Markov-Chain Monte Carlo Methods for the Construction of Phylogenetic Trees

Tessa Rhinehart
Department of Mathematics and Statistics
Swarthmore College

December 17, 2016

1 Introduction

An understanding of the evolutionary history of a group of organisms is crucial to a wide variety of disciplines. Evolutionary relationships can reveal the origins of pathogens, improve evolutionary theory, inform forensic investigations, and more. Typically, these relationships are represented by a phylogenetic tree or *phylogeny*, a branching structure where the tips of branches represent current species and each branch point represents two species' common ancestor (Fig. 1).

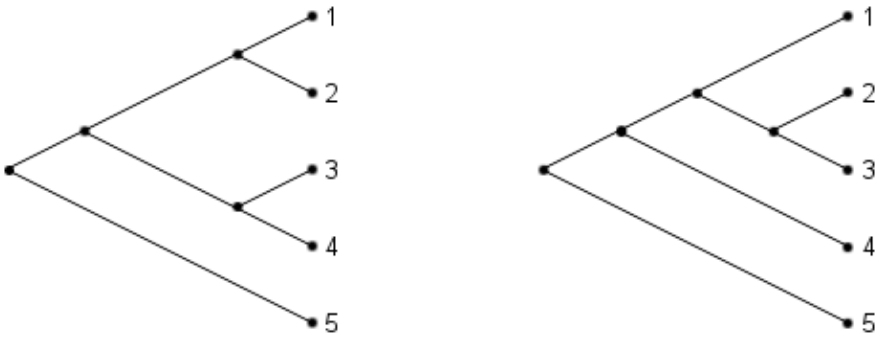

Figure 1: Two possible bifurcating, rooted phylogenetic trees for five species.

For most of the history of the field, biologists studying phylogenetics primarily used species' behavior and physical characteristics to produce phylogenies. Now, recent improvements in technology allow the sequence of an organism's DNA to be determined incredibly

quickly and cheaply. With this improvement, DNA-based methods are increasingly used to reconstruct the most likely phylogenetic trees.

DNA is a chemical code contained within the cells of every organism. This code specifies biological structures, such as proteins, and is composed of four nucleotides: two purines, adenine (A) and guanine (G), and two pyrimidines, cytosine (C) and thymine (T). Random mutations, such as substitutions of one nucleotide for another, accumulate in this code over time; thus, the more generations between an ancestor and its descendants, the less similar the descendant's DNA is likely to be in comparison to its ancestor's. In general, species with more similar genetic codes are predicted to have a more recent common ancestor, while those with fewer similarities are more distantly related.

Methods for constructing phylogenetic trees fall into two broad categories: distance-based methods, and character-based methods. Distance-based methods implement a distance metric to quantify the degree of difference between each pair of sequences. While these methods are computationally speedy, they tend to be inaccurate, as they lose much information about the sequences by compressing phylogenetic information down to a distance [2]. In contrast, character-based methods are more computationally expensive, but can be highly accurate. The most common character-based method, maximum likelihood, finds the single most likely tree given the data. This method has drawbacks, especially in cases where more than one tree has a probability: maximum likelihood will output only the most likely tree, without revealing other candidates.

A relatively recently described character-based method, Bayesian inference, instead estimates the probability that every possible tree explains the genetic data, allowing researchers to compare the most likely trees [6]. Bayesian inference of phylogeny is accomplished using Markov-chain Monte Carlo (MCMC) methods, especially the Metropolis-Hastings algorithm, which create a series of trees called a *Markov chain*. Metropolis-Hastings constructs a this Markov chain such that the number of times each tree is visited by the chain is proportional to its conditional probability given some genetic data.

This paper begins by introducing the mathematical phylogenetic model under consideration in Section 2. Section 3 defines Markov chains, explains a particular algorithm for creating Markov chains, and describes one method of proposing candidate trees, the Li 1996 algorithm. Section 4 proves that the Markov chain produced does visit trees proportionally to their Bayesian probability. Finally, Section 5 discusses some practical considerations for Markov-chain Monte Carlo methods.

2 Phylogenetic models

A phylogenetic model includes a tree and a nucleotide substitution model, a model for how mutations arise in DNA. A tree τ is defined as a combination of an unrooted tree u, a root r, present-day DNA sequences $\boldsymbol{D}$, hypothesized ancestral DNA sequences $\boldsymbol{A}$, and a vector of splitting times $\boldsymbol{s}$ [6]. The space of all possible trees is denoted T.

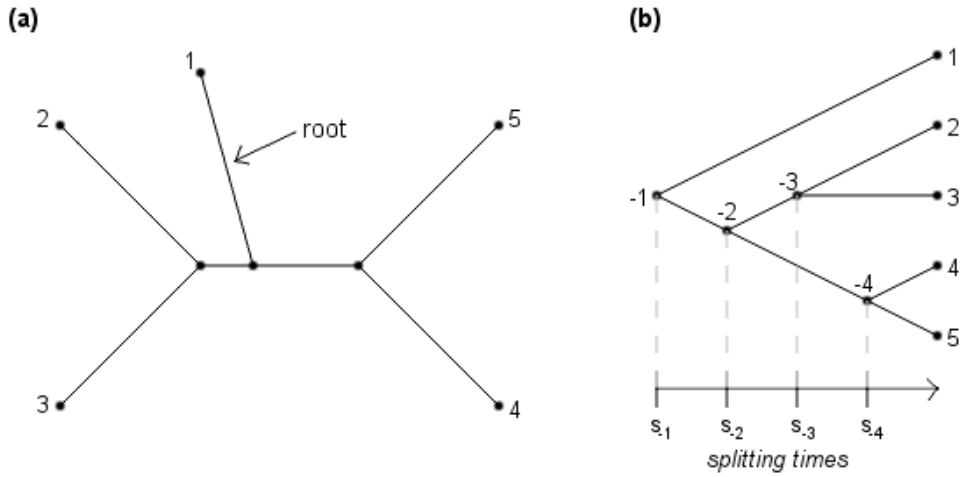

Figure 2: An unrooted tree (a) and a rooted tree (b).

The **unrooted tree** u illustrates the relatedness of species; each species is represented by an external node (Fig. 2a). This tree can be modified to show the direction of evolutionary time by adding a **root**, an extra node that divides one branch and represents the last common ancestor of all the species in the tree (Fig. 2b). Trees are often rooted by including a present-day *outgroup* species, a species known to be less related to the other species in the tree (e.g. Species 1, Fig. 2). For n species, there exist $\frac{(2n-3)!}{2^{n-2}(n-2)!}$ rooted trees (u, r), also known as tree **topologies**.

The external nodes of the n present-day species under consideration are labeled $(1, 2, ..., n)$, and the internal nodes of the rooted tree labeled $(-1, -2, ..., -(n-1))$, where the root is labeled -1. The collected DNA sequences of present-day species are stored in the matrix $\boldsymbol{D}$, where each row $\boldsymbol{d}_i$ is the sequence of species i. Sequences are aligned such that comparable regions of DNA are located at the same position d_{ij} for all species i; all sequences are of uniform length m. Thus, for the set of DNA nucleotides $\mathbb{D} = \{\text{A, C, G, T}\}$, each $\boldsymbol{D}_i$ is an element of $\mathbb{D}^m$.

Each tree includes a hypothesis about the actual DNA sequences of the ancestral species and their splitting times. The matrix $\boldsymbol{A}$ contains the hypothesized DNA sequence of each ancestor $-i$, where $A_{-i} \in \mathbb{D}^m$. The **splitting time** of each ancestor is amount of time since

the ancestor split into two species; the splitting time of present-day species is defined to be 0, and the splitting time of the root is defined as a constant, σ. Ancestral splitting times of each internal node $-i$ are included in the vector $\boldsymbol{s} = \{s_{-1}, s_{-2}, ..., s_{-(n-1)}\}$, so $\boldsymbol{s} \in (0, \sigma)^{n-1}$ (Fig. 2b).

2.1 Nucleotide substitution model

Along with a tree τ, the **nucleotide substitution model** completes a phylogenetic model. This model describes the instantaneous probability that a nucleotide at a given site will mutate to another nucleotide, and crucially affects our determination of relatedness. One of the most commonly used models is the Hasegawa, Kishino, and Yano (1985) [4] model.

Nucleotide substitution involves both *transitions* between purines (A, G) or between pyrimidines (C, T), and *transversions* from a purine to a pyrimidine or vice versa. The HKY85 model allows the rate of transition and transversion to differ where $\kappa > 0$ describes the ratio of transition to transversion. The model also does not assume that the four nucleotides are equally prevalent in the DNA, allowing for four different nucleotide frequences $\lambda_{\mathrm{A}}, \lambda_{\mathrm{C}}, \lambda_{\mathrm{T}}, \lambda_{\mathrm{G}}$. These characteristics make HKY85 appropriate for datasets that have biased base composition, such as animal mitochondrial DNA or GC-rich microbes. For an overall substitution rate $\alpha > 0$, the instantaneous rate of substitution is

$$\frac{\boldsymbol{R}}{\alpha} = \begin{array}{c} \\ \mathrm{A} \\ \mathrm{C} \\ \mathrm{G} \\ \mathrm{C} \end{array} \begin{array}{c} \begin{array}{cccc} \mathrm{A} & \mathrm{C} & \mathrm{G} & \mathrm{T} \end{array} \\ \begin{pmatrix} \sigma_A & \lambda_{\mathrm{A}} & \kappa\lambda_{\mathrm{A}} & \lambda_{\mathrm{A}} \\ \lambda_{\mathrm{C}} & \sigma_C & \lambda_{\mathrm{C}} & \kappa\lambda_{\mathrm{C}} \\ \kappa\lambda_{\mathrm{G}} & \lambda_{\mathrm{G}} & \sigma_G & \lambda_{\mathrm{G}} \\ \lambda_{\mathrm{T}} & \kappa\lambda_{\mathrm{T}} & \lambda_{\mathrm{T}} & \sigma_C \end{pmatrix} \end{array}$$

where diagonals are chosen such that each row sums to 0. Note that this transition matrix describes the instantaneous substitution rate for only one site of DNA. Bayesian phylogenetic reconstruction can be implemented either assuming that each site of DNA evolves independently according to the same distribution, or computing differing rates for different sites. The latter method is more computationally expensive, but may improve accuracy. For one site, the probability of substitution from nucleotide i to j over time t is given by [6]:

$$P_{i\to j}(t) = \begin{cases} \lambda_j + \lambda_j \left(\frac{1}{\nu_j} - 1\right) e^{-\alpha t} + \left(\frac{\nu_j - \lambda_j}{\lambda_j}\right) e^{-\alpha\gamma_j t} & i = j \\ \lambda_j + \lambda_j \left(\frac{1}{\nu_j} - 1\right) e^{-\alpha t} - \left(\frac{\lambda_j}{\nu_j}\right) e^{-\alpha\gamma_j t} & i \neq j, \text{transition} \\ \lambda_j \left(1 - e^{-\alpha t}\right) & i \neq j, \text{transversion} \end{cases}$$

where $\nu_j = \lambda_A + \lambda_G$ if the base is an A or G, $\nu_j = \lambda_C + \lambda_T$ if the base is a T or C, and $\gamma_j = 1+(\kappa-1)\nu_j$. For further explanation of the derivation of $P_{i\to j}(t)$ from the instantaneous rate matrix $\boldsymbol{R}$ see 1001[4, 6]. Assuming that nucleotides evolve independently according to the same distribution, the probability of transitioning from a sequence a_i of length m to a sequence a_j is the product of the probabilities of substitution at each site,

$$P_{a_i\to a_j}(t) = \prod_{k=1}^{m} P_{a_{ik}\to a_{jk}}(t)$$

3 Markov-chain Monte Carlo methods

3.1 Bayesian inference of phylogeny

Which trees are most likely to represent the evolutionary history of a group of organisms of interest? This problem can be investigated using Bayesian inference: given a matrix $\boldsymbol{D}$ of aligned DNA sequences, we find a probability distribution on the space T of all possible trees.

Bayesian inference of phylogeny assumes for a given tree $\tau_i \in T$ a prior probability $f(\tau_i)$. This distribution is often uninformative. For instance, the distribution may consider all rooted trees (u, r) and potential ancestral sequences $\boldsymbol{a}$ to be equally likely, and may assume splitting times s_i are uniformly distributed according to the order constraints of the tree. Given the prior probability and a model of nucleotide substitution, our goal is to find the conditional probability of a tree $\tau \in T$ given the data, $f(\tau|\boldsymbol{D})$, which by Bayes' theorem is

$$f(\tau|\boldsymbol{D}) = \frac{f(\tau)f(\boldsymbol{D}|\tau)}{f(\boldsymbol{D})}.$$

The likelihood $f(\boldsymbol{D})$ is a constant, but it is difficult to calculate analytically: it is a many-dimensional integral over the tree space, involving integration over every possible topology, matrix of ancestral nucleotides, and vector of splitting times [6]. Therefore, instead of calculating the conditional distribution directly, this distribution can be approximated by constructing a Markov chain that samples trees proportional to this probability.

3.2 Markov chains

A **Markov chain** is a sequence of random variables $X = \{x_0, x_1, x_2, ...\}$ where at time $t \geq 0$, the value of the next state x_{t+1} is probabilistically selected based on the current state

x_t according to a certain **transition probability**, $R(x_{t+1} \,|\, x_t)$. The Markov chain can be represented by a matrix $\boldsymbol{R}$ where r_{ab} is the probability of transitioning from state a to state b, $R(x_{t+1} = b \,|\, x_t = a)$.

For instance, consider a collection of states (a, b, c) that can be visited by a Markov chain with transition matrix

$$\boldsymbol{R} = \begin{array}{c} \\ a \\ b \\ c \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{pmatrix} 0.5 & 0.1 & 0.4 \\ 0.2 & 0.4 & 0.4 \\ 0.8 & 0.1 & 0.1 \end{pmatrix} \end{array}$$

Note that each row sums to 1, as each state must transition to some other state. A state vector π_i describes the probability of being in each state at time t_i. Continuing our example, assume that the chain initiates at state a, such that $\pi_0 = (1, 0, 0)$. At time $t = 1$, the Markov chain will be in each state with probability $\pi_1 = \boldsymbol{R} \cdot \pi_0$:

$$\pi_1 = \begin{array}{c} \\ a \\ b \\ c \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{pmatrix} 0.5 & 0.1 & 0.4 \\ 0.2 & 0.4 & 0.4 \\ 0.8 & 0.1 & 0.1 \end{pmatrix} \end{array} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0.5 \\ 0.2 \\ 0.8 \end{pmatrix}$$

We can again multiply π_1 by the transition matrix to find the distribution of states at time $t = 2$, and so on. Therefore, the distribution of states at $t = n$ will be $\pi_n = \boldsymbol{R}^n \cdot \pi_0$. A **stationary distribution** of a Markov chain is a state vector Π such that $\Pi = \boldsymbol{R} \cdot \Pi$.

This concept is applied to phylogenetics by generating a Markov chain where each state is a tree τ, and the stationary distribution of the chain is equal to $f(\tau|\boldsymbol{D})$. While several algorithms exist to generate such a Markov chain, we examine one in particular, the Metropolis-Hastings algorithm. The Metropolis-Hastings algorithm creates a chain by either accepting or rejecting proposed states; unlike the similar Metropolis algorithm, it allows for states to be proposed according to an asymmetrical distribution, which increases the speed of convergence to the stationary distribution. We describe the algorithm itself, then prove its stationary distribution is the desired conditional probability.

3.3 The Metropolis-Hastings algorithm

The Metropolis-Hastings algorithm generates a Markov chain $X = \{x_0, x_1, x_2, ..., x_N\}$, where each state x_i is equal to some tree. We will prove in Section 4 that the stationary distribution of this Markov chain is equal to the conditional distribution $f(\tau|\boldsymbol{D})$, such that

for large enough N, the chain visits trees proportional to their conditional probability. The Markov chain is created by starting at an initial tree, modifying that tree to propose a "candidate" tree x', and accepting or rejecting the candidate with probability

$$\alpha(x'|x_t) = \min\left\{1, \frac{f(x'|\boldsymbol{D})q(x', x_t)}{f(x_t|\boldsymbol{D})q(x_t, x')}\right\},$$

where q is determined by the tree proposal procedure as described in Section 3.4. The q term in this acceptance probability corrects for asymmetry in the proposal distribution, which we will see in Section 4 is important for the convergence of the Markov chain. The Metropolis-Hastings acceptance probability contrasts with that of the Metropolis algorithm, for which the proposal distribution is required to be symmetrical such that $q(x', x_t)/q(x_t, x') = 1$.

Notably, the acceptance probability requires the computation of $\frac{f(x'|\boldsymbol{D})}{f(x_t|\boldsymbol{D})}$. Recall that, although the denominator of the conditional probability, $f(\boldsymbol{D})$, is difficult to calculate, it is a constant. Herein lies the reason that MCMC can be used to estimate Bayesian distributions: due to the division of conditional probabilities in the acceptance probability, the algorithm does not compute $f(\boldsymbol{D})$.

Intuitively, this acceptance probability compares the conditional probability of the proposed tree to the current tree. If the proposed tree is more probable, it is automatically accepted. If it is less probable, it is accepted with probability $\alpha < 1$. Thus, the Markov chain visits progressively more probable trees, such that after a period of "burn-in," it samples trees proportional to their probability. Section 5 provides a more thorough discussion of burn-in and other practical concerns.

Metropolis-Hastings algorithm:

1. Set $t = 0$. Initiate the algorithm by selecting a tree and setting it as x_0.
2. Sample a candidate tree x' according to distribution $q(x_t, \cdot)$.
3. Calculate the acceptance probability $\alpha(x'|x_t)$.
4. Randomly sample a number between 0 and 1. If this number is less than or equal to α, set $x_{t+1} = x'$. Otherwise, set $x_{t+1} = x_t$.
5. Increment t. Repeat steps 2–4 N times.

3.4 Proposing candidate trees

The selection method of the next potential tree, the "candidate" x', determines the proposal density $q(x_t, x')$. Proposal methods may either intentionally rearrange the topology of the

tree in every step, or may rearrange the tree continuously such that occasionally, a topology change occurs. Proposals in the former category tend to converge faster; for a comparison of the efficiency of several different types of proposals, see [5]. We examine an algorithm in the former category, nearly identical to that described by Li, *et al.* (1996) [6], in which the tree at the current state is modified in three steps to produce the candidate. Any ancestral node except the root node is randomly selected for modification. We rearrange the topology around this "target" node, then change its splitting time and nucleotide sequence using the following steps.

Step 1. Randomly select any ancestral node except for the root node. Designate this node the target T, its sibling S, and its two children C_1 and C_2 (Fig. 3a). Randomly select one of S, C_1, or C_2 to be the new sibling node S' (Fig. 3a, 3b, and 3c, respectively). Label the remaining nodes C_1', C_2'. Rearrange the topology, preserving each node's attached subtree.

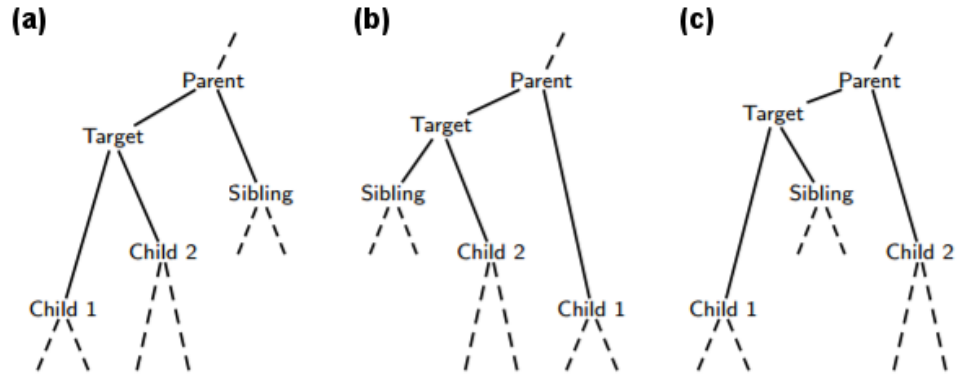

Figure 3: Potential rearrangements of the initial tree (a). Figure adapted from [6].

Step 2. Pick a new splitting time for the target T'. This time, $s_{T'}$, must be between the splitting time s_P of the parent and the time for the nearest child, $s_{C_n} = \max(S_{C_1'}, S_{C_2'})$ and the time of the nearest child–an ancestor cannot be older than its parent or younger than its children. We pick the time $s_{T'}$ from the density

$$g(s_{T'}) = \frac{\sum_{a_{T'}} P_{a_P \to a_{T'}}(s_P - s_{T'}) P_{a_{T'} \to a_{C_1'}}(s_{T'} - s_{C_1'}) P_{a_{T'} \to a_{C_2'}}(s_{T'} - t_{C_2'})}{\int_{s_{C_n}}^{s_P} \sum_{a_{T'}} P_{a_P \to a_{T'}}(s_P - s_{T'}) P_{a_{T'} \to a_{C_1'}}(s_{T'} - s_{C_1'}) P_{a_{T'} \to a_{C_2'}}(s_{T'} - s_{C_2'})\, dt}.$$

Although this density is complex, investigation of each term shows that it is a sensible choice. First, recall from Section 2.1 that expression $P_{a_P \to a_{T'}}(s_P - s_{T'})$ describes the probability of transitioning from the parental nucleotide sequence a_P to the target nucleotide sequence $a_{T'}$ in the amount of time between the parent's splitting time and the target's splitting time. The summand of the numerator corresponds to the probability of three events occurring:

the transition from the parental sequence to the target sequence, and the transition from the parental sequence to each of the child sequences. This probability is summed over all potential target sequences $a_{T'}$, since the new target sequence is yet to be decided. Similarly, the denominator includes this sum, but integrates from the splitting time of the parent to the splitting time of the closest child. Overall, choosing from this density makes it more likely that we will choose high-probability splitting times.

Step 3. Pick a nucleotide sequence for the new target. Similar to Step 2, the sequence $a_{T'}$ is chosen from a distribution such that more likely ancestral sequences are more likely to be chosen:

$$h(a_{T'}) = \frac{P_{a_P \to a_{T'}}(s_P - s_{T'}) P_{a_{T'} \to a_{C_1'}}(s_{T'} - s_{C_1'}) P_{a_{T'} \to a_{C_2'}}(s_{T'} - s_{C_2'})}{\sum_{a_{T'}} P_{a_P \to a_{T'}}(s_P - s_{T'}) P_{a_{T'} \to a_{C_1'}}(s_{T'} - s_{C_1'}) P_{a_{T'} \to a_{C_2'}}(s_{T'} - t_{C_2'})}.$$

Thus, the proposal density $q(x, y)$ is defined as the probability of drawing a particular inner node, multiplied by the probability of selecting one of C_1, C_2, or S to be the new sibling, multiplied by $g(s_{T'})$ and $h(a_{T'})$. Since the denominator of g and the numerator of h cancel,

$$q(x, y) = \frac{1}{n-2} \cdot \frac{1}{3} \cdot \frac{P_{a_P \to a_{T'}}(s_P - s_{T'}) P_{a_{T'} \to a_{C_1'}}(s_{T'} - s_{C_1'}) P_{a_{T'} \to a_{C_2'}}(s_{T'} - t_{C_2'})}{\int_{s_{C_n}}^{s_P} \sum_{a_{T'}} P_{a_P \to a_{T'}}(s_P - s_{T'}) P_{a_{T'} \to a_{C_1'}}(s_{T'} - s_{C_1'}) P_{a_{T'} \to a_{C_2'}}(s_{T'} - t_{C_2'})\, dt}$$

which, finally, defines our acceptance probability.

4 Validity of Metropolis-Hastings

Our goal is to prove that the Metropolis-Hastings algorithm generates a Markov chain with stationary distribution $f(\tau_i|\boldsymbol{D})$. Because a Markov chain may have more than one stationary distribution, we prove first that the Markov chain does converge to a unique stationary distribution. We then show using the detailed balance condition that the stationary distribution is the conditional probability as desired.

4.1 Convergence to unique stationary distribution

A Markov chain converges uniquely from every possible starting position if it is irreducible, aperiodic, and positive recurrent. For a thorough discussion of this fact and the development of the theory behind it, refer to Chapter 4 of Robert & Casella (1999) [7]. A chain is **irreducible** if it can reach any state from any other state. A chain is said to have period

$k > 1$ if it can only return to its current state x_t at times $x_{t+k}, x_{t+2k}, x_{t+3k}, ...$; the chain is **aperiodic** if it does not have a period $k > 1$. One state to which a chain is expected to return an infinite number of times is **recurrent**, and if each return is expected in a finite number of time steps, then the state is **positive recurrent**. If all states are positive recurrent, the chain is said to be positive recurrent.

In the proposal method described by Li, *et al.* (1996), splitting times are sampled from a continuous distribution from $(0, \sigma)$, with the splitting time of the external nodes defined to be 0 and the splitting time of the root defined to be σ. Therefore, the state space for the Markov chain as described is uncountable, making it a Markov *process*. This continuous problem can be approximated well using a discretized state space where times belong to the set $\{0, \frac{\sigma}{z}, \frac{2\sigma}{z}, ..., \frac{z\sigma}{z}\}$ with z large. This modification transforms the uncountable space into a finite one. This distinction is important for the following lemma, which we will later use to prove that the Markov chain is positive recurrent.

Lemma 1. *In a finite state space Markov chain,*

(1) *at least one state is positive recurrent, and*

(2) *all recurrent states are positive recurrent.*

Proof. If a state is positive recurrent, it must also be recurrent. Assume toward a contradiction that no state is recurrent. Thus, every state i is returned to a finite number of times. However, the Markov chain can be arbitrarily long. Once the chain reaches length $1 + \sum_i n_i$, it must return to some state j an additional time: $n_j = n_j + 1$, a contradiction. Therefore, at least one state is recurrent.

As for (2), first note that to guarantee that every recurrent state is returned to an infinite number of times, every recurrent state must be able to return to every other recurrent state. Furthermore, at least one state k in a finite state space Markov chain must be positive recurrent, i.e. there must exist at least one state with a finite expected return time. If this were not the case, all states would have expected return time ∞, meaning that the Markov chain could sample from no state. Finally, since this positive recurrent state k can reach any other recurrent state l, every time the chain visits k, it has a nonzero probability of visiting l. Therefore, every recurrent state l is also positive recurrent. □

Theorem 1. *The Markov chain generated by the Metropolis-Hastings algorithm using the described discretized tree proposal method converges uniquely.*

Proof. We show that the Markov chain is irreducible, aperiodic, and positive recurrent.

The first condition, irreducibility, is proven by demonstrating through induction that the algorithm can reach any topology. The ability to reach any topology is sufficient to reach any

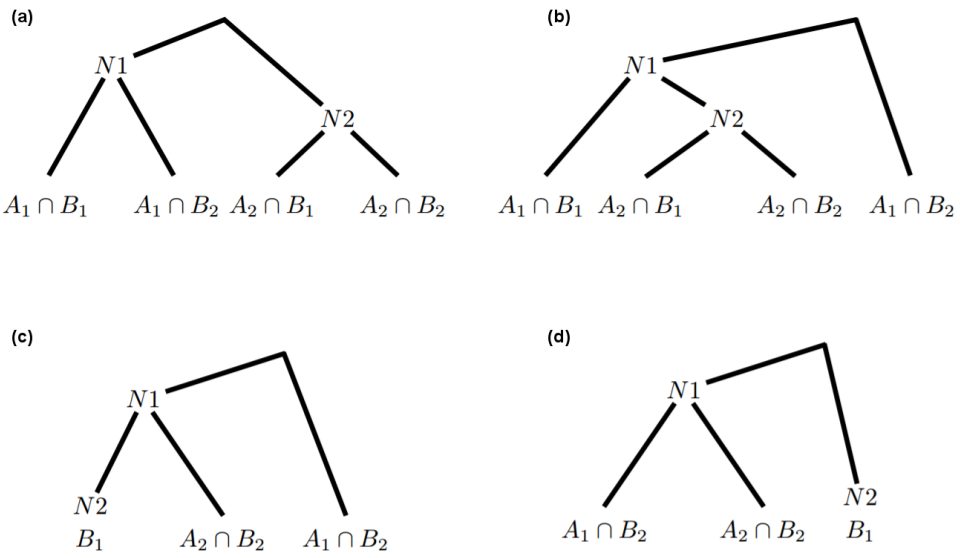

Figure 4: Steps for rearranging a tree with external nodes $A_1 \cup A_2$ to a tree with external nodes $B_1 \cup B_2$. Figure adapted from [6].

tree. Once a desired topology has been reached, the desired splitting times and nucleotide sequences can be reached by successive proposals on each node, where for each topological rearrangement step, the current sibling is selected as the sibling for the next tree.

For a base case of three species, it is clear that any topology can be reached by one rearrangement. Assume that topologies for trees with fewer than n external nodes can be generated through the rearrangement method given. Suppose we start at a tree where the bifurcation at the root node partitions the external nodes into two sets, A_1 and A_2. Furthermore, consider a desired tree, where the bifurcation at the root node similarly partitions the external into B_1 and B_2. Consider the case where either $A_1 = B_1$ or $A_2 = B_2$. By our inductive hypothesis, each subtree with external nodes A_1 and A_2 can be rearranged to match the corresponding subtree of the desired tree.

Next, consider the case where $A_1 \neq B_1$ and $A_1 \neq B_2$. By the inductive hypothesis, the subtree with external nodes A_1 can be rearranged to a subtree with subpartitions $A_1 \cap B_1$ and $A_1 \cap B_2$, which bifurcate from a node $N1$ (Fig. 4a). The A_2 subtree can be rearranged similarly, with subtrees of $A_2 \cap B_1$ and $A_2 \cap B_2$ bifurcating from $N2$. Now, consider a rearrangement where $N1$ is the target node: we can swap the child subtree $A_1 \cap B_2$ with the sibling node N_2 to produce the tree in Figure 4b. To bring the B_1 species into one subtree, we now consider $N2$ the target node, and swap the $A_1 \cap B_1$ with the $A_2 \cap B_2$ subtree such that all the B_1 species are children of the N_2 node (Fig. 4c). Finally, consider $N1$ the target node again, and swap N_2 with $A_1 \cap B_2$. Now, the bifurcation at the root partitions

the species into sets B_1 and B_2, as in our desired tree. By the inductive hypothesis, these subtrees can be rearranged as desired. Therefore, all topologies can be reached using the provided rearrangement method.

It is easy to see the other two conditions are true. First, the chain cannot have period $k > 1$, because it may visit the same tree at x_t and x_{t+1} if it rejects the candidate tree. Second, an irreducible, aperiodic finite Markov chain must be positive recurrent. By Lemma 1.1, at least one state in the Markov chain is recurrent. Since the chain is irreducible, there is a positive probability of transitioning from this recurrent state to any other state. Therefore, every state is recurrent. Finally, since all recurrent states are positive recurrent by Lemma 1.2, the chain itself is positive recurrent. □

4.2 Stationary distribution is conditional distribution

A Markov chain $x_1, x_2, ...$ with a transition probability R satisfies the **detailed balance condition** if there exists a distribution π satisfying

$$R(b|a)\pi(a) = R(a|b)\pi(b) \qquad [7]$$

where $\pi(a)$ is the probability of being at a state a and $R(b|a)$ is the probability of transitioning from state a to state b. In other words, detailed balance indicates that the probability of being at a state a and then transitioning to state b equals the probability of being at state b and transitioning to a.

Lemma 2. *For a Markov chain X with transition probability R, if π satisfies the detailed balance condition, it is a stationary distribution of X.*

Proof. Let X with transition probability R be such a Markov chain, where π satisfies the detailed balance condition. For π to be a stationary distribution, we must have that

$$\pi = R\pi = \begin{array}{c} \\ a_1 \\ \vdots \\ a_x \end{array} \begin{array}{c} \begin{array}{ccc} a_1 & \dots & a_x \end{array} \\ \begin{pmatrix} R(a_1|a_1) & \dots & R(a_x|a_1) \\ & \ddots & \\ R(a_1|a_x) & \dots & R(a_x|a_x) \end{pmatrix} \end{array} \cdot \begin{pmatrix} \pi(a_1) \\ \vdots \\ \pi(a_x) \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{x} R(a_i|a_1)\pi(a_i) \\ \vdots \\ \sum_{i=1}^{x} R(a_i|a_x)\pi(a_i) \end{pmatrix}$$

So, for any state a, the stationary probability $\pi(a)$ equals the product of the vector π and the ath row of the transition matrix: $\pi(a) = \sum_i R(a_i|a)\pi(a_i)$.

Take the summation of both sides of the detailed balance condition over all states b. Since the chain has to transition to some state, $\sum_b R(b|a) = 1$. Therefore,

$$\pi(a) = \sum_b R(a|b)\pi(b),$$

so π is a stationary distribution of X as desired. □

Theorem 2. *The Markov chain generated by the Metropolis-Hastings algorithm has unique stationary distribution $f(\tau|\boldsymbol{D})$.*

Proof. We will prove that the Markov chain generated by the Metropolis-Hastings algorithm satisfies the detailed balance criterion when trees are drawn from the conditional distribution $f(\tau|\boldsymbol{D})$ [7]. This result shows by Lemma 2 that $f(\tau|\boldsymbol{D})$ is a stationary distribution of the Markov chain. Since the Markov chain has a unique stationary distribution by Theorem 1, the detailed balance criterion is sufficient to prove Theorem 2.

Consider two trees τ_a and τ_b, labeled without loss of generality such that the acceptance probability of transitioning to τ_b from τ_a is $\alpha(\tau_a, \tau_b) = 1$. If we were to draw τ_a from the desired distribution, the unconditional probability of moving from τ_a to τ_b would be the probability of being at τ_a, multiplied by the probability of proposing tree τ_b while at τ_a, multiplied by the probability of accepting the transition from τ_a to τ_b:

$$\begin{aligned} f(\tau_a|\boldsymbol{D})R(\tau_b|\tau_a) &= f(\tau_a|\boldsymbol{D})\, q(\tau_a, \tau_b)\, \min\left\{1, \frac{f(\tau_b|\boldsymbol{D})q(\tau_b, \tau_a)}{f(\tau_a|\boldsymbol{D})q(\tau_a, \tau_b)}\right\} \\ &= f(\tau_a|\boldsymbol{D})\, q(\tau_a, \tau_b) \cdot 1 \end{aligned}$$

Now, consider the opposite draw, where we want to know the probability of being at $x_t = \tau_b$ and accepting a transition to $x_{t+1} = \tau_b$. Because of our labeling, $\alpha(\tau_b, \tau_a) < 1$.

$$\begin{aligned} f(\tau_b|D)R(\tau_a|\tau_b) &= f(\tau_b|\boldsymbol{D})\, q(\tau_b, \tau_a)\, \min\left\{1, \frac{f(\tau_a|\boldsymbol{D})q(\tau_a, \tau_b)}{f(\tau_b|\boldsymbol{D})q(\tau_b, \tau_a)}\right\} \\ &= f(\tau_b|\boldsymbol{D})\, q(\tau_b, \tau_a)\, \frac{f(\tau_a|\boldsymbol{D})q(\tau_a, \tau_b)}{f(\tau_b|\boldsymbol{D})q(\tau_b, \tau_a)} \\ &= f(\tau_a|\boldsymbol{D})\, q(\tau_a, \tau_b) \end{aligned}$$

Therefore, $f(\tau_a|D)R(\tau_b|\tau_a) = f(\tau_b|D)R(\tau_a|\tau_b)$, so the detailed balance criterion is satisfied when trees are drawn from the conditional distribution. Thus, the conditional distribution is a stationary distribution of the Markov chain (Lemma 2). Since the Markov chain generated using this tree proposal method has only one stationary distribution (Theorem 1), $f(\tau|\boldsymbol{D})$ is the unique stationary distribution of the Markov chain. □

5 Practical considerations

Implementing MCMC requires not only that a Markov chain be initiated, but also that at some point, it is stopped to observe the distribution it has reached. Crucially, we must be careful to ensure that when the chain is stopped, it actually has converged to the stationary distribution, or else we may arrive at false conclusions. For a general review of diagnostics used to assess convergence, see [3].

The more important barrier to the convergence of MCMC arises when the proposal distribution includes high probability "peaks" separated by low probability "valleys" (Fig. 5a). Consider a Markov chain where state X_t is a high-probability tree τ_t. Perhaps this tree represents a local, but not global, maximum on q, such that any other proposed tree will have a lower probability than τ_t. Thus, the $q(x', x_t)/q(x_t, x')$ term of the acceptance probability will be low, and Metropolis-Hastings will be unlikely to accept the proposal. In these instances, the Markov chain may take a long time to traverse valleys; if the algorithm is stopped prematurely, the chain may fail to sample the highest probability trees.

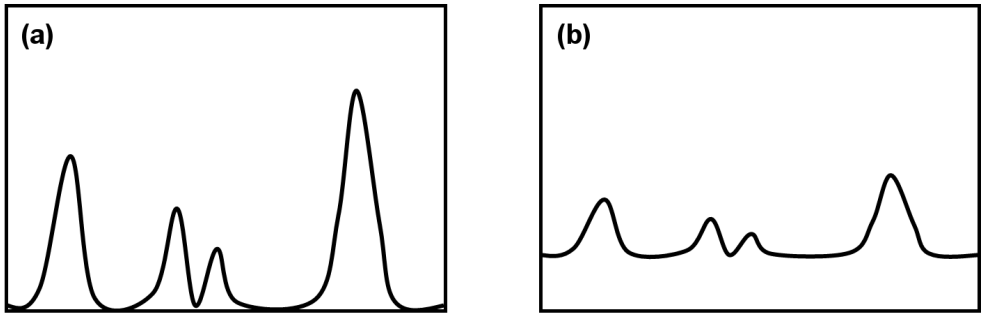

Figure 5: An unheated (a) and heated (b) proposal density

To increase the convergence speed of the chain, the statistician can perform Metropolis-coupled MCMC $(MC)^3$, an algorithm that uses "heated" chains to improve the convergence of unheated chains. To heat a chain is to flatten its proposal density, reducing the relative height of its peaks and valleys (Fig. 5b). Thus, heated chains can more readily cross valleys to explore alternative peaks. The algorithm proceeds by generating heated chains simultaneously alongside unheated chains, at each time step attempting to swap the state of the heated trees with unheated trees [1]. Thus, if the state proposed by the heated chain is across a deep valley, the unheated chain may jump over this valley in a single step, increasing the convergence time of the chain.

References

[1] G. Altekar, S. Dwarkadas, J. P. Huelsenbeck, and F. Ronquist. Parallel Metropolis coupled Markov chain Monte Carlo for Bayesian phylogenetic inference. *Bioinformatics*, 20:407–415, 2004.

[2] P. Besse. Phylogenetic reconstruction methods: an overview. In J. M. Walker, editor, *Molecular Plant Taxonomy: Methods and Protocols*, chapter 13, pages 257–277. Springer Science, New York, 2014.

[3] M. Cowles and B. Carlin. Markov chain Monte Carlo convergence diagnostics: a comparative review. *Journal of the American Statistical Association*, 91:883–904, 1996.

[4] M. Hasegawa, H. Kishino, and T. Yano. Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. *J Mol Evol*, 22:160–174, 1985.

[5] C. Lakner, P. van der Mark, J. P. Huelsenbeck, B. Larget, and F. Ronquist. Efficiency of Markov Chain Monte Carlo tree proposals in Bayesian phylogenetics. *Syst. Biol.*, 57:86–103, 2008.

[6] S. Li. *Phylogenetic Tree Construction using Markov Chain Monte Carlo*. PhD thesis, Ohio State University, Columbus, 1996.

[7] C.P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, New York, 1998.