The University of Adelaide

School of Mathematical Sciences

# The pairwise sequentially Markovian coalescent model - background information

*Author:*
Alex Jackson

*Supervisors:*
Prof. Nigel Bean
+ his 2 sidekicks


THE UNIVERSITY of ADELAIDE

# Contents



# List of Algorithms

# List of Figures



# List of Tables

**TODO LIST:**

- Sorry, been having some trouble with BibTeX, will get it working over the weekend.

**The Beantles - Let It Bean**

Let it bean, let it bean
Let it bean, let it bean
Red-penned words of wisdom, let it bean

# 1 Introduction

The aim of this project is to reconstruct ancient population dynamics of ~~iconic Australian animals~~ the iconic Australian bison **(WHAT TYPE?)**, to investigate past bottleneck events and the possible resulting lack of genetic diversity. This will be undertaken by using the *pairwise sequentially Markovian coalescent* (PSMC) model suggested by Li and Durban (**REF...**).

Before the PSMC model can be discussed, we will cover the background theory: the Wright-Fisher model, tree likelihoods, Bayesian statistics, Skyline Plots, recombination, Hidden Markov models and maximisation algorithms.

## 1.1 Coalescent Theory

### 1.1.1 The Wright-Fisher Model

In order to model population changes over time, we need some method of modelling genetic drift. One way we can do this is with the Wright-Fisher model. This method has the frequency of alleles in one generation drawn at random from the frequency of alleles in the previous generation.

### 1.1.2 The Coalescent Model

The coalescent model aims to trace individuals in a population back to the *most recent common ancestor* (MRCA). The pioneering mathematician in this field was Kingman **REF**. The theory involves going back in time, by having children "select" their parents from the previous generation according to the Wright-Fisher model. A coalescent event occurs when two children select the same parent, until all modern individuals can be shown to have descended from a single ancestor.

The assumptions are

- children choose their parents uniformly at random,
- time is discretised into non-overlapping generations of constant size $N$, and
- and mutations do not affect an individual's fitness.

Time is rescaled such that one unit of scaled time corresponds to $N$ generations passing. It can be shown that for $k$ lineages in a population of $N$, the time until a coalescent event is distributed exponentially with rate $\binom{k}{2}$.

Other elements can be added to this simple model, such as population dynamics, population substructure, recombination, selection, and positive or negative mutations. For example, in section 1.4, we discuss Skyline Plots, which allow use to estimate ancient population sizes, based upon the above theory.

**yeah i still don't really understand where wright-fisher ends and coalescent begins...**

## 1.2 Tree Likelihood Calculations

To assess if the coalescent trees we construct are sensible, we use likelihood calculations.

### 1.2.1 Terminology

- A *bifurcating* genealogical tree has at most two children selecting any one parent. We assume this is true in coalescent theory as the population is large, so the probability of three or more children selecting a single parent is negligible..
- A *node* is any point on the tree where two branches meet or a branch terminates.
- A *leaf* is a node without any children (i.e. known sequences).
- An *ancestral node* is an internal node (i.e. with children, and having an unknown sequence).
- The *root node* is the internal node furthest back in time, corresponding to the MRCA.
- *Branch lengths* give the times between coalescent events (or recombination events - see Section 1.5) for one lineage. Mutations can occur in this time.

### 1.2.2 Likelihood Functions

The *likelihood* of a set of parameters $\boldsymbol{\theta}$, given observations $\boldsymbol{x}$, is equal to the probability of those $\boldsymbol{x}$ given parameters $\boldsymbol{\theta}$. In equation form, the *likelihood function* $L(\boldsymbol{\theta}; \boldsymbol{x})$ (a function

of $\boldsymbol{\theta}$) satisfies

$$L(\boldsymbol{\theta}; \boldsymbol{x}) = P(\boldsymbol{x}|\boldsymbol{\theta}). \tag{1}$$

For these problems, $\boldsymbol{x}$ is the observed sequences, $\boldsymbol{\theta}$ is model parameters (e.g. mutation rates), and the tree topology and branch lengths are given.

An assumption is that individual nucleotides evolve independently and identically of one another.

Finally, let

- $s_i$ be the DNA sequence of the individual at node $i$;
- $v_i$ be the branch length extending backward in time from node $i$; and
- $P_{i,j}(v)$ be the probability of going from sequence $s_i$ to $s_j$ over time $v$.

One possible model of nucleotide substitution is the *Jukes-Cantor model* **(REFERENCE)**, defined by

$$P_{x,y}(v) = \begin{cases} \frac{1}{4} + \frac{3}{4}e^{-\frac{4}{3}v}, & x = y \\ \frac{1}{4} - \frac{1}{4}e^{-\frac{4}{3}v}, & x \neq y \end{cases} \tag{2}$$

to model mutations, where $x$ and $y$ are the identity of single nucleotides (i.e. A, C, G or T), and $v$ is time. In the Jukes-Cantor model, when a mutation occurs there is an equal chance of selecting any other nucleotide.

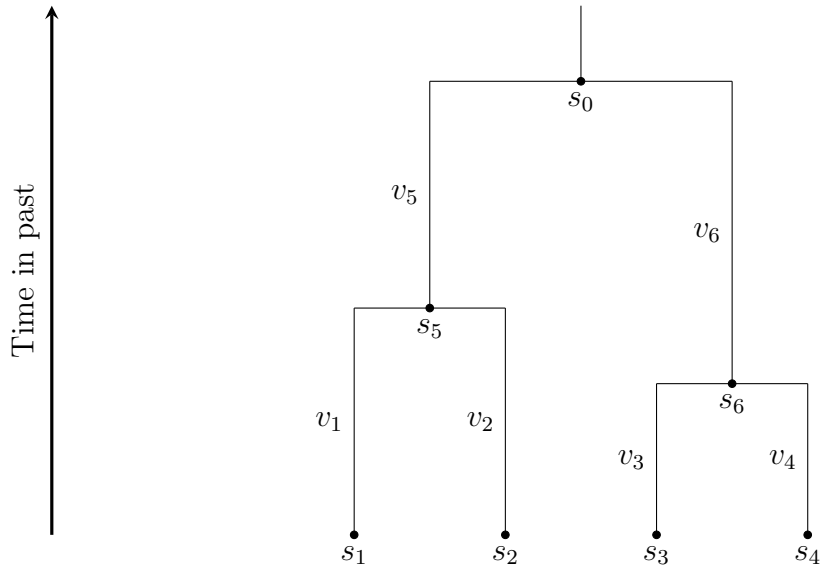


*Figure 1: A simple four-leaf tree.*

For a simple 4-leaf tree (see Figure 1), the likelihood function is given by

$$L(\boldsymbol{\theta}; \boldsymbol{x}) \;=\; \pi_0 P_{s_0,s_5}(v_5) P_{s_5,s_1}(v_1) P_{s_5,s_2}(v_2) P_{s_0,s_6}(v_6) P_{s_6,s_3}(v_3) P_{s_6,s_4}(v_4).$$

The probability of the root node sequence is denoted $\pi_0$. As beyond the MRCA is assumed to be in equilibrium (under the Jukes-Cantor model), all possibilities of $s_0$ are equally likely (and thus $\pi_0$ doesn't actually depend on $s_0$). The sequences at the leaves $\{s_1, s_2, s_3, s_4\}$ are known. But we don't know the sequences at the internal nodes, so we must use the Law of Total Probability (LOTP) to account for all of the possible internal sequences. This expression can then be simplified by grouping the different internal states, to give the following expression.

$$L(\boldsymbol{\theta}; \boldsymbol{x}) \quad = \quad \sum_{\text{all } s_0} \pi_0 \left\{ \sum_{\text{all } s_5} P_{s_0,s_5}(v_5) P_{s_5,s_1}(v_1) P_{s_5,s_2}(v_2) \right\} \left\{ \sum_{\text{all } s_6} P_{s_0,s_6}(v_6) P_{s_6,s_3}(v_3) P_{s_6,s_4}(v_4) \right\}.$$

This allows us to find the most likely set of sequences to fit the tree.

## 1.3 Bayesian Statistics

Section 1.2.2 has discussed tree calculations with known properties, such as branch lengths and tree topology. But these parameters are unknown in advance. An alternative approach to statistics, *Bayesian statistics*, can assist with this problem.

There are two main approaches to statistics - *frequentist*, and Bayesian. The differing schools of thought arise from different interpretations of probability. From a frequentist perspective, probability is the relative frequency of an event, if the event was repeated many times. The parameters which determine the outcome of random variables are fixed, but unknown. On the other hand, the Bayesian view of probability is a degree of belief in a proposition. Random variable parameters are characterised by a density function, which assigns degrees of belief to possible values of the parameters.

Bayesian statistics uses a generalisation of Bayes' Theorem,

$$f(\boldsymbol{\theta}|\boldsymbol{y}, C) = \frac{L(\boldsymbol{\theta}; \boldsymbol{y}, C) f(\boldsymbol{\theta}|C)}{P(\boldsymbol{y}|C)}. \tag{3}$$

We start with the *prior distribution* $f(\boldsymbol{\theta}|C)$, which represents our belief about the parameters $\boldsymbol{\theta}$ before we've observed the data (based upon some background information $C$). Once we've observed our data $\boldsymbol{y}$, we find the *likelihood function* $L(\boldsymbol{\theta}; \boldsymbol{y}, C)$ of observing the data for different values of the parameters. By modifying the prior with the likelihood, we obtain the *posterior distribution* $f(\boldsymbol{\theta}|\boldsymbol{y}, C)$, which is our updated beliefs about $\boldsymbol{\theta}$ once we've taken the observations $\boldsymbol{y}$ into account. The denominator $P(\boldsymbol{y}|C)$ is just a normalising constant known as the *marginal likelihood*. See Figure 2.

This process can be iterated. When more data arises, the old posterior becomes the new prior. We can then work out a new likelihood function, and update our beliefs about the parameters again by finding a new posterior.

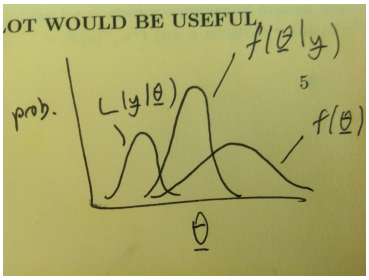


*Figure 2: Bayesian statistics involves updating prior beliefs with data. Starting from the prior $f(\boldsymbol{\theta}|C)$, we find the likelihood $L(\boldsymbol{\theta}; \boldsymbol{y}, C)$and "combine" them to give our posterior $f(\boldsymbol{y}|\boldsymbol{\theta}, C)$, which is a distribution that gives us information about $\boldsymbol{\theta}$ with higher confidence.* ***I NEED TO FIGURE OUT HOW TO DRAW THIS PROPERLY, ALSO MY LIKELIHOOD IS THE WRONG WAY ROUND***

In this field, the parameters of interest may be things like tree branch lengths.

### 1.3.1 Approximate Bayesian Computing

In the majority of real-life problems, it is difficult or impossible to calculate the exact posterior distribution. This may be due to the marginal likelihood being computationally intractable, or the likelihood function may not exist. However, there are a variety of methods for sampling from the posterior distribution. One of these is *Approximate Bayesian Computing* (ABC), which relies upon efficient data simulation and avoids calculating the likelihood. This technique is derived from another method called the *Rejection-Acceptance Algorithm*. **REF THESE TWO.** ABC starts with some observed data $\boldsymbol{x}_{\text{obs}}$. The steps in Algorithm 1 are then followed.

If in Algorithm 1, $\epsilon = 0$ and $S$ is a sufficient statistic, then a sample from the exact posterior distribution is obtained. If not, then an approximate posterior sample is obtained.

### 1.3.2 Markov Chain Monte Carlo

*Markov Chain Monte Carlo* (MCMC) is another method of sampling from the posterior distribution. It avoids calculating the marginal likelihood by taking the ratio of posterior densities. MCMC works by using a Markov chain which has the posterior distribution of the parameter $\boldsymbol{\theta}$, as the equilibrium distribution of the chain. With a set of observations $\boldsymbol{y}$, the method is as follows in Algorithm 2. **REF METROPOLIS HASTINGS**

**Algorithm 1** Approximate Bayesian Computing.

1: $n = 1$.
2: **while** $n \leq N$ (where $N$ is the desired number of samples) **do**
3: Sample $\boldsymbol{\theta^*}$ from the prior distribution $f(\boldsymbol{\theta})$.
4: Simulate data $\boldsymbol{x^*}$ from your random variable of interest $X_{\boldsymbol{\theta^*}}$, using your sampled parameters $\boldsymbol{\theta^*}$.
5: Using a summary statistic $S(\boldsymbol{x})$ of the parameter of interest $\boldsymbol{\theta}$ and a distance function $\rho(S(\boldsymbol{x}), S(\boldsymbol{y}))$, calculate $\rho(S(\boldsymbol{x^*}), S(\boldsymbol{x}_{\text{obs}}))$.
6: **if** $\rho(S(\boldsymbol{x^*}), S(\boldsymbol{x}_{\text{obs}})) \leq \epsilon$ for some tolerance value $\epsilon$ **then**
7: $\boldsymbol{\theta_n} = \boldsymbol{\theta^*}$ (accept $\boldsymbol{\theta^*}$ as a sample from the posterior distribution.)
8: $n \leftarrow n + 1$
9: **else**
10: Discard $\boldsymbol{\theta^*}$.
11: **end if**
12: **end while**

The algorithm gives a set of posterior samples $\{\boldsymbol{\theta_1}, \boldsymbol{\theta_2}, \cdots, \boldsymbol{\theta_T}\}$. Since we started the Markov chain from an arbitrary starting point, the early posterior samples are not taken from the Markov chain's equilibrium distribution. Thus, they do not accurately represent the posterior distribution. Therefore, early parameters are removed to give a sample which represents the true posterior distribution. This is known as the *burn-in*. The samples can also be *thinned* (only retaining every $n^{\text{th}}$ point) to reduce autocorrelation if necessary, and also to bring the data set down to a processable size.

From these calculations, we cam work out information such as population dynamics.

## 1.4 Bayesian Skyline Plots

An assumption of coalescent theory is that the population size remains constant over time. However, in reality this is not the case. *Bayesian Skyline Plots* can be used to model ancient populations dynamics.

Using procedures such as MCMC, a tree with branch lengths can be calculated from a sample of sequences. The $n-1$ coalescent times $t_2, t_3, \cdots, t_n$ can be obtained, as can the inter-event times $g_2, g_3, \cdots, g_n$ and the mean population size estimates $\widehat{M_2}, \widehat{M_3}, \cdots, \widehat{M_n}$.

**PIC OF GETTING THESE VALUES.**

Under the assumption that the population size $N(t)$ can only change at coalescent events (i.e. $N(t) = M_i$ for $t_i \leq t < t_{i+1}$), the modified likelihood function becomes

$$P(g_i|t_i) = \frac{\binom{i}{2}}{M_i} e^{g_i \binom{i}{2}/M_i}, \text{ likelihood? why the P and what's the param/data?} \quad (5)$$

**Algorithm 2** Markov Chain Monte Carlo.

1: $t = 1$.
2: Choosing an sensible starting parameter set $\boldsymbol{\theta_1}$.
3: **while** $t < T$ (where $T$ is the desired number of samples) **do**
4: Use the *proposal density* $Q(\boldsymbol{\theta'}|\boldsymbol{\theta_t})$ to generate a new set of parameters $\boldsymbol{\theta'}$, based upon the previous set $\boldsymbol{\theta_t}$ (this is the Markov property). For example, if we were looking for posterior samples of branch length $v_i$, we could have the $t+1^{\text{th}}$ branch $v_i^{(t+1)} \sim N(v_i^{(t)}, \sigma^2)$ (for a sensible value of $\sigma^2$).
5: Calculate the ratio

$$\begin{aligned} \alpha &= \frac{f(\boldsymbol{\theta'}|\boldsymbol{y})}{f(\boldsymbol{\theta_t}|\boldsymbol{y})} \\ &= \frac{L(\boldsymbol{\theta'};\boldsymbol{y})f(\boldsymbol{\theta'})/P(\boldsymbol{y})}{L(\boldsymbol{\theta_t};\boldsymbol{y})f(\boldsymbol{\theta_t})/P(\boldsymbol{y})} \\ &= \frac{L(\boldsymbol{\theta'};\boldsymbol{y})f(\boldsymbol{\theta'})}{L(\boldsymbol{\theta_t};\boldsymbol{y})f(\boldsymbol{\theta_t})} \text{ (Observe the marginal likelihoods cancel.)} \end{aligned} \tag{4}$$

of the posterior probabilities (i.e. evaluate the relative posterior density functions at $\boldsymbol{\theta'}$ and $\boldsymbol{\theta_t}$).
6: **if** $\alpha \geq 1$ **then**
7: Accept $\boldsymbol{\theta_{t+1}} = \boldsymbol{\theta'}$.
8: **else**
9: Take

$$\boldsymbol{\theta_{t+1}} = \begin{cases} \boldsymbol{\theta'}, & \text{with probability } \alpha \\ \boldsymbol{\theta_t}, & \text{with probability } 1-\alpha \end{cases}.$$

10: **end if**
11: $t \leftarrow t+1$.
12: **end while**

which has MLE

$$\widehat{M_i} = \binom{i}{2} g_i. \tag{6}$$

The estimated population sizes $\widehat{M_i}$ are then plotted over their respective intervals to give Skyline Plots. Programs such as BEAST simulate thousands of Skyline Plots and average them to give a smooth plot known as a *Skyride*.

**SKYLINE/RIDE PIC.**

We can update the estimates with the belief that $M_i \sim \text{Exp}(M_{i-1})$, with a scale-invariant prior for $M_1$.

## 1.5 Ancestral Recombination Graphs

Thus far, we have only considered mutations as driving genetic change. However, this neglects another important phenomenon - *recombination*.

*Diploid* organisms receive genetic material from both parents, and "mix" it together via recombination (see Figure 3). **USE ARG PAPER I DOWNLOADED AS REF.**

Each parent gives their offspring a piece of genetic material.

The offspring recombines the genetic material.

*Figure 3: A highly simplified example of recombination, which occurs at the point marked by dashed lines.*

We need to incorporate recombination into the coalescent model. In the coalescent approximation, the "death rate" (i.e. the rate of coalescent events), is $\mu_k = \binom{k}{2} = \frac{k(k-1)}{2}$ when there are $k$ lineages. The rate of recombination is $\lambda_k = \frac{k\rho}{2}$, for $\rho$ small.

When a recombination event occurs, the place on the genome at which the recombination happens is randomly assigned by a realisation of a distribution on (0,1), where we take 0 and 1 to be the start and end of the DNA molecule). The coalescent process with recombination is visualised with ARGs, *ancestral recombination graphs* (not trees, as there can be loops in the ARGs). See Figure 4.

Each individual site (or segment of genetic material between recombination points) can have its own *marginal tree* (see Figure 4), with corresponding MRCA. The marginal trees are found by the following procedure.

- Define recombination point $j$ to be $r_j \in (0, 1)$.

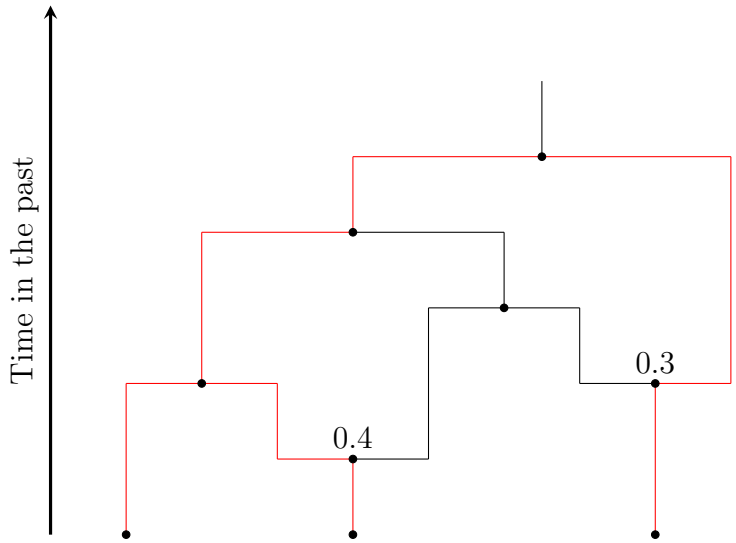


*Figure 4: A simple ARG with three leaves and two recombination events. The marginal tree for some site $x \in (0.3, 0.4)$ is shown in red.*

- When identifying a marginal tree at site $x \in (0, 1)$, head left (on the graph) at $r_j$ if $x < r_j$ and right if $x > r_j$.

As a recombination event increases the number of lineages, this allows us to probe deeper into species' past.

## 1.6 Hidden Markov Models

In standard Markov chains, the state of the system is observed directly. However, in *Hidden Markov models* (HMMs), the states are not observed - instead, observations are probabilistic functions of the states. In our case, states could be the time until most recent common ancestor, and observations could be heterozygousity, for different sections of the genome. Conditional upon knowing the order of states, observations are independent. **REF THE SPEECH RECOG. PAPER.**

The elements of a discrete-time Hidden Markov model are listed below.

- The $N$ possible states are

$$S = \{S_1, S_2, \cdots, S_N\}.$$

  We denote the state the chain is at time $t$ as $q_t$.
- The $M$ possible observations (otherwise known as *symbols*) for each state,

$$V = \{v_1, v_2, \cdots, v_M\}.$$

- The *state transition probability distribution*, $A = \{a_{i,j}\}$, where

$$a_{i,j} = P(q_{t+1} = S_j | q_t = S_i)$$

  for $1 \leq i, j \leq N$. Note this implies a time-homogeneous Markov Chain.
- The *observation symbol probability distribution*, $B = \{b_j(k)\}$, where

$$b_j(k) = P(v_k | q_t = S_j)$$

  for $1 \leq j \leq N$ and $1 \leq k \leq M$. $B$ is also known as the *emission probability matrix*.
- The *initial state distribution*, $\boldsymbol{\pi} = \{\pi_i\}$, where

$$\pi_i = P(q_1 = S_i)$$

  for $1 \leq i \leq N$.

There are three main problems in HMM analyses.

1. *Evaluation*: Given a sequence of observations $\boldsymbol{O} = (O_1, O_2, \cdots, O_T)$ and parameters $\boldsymbol{\lambda} = \{A, B, \boldsymbol{\pi}\}$, how do we calculate the probability of seeing the observations $P(\boldsymbol{O}|\boldsymbol{\lambda})$?
2. *Decoding*: Given $\boldsymbol{O}$ and $\boldsymbol{\lambda}$, how do we choose an "optimal" state sequence $\boldsymbol{Q} = (q_1, q_2, \cdots, q_T)$?
3. *Parameter estimation*: How do we choose $\boldsymbol{\lambda}$ to maximise $L(\boldsymbol{\lambda}; \boldsymbol{O})$?

**I'm not sure whether we are calculating a likelihood or a probability?**

We will just consider problem 1 for now (methods for solving problem 3 can be found in Sections 1.7 and 1.8). We begin by considering a state sequence $\boldsymbol{Q} = (q_1, q_2, \cdots, q_T)$. As the observations are independent given the states, we have

$$\begin{aligned} P(\boldsymbol{O}|\boldsymbol{Q}, \boldsymbol{\lambda}) &= \prod_{t=1}^{T} P(O_t | q_t, \boldsymbol{\lambda}) \\ &= b_{q_1}(O_1) b_{q_2}(O_2) \cdots b_{q_T}(O_T). \end{aligned}$$

Additionally, consider the probability of that particular state sequence,

$$P(\boldsymbol{Q}|\boldsymbol{\lambda}) = \pi_{q_1} a_{q_1,q_2} a_{q_2,q_3} \cdots a_{q_{T-1},q_T}.$$

Therefore, by the Law of Total Probability we obtain the result

$$\begin{aligned} P(\boldsymbol{O}|\boldsymbol{\lambda}) &= \sum_{\text{all } \boldsymbol{Q}} P(\boldsymbol{O}, \boldsymbol{Q}|\boldsymbol{\lambda}) \\ &= \sum_{\text{all } \boldsymbol{Q}} P(\boldsymbol{O}|\boldsymbol{Q}, \boldsymbol{\lambda}) P(\boldsymbol{Q}|\boldsymbol{\lambda}). \end{aligned}$$

However, this cannot be computed in a feasible length of time.

### 1.6.1 Forward Algorithm

Instead, the *forward algorithm* can be used to calculate the likelihood in a short time. For the forward algorithm, consider the *forward variable*

$$\alpha_t(i) = P(O_1, O_2, \cdots, O_t, q_t = S_i|\boldsymbol{\lambda}). \tag{7}$$

This is calculated inductively, by first initialising with

$$\alpha_1(i) = \pi_i b_i(O_1) \tag{8}$$

for $1 \leq i \leq N$. Next, $\alpha_{t+1}(j)$ is calculated inductively (for $1 \leq t \leq T-1$ and $1 \leq j \leq N$) via the formula

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{i,j}\right] b_j(O_{t+1}). \tag{9}$$

Finally, we obtain the probability

$$\begin{aligned} P(\boldsymbol{O}|\boldsymbol{\lambda}) &= \sum_{i=1}^{N} P(\boldsymbol{O}, q_T = S_i|\boldsymbol{\lambda}) \text{ (by the LOTP)} \\ &= \sum_{i=1}^{N} \alpha_T(i) \text{ (by definition of } \alpha_T(i)). \end{aligned} \tag{10}$$

**NEED TO ADD MORE HERE APPARENTLY.**

## 1.7 The Expectation-Maximisation Algorithm

The Expectation-Maximisation (EM) Algorithm is a technique for finding MLEs which cannot be solved analytically. It is a useful method when there are missing data or "latent variables", e.g. for a HMM when we don't know the parameters $\boldsymbol{\theta}$ and the sequence of hidden states $\boldsymbol{Q}$. For the list of steps, see Algorithm 3.

## 1.8 The Baum-Welch Algorithm

The Baum-Welch Algorithm is also used to find the unknown parameters of a HMM (see Algorithm 4).

In the next section, we use the Baum-Welch Algorithm to calculate the parameters in the PSMC model.

**Algorithm 3** Expectation-Maximisation.

1: $t = 1$.
2: Pick an arbitrary starting set of parameters $\boldsymbol{\theta^{(1)}}$ (or base the set upon prior information).
3: Expectation step: Calculate

$$\mathcal{E}(\boldsymbol{\theta}|\boldsymbol{\theta^{(t)}}) = E_{\boldsymbol{Q}|\boldsymbol{X},\boldsymbol{\theta^{(t)}}}[\log L(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{Q})], \tag{11}$$

which is the expectation of the log-likelihood function with respect to the conditional distribution of $\boldsymbol{Q}$, given observations $\boldsymbol{X}$ and the current parameter estimates $\boldsymbol{\theta^{(t)}}$.
4: Maximisation step: Calculate

$$\boldsymbol{\theta^{(t+1)}} = \arg\max_{\boldsymbol{\theta}} \mathcal{E}(\boldsymbol{\theta}|\boldsymbol{\theta^{(t)}}), \tag{12}$$

which finds the next parameter set $\boldsymbol{\theta^{(t+1)}}$ based upon the old parameter set $\boldsymbol{\theta^{(t)}}$ via maximisation.
5: $t \leftarrow t + 1$.
6: Iterate steps 3, 4 and 5 until a desired level of convergence is reached.

## 1.9 The Pairwise Sequentially Markovian Coalescent Model

The *pairwise sequentially Markovian coalescent* (PSMC) model is used to find relative population sizes in the past. Using diploid genome sequences, it estimates past population sizes relative to $N_0$, (the time 0 population size) in a discrete-time manner. **REF PAPER.**

The PSMC model uses a HMM to describe events. To begin with, the genome is divided into non-overlapping 100 base pair (bp) "bins". Additionally, continuous time is discretised such that the intervals are evenly spaced on a log scale. For each bin $\mathcal{B}$, its corresponding hidden state is the discrete time interval $k$ into which the bin $\mathcal{B}$'s *time until most recent common ancestor* (TMRCA) can be found. Thus, the state space $S$ of the HMM is

$$S = \{k : k = [t_k, t_{k+1})\}, \tag{21}$$

for $k = 0, 1, \cdots, n$ and $t_{n+1} = \infty$ ($n$ defines the number of intervals we want to split time into). In other words, a bin is assigned state $k$ if its TMRCA falls within $[t_k, t_{k+1})$.

The observations we observe are heterozygousity (denoted "1") or homozygousity (denoted "0"). Heterozygousity is defined by a difference of at least one nucleotide between the two chromosomes. The model can also account for missing data (denoted ".") but we will ignore this for now. Thus, the set of observation symbols is

$$V = \{0, 1\}. \tag{22}$$

**Algorithm 4** Baum-Welch.

1: $n = 1$.

2: Set parameters $\boldsymbol{\lambda} = (A, B, \boldsymbol{\pi})$ with arbitrary initial conditions $\boldsymbol{\lambda_1} = (A_1, B_1, \boldsymbol{\pi_1})$ (or use prior information if available). $A$, $B$ and $\boldsymbol{\pi}$ are the HMM parameters as defined in Section 1.6.

3: For the forward variable as defined in 1.6.1 and the *backward variable* $\beta_i(t)$ as defined and characterised below by

$$\begin{aligned} \beta_i(t) &= P(O_{t+1}, O_{t+2}, \cdots, O_T | q_t = S_i, \boldsymbol{\lambda}) && (13) \\ \beta_i(t) &= \sum_{j=1}^{N} \beta_j(t+1) a_{i,j} b_j(O_{t+1}) && (14) \\ \beta_i(T) &= 1, && (15) \end{aligned}$$

we can calculate *temporary variables* $\gamma_i(t)$ (the probability of being in in state $S_i$ at time $t$, given observations $\boldsymbol{O}$ and parameters $\boldsymbol{\lambda_n}$) and $\zeta_{i,j}(t)$ (the probability of being in states $S_i$ and $S_j$ at times $t$ are $t+1$ respectively, given $\boldsymbol{O}$ and $\boldsymbol{\lambda_n}$). Using Bayes' Theorem, we find them as

$$\begin{aligned} \gamma_i(t) &= P(q_t = S_i | \boldsymbol{O}, \boldsymbol{\lambda_n}) = \frac{\alpha_i(t)\beta_i(t)}{\sum_{k=1}^{N} \alpha_k(T)}, && (16) \\ \zeta_{i,j}(t) &= P(q_t = S_i, q_{t+1} = S_j | \boldsymbol{O}, \boldsymbol{\lambda_n}) = \frac{\alpha_i(t) a_{i,j} \beta_j(t+1) b_j(O_{t+1})}{\sum_{k=1}^{N} \alpha_k(T)}. && (17) \end{aligned}$$

We use our estimates of $a_{i,j}$ etc. to calculate these values.

4: Obtain an updated estimate $\boldsymbol{\lambda_{n+1}} = (A_{n+1}, B_{n+1}, \boldsymbol{\pi}_{n+1})$ of parameters $\boldsymbol{\lambda}$, where

$$\begin{aligned} a_{i,j} &= \frac{\sum_{i=1}^{T-1} \zeta_{i,j}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)}, && (18) \\ b_i(v_k) &= \frac{\sum_{t=1}^{T} \boldsymbol{\mathcal{I}}_{\{O_t = v_k\}} \gamma_i(t)}{\sum_{t=1}^{T} \gamma_i(t)} \text{ for } \boldsymbol{\mathcal{I}}_{\{O_t = v_k\}} = \begin{cases} 1, & O_t = v_k \\ 0, & O_t \neq v_k \end{cases}, && (19) \\ \pi_i &= \gamma_i(1). && (20) \end{aligned}$$

5: $n \leftarrow n + 1$.

6: Iterate steps 3, 4 and 5 until a desired level of convergence is reached.

The emission probabilities $e_k(i)$ of observation $i \in V$ given a bin has state $k$ (i.e. hetero/homozygousity given a TMRCA interval), are found by

$$\begin{aligned} e_k(1) &= \left(1 - \frac{\pi_k}{C_\sigma \sigma_k}\right)^{\theta/\rho} && (23) \\ e_k(0) &= 1 - e_k(1), && (24) \end{aligned}$$

where $\pi_k, C_\sigma$ and $\sigma_k$ are functions of $\lambda_k$, and $\lambda_t$ is a piecewise-constant function describing the relative population size at time $t$. We assume the mutation rate $\theta$ and recombination rate $\rho$ are known. **REF S.I. of PSMC paper**

The state transition probability (i.e. a recombination event), for going from bin $\mathcal{B}$ in state $k$ to bin $\mathcal{B}+1$ in state $l$ is

$$\begin{aligned} p_{k,l} &= P(\text{bin } \mathcal{B}+1 \text{ falls in state } l | \text{bin } \mathcal{B} \text{ falls in state } k) \\ &= \frac{\pi_k}{C_\sigma \sigma_k} q_{k,l} + \delta_{k,l} \frac{\pi_k}{C_\sigma \sigma_k}, \end{aligned} \tag{25}$$

where $q_{k,l}$ is a conditional transition probability and **I think $\delta$ is the Dirac delta function or something?**

To estimate the parameters, an expectation-maximisation algorithm is used. The expectation step was begun for a constant population size over time, that is $\lambda_t$ constant for all $t$. The expectation step was found analytically, while the Baum-Welch Algorithm was used for the maximisation step **(I dunno how these work at the moment)**.

Information on population dynamics and past bottleneck events can be useful for conservation efforts.