methods have provided several breakthroughs in our understanding of the relationship between different human populations [11] [8]. qpgraph [9] employs information similar to that used in the Patterson’s D [6], allowing users to sequentially identify the best position of a possibly admixed population in a previously established admixture graph. Treemix estimates an admixture graph de novo using a Gaussian model of the distribution of allele frequencies among populations. The implicit assumption in the Gaussian model is that the genetic drift process of change in allele frequencies can be approximated as a Brownian motion process. This assumption dates back to the early work by Edwards and cavalli-Sforza [?] and has recently re-emerged as a computationally attractive alternative to the full Wright–Fisher process. It is not only used in Treemix, but also in several other methods aimed at modeling the joint distribution of allele frequencies among populations [3].

The estimation of admixture graphs is not computationally trivial. They represent an extension of trees to a much larger space of topologies. The only current method that allows a full optimization for the optimal graph (Treemix) has implemented advance optimization algorithms [10], but users nonetheless typically run the program many times in an attempt to avoid optimization errors [?]. Furthermore, users typically only report a single maximum likelihood estimate of the admixture graph. We note that reporting only a single admixture graph has the potential to be misleading because there may be a large number of possible admixture graphs with very similar degree of statistical support. Therefore, the reporting of single graphs has the potential to be misinterpreted as strong confidence in particular historical relationship among populations.

We will here provide an alternative solution to this problem. Based on a model similar to the one used by Treemix, we will use a Bayesian approach based on reversible jump Markov Chain Monte Carlo (MCMC) for estimating admixture graphs. The method can report sets of admixture graphs and associated measures of statistical support. This approach has one additional practical advantage; it can provide posterior probabilities (or Bayes factors) for particular subgraphs. These subgraphs provide solutions to topological questions of interest while integrating over other features of the admixture graph. The method can, therefore, be used to investigate the relationship between specific focal populations while taking their ancestry and the possibility of admixture from other populations into account.

We illustrate the utility of the method using simulations. We also reanalyze a previously published genomic data set of Siberians and Native Americans. We use the method to re-visit two important and controversial questions in the history of the peopling of the Americas. First, we analyze the origin of Inuit and show that they are modeled best as an admixture between a population related to so-called paleo-Eskimos represented by the Saqqaq genome, and native Americans represented by Athabascans. Secondly, we show that Athabascans best can be represented as admixed between a native American population and a Siberian population most closely related to the Koryak, but not the Saqqaq. We have implemented the method in a open source program, AdmixtureBayes, available on GitHub.
Results

We have implemented a Markov Chain Monte Carlo (MCMC) algorithm, AdmixtureBayes, that samples admixture graphs from their posterior distribution. We summarize genetic data from multiple populations or multiple individuals as a matrix that captures how allele frequencies in the data covaries between samples. AdmixtureBayes samples graphs that explains this covariance matrix. The topology of the graphs capture the relationship between samples; branch lengths capture the divergence between samples, measured by drift, and admixture events translate into mixture distributions in the covariance matrix. As a consequence of the MCMC, each graph is sampled at a frequency corresponding to its posterior probability.

Simulation experiments

To validate the method, we simulated genetic data according to known admixture graphs, and we applied AdmixtureBayes to this data to collect graph samples. To compare AdmixtureBayes with Treemix, which computes a single best graph, we first reduced our samples to the most seen graph topology, the maximum a posterior topology. Treemix must be told the number of admixture events to fit while can AdmixtureBayes infer this or be confined to a fixed number of events similar to TreeMix and we explored both setups. The comparison is presented in Figure 1. It shows that the probability of estimating the correct graph is almost the same for the two methods. We also explored the sensitivity of AdmixtureBayes to its initial state by either starting the Markov chain in the true topology or in a random topology (Supplementary Figure 10).

Simply counting how often we infer the true graph as the maximum a posterior graph does not capture how close the inferred graph is to the true graph when the topologies differ. Very similar graphs are considered just as incorrect as graphs that differ from the true graphs by small differences. To measure how well our inference performs when we do not get the exact true graph, we developed a graph comparison measure that summarizes each inner node in a graph as the set of its descendants. To compare two graphs, we then count how many of these descendants-sets are found in one and not the other graph (Supplementary Figure 5). This measure can be seen as a generalization of the Robinson-Foulds’ distance for tree topologies. We compared the performance of AdmixtureBayes’ maximum a posterior graphs with TreeMix’ maximum likelihood graphs, see Figure 2. For AdmixtureBayes, the output is a sample of graphs, we can also compute the average distance from true graph, as a measure of how the samples are to the true graph, as apposed to just the maximum a posterior. Results for this measure are also shown in Figure 2.

The sets of descendants we use to compare samples of graphs against the true graph can also be used to summarize a set of graph samples. Given a sample of graphs we can extract all inner nodes from all graphs, summarized by their descendants-sets. We can filter these sets according to the fraction of samples they appear in to get a smaller set of descendants-sets, and we can build a summary graph from this, (See Methods). The result is not necessarily an admixture graph that AdmixtureBayes could sample. In AdmixtureBayes, inner nodes will either have one parent and two children or two parents and one child; in the summary graph, inner nodes can have any number of parents and
Figure 1: Comparison of AdmixtureBayes (uAdmBayes, blue), constrained AdmixtureBayes (CAdmBayes, red) and Treemix (green) by how often they find the correct topology. On the left, the average AdmixtureBayes probability of the true topology is plotted, whereas the right plot shows the probability that the highest posterior topology is the correct topology. Treemix only estimates one graph, so the Treemix bar is identical in the two plots.
Figure 2: Comparison of AdmixtureBayes (uAdmBayes, blue), constrained AdmixtureBayes (CAdmBayes, red) and Treemix (green) by how often they find the correct topology. On the left, we show the average distance between the highest posterior topology and the true topology, whereas the right plot shows the posterior distance between an AdmixtureBayes topology and the true topology. As in Figure 1, the Treemix bar is the same in the two graphs.
children. Nevertheless, the summary graph captures aspects of the sample of graphs, and inner nodes can be annotated with the frequency at which we see them in the samples, giving us a measure of support for a node similar to the bootstrap values given to edges in phylogenetic tree inference.

**Summarizing graph-samples**

Ideally, we want to draw inference from the entire sample of graphs, but considering the space of possible graphs and the variation we will inevitably see in the samples, it is infeasible in practice to consider the entire sample. We can, however, extract sub-graphs from the samples to consider specific questions about relatedness. Given a small subset of leaves, we can extract all subgraphs containing only these leaves, and for each subgraph we get a posterior probability as the frequency by which we see a subgraph. We can rank these subgraphs by their probability and directly see how much support each population history has. Note that this is different from restricting the data to a subset of leaves and sample graphs from the smaller data set. Gene flow from an unsampled population into a sampled population can look like gene flow within the samples. Consider e.g. the topology (((A,B),C,(D,E)) where there is gene flow from E to B. This will make A look more like C than B does. If we remove E from the topology, this stronger correlation between A and C than between B and C can be interpreted both as gene flow from C into A or gene flow from D into B. Including E in the analysis, resolves which of the scenarios we have. By sampling graphs over large sets of leaves we can extract subgraphs that contain information about populations not included in the subgraph. We will be able to combine data from all populations to resolve specific questions about a subset of the populations that we would not be able to resolve if we restricted the analysis to only those populations. The benefit from can also be seen in Supplementary Figure 9, where we inferred subgraphs in the full data and compared it to graphs inferred from subdata.

**Exploring the genetic history of Paleo-Eskimos, Inuits and Native Americans**

We applied AdmixtureBayes to a set of Siberian and Native American samples to explore the relationship between Siberian Chukotko-Kamchatcan speakers (Koryak), Paleo-Eskimos (Saqqaq), Eskimo-Aleut speakers (Greenlandic Inuits), and Na-Dene speakers (Athabascan). The dataset also contained North and South Americans (Anzick, TA6), The data contains 2171266 SNPs. Due to linkage between SNPs, the variance of block-bootstrapped estimates of the covariance matrix corresponded to $\approx 60000$ independent SNPs. We chose the Yoruba population as outgroup. Running time of AdmixtureBayes was 20 hours on 20 cores. The output graph sample was extremely diverse due to a high number of admixture events (95% credible interval [7,16]). The highest posterior topology had a posterior probability of less than 1%. From a topology set one can construct a minimal topology giving the same topology set(Methods). The two minimal topologies with the highest posterior probability (2.7 and 1.4 %) are shown in Figure 3. Both contain inner nodes which themselves have a low posterior probability. We combined the sample into one summary graph as described in the previous section (Figure 4). The graphs support that both
Athabascans and Greenlandic Inuits are admixtures of the Anzick-TA6 and Koryak lineages. There is also strong evidence of gene flow from the Saqqaq lineage to the Greenlandic Inuit population. There is large uncertainty about the order of the splits between the 5 lineages Ket, TA6-Anzik, USR1, Koryak, Saqqaq.

![Diagram of admixture graph](image)

Figure 3: The two minimal topologies with the highest posterior probability. Each inner node is colored according to the posterior probability (in the parentheses) that a node with the same descendants exists.

Conclusion

We have developed AdmixtureBayes which is a method for inferring admixture graphs using MCMC. On simulated data, it infers graphs that are closer to the true graph than TreeMix using the Set distance measure. The posterior probability of the true topology is similar to the probability that TreeMix finds the true graph.

On the Native American and Siberian samples we found many admixture events supporting previous studies [8]. Furthermore, we find a similar but not identical topology to a previous admixture topology [8]. However, our results also indicate that several features of the tree are too uncertain to draw conclusions on. For example, we could not estimate the order of coalescence events between the 5 lineages USR1, Ket, Koryak, Anzick-TA6 and Saqqaq.

Methods

AdmixtureBayes Model

The AdmixtureBayes model searches the posterior distribution of admixture graphs given observed data with a Markov Chain Monte Carlo procedure. The
Figure 4: Considering the posterior probability of all descendant sets, this is the recipeA node summary of the posterior sample of admixture graphs on the Native American dataset. In the parentheses of each node is the posterior prevalence of the node.
admixture graphs and observed data are summarized as covariance matrices of allele frequency changes. We use the Treemix method of summarizing the admixture graph which we will explain in the following; Consider the tree structure in (1) where population 2 is a mix of two populations with proportion $w$ and $1-w$.

The allele frequency in the 4 populations, $P_0$, $P_1$, $P_2$ and $P_3$ are related through the allele frequency changes $x_0, \ldots, x_7$.

$$
\begin{pmatrix}
P_1 \\
P_2 \\
P_3
\end{pmatrix} =
\begin{pmatrix}
P_0 + x_0 + x_1 + x_2 \\
P_0 + x_0 + x_7 + w(x_6 + x_4) + (1-w)(x_5 + x_2) \\
P_0 + x_0 + x_3 + x_4
\end{pmatrix} =
\begin{pmatrix}
P_0 \\
P_0 \\
P_0
\end{pmatrix} + A
\begin{pmatrix}
x_0 \\
x_7
\end{pmatrix}
$$

where $A$ is a matrix depending on the graph structure and admixture proportions. In the neutral Wright-Fisher model allele frequency drift can be approximated by a normal distribution when they are far from 0 and 1. If $x_i$ is a drift from a node with allele frequency $p_i$, it follows that $x_i \sim≈ N(0, (1-e^{-d_i})p_i(1-p_i))$ where $d_i = t_i/2N_i$ is the number of generations scaled with the population size. We collect the term $(1-e^{-d_i})$ into a single term $c_i$ and make the further approximation

$$x_i \sim≈ N(0, c_i p_i(1-p_i)).$$

where $p$ is a base allele frequency. Therefore we can conclude that

$$\begin{pmatrix}
P_1 - P_0 \\
P_2 - P_0 \\
P_3 - P_0
\end{pmatrix} \sim≈ N(0, p(1-p)\Sigma), \quad \Sigma = A \cdot \text{diag}(c_0, \ldots, c_7) \cdot A^*$$

(2)

Let $p_{ij}$ be the sample allele frequency in the $i$'th population at the $j$'th SNP. Assume that population 0 is an outgroup. The base estimate of the covariance matrix is

$$S_{k,l} = \frac{1}{N} \sum_{j=1}^{N} (p_{kj} - p_{0j})(p_{lj} - p_{0j})$$

(3)

The sampled allele frequencies are not direct observations from the model in (2). Instead they are binomially sampled

$$p_{ij} \sim \frac{1}{m_{ij}} \text{bi}(m_{ij}, P_{ij})$$
where \( m_{ij} \) is the number of haplotypes sampled and \( P_{ij} \) is the true allele frequency in population \( i \) at SNP \( j \). This inflates the expected value of \((p_{kj} - p_{0j})(p_{lj} - p_{0j})\) with the quantity

\[
1(k = l) \frac{P_{kj}(1 - P_{kj})}{m_{ij}} + \frac{P_{0j}(1 - P_{0j})}{m_{ij}}
\]

which suggests the correction on \( S_{k,l} \)

\[
\hat{B}_{kl} = 1(k = l) \frac{1}{N} \sum_{j=1}^{N} \frac{p_{kj}(1 - p_{kj})}{m_{ij} - 1} + \frac{1}{N} \sum_{j=1}^{N} \frac{p_{0j}(1 - p_{0j})}{m_{ij} - 1}
\]

Next, we normalize with

\[
\hat{h} = \frac{1}{N} \sum_{j=1}^{N} \bar{p}_j(1 - \bar{p}_j), \quad \text{where} \quad \bar{p}_j = \frac{1}{n+1} \sum_{i=0}^{n} p_{ij}
\]

to take the term \( p(1 - p) \) from (2) into account.

If the sample allele frequencies were normally distributed and independent across markers, the estimator in (3), would be Wishart distributed and the degrees of freedom would be the number of markers. The sample allele frequencies are not independent and only approximately normal, yet we use the likelihood.

\[
W \left( S/\hat{h}; \Sigma + \hat{B}/\hat{h}, \text{df} \right)
\]

Because we put a prior on \( \Sigma \), certain values of \( \Sigma \) will be preferred. The inference could therefore be perturbed by influenced by the absolute value of \( S/\hat{h} \). To avoid this, we normalize the input such that the trace of \( S/\hat{h} \) is \( \log_2(L)\), where \( L \) is the number of leaves in the graph,

\[
W \left( c_S S/\hat{h}; c_S(\Sigma + \hat{B}/\hat{h}), \text{df} \right).
\]

We estimate the degrees of freedom, \( \text{df} \), using bootstrapped values of \( S/\hat{h} \) which we will denote \( X^{(1)}, \ldots, X^{(B)} \). Let \( \bar{X} \) be the average of the bootstrap sample. It would be natural to estimate \( \text{df} \) with the maximum likelihood of the model

\[
X^{(1)}, \ldots, X^{(B)} \sim W(\bar{X}, \text{df})
\]

However the estimates are suboptimally low(Supplementary Figure 11). The matrices \( X^{(i)} \) deviates too much from the wishart distribution. In stead we use that the variance of the \( k,l \)’th entry of a Wishart distribution with scale parameter \( \Psi \) and degrees of freedom, \( \text{df} \), is

\[
\frac{1}{\text{df}}(\Psi_{kk}^2 + \Psi_{kk}\Psi_{ll})
\]

We estimate \( \text{df} \)

\[
\text{arg min}_{\text{df}} \sum_{k=1}^{n} \sum_{l=1}^{n} \left( \text{Var}(X_{kl}^{(1)}, \ldots, X_{kl}^{(B)}) - \frac{1}{\text{df}}(\bar{X}_{kl}^2 + \bar{X}_{kk}\bar{X}_{ll}) \right)^2
\]
where $\hat{\text{Var}}$ is the sample variance. This estimator is less vulnerable to deviations from the wishart distribution.

An admixture graph consists of a topology and a set of continuous parameters. The space of topologies for a given number of leaves, $B$, consists of all uniquely labelled graphs of the set of all directed acyclic graphs which fulfills

1. There exists one and only one root. That is a node with no parents and only one child.

2. The number of nodes with no children is $B$. All these nodes have only one parent and are called leaves.

3. If a node is not a root or a leaf, it has either
   (a) 1 parent and 2 children in which case we call it coalescence node.
   (b) 2 parents and 1 child in which case we call it admixture node

4. No node has the same node as both parents.

The labelling includes

1. All leaves are given a unique label

2. Parent edges from an admixture are labelled 0 and 1

This means that one can interpret admixture nodes as having a ‘main’ parent and an admixture parent. In addition, all edges have a length in the interval $(0, \infty)$ and all admixture nodes are given an admixture proportion in the interval $(0, 1)$.

**Prior**

We chose the prior AdmixtureBayes to split up into a prior on the topology, $G$, and a prior on the continuous parameters. The continuous parameters include the branch drift lengths, $\overrightarrow{c} = (c_1, \ldots, c_B)$, and the admixture proportions $\overrightarrow{w} = (w_1, \ldots, w_A)$. Let $K$ denote the number of admixtures and $B$ the number of branches and $L$ the number of leaves. The prior is

$$P(G|K)P(\overrightarrow{w}, \overrightarrow{c} | K, B) = P(G|K)P(K)P(\overrightarrow{c} | K, B)P(\overrightarrow{w} | K)$$

The prior on the number of admixture events is a geometrical distribution with parameter 0.5 (truncated to max 20). For the prior $P(G|K)$ we chose the uniform prior. For this we need the number of possible topologies for a given number of
admixture events, $K$. We use the recursion

$$N(L, P, A, E) = \left( (E+1)N(L-1, P, A, E+1) 
+ (L-2P+1)N(L-1, P-1, A, E) 
+ (L+2P+3A-2E-2)N(L-1, P, A, E) 
+ \frac{2(P+1)}{L(L+1)}N(L+1, P+1, A-1, E-1) 
+ \frac{2(P+1)(P+2)}{L(L+1)}N(L+1, P+2, A-1, E) 
+ \frac{2(P+1)(L-2P-1)}{L(L+1)}N(L+1, P+1, A-1, E) 
+ \frac{(L-2P)(L-2P-1)}{2L(L+1)}N(L+1, P, A-1, E) \right),$$

where $L$ is the number of leaves, $P$ is the number of pairs of leaf branches that coalesce with each other, $A$ is the number of admixture events and $E$ is the number of eyes (that is nodes with two identical parents). $N(L, P, A, E)$ is the number of topologies with $L$ leaves, $P$ pairs, $A$ admixtures and $E$ eyes. Then

$$P(\mathcal{G}|K) = \frac{1}{\sum_{P=0}^{\lfloor L/2 \rfloor} N(L, P, K, 0)}.$$

For the prior on the branch lengths, $\mathcal{c}$, we have to be aware that the number of branches depend on the graph. Graphs with many admixture events have more branches. The covariance matrix $c_{SS}/\hat{h}$ is scaled such that a graph without admixture events is expected to have an average branch length of 1. In an admixture graph with admixture events the expected branch length would therefore be

$$\frac{2n-2}{B} \quad \text{(8)}$$

We therefore use an exponential prior on the branch lengths with rate (8).

For the admixture proportions, we chose the uniform prior on the interval $(0, 1)$.

**MCMC**

The MCMC is implemented as a parallel MC$^3$ algorithm [4] to jump faster between modes of the posterior surface. Because the admixture graphs have different numbers of continuous parameters, we use the Reversible Jump generalization of the MCMC algorithm [5]. The proposal distribution is a mix of 7 smaller proposals. They are

1. Add an admixture branch to the admixture graph. A random sink branch, $s$, is chosen with probability $\frac{1}{B}$ where $B$ is the number of branches in the graph (not including the branch to the outgroup). Then a random source branch, $s'$, is chosen amongst the remaining branches (including the root/outgroup branch) which would not cause a loop in the admixture graph. If the number of possible sink branches is $B'(s)$ the probability is $\frac{1}{B'(s)}$. Next the attachment points of the admixture graph is simulated
uniformly. If the branch lengths of \(s\) and \(s'\) is \(c(s)\) and \(c(s')\) the attachment outcome has density \(\frac{1}{B B'(s) c(s)c(s')}\). If the source branch is the root branch, we simulate the attachment point with an exponential distribution, \(e(1)\), instead. The new admixture proportion is simulated uniformly between 0 and 1, and the admixture branch length is simulated from \(e(1)\) with density \(e^{-1}\). Lastly, the labelling of the two parent branches of the new admixture node is simulated. The density is \(\frac{1}{2}\). In conclusion, the density is

\[
\frac{1}{B B'(s) c(s)c(s')} e^{-1} \frac{1}{2}. \tag{9}
\]
When accepting this proposal, one should calculate the reverse move which is proposal number 2. The Reversible Jump Jacobian factor is 1.

2. Remove an admixture branch from the admixture graph. An admixture branch can be removed if its parent is not an admixture branch. Let the number of admixture branches with no admixture parent be \(K'\). We choose uniformly in that set and remove the admixture branch. The density is

\[
\frac{1}{K'} \tag{10}
\]

3. Node sliding. A random branch whose parent is not an admixture node is chosen. We move the attachment point a distance \(x\) in the graph where \(x \sim e(\lambda)\). The backward density is identical. We adapt \(\lambda\) on-the-go following guidelines for adaptive proposals in MCMC [1].

4. Random walk on the branch lengths. We add a normally distributed noise to all the parameters. The backward density is again identical to the forward parameter. The variance is controlled by parameter \(s\) which we also adapt on-the-go.

5. Random walk on the admixture proportions as in step 4.

6. Random walk on the branch to the outgroup as in step 4.

7. Random walk on the branch lengths but inside the null space of matrix \(A\). This means that the proposed admixture branch will have the same likelihood as the previous graph. This proposal is also adaptive as in step 4.

Graph Summaries
To summarize a sample of admixture graphs, \(g_1, \ldots, g_R\), we first transform all of them into their topology sets (Figure 5) and obtain a sample \(t_1, \ldots, t_R\). From the frequency of topology sets,

\[
f(s) = \frac{\# \left\{ t \in \{t_1, \ldots, t_n \} : s \in t \right\}}{R}
\]
we define the summary topology sets at level \(\alpha\) as

\[
T^\alpha\left\{ s \in \bigcup_{i=1}^{R} t_i : f(s) > \alpha \right\}. \tag{11}
\]
Next, $\mathcal{T}$ can be turned into a graph structure. First we expand it with the root node and leaf nodes topology sets. Next we let all topology sets in the expanded $\mathcal{T}$ correspond to a node in the summary graph. There will be a connection from node $t$ to $t'$ if

\[
\begin{align*}
& t, t' \in \mathcal{T}, \\
& t \neq t', \\
& t' \subseteq t, \\
& \not\exists t'' \in T \setminus \{t, t'\}: t' \subseteq t'' \subseteq t.
\end{align*}
\]
Figure 5: This is a scheme of how to calculate the Set distance between two admixture graph topologies (left). First, they are transformed in their descendant sets/topology sets(middle). The distance is the symmetric set distance between the two topology sets.

References


Figure 6: Similar to Figure 5, except the topology sets are transformed into minimal topologies (right).
A Supplemental Text

Simulation Results

AdmixtureBayes uses Markov Chain Monte Carlo (MCMC) on the posterior distribution of admixture graphs. To determine the requirements for convergence, we simulated admixture graph covariance matrices with 5, 10 or 20 leaves, and 0, 1 or 2 admixture events. We compared AdmixtureBayes chains started in the true admixture graphs and in a random graph. In less than 6 hours on 15 cores, the randomly started chains achieved the same accuracy as the perfectly started chains for admixture graphs with up to 10 chains and 2 admixture events (Supplementary Figure 10).

B Supplementary Figures
Figure 7: Using admixture graphs with 10 leaves and 0, 1, 2, 5 admixture events, we simulated 600 different datasets with ms in a grid of simulation parameters. We separated them on the number of admixture events and measured the distance to the true graph in 5 ways. a) The average distance between the covariance b) The set distance between the highest posterior topology and the true topology. c) the average set distance. d) the frequency of topology identity. e) the frequency that highest posterior topology is identical to the true topology. The blue column labelled uAdmBayes contains AdmixtureBayes graphs and the red column labelled CAdmBayes contains AdmixtureBayes graphs with the true number of admixture events.
Figure 8: For the same simulations as in Figure 13, we separated the simulations on the sample size in each population.

Figure 9: Using admixture graphs with 10 leaves and 0, 1 and 2 admixture events, we simulated 60 different admixture graphs with ms. We estimated subgraphs of size 3, 4 and 5 from each dataset. The Small column contains graphs build from the marginal dataset and the Big column contains subgraphs of graphs obtained from the full data set.
Figure 10: We simulated admixture graphs with 0, 1 and 2 admixture events and 5, 10 and 20 leaves. We used the true covariance matrices as input in AdmixtureBayes and degrees of freedom $10^3$, $10^4$, $10^5$ and $10^6$. AdmixtureBayes was initialised in a random tree (blue) or in the true, perfect tree (red). Two adjacent bars of the same height is an indication that AdmixtureBayes converges when initialized in a random tree.
Figure 11: We simulated admixture graphs with 10 leaves and 0, 1 and 2 admixture events. Using these graphs, we simulated datasets with ms and different sample sizes. The top plot illustrates the ratio between the maximum likelihood degrees of freedom estimate from (6) and the variance estimator in (7). We ran AdmixtureBayes with the maximum likelihood estimate, the variance estimate, and 2 and 4 times the variance estimate. We calculated the mean posterior of the true topology for each run. The bottom plot shows the averages of those values.
Figure 12: In Figure 9, we calculated how Treemix and AdmixtureBayes performed when estimating subgraphs. Here we have stratified the plot according to subgraph size (in the columns), measure of accuracy (in the rows) and the colors are the method.

Figure 13: For the same simulations as in Figure 13, we separated the simulations on the number of SNPs.
Figure 14: From a posterior AdmixtureBayes sample, we computed the posterior probability of all minimal topologies for two subsets of the populations. BF is the unnormalized likelihood of the topology and can be used to calculate the Bayes Factor. The Bayes Factor between the highest and second highest posterior minimal topology for Greenlander, Koryak and Saqqaq is $45/0.57 = 78.9$. 

<table>
<thead>
<tr>
<th>Sub topology:</th>
<th>Sub topology:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Athabascan, Koryak and Saqqaq.</td>
<td>Greenlander, Koryak and Saqqaq.</td>
</tr>
<tr>
<td><img src="image1.png" alt="Diagram" /></td>
<td><img src="image2.png" alt="Diagram" /></td>
</tr>
<tr>
<td>Posterior: 59%, BF=1.9</td>
<td>Posterior: 72%, BF=45</td>
</tr>
<tr>
<td><img src="image3.png" alt="Diagram" /></td>
<td><img src="image4.png" alt="Diagram" /></td>
</tr>
<tr>
<td>Posterior: 13%, BF=8.1</td>
<td>Posterior: 18%, BF=0.57</td>
</tr>
<tr>
<td><img src="image5.png" alt="Diagram" /></td>
<td><img src="image6.png" alt="Diagram" /></td>
</tr>
<tr>
<td>Posterior: 11%, BF=6.9</td>
<td>Posterior: 7%, BF=0.22</td>
</tr>
</tbody>
</table>
Sub topology: Greenlander, Koryak and Athabaschan.

Sub topology: Greenlander, Koryak, Athabaschan and Saqqaq.

Posterior: 56%, BF=1.8

Posterior: 43%, BF=241

Posterior: 19%, BF=12

Posterior: 10%, BF=56

Posterior: 12%, BF=7.5

Posterior: 6%, BF=957

Figure 15: The same as Figure 14 but for other subsets of populations.