Technische Universität München

Zentrum Mathematik

Stochastic Models for Speciation Events in Phylogenetic Trees

Diplomarbeit

von

Tanja Gernhard

Aufgabensteller:	Prof. Dr. Rupert Lasser
Betreuer:	Prof. Dr. Mike Steel
Abgabetermin:	7. April 2006

Hiermit erkläre ich, dass ich die Diplomarbeit selbständig angefertigt und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 7. April 2006

.....

Tanja Gernhard

Acknowledgements

First and foremost, I would like to thank my supervisor Mike Steel for making it possible for me to come to New Zealand, for the great support throughout my stay, for suggesting great problems to work on and for very helpful discussions and advice. Through my stay in New Zealand and my work with Mike, I finally found my area in research.

My thesis abroad and the great experience I had during that time would not have been possible without the support of my German supervisor Rupert Lasser. He encouraged me in any of my plans and let me have all the freedom I needed in choosing a topic for my thesis.

The three days of Daniel Ford's stay in Canterbury were probably the three most productive days of my thesis, while we implemented and optimized my algorithms. Daniel introduced me to Python which was a very convenient language for my research.

Talking to Erick Matsen during coffee breaks helped me to see things I was working on in a broader scientific perspective. Mareike Fischer had very helpful comments for last improvements of my thesis.

I would also like to thank Craig Moritz, Andrew Hugall, Arne Mooers and Rutger Vos who posed the questions which led to my thesis.

The people and the friendly environment in the Biomath Department at Canterbury University made my stay most enjoyable. Special thanks go to Charles Semple who helped me very much when I first arrived so that I felt comfortable in New Zealand right away.

Further, thanks to the Friedrich-Ebert-Stiftung for the support throughout my time at university and the Allan Wilson Center for hosting me as a summer student while I was in New Zealand.

Last but not least, I would like to thank my family and my boyfriend for supporting me in any possible way, for giving me good advice whenever I had to make a key decision, for always encouraging me and for providing me a home I always look forward going back to.

Contents

Chapter 1

Introduction

1.1 Overview

In 1837, Darwin published a first sketch of an evolutionary tree, see Fig. 1.1. This new idea that all species evolved over time was under a lot of discussion and not until the early 20th century was evolution generally accepted by the scientific community. Since then, much research went into the field of evolution. With the help of fossils, and by comparing the anatomy as well as the geographic occurrence of species, complex evolutionary trees have been created.

In an evolutionary tree, each leaf represents an existing species and all the interior vertices represent the ancestors. The edges of the tree show the relationships between the species.

The first step to modern evolutionary research was the discovery of the double helix structure of DNA (deoxyribonucleic acid) by Watson and Crick in 1953. The genetic code is a long chain of bases (Adenine, Cytosine, Guanine, Thymine) and triplets of these bases encode the 20 amino acids. A backbone of sugars and phosphates holds the bases together, see Fig. 1.2. The amino acids in a cell form

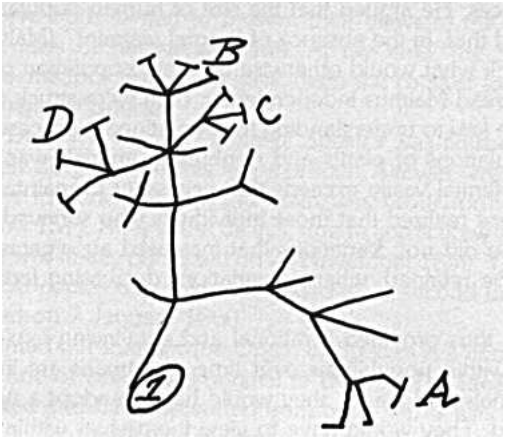


Figure 1.1: Darwin's first diagram of an evolutionary tree from his 'First Notebook on Transmutation of Species' (1837).

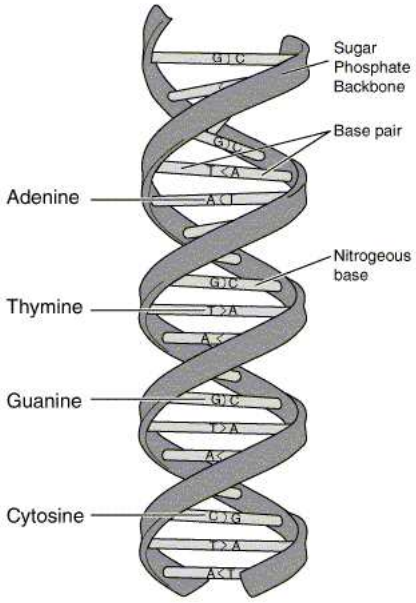


Figure 1.2: The DNA - a double helix

proteins according to the DNA code. From a chemical point of view, life is nothing else than the functioning of proteins. Since the DNA determines which proteins are built, a living organism can chemically be described by its DNA, the genetic information [17].

Each cell of an organism has an identical copy of the DNA. In eukaryotes, the DNA is found in a cell nucleus whereas in prokaryotes (archaea and bacteria), the DNA is not separated from the rest of the cell.

During reproduction, the DNA is transmitted to the offspring, so parents and children are similar in many ways (e.g. hair color, blood group, disease susceptibility).

It was not until 2003 that the complete human DNA code was described. Currently, the complete DNA sequence of several different species is known (358 bacteria, 27 archae, 95 eukaryotes, see http://www.ncbi.nih.gov/). By aligning the DNA of different species, the similarities and differences of the DNA allow us to reconstruct lineages with more accuracy than before; for an example see Fig. 1.3.

It is noticeable that the same four DNA bases and the 20 amino acids are found in all organisms. This is strong evidence for having one common ancestor to all the species.

Evolutionary trees are also called 'phylogenetic trees'. If all the species in the tree have a common ancestor, we call the tree a 'rooted tree', the common ancestor is called the 'root'.

I take a closer look at rooted phylogenetic trees. The shape of the tree is determined by how speciation occurred. But since speciation is not understood well and is dependent on historical events which we might never be able to

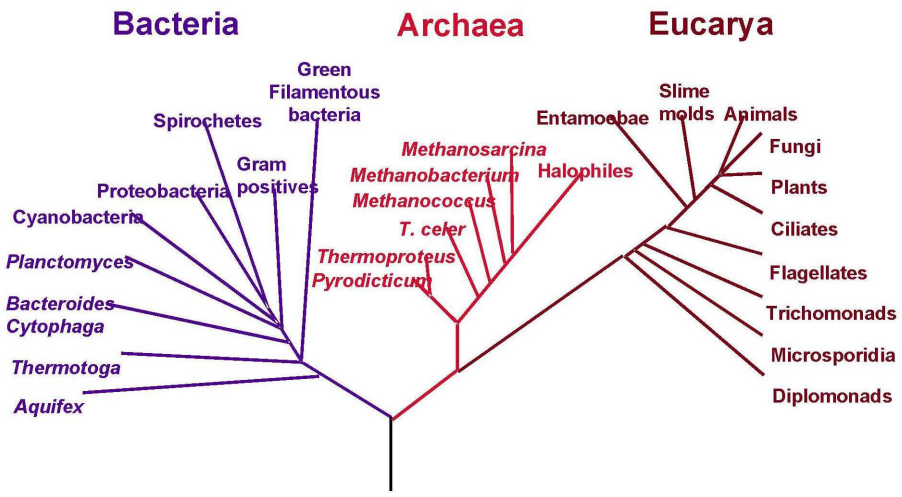


Figure 1.3: Illustration of the tree of life by Carl R. Woese. There are three main branches, the bacteria, archaea and eucarya, source http://www.life.uiuc.edu/micro/faculty/faculty_woese.htm.

reconstruct, a stochastic model for speciation is needed. I investigate the Yule model and the uniform model, two very common models.

In my thesis, I develop the theory with a view to the following applications in biology.

Rutger Vos and Arne Mooers from the Simon Fraser University (Vancouver) recently constructed a supertree for the primates (i.e. lemurs, monkeys, apes and humans) as shown in Appendix C.

In Section 2.2.1, we will see that the primate tree is much more likely to have evolved under the Yule than under the uniform model.

With the supertree method, the shape of the primate tree could be determined, but there was no information about the edge lengths, i.e. the time between speciation events. In [16], edge lengths were estimated by simulations, assuming the (super)tree evolved under the Yule model. The authors concluded by asking for an analytical approach which I develop in Chapter 4.

Craig Moritz (UC Berkeley) and Andrew Hugall (University of Adelaide) worked with an evolutionary tree which had edge lengths assigned. The leaves were different types of snails. The snails either live in open forest or rain forest. Moritz and Hugall asked (pers. comm.) if the rate of speciation for open forest snails differs from the rate of speciation for rain forest snails. The rate of speciation is a measure of how fast a class of species produces splits in the evolutionary tree. Chapter 5 provides a linear algorithm for solving that problem.

1.2 Short guide to the thesis

In Chapter 2, two important stochastic models for binary phylogenetic trees are introduced - the uniform and the Yule model. Those two models are discussed and the Kullbach-Liebler-distance between them is calculated. The Kullbach-Liebler-distance turns out to be very useful in deciding whether a given tree evolved under the Yule or the uniform model.

Chapter 3 formulates a test statistic for that decision problem, the log-likelihood-ratio test. Instead of estimating the power of the test by simulations, we provide an analytic bound for the power by introducing a martingale process on trees and applying the Azuma inequality.

The algorithms in Chapter 4 work in particular for trees under the Yule model. In order to verify that a tree evolved under Yule, the test provided in Chapter 3 can be applied before running the algorithms.

After having established all the necessary stochastic background, Chapter 4 provides a quadratic algorithm for calculating the probability distribution of the rank for a given interior vertex in a phylogenetic tree. The algorithm is called RankProb and we assume that every rank function on a given tree is equally likely. That is in particular the case for the Yule model. The algorithm RankProb is extended to non-binary trees as well, again we assume that every rank function is equally likely. We call that algorithm RankProbGen. Calculating the probability of having an interior vertex u earlier in the tree than an interior vertex v is calculated with the algorithm Compare in quadratic time. We coded up the algorithms RankProb and Compare in Python, see Appendix B. The chapter concludes with an analytical approach of estimating edge lengths in a given tree under the Yule model. This approach makes use of the algorithm RankProb.

Chapter 5 looks at the rate of speciation. Given is a phylogenetic tree with the leaves being divided into two classes α and β. The edge lengths shall represent the time between two events. We provide a linear algorithm for the expected time a species of class α exists until it speciates and two new species evolve. The average edge length is an estimate for the inverse of the rate of speciation. An example for the classes α and β could be rain forest snails and open forest snails.

After introducing the stochastic models in Chapter 2, the remaining results in that Chapter are new. The results in Chapter 3, 4 and 5 are new unless otherwise stated. Improvements on the algorithms in Chapter 4 and coding them up in Python was joint work with Daniel Ford. Chapter 4 was the topic of my talk at the New Zealand Phylogenetics Conference in Kaikoura in February 2006 (http://www.math.canterbury.ac.nz/bio/kaikoura06/).

The rest of this Chapter introduces the basic definitions from graph theory and phylogenetics needed for the thesis. Further, some basic results for phylogenetic trees are stated.

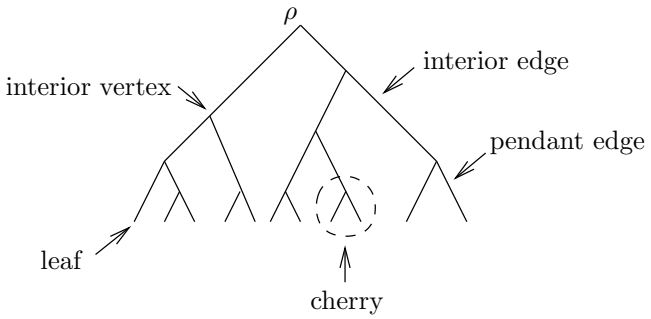


Figure 1.4: A rooted binary tree

1.3 Graphs and Trees

Definition 1.3.1. A *graph* G is an ordered pair (V, E) consisting of a non-empty set V of *vertices* and a multiset E of *edges* each of which is an element of $\{\{x, y\} : x, y \in V\}$. The degree $\delta(v)$ of a vertex $v \in V$ is the number of edges in G that are incident with v. A *path* p in G from vertex $x \in V$ to vertex $y \in V$ is a sequence $p = (v_i)_{i=1,\dots n}$, $v_i \in V$, such that $x = v_1$, $y = v_n$, and $\{v_i, v_{i+1}\} \in E$ for $i = 1, \dots n-1$. A graph G is *connected* precisely if there exists a path from x to y for all $x, y \in V$. A *cycle* in a graph is a path $p = (v_i)_{i=1,\dots n}$ with $v_1 = v_n$. The graph $G' = (V', E')$ is a *subgraph* of G if $V' \subseteq V$ and $E' \subseteq E$.

Definition 1.3.2. A *tree* $T = (V, E)$ is a connected graph with no cycles. A connected subgraph of T is a *subtree* of T. A *rooted tree* is a tree that has exactly one distinguished vertex called the *root* which we denote by the letter ρ. A vertex $v \in V$ with $\delta(v) \leq 1$ is called a *leaf* . The set of all leaves of T is denoted by L. A vertex which is not a leaf is called an *interior vertex*. Let $\mathring{V}$ denote the set of all interior vertices of T. A *binary tree* is a tree with $\delta(v) = 3$ for all $v \in \mathring{V}$. A *rooted binary tree* is a rooted tree with $\delta(v) = 3$ for all $v \in \mathring{V} \setminus \rho$ and $\delta(\rho) = 2$. Let $V' \subset V$. The subtree $T' = T|_{V'}$ is the minimal (w.r.t. the number of vertices) connected subgraph of T containing V'. An edge which is incident with a leaf is called a *pendant* edge. A non-pendant edge is called an *interior* edge. Two distinct leaves of a tree form a *cherry* if they are adjacent to a common ancestor. Let $v \in \mathring{V} \setminus \rho$ with $\delta(v) = 2$. The vertex v is *suppressed* in T if we delete v with its two incident edges $e_1 = (v_1, v), e_2 = (v, v_2)$ and then add a new edge $e = (v_1, v_2)$. For an example of a tree see Fig. 1.4.

Definition 1.3.3. Let $T = (V, E)$ be a rooted tree with leaf set $L \subset V$ and for all $v \in \mathring{V} \setminus \rho$ is $\delta(v) \neq 2$. Let X be a non-empty finite set with $|X| = |L|$. Let $\phi : X \to L$ be a bijection. Then $\mathcal{T} = (T, \phi)$ is called a *phylogenetic* $(X-)$ *tree* with *labeling function* ϕ. X is called the *label set.* A phylogenetic tree is also called a *labeled tree*. A *tree shape* is a phylogenetic tree without the labeling.

Remark 1.3.4. In the following, for a phylogenetic tree $\mathcal{T}$, we sometimes write $E_{\mathcal{T}}$ instead of E, $V_{\mathcal{T}}$ instead of V, $\mathring{V}_{\mathcal{T}}$ instead of $\mathring{V}$ and $L_{\mathcal{T}}$ instead of L. This notation clarifies to which tree the sets refer whenever we talk about several different trees.

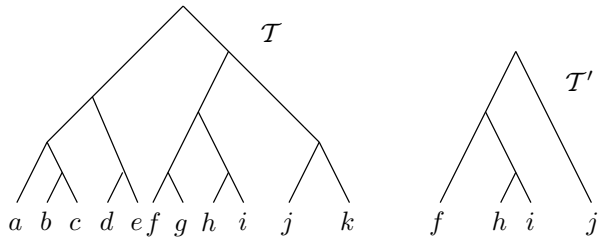


Figure 1.5: A rooted binary phylogenetic X-tree $\mathcal{T}$ with $X = \{a, b, \ldots, k\}$ and the subtree $\mathcal{T}' = \mathcal{T}|_{\{f,h,i,j\}}$.

Definition 1.3.5. Let T be a rooted tree. A partial order $\leq_T$ on V is obtained by setting $v_1 \leq_T v_2$ $(v_1, v_2 \in V)$ precisely if the path from the root ρ to v_2 includes v_1. If $v_1 \leq_T v_2$, we say v_2 is a *descendant* of v_1 and v_1 is an *ancestor* of v_2. If $v_1 \leq_T v_2$ and there is no $v_3 \in V$ with $v_1 \leq_T v_3 \leq_T v_2$, we say v_2 is a *direct descendant* of v_1 and v_1 is a *direct ancestor* of v_2. The *number of direct descendants* of v is $d(v)$. When we talk about a phylogenetic tree, we often write $\leq_{\mathcal{T}}$ instead of $\leq_T$.

Definition 1.3.6. Let $\mathcal{T} = (T, \phi)$ be a phylogenetic X-tree. Let $X' \subset X$. The phylogenetic subtree $\mathcal{T}' = \mathcal{T}|_{X'} = (T', \phi')$ is a phylogenetic tree where T' is the tree $T|_{\phi(X')}$ with all degree-two vertices suppressed (except for the root). The labeling function is $\phi' = \phi|_{X'}$. The root of $\mathcal{T}'$ is the vertex ρ' which is minimal in the tree T' under the partial order $\leq_{\mathcal{T}}$ (see Fig. 1.5). Let $\mathcal{T}'$ be a subtree of $\mathcal{T}$. Denote the subtree $\mathcal{T}|_{L_{\mathcal{T}} \setminus L_{\mathcal{T}'}}$ by $\mathcal{T} \setminus \mathcal{T}'$.

Let $v \in \mathring{V}$ and let X_v be the label set of all the leaves in $\mathcal{T}$ which are descendants of v. The subtree $\mathcal{T}_v$ is *induced by* v if $\mathcal{T}_v = \mathcal{T}|_{X_v}$. A binary phylogenetic tree is *balanced* if the two subtrees induced by the two direct descendants of the root have the same shape. Otherwise, the tree is *unbalanced*.

Definition 1.3.7. Let $\mathcal{T}$ be a rooted phylogenetic tree. Let the function r be a bijection from the set of interior vertices $\mathring{V}$ of $\mathcal{T}$ into $\{1, 2, \ldots, |\mathring{V}|\}$ that satisfies the following property:

$$if\ v_1 \leq_{\mathcal{T}} v_2,\ then\ r(v_1) \leq r(v_2)$$

$(\mathcal{T}, r)$ is called a *phylogenetic ranked tree* (see Fig. 1.6). The function r is called a *rank function* for $\mathcal{T}$. A vertex v with $r(v) = i$ is said to be in the $i-th$ *position* of $\mathcal{T}$ or v has rank i. We write $r_{\mathcal{T}}$ instead of r when it is not clear from the context to which tree the rank function r refers. Note that r induces a linear order on the set $\mathring{V}$. We define the set $r(\mathcal{T})$ as

$$r(\mathcal{T}) = \{r :\ r \text{ is a rank function on } \mathcal{T}\}.$$

The following Lemma has been shown in [14] using poset theory. We will give an elementary proof using induction.

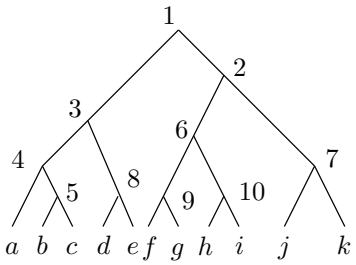


Figure 1.6: A rooted binary phylogenetic ranked X-tree with $X = \{a, b, \ldots, k\}$

Lemma 1.3.8. *Let $\mathcal{T}$ be a rooted phylogenetic tree. For each $v \in \mathring{V}$, let λ_v denote the number of elements of $\mathring{V}$ that are descendants of v. Then the number of rank functions for $\mathcal{T}$ is*

$$|r(\mathcal{T})| = \frac{|\mathring{V}|!}{\prod_{v \in \mathring{V}} \lambda_v} \tag{1.1}$$

Note that a vertex v is a descendant of itself by definition, so λ_v also counts the vertex v.

Proof. This proof is done by induction over the number n of interior vertices of a tree. For $n = 1$, there is only one rank function, the only interior vertex has rank 1, which equals to $\frac{|\mathring{V}|!}{\prod_{v \in \mathring{V}} \lambda_v} = \frac{1!}{1} = 1$. Suppose that (1.1) is true for all trees with $n < k$ interior vertices. Let $\mathcal{T}$ be a tree with k interior vertices. The degree of root ρ is $\delta(\rho) = m$ where $m < k$. $\mathcal{T}$ has m vertex-disjoint rooted subtrees $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_m$ induced by the direct descendants of ρ, and with $|\mathring{V}_{\mathcal{T}_i}| < k$. Each subtree $\mathcal{T}_i$ has $\frac{|\mathring{V}_{\mathcal{T}_i}|!}{\prod_{v \in \mathring{V}_{\mathcal{T}_i}} \lambda_v}$ different rank functions by the induction assumption. Counting all the rank functions on $\mathcal{T}$ is equivalent to counting the rank functions on each subtree $\mathcal{T}_i$ and then combining the positions of the vertices of all the $\mathcal{T}_i$ to get a linear order on $\mathring{V}_{\mathcal{T}}$, by preserving the order of the vertices of each $\mathcal{T}_i$. For a given rank function on each $\mathcal{T}_i$, we can order all the interior vertices in $\frac{\left(\sum_i |\mathring{V}_{\mathcal{T}_i}|\right)!}{\prod_i \left(|\mathring{V}_{\mathcal{T}_i}|!\right)}$ different ways where the order within each $\mathcal{T}_i$ is preserved. Multiplying by all the possible

rank functions for each $\mathcal{T}_i$ yields to

$$\begin{aligned}
|r(\mathcal{T})| &= \frac{\left(\sum_{i=1}^{m} |\mathring{V}_{\mathcal{T}_i}|\right)!}{\prod_{i=1}^{m}\left(|\mathring{V}_{\mathcal{T}_i}|!\right)} \left(\prod_{i=1}^{m} |r(\mathcal{T}_i)|\right) \\
&= \frac{\left(\sum_{i=1}^{m} |\mathring{V}_{\mathcal{T}_i}|\right)!}{\prod_{i=1}^{m}\left(|\mathring{V}_{\mathcal{T}_i}|!\right)} \left(\prod_{i=1}^{m} \frac{|\mathring{V}_{\mathcal{T}_i}|!}{\prod_{v\in\mathring{V}_{\mathcal{T}_i}} \lambda_v}\right) \\
&= \left(\sum_{i=1}^{m} |\mathring{V}_{\mathcal{T}_i}|\right)! \prod_{i=1}^{m} \frac{1}{\prod_{v\in\mathring{V}_{\mathcal{T}_i}} \lambda_v} \\
&= \frac{(|\mathring{V}_{\mathcal{T}}|-1)!}{\prod_{v\in\mathring{V}_{\mathcal{T}\setminus\rho}} \lambda_v} \\
&= \frac{|\mathring{V}_{\mathcal{T}}|!}{\prod_{v\in\mathring{V}_{\mathcal{T}}} \lambda_v}.
\end{aligned}$$

This establishes the induction step, and thereby the theorem. □

Remark 1.3.9. In the following, all trees shall be rooted. The set of all binary rooted phylogenetic trees with label set X is denoted by $RB(X)$. The set of all ranked binary rooted phylogenetic trees with label set X is denoted by $rRB(X)$.

Remark 1.3.10. A rooted binary phylogenetic tree with n leaves has $|\mathring{V}| = n-1$ interior vertices and $|E| = 2(n-1)$ edges, which is shown by induction in [14].

Chapter 2

Stochastic Models on Trees

Given a phylogenetic X-tree, we are interested in the probability of that tree from the set $RB(X)$ or $rRB(X)$, depending on whether the given tree is ranked or not. When defining a probability distribution on trees, the probability of a labeled tree should be invariant under a different labeling. This property is called *exchangeability.*

There are several stochastic models for binary phylogenetic X-trees, the most common are the uniform and Yule model which we will introduce and compare.

In the following, for simplifying notation, any X with $|X| = n$ shall be $X = \{1, 2, \ldots, n\}$ and we write $RB(n)$, $rRB(n)$ instead of $RB(X)$, $rRB(X)$.

2.1 The uniform model

Under the uniform model, a random element of $RB(n)$ is generated in the following way (*cf.* Figure 2.1):

- Label the two leaves of a cherry with 1 and 2.
- Add to the cherry a third edge connecting the root ρ of the cherry and a new vertex ρ' which is earlier than ρ. This extended cherry is denoted by $\mathcal{T}$.
- In each step, modify $\mathcal{T}$ in the following way, until $\mathcal{T}$ has n leaves:
  - Let the number of leaves of $\mathcal{T}$ be k. Choose an edge of $\mathcal{T}$ randomly and with uniform probability and subdivide this edge to create a new vertex.
  - Add an edge from the new vertex to a new leaf.
  - Label the new leaf by $k + 1$.
- Remove from the tree $\mathcal{T}$ the vertex ρ' and its incident edge to get the binary rooted tree $\mathcal{T}$.

In this way, each rooted binary phylogenetic X-tree has equal probability (see [11]). Obviously, the probability of a tree is invariant under a different leaf labeling.

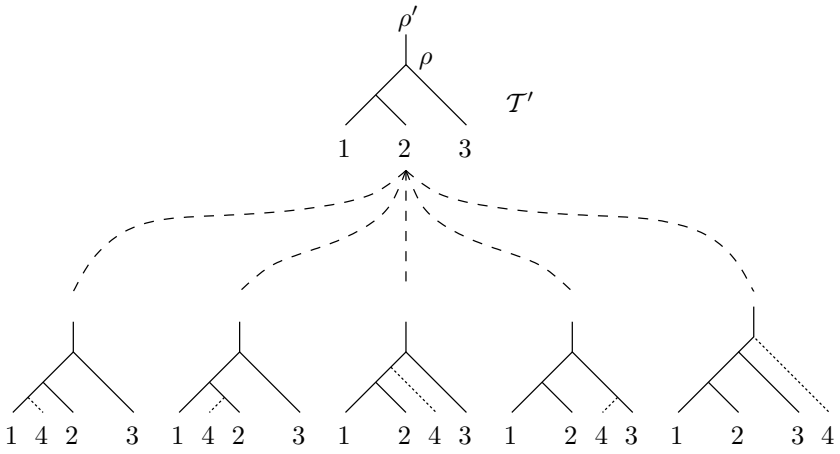


Figure 2.1: Tree evolving under the uniform model. Let $X = \{1, 2, 3, 4\}$. Given the tree $\mathcal{T}'$ with label set $\{1, 2, 3\}$, which has probability $1/3$ under the uniform model, there are five possible edges to attach the leaf with label 4. Each of the five trees with label set $\{1, 2, 3, 4\}$ has probability $1/5$ given $\mathcal{T}'$. So the overall probability of each tree with four leaves is $1/15$ under the uniform model.

Note that it is not necessary to choose the elements of X in the given order $1, 2, \ldots, n$. We could choose the leaf labels in any order. This will not be the case for the Yule model.

Lemma 2.1.1. *For each $n \geq 2$,*

$$(2n-3)!! = \frac{n! c_{n-1}}{2^{n-1}}$$

with $(2n-3)!! = (2n-3)\cdot(2n-5)\ldots 5\cdot 3\cdot 1$ and c_n being the n-th Catalan number, $c_n = \frac{1}{n+1}\binom{2n}{n}$.

Proof.

$$\begin{aligned}(2n-3)!! &= \frac{(2n-3)!}{2^{n-2}\left(\frac{2n-4}{2}\right)!} = \frac{(2n-3)!}{2^{n-2}(n-2)!} \\ &= \frac{(2n-2)!}{2^{n-1}(n-1)!} = \frac{\frac{(n-1)!(2(n-1))!}{2(n-1)!}}{2^{n-1}} = \frac{n!\frac{1}{n}\binom{2(n-1)}{n-1}}{2^{n-1}} \\ &= \frac{n! c_{n-1}}{2^{n-1}}.\end{aligned}$$

□

The following result is already shown in [14] by considering unrooted trees and defining a bijection from unrooted to rooted trees. This proof is direct.

Theorem 2.1.2. *The number of binary rooted phylogenetic trees is*

$$|RB(n)| = (2n-3)!!$$

Proof. The proof is done by induction over n. For $n = 2$, we have $|RB(2)| = 1$ and $(2 \cdot 2 - 3)!! = 1$. Assume $|RB(n)| = (2n-3)!!$ holds for all $n \leq k$, where $k \geq 2$. A tree $\mathcal{T}_k$ with k leaves has $2(k-1)$ edges (see Remark (1.3.10)). Denote the root of $\mathcal{T}_k$ by ρ_k. The $(k+1)$-th leaf x can be attached to $\mathcal{T}_k$ to any of the $2(k-1)$ edges or a new root ρ with edges $e_1 = (\rho, \rho_k)$ and $e_2 = (\rho, x)$ is added. So we can construct $2(k-1)+1 = 2k-1$ different trees from $\mathcal{T}_k$. By the induction assumption, we have $|RB(k)| = (2k-3)!!$. Therefore, $|RB(k+1)| = (2k-3)!! \cdot (2k-1) = (2(k+1)-3)!!$ which proves the theorem. □

Corollary 2.1.3. *Under the uniform model, the probability $\mathbb{P}[\mathcal{T}]$ of a tree $\mathcal{T}$ chosen from the set $RB(n)$ is*

$$\mathbb{P}[\mathcal{T}] = \frac{1}{(2n-3)!!} = \frac{2^{n-1}}{n! c_{n-1}}.$$

Proof. Since a phylogenetic tree $\mathcal{T}$ is chosen from $RB(n)$ uniformly at random in the uniform model, we have

$$\mathbb{P}[\mathcal{T}] = \frac{1}{|RB(n)|}.$$

By Theorem (2.1.2) and Lemma (2.1.1), we get $\mathbb{P}[\mathcal{T}] = \frac{1}{(2n-3)!!} = \frac{2^{n-1}}{n! c_{n-1}}$. □

2.2 The Yule model

Under the Yule model [18, 8], a random element of $rRB(n)$ is generated in the following way (*cf.* Figure 2.2):

- Two elements of X are selected uniformly at random and the two leaves of a cherry are labeled by them. This cherry is denoted by $\mathcal{T}$ and its root has rank 1.
- In each step, modify $\mathcal{T}$ in the following way, until $\mathcal{T}$ has n leaves:
  - Let the number of leaves of $\mathcal{T}$ be k. Choose a pendant edge of $\mathcal{T}$ uniformly at random and subdivide this edge to create a new interior vertex with rank k.
  - Add an edge from the new vertex to a new leaf.
  - Select an element of X which is not in the label set of $\mathcal{T}$ uniformly at random and label the new leaf by that element.

In other words, any pendant edge of a binary tree is equally likely to split and give birth to two new pendant edges. The Yule model is therefore an explicit model of the process of speciation. This makes it a very important model for the distribution on trees. Since the labels are added uniformly at random, the

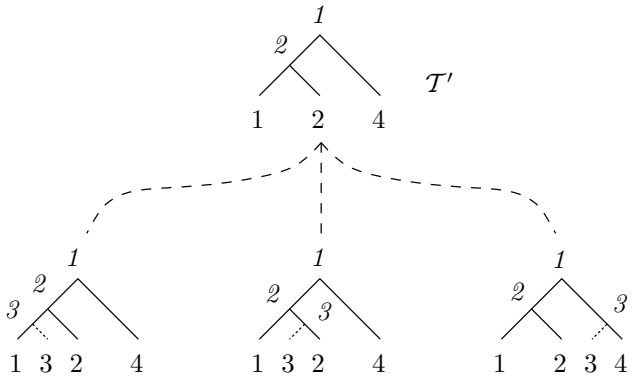


Figure 2.2: Ranked tree evolving under the Yule model. Let $X = \{1, 2, 3, 4\}$. Suppose the ranked tree $\mathcal{T}'$ with label set $\{1, 2, 4\}$ evolved under the Yule model. There are three possible pendant edges to attach the leaf with the remaining label 3. Each ranked tree with label set $\{1, 2, 3, 4\}$ has probability $\frac{2^{4-1}}{4!(4-1)!} = 1/18$ according to Theorem (2.2.1).

probability of a tree is invariant under a different leaf labelling (i.e. dependent only on the 'shape' of the tree).

Note that under the Yule model, at each moment in time, the probability of a speciation event is equal for all the current species. For different points in time, these probabilities can be quite different though.

Under the Yule model, balanced trees are more likely than unbalanced trees whereas under the uniform model, every tree is equally likely. Phylogenetic trees constructed for most sets of species tend to be more balanced than predicted by the uniform model, but less balanced than predicted by the Yule model. That can be explained in the following way. In nature, we observe that a species, which has not given birth to new species for a long time, is not very likely to give birth in the future either. The Yule model does not take this fact into account. In [15], there is an extension of the Yule model described which takes care of that biological observation. One special case of the extended Yule model assumes, that unless a species has undergone a speciation event within the last ϵ time interval, it will never do so. It is shown in [15] that for sufficient small ϵ, this model induces the uniform distribution. So the uniform model can also be interpreted as a process of speciation.

The Yule and the uniform model can be put in a more general framework. In [1], the beta-splitting model is introduced, where the Yule and the uniform model are special cases. In [7], the alpha model is introduced and again, the Yule and the uniform model are special cases. In both papers, a one parameter family of probability models on binary phylogenetic trees is introduced which interpolates continuously between the Yule and the uniform model.

These models are far more complicated than the uniform and Yule model though, and since especially the Yule model is still a reasonably good model for speciation, we will now focus on properties of the Yule model. Theorem (2.2.1)

and Corollary (2.2.2) have been established in [5]. Here we provide an alternative proof.

Theorem 2.2.1. *The probability under the Yule model of generating a ranked binary phylogenetic tree* $(\mathcal{T}, r) \in rRB(n)$ *is*

$$\mathbb{P}[\mathcal{T}, r] = \frac{2^{n-1}}{n!(n-1)!}.$$

That is a uniform distribution over $rRB(n)$.

Proof. We calculate the probability $\mathbb{P}[\mathcal{T}, r]$ by looking at the generation of the tree $\mathcal{T}$. In the first step of the generation, we have n possibilities to choose the label for the left leaf of the cherry and $n-1$ possibilities to choose the label for the right leaf of the cherry. So the probability for a certain cherry, with distinguishing between left and right vertex, is $\frac{1}{n(n-1)}$, since the selection of the labels is uniformly at random. The root of the cherry has rank 1. When adding a new leaf to a tree $\mathcal{T}_k$ with k leaves, we have k possibilities to choose a pendant vertex and $n-k$ possibilities to choose a label. So the probability of attaching a new labeled leaf to a certain edge is $\frac{1}{k(n-k)}$ since we choose the pendant edge and the label uniformly at random. The new interior vertex has rank k. Let the new leaf be x. The leaf x shall be on the right side of the new cherry. With the process above, we get two equal trees precisely if every step of the tree generation process is equal for both trees. While distinguishing between left and right child of an interior vertex, we count each phylogenetic tree $2^{|\mathring{V}|} = 2^{n-1}$ times. Therefore, we get the following probability for the ranked phylogenetic tree $(\mathcal{T}, r)$

$$\mathbb{P}[\mathcal{T}, r] = 2^{n-1}\frac{1}{n(n-1)}\frac{1}{2(n-2)}\frac{1}{3(n-3)}\cdots\frac{1}{(n-1)1} = \frac{2^{n-1}}{n!(n-1)!}$$

Since $\mathbb{P}[\mathcal{T}, r]$ is independent of $\mathcal{T}$ and r, we have a uniform distribution. □

Corollary 2.2.2. *The number of ranked phylogenetic trees is*

$$|rRB(n)| = \frac{n!(n-1)!}{2^{n-1}}$$

Proof. Since $\mathbb{P}[\mathcal{T}, r] = \frac{2^{n-1}}{n!(n-1)!}$ is uniform under the Yule model and probabilities add up to 1, we have $\frac{n!(n-1)!}{2^{n-1}}$ different ranked phylogenetic trees. □

Lemma 2.2.3. *Let* A *be a finite set and for each* $a \in A$*, let* $B(a)$ *be a finite set and let* $\Omega = \{(a, b) : a \in A, b \in B(a)\}$*. Let* $C = (C_1, C_2)$ *be the (two-dimensional) random variable which takes a value in* Ω *selected uniformly at random, i.e.* $\mathbb{P}[C = (a, b)] = 1/|\Omega|$ *for all* $(a, b) \in \Omega$*. Then the conditional probability distribution* $\mathbb{P}[C = (a, b)|C_1 = a]$ *is uniform on* $B(a)$.

Proof. We have

$$\mathbb{P}[C = (a,b)|C_1 = a] = \frac{\mathbb{P}[C = (a,b)]}{\mathbb{P}[C_1 = a]} = \frac{1}{|\Omega|\mathbb{P}[C_1 = a]}$$

which is independent of b and therefore is uniform on $B(a)$. □

Theorem 2.2.4. *Assume a given binary phylogenetic tree $\mathcal{T}$ with n leaves evolved under the Yule model. Then the probability of a rank function r on a given tree $\mathcal{T}$ is*

$$\mathbb{P}[r|\mathcal{T}] = \frac{\prod_{v\in\mathring{V}} \lambda_v}{(n-1)!}$$

i.e. $\mathbb{P}[r|\mathcal{T}]$ is uniform over all rankings r of $\mathcal{T}$.

Proof. Consider the probability distribution induced by the Yule model on $A = RB(n)$. Let $B(a)$ be the set of all rankings for a tree $a \in A$ and let $\Omega = \{(a,b) : a \in A, b \in B(a)\}$. Let $C = (C_1, C_2)$ be the (two-dimensional) random variable which takes a value in Ω. The random variable C is uniform on the set Ω by Theorem (2.2.1) and we can apply Lemma (2.2.3) to obtain

$$\mathbb{P}[C = (\mathcal{T}, r)|C_1 = \mathcal{T}] = \mathbb{P}[r|\mathcal{T}] = \frac{1}{|\Omega|\mathbb{P}[C_1 = \mathcal{T}]}$$

which shows that $\mathbb{P}[r|\mathcal{T}]$ is uniform over all rankings r of $\mathcal{T}$. Since for a tree $\mathcal{T}$, we have $\frac{|\mathring{V}|!}{\prod_{v\in\mathring{V}} \lambda_v}$ possible rankings by (1.3.8), and $|\mathring{V}| = n-1$ for binary trees, we get

$$\mathbb{P}[r|\mathcal{T}] = \frac{1}{\frac{|\mathring{V}|!}{\prod_{v\in\mathring{V}} \lambda_v}} = \frac{\prod_{v\in\mathring{V}} \lambda_v}{(n-1)!}.$$

□

The following Corollary was established in [4] using induction.

Corollary 2.2.5. *The probability of a binary phylogenetic tree $\mathcal{T} \in RB(n)$ under the Yule model is*

$$\mathbb{P}[\mathcal{T}] = \frac{2^{n-1}}{n! \prod_{v\in\mathring{V}} \lambda_v}$$

where λ_v is as defined in Lemma (1.3.8).

Proof. With Theorem (2.2.1) and Theorem (2.2.4) we get

$$\mathbb{P}[\mathcal{T}] = \frac{\mathbb{P}[\mathcal{T}, r]}{\mathbb{P}[r|\mathcal{T}]} = \frac{2^{n-1}}{n!(n-1)!} \cdot \frac{(n-1)!}{\prod_{v\in\mathring{V}} \lambda_v} = \frac{2^{n-1}}{n! \prod_{v\in\mathring{V}} \lambda_v}.$$

□

Example 2.2.6. Recall again the ranked tree $(\mathcal{T}, r)$ in Fig. 1.6. In that tree, $X = \{a, b, \ldots, k\}$ and $n = |X| = 11$. Let $\mathbb{P}_Y[\mathcal{T}, r]$ be the probability that the ranked tree $(\mathcal{T}, r)$ evolved under the Yule model. With Theorem (2.2.1), we get

$$\mathbb{P}_Y[\mathcal{T}, r] = \frac{2^{n-1}}{n!(n-1)!} = \frac{2^{10}}{11! \times 10!} \approx 0.71 \times 10^{-11}$$

With Corollary (2.2.5), we get

$$\mathbb{P}_Y[\mathcal{T}] = \frac{2^{n-1}}{n! \prod_{v \in \mathring{V}} \lambda_v} = \frac{2^{10}}{11! \times 1^5 \times 2 \times 3 \times 4 \times 5 \times 10} \approx 0.21 \times 10^{-7}$$

With Theorem (2.2.4), we get

$$\mathbb{P}_Y[r|\mathcal{T}] = \frac{\prod_{v \in \mathring{V}} \lambda_v}{(n-1)!} = \frac{1^5 \times 2 \times 3 \times 4 \times 5 \times 10}{10!} \approx 0.33 \times 10^{-3}$$

Let $\mathbb{P}_U[\mathcal{T}]$ be the probability that $\mathcal{T}$ evolved under the uniform model. Then,

$$\mathbb{P}_U[\mathcal{T}] = 1/(2n-3)!! \approx 0.15 \times 10^{-8}$$

Since $\frac{\mathbb{P}_Y[\mathcal{T}]}{\mathbb{P}_U[\mathcal{T}]} \approx \frac{0.21}{0.15} \times 10^1 = 14 > 1$, i.e. $\mathbb{P}_Y[\mathcal{T}] > \mathbb{P}_U[\mathcal{T}]$, the tree $\mathcal{T}$ (without a ranking) is more likely to have evolved under the Yule model.

Remark 2.2.7. In Chapter 4, we want to calculate for a given phylogenetic tree $\mathcal{T}$ the probability $\mathbb{P}[r(v) = i, r \in r(\mathcal{T})|\mathcal{T}]$ for a $v \in \mathring{V}$ under the Yule model where $r(\mathcal{T})$ as defined in (1.3.7). By Theorem (2.2.4), the rankings for $\mathcal{T}$ all have the same probability, and therefore

$$\mathbb{P}[r(v) = i, r \in r(\mathcal{T})|\mathcal{T}] = \frac{|\{r \in r(\mathcal{T}) : r(v) = i\}|}{|r(\mathcal{T})|}.$$

For the value $|r(\mathcal{T})|$, a formula is stated in Lemma 1.3.8. The value $|\{r \in r(\mathcal{T}) : r(v) = i\}|$ will be calculated with the algorithm RankCount.

Remark 2.2.8. Another stochastic model on trees is the coalescent model. The coalescent model starts with n species and goes back in time. At each event, two species are selected uniformly at random and the two species are joint together, the joint being a new species, the ancestor. So after $n-1$ joining events, we are left with one species, the root of the tree.

With i remaining species, we have $\binom{i}{2}$ possibilities to choose two species for the joint. The probability for a specific ranked tree is therefore

$$\mathbb{P}[\mathcal{T}, r] = \frac{1}{\binom{n}{2}\binom{n-1}{2}\ldots\binom{2}{2}} = \frac{2^{n-1}}{n!(n-1)!}$$

which is equivalent to the Yule model.

Thus, the Yule model and the coalescent model are equivalent as long as edge lengths are not considered.

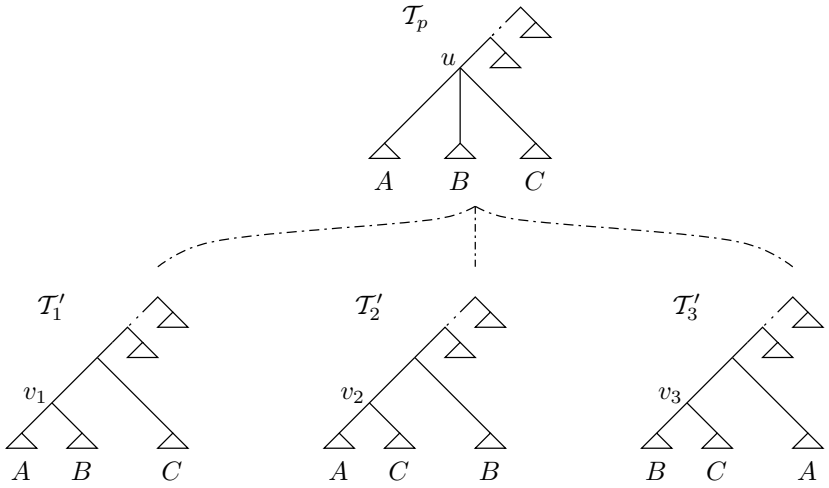


Figure 2.3: Vertex in $\mathcal{T}_p$ with three direct descendants. There are three possible binary resolutions.

2.2.1 Did the primate tree evolve under Yule?

Consider the primate tree $\mathcal{T}_p$ in Appendix C. $\mathcal{T}_p$ has $n = 218$ leaves. We want to calculate the value $\frac{\mathbb{P}_Y[\mathcal{T}_p]}{\mathbb{P}_U[\mathcal{T}_p]}$ in order to decide whether to favor the Yule model over the uniform model. Note that $\mathbb{P}_U[\mathcal{T}] = \frac{2^{n-1}}{n! c_{n-1}}$ and $\mathbb{P}_Y[\mathcal{T}] = \frac{2^{n-1}}{n! \prod_{v \in \mathring{V}} \lambda_v}$.

In $\mathcal{T}_p$, there are six vertices (vertex labels $48, 63, 148, 153, 157$ and 200) with more than two direct descendants because the exact resolution is unclear. Five of those vertices have three direct descendants.

For each vertex with three direct descendants, there are three possible binary resolutions, see Fig. 2.3.

Let u be a vertex of $\mathcal{T}_p$ with three direct descandants. Let v be the additional vertex for a binary resolution of vertex u. For the three different binary resolutions of vertex u, we also write v_1, v_2, v_3 instead of v, see Fig. 2.3.

Let $\mathcal{T}'$ be a binary resolution of $\mathcal{T}_p$. Let $\mathcal{T}_i'$, $i = 1, 2, 3$, be a binary resolution of $\mathcal{T}_p$ where vertex u is resolved as displayed in Fig. 2.3. Let $\lambda_{v(\mathcal{T}')}$ be the number

of descendants of v in resolution $\mathcal{T}'$. We want to estimate $\tilde{\lambda_v}$.

$$\begin{aligned}
\tilde{\lambda_v} &= \frac{\sum_{\mathcal{T}'} \lambda_{v(\mathcal{T}')} \mathbb{P}[\mathcal{T}']}{\sum_{\mathcal{T}'} \mathbb{P}[\mathcal{T}']} \\
&= \frac{\sum_{i=1}^{3} \sum_{\mathcal{T}_i'} \lambda_{v_i} \mathbb{P}[\mathcal{T}_i']}{\sum_{i=1}^{3} \sum_{\mathcal{T}_i'} \mathbb{P}[\mathcal{T}_i']} \\
&= \frac{\sum_{i=1}^{3} \sum_{\mathcal{T}_i'} \lambda_{v_i} \frac{2^n}{n! \prod_{w \in \mathring{V}_{\mathcal{T}_i'}} \lambda_w}}{\sum_{i=1}^{3} \sum_{\mathcal{T}_i'} \frac{2^n}{n! \prod_{w \in \mathring{V}_{\mathcal{T}_i'}} \lambda_w}} \\
&= \frac{\sum_{i=1}^{3} \sum_{\mathcal{T}_i'} \frac{2^n}{n! \prod_{w \in \{\mathring{V}_{\mathcal{T}_i'} \setminus v_i\}} \lambda_w}}{\sum_{i=1}^{3} \frac{1}{\lambda_{v_i}} \sum_{\mathcal{T}_i'} \frac{2^n}{n! \prod_{w \in \{\mathring{V}_{\mathcal{T}_i'} \setminus v_i\}} \lambda_w}}
\end{aligned}$$

Note that the inner sum is constant for all i, so we get

$$\begin{aligned}
\tilde{\lambda_v} &= \frac{\sum_{\mathcal{T}_1'} \frac{2^n}{n! \prod_{w \in \{\mathring{V}_{\mathcal{T}_1'} \setminus v_1\}} \lambda_w} \sum_{i=1}^{3} 1}{\sum_{\mathcal{T}_1'} \frac{2^n}{n! \prod_{w \in \{\mathring{V}_{\mathcal{T}_1'} \setminus v_1\}} \lambda_w} \sum_{i=1}^{3} \frac{1}{\lambda_{v_i}}} \\
&= \frac{3}{\sum_{i=1}^{3} \frac{1}{\lambda_{v_i}}}
\end{aligned}$$

With this formula, we estimate the values $\tilde{\lambda_v}$ for the new vertex v in the binary resolution of vertex $48, 63, 153, 157$ and 200.

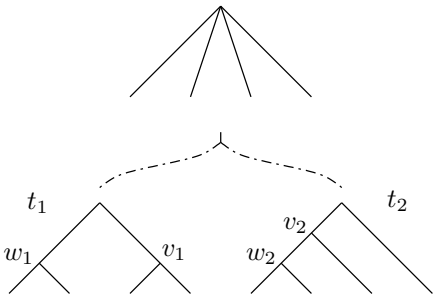


Figure 2.4: Vertex in $\mathcal{T}_p$ with four leaf-descendants.

The interior vertex with label 148 has four leaves as direct descendants. There are two different shapes t_1 and t_2 for a binary tree with four leaves, see Fig. 2.4. In t_1, the new interior vertices v_1 and w_1 have the value $\lambda_{v_1} = 1$ and $\lambda_{w_1} = 1$. In t_2, the new vertex v_2 has $\lambda_{v_2} = 2$, the new vertex w_2 has $\lambda_{w_2} = 1$. We set $\tilde{\lambda_w} = 1$ in $\mathcal{T}_p$ since $\lambda_{w_1} = 1$ and $\lambda_{w_2} = 1$. We want to estimate $\tilde{\lambda_v}$, the value $\tilde{\lambda_v}$ shall be the weighted sum of the λ_{v_i},

$$\tilde{\lambda_v} = \frac{\mathbb{P}_Y[t_1]\lambda_{v_1} + \mathbb{P}_Y[t_2]\lambda_{v_2}}{\mathbb{P}_Y[t_1] + \mathbb{P}_Y[t_2]} = 1/3 \cdot 1 + 2/3 \cdot 2 = 5/3.$$

With those estimated values for $\tilde{\lambda_v}$, we now estimate $\frac{\mathbb{P}_Y[\mathcal{T}]}{\mathbb{P}_U[\mathcal{T}]}$. Let $\mathcal{T}_i, i = 1, \ldots, m$, be the binary resolutions of $\mathcal{T}$. We get

$$\frac{\mathbb{P}_Y[\mathcal{T}]}{\mathbb{P}_U[\mathcal{T}]} = \frac{\sum_i \mathbb{P}_Y[\mathcal{T}_i]}{\sum_i \mathbb{P}_U[\mathcal{T}_i]} \approx \frac{c_{n-1}}{\prod_{v \in \mathring{V}_\mathcal{T}} \lambda_v \cdot \prod \tilde{\lambda_v}} \approx 0.25 \times 10^{14}$$

which favors the Yule model over the uniform model. Note that without the estimates for $\tilde{\lambda_v}$, we would have to calculate $\mathbb{P}_Y[\mathcal{T}_i]$ and $\mathbb{P}_U[\mathcal{T}_i]$ for the $3^5 \times 15$ linear resolutions of $\mathcal{T}$.

In Section 4.3, we will assume that the primate tree $\mathcal{T}_p$ evolved under the Yule model.

2.3 Yule model vs. uniform model

As we have seen in Corollary (2.1.3), the probability of generating a given tree $\mathcal{T}$ with n leaves under the uniform model is

$$\mathbb{P}_U[\mathcal{T}] = \frac{2^{n-1}}{n! c_{n-1}}.$$

By Corollary (2.2.5), the probability of generating a given tree $\mathcal{T}$ under the Yule model is

$$\mathbb{P}_Y[\mathcal{T}] = \frac{2^{n-1}}{n! \prod_{v \in \mathring{V}} \lambda_v}.$$

The fraction of the two probabilities, the 'Bayes factor' [6], is

$$\frac{\mathbb{P}_Y[\mathcal{T}]}{\mathbb{P}_U[\mathcal{T}]} = \frac{c_{n-1}}{\prod_{v\in\mathring{V}} \lambda_v}.$$

Given a tree $\mathcal{T}$, we want to know if it evolved under the Yule or the uniform model. The fraction $\frac{\mathbb{P}_Y[\mathcal{T}]}{\mathbb{P}_U[\mathcal{T}]}$ being bigger than 1 suggests favoring the Yule model, the fraction being smaller than 1 suggests favoring the uniform model. So $\ln\left(\frac{\mathbb{P}_Y[\mathcal{T}]}{\mathbb{P}_U[\mathcal{T}]}\right)$ being bigger than 0 suggests favoring the Yule model, the logarithm being smaller than 0 suggests favoring the uniform model. In the following, we want to calculate the expected value $\mathbb{E}_Y[\ln\left(\frac{\mathbb{P}_Y[\mathcal{T}]}{\mathbb{P}_U[\mathcal{T}]}\right)]$, given the tree $\mathcal{T}$ evolved under the Yule model. We will see that $\mathbb{E}_Y\left[\ln\left(\frac{\mathbb{P}_Y[\mathcal{T}]}{\mathbb{P}_U[\mathcal{T}]}\right)\right]$ is the 'Kullbach-Liebler' distance (defined below) between $\mathbb{P}_Y$ and $\mathbb{P}_U$, and show that it goes to infinity with increasing n. Further, $\mathbb{E}_U\left[\ln\left(\frac{\mathbb{P}_U[\mathcal{T}]}{\mathbb{P}_Y[\mathcal{T}]}\right)\right]$ goes to infinity with increasing n. Therefore, for n large enough, the value $\ln\left(\frac{\mathbb{P}_Y[\mathcal{T}]}{\mathbb{P}_U[\mathcal{T}]}\right)$ is relevant to the question of testing whether a tree evolved under the Yule or uniform model. In Section 3.4, we will actually test the Yule model against the uniform model.

2.3.1 The Kullbach-Liebler distance

Definition 2.3.1. Let X be a discrete random variable which takes values in the finite set $\Omega = \{w_1, w_2, \ldots, w_n\}$ with associated probabilities $\{p(\omega_1), p(\omega_2), \ldots, p(\omega_n)\}$. We call this probability distribution p. The *information content* of an event $\omega \in \Omega$ is

$$I(\omega) = -\ln p(\omega)$$

The *entropy* $\mathbb{J}_p$ of the probability distribution p is defined as

$$\mathbb{J}_p = \mathbb{E}[I(X)] = -\sum_{\omega\in\Omega} p(\omega)\ln p(\omega)$$

In [9], Chapter 7, the entropy $\mathbb{J}_Y$ for the Yule distribution over $RB(n)$ and the entropy $\mathbb{J}_U$ for the uniform distribution over $RB(n)$ are calculated. Recall that for two functions $f(n)$ and $g(n)$, we write $f(n) \sim g(n)$ precisely if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 1$.

For $\mathbb{J}_Y$, one has (from [9])

$$\mathbb{J}_Y = n\sum_{k=2}^{n-1} \frac{g(k)}{k+1} \tag{2.1}$$

where $g(k) = \frac{1-k}{k}\ln\frac{k-1}{2} + \ln\frac{k}{2} + \ln(k+1) - \frac{1}{k}\ln k!$. Asymptotically, one has

$$\mathbb{J}_Y - n\ln(n) + c_1 n \sim -\frac{1}{2}\ln(n) \tag{2.2}$$

where $c_1 = \ln(2)\ln(\frac{200}{49e}) + \ln(9)\ln(\frac{7}{10}) + 2\mathrm{Li}_2(\frac{7}{4}) - 2\mathrm{Li}_2(\frac{5}{2}) - 1 \approx 0.493$ and $\mathrm{Li}_2(x) = \int_1^x \frac{\ln t}{1-t} dt$.

For $\mathbb{J}_U$, one has (again from [9])

$$\mathbb{J}_U = \ln |RB(n)| = \ln(2n-3)!! \tag{2.3}$$

and asymptotically

$$\mathbb{J}_U - n\ln(n) + c_2 n \sim -\ln(n) \tag{2.4}$$

where $c_2 = 1 - \ln(2) \approx 0.307$.

Definition 2.3.2. Let p and q be probability distributions over a finite set Ω. The *Kullbach-Liebler distance* between p and q is defined as

$$d_{KL}(p,q) = \sum_{\omega\in\Omega} p(\omega)\ln\frac{p(\omega)}{q(\omega)}.$$

Remark 2.3.3. The Kullbach-Liebler distance is positive definite, i.e. $d_{KL}(p,q) \geq 0$ with $d_{KL}(p,q) = 0$ iff $p = q$. Notice that $d_{KL}(p,q) = \infty$ iff there exists a $u \in \Omega$ with $p(u) > 0$, $q(u) = 0$. For $p = \mathbb{P}_Y$ and $q = \mathbb{P}_U$, both $d_{KL}(p,q)$ and $d_{KL}(q,p)$ are finite, since $\mathbb{P}_Y[\mathcal{T}] > 0$ and $\mathbb{P}_U[\mathcal{T}] > 0$ for all $\mathcal{T} \in RB(n)$. Note that the Kullbach-Liebler distance between p and q is not symmetric, i.e. we have $d_{KL}(p,q) \neq d_{KL}(q,p)$ in general.

Remark 2.3.4. Note that the Kullbach-Liebler distance between the probability distributions p and q over the set Ω equals the following expected value

$$d_{KL}(p,q) = \sum_{\omega\in\Omega} p(\omega)\ln\frac{p(\omega)}{q(\omega)} = \mathbb{E}_p[\ln\frac{p}{q}].$$

Lemma 2.3.5. *Let Ω be a finite set. Let p be any probability distribution over Ω, and let q be the uniform distribution over Ω. Then*

$$d_{KL}(p,q) = \mathbb{J}_q - \mathbb{J}_p.$$

Proof. By assumption, $q(\omega) = 1/|\Omega|$ for all $\omega \in \Omega$. From the definition of $d_{KL}(p,q)$,

it follows that

$$\begin{aligned} d_{KL}(p,q) &= \sum_{\omega\in\Omega} p(\omega)\ln\frac{p(\omega)}{q(\omega)} \\ &= \sum_{\omega\in\Omega} p(\omega)\ln p(\omega) - \sum_{\omega\in\Omega} p(\omega)\ln q(\omega) \\ &= -\mathbb{J}_p - \sum_{\omega\in\Omega} p(\omega)\ln\frac{1}{|\Omega|} \\ &= -\mathbb{J}_p - \left(\ln\frac{1}{|\Omega|}\right)\sum_{\omega\in\Omega} p(\omega) \\ &= -\mathbb{J}_p - \ln\frac{1}{|\Omega|} \\ &= -\mathbb{J}_p - \sum_{\omega\in\Omega}\frac{1}{|\Omega|}\ln\frac{1}{|\Omega|} \\ &= \mathbb{J}_q - \mathbb{J}_p. \end{aligned}$$

□

2.3.2 Kullbach-Liebler distance between $\mathbb{P}_Y$ and $\mathbb{P}_U$

In the following, we calculate the Kullbach-Liebler distance between the Yule distribution $\mathbb{P}_Y$ and the uniform distribution $\mathbb{P}_U$ over $RB(n)$.

Theorem 2.3.6. *Let $\mathbb{P}_Y$ be the Yule distribution and $\mathbb{P}_U$ be the uniform distribution over $RB(n)$. The Kullbach-Liebler-distance between those two distributions is*

$$d_{KL}(\mathbb{P}_Y,\mathbb{P}_U) = \ln(2n-3)!! - n\sum_{k=2}^{n-1}\frac{g(k)}{k+1}$$

where $g(k)$ is again defined as $g(k) = \frac{1-k}{k}\ln\frac{k-1}{2} + \ln\frac{k}{2} + \ln(k+1) - \frac{1}{k}\ln k!$. Asymptotically, we have

$$d_{KL}(\mathbb{P}_Y,\mathbb{P}_U) - c_Y n \sim -1/2\ln(n)$$

with $c_Y \approx 0.186$.

Proof. From Lemma (2.3.5), we have $d_{KL}(\mathbb{P}_Y,\mathbb{P}_U) = \mathbb{J}_U - \mathbb{J}_Y$. With Equations (2.1) and (2.3), we get $d_{KL}(\mathbb{P}_Y,\mathbb{P}_U) = \ln(2n-3)!! - n\sum_{k=2}^{n-1}\frac{g(k)}{k+1}$. For the asymptotic behavior, we get with Equation (2.2) and (2.4)

$$\begin{aligned} \mathbb{J}_U - n\ln(n) + c_2 n - (\mathbb{J}_Y - n\ln(n) + c_1 n) &\sim -\ln(n) + 1/2\ln(n) \\ \mathbb{J}_U - \mathbb{J}_Y - (c_1 - c_2)n &\sim -1/2\ln(n) \\ \mathbb{J}_U - \mathbb{J}_Y - c_Y n &\sim -1/2\ln(n) \end{aligned}$$

where $c_Y = c_1 - c_2 \approx 0.186$. □

Corollary 2.3.7. *For the expected value* $\mathbb{E}_Y[\ln \frac{\mathbb{P}_Y}{\mathbb{P}_U}]$*, we get*

$$\mathbb{E}_Y[\ln \frac{\mathbb{P}_Y}{\mathbb{P}_U}] - c_Y n \sim -1/2 \ln(n)$$

So $\mathbb{E}_Y[\ln \frac{\mathbb{P}_Y}{\mathbb{P}_U}] \to \infty$ *for* $n \to \infty$*.*

Proof. With Theorem (2.3.6), we get

$$\begin{aligned} \mathbb{E}_Y[\ln \frac{\mathbb{P}_Y}{\mathbb{P}_U}] - c_Y n &= \sum_{\mathcal{T} \in RB(n)} \mathbb{P}_Y[\mathcal{T}] \ln \frac{\mathbb{P}_Y[\mathcal{T}]}{\mathbb{P}_U[\mathcal{T}]} - c_Y n \\ &= d_{KL}(\mathbb{P}_Y, \mathbb{P}_U) - c_Y n \\ &\sim -1/2 \ln(n) \end{aligned}$$

That implies $d_{KL}(\mathbb{P}_Y, \mathbb{P}_U) \sim c_Y n$ and since $c_Y > 0$, we have $\mathbb{E}_Y[\ln \frac{\mathbb{P}_Y}{\mathbb{P}_U}] \to \infty$ for $n \to \infty$. □

2.3.3 Kullbach-Liebler distance between $\mathbb{P}_U$ and $\mathbb{P}_Y$

In the following, we calculate the Kullbach-Liebler distance between the uniform distribution $\mathbb{P}_U$ and the Yule distribution $\mathbb{P}_Y$ over $RB(n)$.

Lemma 2.3.8. *The central binomial coefficient* $\binom{2m}{m}$ *can be written as*

$$\binom{2m}{m} = 2^{2m} \prod_{j=1}^{m} \frac{2j-1}{2j}.$$

Proof.

$$\begin{aligned} \binom{2m}{m} &= \frac{(2m)!}{m!m!} = \frac{2^{2m} \cdot 2m \cdot (2m-1) \cdot (2m-2) \ldots 3 \cdot 2 \cdot 1}{2m \cdot 2m \cdot 2(m-1) \cdot 2(m-1) \ldots 4 \cdot 4 \cdot 2 \cdot 2} \\ &= 2^{2m} \prod_{j=0}^{m-1} \frac{2m-2j-1}{2(m-j)} \\ &= 2^{2m} \prod_{j=1}^{m} \frac{2j-1}{2j}. \end{aligned}$$

□

Lemma 2.3.9. *For the set* $RB(n)$*, we have*

$$\sum_{\mathcal{T} \in RB(n)} \sum_{v \in \mathring{V}_{\mathcal{T}}} \ln \lambda_v = \sum_{i=1}^{n-1} \ln i \binom{n}{i+1} |RB(i+1)||RB(n-i)|$$

where λ_v *is defined as in Lemma (1.3.8).*

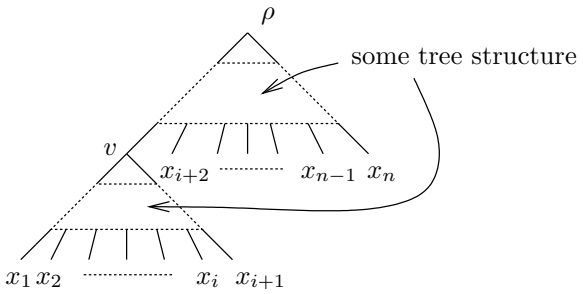


Figure 2.5: Counting the pairs $(\mathcal{T}, v)$ in Lemma (2.3.9). The variables $(x_1, \ldots, x_{i+1})$ take any distinct values from X', the variables $(x_{i+2}, \ldots, x_{n-1}, x_n)$ take any distinct values from X''.

Proof. We have $\lambda_v \in \{1, 2, \ldots, (n-1)\}$ since a binary tree $\mathcal{T}$ with n leaves has $n-1$ interior vertices. We rewrite the double sum as

$$\sum_{\mathcal{T} \in RB(n)} \sum_{v \in \mathring{V}_\mathcal{T}} \ln \lambda_v = \sum_{i=1}^{n-1} \ln i \cdot |\{(\mathcal{T}, v) : \mathcal{T} \in RB(n), v \in \mathring{V}_\mathcal{T}, \lambda_v = i\}|$$

To calculate $|\{(\mathcal{T}, v) : \mathcal{T} \in RB(n), v \in \mathring{V}_\mathcal{T}, \lambda_v = i\}|$, we have to count all the pairs $(\mathcal{T}, v)$ with $v \in \mathring{V}_\mathcal{T}$ having exactly i interior nodes as descendants. For a binary tree, this is equivalent to v having $i+1$ leaves as descendants (*cf.* Figure 2.5). So for an interior vertex v, we choose a subset X' of X consisting of $i+1$ elements, which shall label the leaf descendants of v. We have $\binom{n}{i+1}$ possibilities to choose those $i+1$ elements. There are $|RB(i+1)|$ possibilities to build up a binary tree with leaf set X' and root v. Let $X'' = (X \setminus X') \cup v$, so $|X''| = n - i$. For the set X'', there are $|RB(n-i)|$ possible binary trees. Combining all those possibilities yields

$$|\{\mathcal{T}, v : \mathcal{T} \in RB(n), v \in \mathring{V}_\mathcal{T}, \lambda_v = i\}| = \binom{n}{i+1} |RB(i+1)||RB(n-i)|$$

which proves the Lemma. □

Theorem 2.3.10. *For the distance $d_{KL}(\mathbb{P}_U, \mathbb{P}_Y)$, it holds that*

$$d_{KL}(\mathbb{P}_U, \mathbb{P}_Y) = nS_n - \ln c_{n-1}$$

where $S_n = \sum_{i=2}^{n-1} \left[\frac{\ln i}{i+1} \prod_{j=1}^{n-i-1} \frac{1-\frac{1}{2j}}{1-\frac{1}{2(j+i)}} \right]$ and c_n are the Catalan numbers as defined in Lemma (2.1.1).

Proof. By definition of the Kullbach-Liebler distance and with Corollary (2.1.3)

and (2.2.5) and setting $N = |RB(n)|$, we have,

$$\begin{aligned} d_{KL}(\mathbb{P}_U, \mathbb{P}_Y) &= \sum_{\mathcal{T}\in RB(n)} \mathbb{P}_U[\mathcal{T}] \ln \frac{\mathbb{P}_U[\mathcal{T}]}{\mathbb{P}_Y[\mathcal{T}]} \\ &= \sum_{\mathcal{T}\in RB(n)} \frac{2^{n-1}}{n! c_{n-1}} \ln \left[\frac{\frac{2^{n-1}}{n! c_{n-1}}}{\frac{2^{n-1}}{n! \prod_{v\in \mathring{V}_\mathcal{T}} \lambda_v}} \right] \\ &= \sum_{\mathcal{T}\in RB(n)} \frac{1}{N} \ln \left[\frac{\prod_{v\in \mathring{V}_\mathcal{T}} \lambda_v}{c_{n-1}} \right] \\ &= \frac{1}{N} \left[\sum_{\mathcal{T}\in RB(n)} \sum_{v\in \mathring{V}_\mathcal{T}} \ln \lambda_v \right] - \ln c_{n-1} \\ &= \frac{1}{N} s - \ln c_{n-1} \end{aligned} \tag{2.5}$$

where $s = \sum_{\mathcal{T}\in RB(n)} \sum_{v\in \mathring{V}_\mathcal{T}} \ln \lambda_v$. With Lemma (2.3.9) and Lemma(2.1.1), we get

$$\begin{aligned} s &= \sum_{\mathcal{T}\in RB(n)} \sum_{v\in \mathring{V}_\mathcal{T}} \ln \lambda_v \\ &= \sum_{i=2}^{n-1} \ln i \binom{n}{i+1} |RB(i+1)||RB(n-i)| \\ &= \sum_{i=2}^{n-1} \ln i \binom{n}{i+1} \frac{c_i (i+1)!}{2^i} \cdot \frac{c_{n-i-1}(n-i)!}{2^{n-i-1}} \\ &= \frac{n!}{2^{n-1}} \sum_{i=2}^{n-1} \ln i \frac{(i+1)!(n-i)!}{(i+1)!(n-i-1)!} c_i c_{n-i-1} \\ &= \frac{N}{c_{n-1}} \sum_{i=2}^{n-1} \ln i \cdot (n-i) \cdot c_i c_{n-i-1} \\ &= \frac{Nn}{\binom{2(n-1)}{n-1}} \sum_{i=2}^{n-1} \frac{\ln i}{i+1} \binom{2i}{i} \binom{2(n-i-1)}{n-i-1} \end{aligned}$$

With Lemma (2.3.8) we get

$$\begin{aligned}
s &= \frac{Nn}{2^{2(n-1)}\prod_{j=1}^{n-1}\frac{2j-1}{2j}}\sum_{i=2}^{n-1}\left[\frac{\ln i}{i+1}2^{2i}\prod_{j=1}^{i}\frac{2j-1}{2j}2^{2(n-i-1)}\prod_{j=1}^{n-i-1}\frac{2j-1}{2j}\right] \\
&= Nn\sum_{i=2}^{n-1}\left[\frac{\ln i}{i+1}\prod_{j=1}^{n-1}\frac{2j}{2j-1}\prod_{j=1}^{i}\frac{2j-1}{2j}\prod_{j=1}^{n-i-1}\frac{2j-1}{2j}\right] \\
&= Nn\sum_{i=2}^{n-1}\left[\frac{\ln i}{i+1}\prod_{j=i+1}^{n-1}\frac{2j}{2j-1}\prod_{j=1}^{n-i-1}\frac{2j-1}{2j}\right] \\
&= Nn\sum_{i=2}^{n-1}\left[\frac{\ln i}{i+1}\prod_{j=1}^{n-i-1}\frac{2(j+i)}{2(j+i)-1}\prod_{j=1}^{n-i-1}\frac{2j-1}{2j}\right] \\
&= Nn\sum_{i=2}^{n-1}\left[\frac{\ln i}{i+1}\prod_{j=1}^{n-i-1}\frac{(j+i)(2j-1)}{(2(j+i)-1)j}\right] \\
&= Nn\sum_{i=2}^{n-1}\left[\frac{\ln i}{i+1}\prod_{j=1}^{n-i-1}\frac{2j-1}{2j-\frac{2j}{2(j+i)}}\right] \\
&= Nn\sum_{i=2}^{n-1}\left[\frac{\ln i}{i+1}\prod_{j=1}^{n-i-1}\frac{1-\frac{1}{2j}}{1-\frac{1}{2(j+i)}}\right]
\end{aligned}$$

Combining this result with Equation (2.5) establishes the theorem. □

Lemma 2.3.11. *The asymptotic behavior of the n-th Catalan number c_n is*

$$c_n \sim n\ln 4$$

Proof. With the Stirling formula, $\ln n! \sim n\ln n - n$ (see [3]),we get

$$\begin{aligned}
\ln c_n &= -\ln(n+1) + \ln\binom{2n}{n} \\
&= -\ln(n+1) + \ln(2n)! - 2\ln n! \\
&\sim -\ln(n+1) + 2n\ln 2n - 2n - 2n\ln n + 2n \\
&= -\ln(n+1) + 2n\ln 2 \\
&\sim n\ln 4
\end{aligned}$$

□

Theorem 2.3.12. *The Kullbach-Liebler distance between $\mathbb{P}_U$ and $\mathbb{P}_Y$ is asymptotically*

$$d_{KL}(\mathbb{P}_U, \mathbb{P}_Y) \sim c_U n$$

where c_U is a positive constant.

Proof. From Theorem (2.3.10), we have

$$d_{KL}(\mathbb{P}_U, \mathbb{P}_Y) = n\mathcal{S}_n - \ln c_{n-1}$$

with $\mathcal{S}_n = \sum_{i=2}^{n-1} \left[\frac{\ln i}{i+1} \prod_{j=1}^{n-i-1} \frac{1-\frac{1}{2j}}{1-\frac{1}{2(j+i)}} \right]$ and c_n being the n-th Catalan number. By Lemma (2.3.11), it holds $c_{n-1} \sim n \ln 4$. In Section 2.3.4, we show that

$$\ln 4 < 1.44 < S_n < S' + N$$

for all $n \geq 200$ with S' and N being some fixed constants. This yields to

$$d_{KL}(\mathbb{P}_U, \mathbb{P}_Y) = nS_n - \ln c_{n-1} \sim nS_n - n \ln 4 \sim c_U n$$

with c_U being a positive constant. □

Corollary 2.3.13. *We obtain*

$$\mathbb{E}_U[\ln \frac{\mathbb{P}_U}{\mathbb{P}_Y}] \to \infty \qquad \text{for } n \to \infty$$

since $\mathbb{E}_U[\ln \frac{\mathbb{P}_U}{\mathbb{P}_Y}] = d_{KL}(\mathbb{P}_U, \mathbb{P}_Y)$ *by Remark (2.3.4).*

2.3.4 Calculating S_n

In Theorem (2.3.10), we obtain the following formula for the Kullbach-Liebler distance between $\mathbb{P}_U$ and $\mathbb{P}_Y$:

$$d_{KL}(\mathbb{P}_U, \mathbb{P}_Y) = nS_n - \ln c_{n-1}$$

with $S_n = \sum_{i=2}^{n-1} \left[\frac{\ln i}{i+1} \cdot a_{n,i} \right]$ and $a_{n,i} = \prod_{j=1}^{n-i-1} \frac{1-\frac{1}{2j}}{1-\frac{1}{2(j+i)}}$. In the following, we will calculate an upper and a lower bound for S_n. Note that $\{a_{n,i}, n \in \mathbb{N}\}$ is monotone decreasing for fixed i and $a_{n,i} > 0$. So $\lim_{n\to\infty} a_{n,i}$ exists.

$$a_i := \lim_{n\to\infty} a_{n,i} = \prod_{j=1}^{\infty} \frac{1-\frac{1}{2j}}{1-\frac{1}{2(j+i)}} = \prod_{j=1}^{i} \left(1 - \frac{1}{2j}\right) > 0$$

$$S'_n := \sum_{i=2}^{n-1} \left[\frac{\ln i}{i+1} \cdot a_i \right]$$

With the property

$$\ln(1-x) = -x - \sum_{i=2}^{\infty} \frac{x^i}{i} \leq -x$$

for $0 \leq x < 1$ (see [19]) and the property

$$\sum_{j=1}^{i} \frac{1}{j} \geq \int_1^i \frac{1}{x} dx = \ln(i)$$

we get the following:

$$\begin{aligned}\ln a_i &= \sum_{j=1}^{i} \ln(1-\frac{1}{2j}) \\ &\leq -\frac{1}{2}\sum_{j=1}^{i}\frac{1}{j} \\ &\leq -\frac{1}{2}\ln(i)\end{aligned}$$

So we have

$$a_i \leq \frac{1}{\sqrt{i}}$$

In the following, we show that S'_n converges.

$$\begin{aligned}S'_n &= \sum_{i=2}^{n-1}\left[\frac{\ln i}{i+1}\cdot a_i\right] \\ &\leq \sum_{i=2}^{n-1}\frac{\ln i}{i^{3/2}}\end{aligned}$$

Since $\sum_{i=2}^{\infty}\frac{\ln i}{i^{3/2}}$ converges, it follows that $\{S'_n, n\in\mathbb{N}\}$ is bounded. The sequence $\{S'_n, n\in\mathbb{N}\}$ is monotone increasing since $\frac{\ln i}{i+1}\cdot a_i > 0$ for all $i\in\mathbb{N}, i\geq 2$. So $\lim_{n\to\infty} S'_n$ exists and we define

$$\lim_{n\to\infty} S'_n := S'.$$

Now we calculate an upper and a lower bound for S_n. Since $a_{i,n}\to a_i$, there exists an $N\in\mathbb{N}$ s.t. $a_{i,n} < (1+1/S')a_i$ for all $n > N$.

$$S_n = \sum_{i=2}^{n-1}\left[\frac{\ln i}{i+1}\cdot a_{n,i}\right] < (N-1) + \sum_{i=N+1}^{n-1}\left[\frac{\ln i}{i+1}\cdot(1+1/S')a_i\right] < (N-1)+(1+1/S')S'_n$$

Since S'_n is monotone increasing, we get

$$S_n < (N-1) + (1+1/S')S'_n < (N-1) + (1+1/S')S'$$

which yields to

$$S_n < S' + N.$$

Since $a_{i,n} > a_i$, we have

$$S_n = \sum_{i=2}^{n-1}\left[\frac{\ln i}{i+1}\cdot a_{n,i}\right] > \sum_{i=2}^{n-1}\left[\frac{\ln i}{i+1}\cdot a_i\right] = S'_n$$

So we get $S_n > S'_n$ for all n. With Maple, I calculated $S'_{200}\approx 1.44 > \ln 4$. Overall, we have

$$\ln 4 < 1.44 < S_n < S' + N$$

for all $n\geq 200$.

Chapter 3

Trees and Martingales

In this chapter, we have a closer look at the process of the tree generation. We will see that the tree generation is a certain stochastic process, a martingale. Under the uniform model, the martingale fulfills the conditions for the Azuma inequality.

We make use of this property at the end of the chapter. We test the Yule model against the uniform model with the log-likelihood-ratio test. With the Azuma inequality, we find an analytical bound for the power of the test. Since the algorithms in Chapter 4 work in particular for trees under the Yule model, it will be useful to have a test for deciding whether a tree evolved under Yule.

First, we provide some basic definitions and properties on conditional probability and martingales.

3.1 Conditional probability and martingales

Definition 3.1.1. Let X (resp. Y) be a discrete random variable which takes values $\{x_i, i \in \mathbb{N}\}$ (resp. $\{y_i, i \in \mathbb{N}\}$). The *conditional expectation*

$$Z = \mathbb{E}[X|Y] = \sum_j x_j \mathbb{P}[X = x_j|Y]$$

is a random variable. Z takes values

$$z_i = \sum_j x_j \mathbb{P}[X = x_j|Y = y_i]$$

on the set $\{Y = y_i\}$ with probability $\mathbb{P}[Z = z_i] = \mathbb{P}[Y = y_i]$.

The two equations in the next Lemma are stated in [13] with a brief verification. We will give a full proof.

Lemma 3.1.2. *Let X (resp. Y, U) be a discrete random variable which takes values $\{x_i, i \in \mathbb{N}\}$ (resp. $\{y_i, i \in \mathbb{N}\}$, $\{u_i, i \in \mathbb{N}\}$). Further, assume $\mathbb{E}[|X|] < \infty$. Then, we get the following two equalities:*

$$\mathbb{E}[X] = \mathbb{E}[\mathbb{E}[X|Y]] \tag{3.1}$$

$$\mathbb{E}[X|U] = \mathbb{E}[\mathbb{E}[X|Y,U]|U] \tag{3.2}$$

Proof. Let $Z = \mathbb{E}[X|Y]$. We obtain Equation (3.1) from

$$\begin{aligned}
\mathbb{E}[\mathbb{E}[X|Y]] &= \sum_i z_i \mathbb{P}[Z = z_i] \\
&= \sum_i \sum_j x_j \mathbb{P}[X = x_j | Y = y_i] \mathbb{P}[Y = y_i] \\
&= \sum_i \sum_j x_j \mathbb{P}[X = x_j, Y = y_i] \qquad (*) \\
&= \sum_j \sum_i x_j \mathbb{P}[X = x_j, Y = y_i] \\
&= \sum_j x_j \mathbb{P}[X = x_j] \\
&= \mathbb{E}[X]
\end{aligned}$$

The summation order in $(*)$ can be changed since $\mathbb{E}[|X|] < \infty$.

It is left to verify (3.2). Let $W = \mathbb{E}[X|Y, U]$. The random variable W takes a value

$$w_{j_1,j_2} = \sum_k x_k \mathbb{P}[X = x_k | Y = y_{j_1}, U = u_{j_2}]$$

with probability $\mathbb{P}[Y = y_{j_1}, U = u_{j_2}]$ where $j_1 \in \mathbb{N}$ and $j_2 \in \mathbb{N}$. Let $Z = \mathbb{E}[W|U]$. The random variable Z takes a value

$$z_i = \mathbb{E}[W|U = u_i]$$

with probability $\mathbb{P}[U = u_i]$ where $i \in \mathbb{N}$. We transform z_i to

$$\begin{aligned}
z_i &= \mathbb{E}[W|U = u_i] \\
&= \sum_{j_1,j_2} w_{j_1,j_2} \mathbb{P}[W = w_{j_1,j_2} | U = u_i] \\
&= \sum_{j_1,j_2} \sum_k x_k \mathbb{P}[X = x_k | Y = y_{j_1}, U = u_{j_2}] \mathbb{P}[Y = y_{j_1}, U = u_{j_2} | U = u_i] \\
&= \sum_{j_1} \sum_k x_k \mathbb{P}[X = x_k | Y = y_{j_1}, U = u_i] \mathbb{P}[Y = y_{j_1} | U = u_i] \\
&= \sum_{j_1} \sum_k x_k \mathbb{P}[X = x_k, Y = y_{j_1}, U = u_i] / \mathbb{P}[U = u_i] \qquad (**) \\
&= \sum_k \sum_{j_1} x_k \mathbb{P}[X = x_k, Y = y_{j_1}, U = u_i] / \mathbb{P}[U = u_i] \\
&= \sum_k x_k \mathbb{P}[X = x_k, U = u_i] / \mathbb{P}[U = u_i] \\
&= \sum_k x_k \mathbb{P}[X = x_k | U = u_i] \\
&= \mathbb{E}[X|U = u_i]
\end{aligned}$$

The summation order in $(**)$ can be changed since $\mathbb{E}[|X|] < \infty$. So we obtain

$$\mathbb{E}[\mathbb{E}[X|Y, U]|U = u_i] = \mathbb{E}[X|U = u_i]$$

for all $i \in \mathbb{N}$, i.e. $\mathbb{E}[\mathbb{E}[X|Y,U]|U] = \mathbb{E}[X|U]$. □

Definition 3.1.3. A stochastic process $\{Z_n, n \in \mathbb{N}\}$ is called a *martingale* if

$$\mathbb{E}[|Z_n|] < \infty \qquad \forall n \in \mathbb{N}$$

and

$$\mathbb{E}[Z_{n+1}|Z_1, Z_2, \ldots, Z_n] = Z_n. \tag{3.3}$$

Remark 3.1.4. Taking expectations of (3.3) with Equation (3.1) gives

$$\mathbb{E}[Z_{n+1}] = \mathbb{E}[Z_n].$$

The results of Lemma (3.1.5) and Theorem (3.1.6) are already stated in [13]. Again, the following proofs are more detailed.

Lemma 3.1.5. *Let $\{Z_n, n \in \mathbb{N}\}$ be a discrete stochastic process with $\mathbb{E}[|Z_n|] < \infty$. Let $\boldsymbol{Y}$ be a vector of discrete random variables. If*

$$\mathbb{E}[Z_{n+1}|Z_1, \ldots, Z_n, \boldsymbol{Y}] = Z_n$$

then $\{Z_n\}$ is a martingale.

Proof. It holds $\mathbb{E}[Z_n|Z_1, \ldots, Z_n] = Z_n$ since $\mathbb{E}[Z_n|Z_1 = z_1, \ldots, Z_n = z_n] = z_n$. With that property and with Equation (3.2), we get

$$\begin{aligned} \mathbb{E}[Z_{n+1}|Z_1, \ldots, Z_n] &= \mathbb{E}[\mathbb{E}[Z_{n+1}|Z_1, \ldots, Z_n, \mathbf{Y}]|Z_1, \ldots, Z_n] \\ &= \mathbb{E}[Z_n|Z_1, \ldots, Z_n] \\ &= Z_n. \end{aligned}$$

□

Theorem 3.1.6. *Let $X, Y_1, Y_2, \ldots$ be discrete random variables such that $\mathbb{E}[|X|] < \infty$ and let*

$$Z_n = \mathbb{E}[X|Y_1, \ldots Y_n]$$

for all $n \in \mathbb{N}$. Then $\{Z_n, n \in \mathbb{N}\}$ is a martingale.

Proof. With Equation (3.1), we get $\mathbb{E}[|Z_n|] = \mathbb{E}[|\mathbb{E}[X|Y_1, \ldots, Y_n]|] \leq \mathbb{E}[\mathbb{E}[|X||Y_1, \ldots, Y_n]] = \mathbb{E}[|X|] < \infty$. To check the second condition for a martingale, it is, by Lemma (3.1.5), sufficient to show that $\mathbb{E}[Z_{n+1}|Z_1, \ldots Z_n, Y_1, \ldots, Y_n] = Z_n$. We have

$$\begin{aligned} \mathbb{E}[Z_{n+1}|Z_1, \ldots Z_n, Y_1, \ldots, Y_n] &= \mathbb{E}[Z_{n+1}|Y_1, \ldots, Y_n] \\ &= \mathbb{E}[\mathbb{E}[X|Y_1, \ldots, Y_{n+1}]|Y_1, \ldots, Y_n] \\ &= \mathbb{E}[X|Y_1, \ldots, Y_n] \qquad \text{(from (3.2))} \\ &= Z_n \end{aligned}$$

which proves the theorem. □

3.1.1 The Azuma inequality

Let $\{Z_i, i \in \mathbb{N}\}$ be a martingale. If the random varialbes Z_i do not change too fast over time, Azuma's inequality gives us some bounds on their probabilities.

The following theorem, the Azuma inequality, is stated in [13] with a detailed proof.

Theorem 3.1.7 (Azuma's Inequality)**.** *Let $\{Z_i, i \in \mathbb{N}\}$ be a martingale with $\mathbb{E}[Z_i] = \mu$. Let $Z_0 = \mu$ and suppose that for nonnegative constants α_j, β_j, $j \geq 1$,*

$$-\alpha_j \leq Z_j - Z_{j-1} \leq \beta_j.$$

Then for any $i \geq 0$, $a > 0$:

$$(i) \qquad \mathbb{P}[Z_i - \mu \geq a] \leq \exp\{-\frac{2a^2}{\sum_{j=1}^{i}(\alpha_j + \beta_j)^2}\}$$

$$(ii) \qquad \mathbb{P}[Z_i - \mu \leq -a] \leq \exp\{-\frac{2a^2}{\sum_{j=1}^{i}(\alpha_j + \beta_j)^2}\}$$

The following corollary will be very useful for the next section.

Corollary 3.1.8. *Let $\{Z_i, i \in \mathbb{N}\}$ be a martingale with $\mathbb{E}[Z_i] = \mu$. Let $Z_0 = \mu$ and suppose that for a nonnegative constant $\mathscr{C}$, $j \geq 1$,*

$$|Z_j - Z_{j-1}| \leq \mathscr{C}$$

Then for any $i \in \mathbb{N}$:

$$\mathbb{P}[Z_i \leq 0] \leq \exp\{-\frac{\mu^2}{2i\mathscr{C}^2}\}$$

Proof. Let $\alpha_i = \beta_i = \mathscr{C}$ for all $i \in \mathbb{N}$ and $a = \mu$. Then inequality (ii) in Theorem (3.1.7) establishes the corollary. □

3.2 A martingale process on trees under the uniform model

In this section, we assume that a tree $\mathcal{T} \in RB(n)$ evolved under the uniform model. Consider the following setting:

- Let $h_U : RB(n) \to \mathbb{R}$ with $h_U(\mathcal{T}) = \ln \frac{\mathbb{P}_U[\mathcal{T}]}{\mathbb{P}_Y[\mathcal{T}]} = \ln \frac{\prod_{v \in \mathring{V}_\mathcal{T}} \lambda_v}{c_{n-1}}$.
- For $j \in \{1, \ldots, n\}$, let $Y_j : RB(n) \to RB(j)$ with $Y_j(\mathcal{T}) = \mathcal{T}|_{\{1 \ldots j\}}$.
- For $j > n$, let $Y_j : RB(n) \to RB(n)$ with $Y_j(\mathcal{T}) = \mathcal{T}$.
- Let $Z_i = \mathbb{E}[h_U | Y_1, \ldots Y_i]$.

We have $\mathbb{E}[|h_U(\mathcal{T})|] < \infty$ since $\mathcal{T}$ is chosen from the finite set $RB(n)$ and $\max_{\mathcal{T}\in RB(n)} |h_U(\mathcal{T})| < \infty$. With Theorem (3.1.6), we obtain that $\{Z_i, i \in \mathbb{N}\}$ is a martingale. Note that

$$Z_i = \mathbb{E}[h_U|Y_1, \dots Y_i] = \mathbb{E}[h_U|Y_i].$$

For all $i \geq n$, we have

$$Z_i = \mathbb{E}[h_U(\mathcal{T})|Y_i = \mathcal{T}] = h_U(\mathcal{T}).$$

The expected value μ_U of Z_n is, with Remark (2.3.4),

$$\mu_U = \mathbb{E}[Z_n] = \mathbb{E}[h_U(\mathcal{T})] = d_{KL}(\mathbb{P}_U, \mathbb{P}_Y).$$

Theorem (2.3.12) shows

$$d_{KL}(\mathbb{P}_U, \mathbb{P}_Y) \sim c_U n$$

which means

$$\mu_U \sim c_U n.$$

In the following, we want to apply Azuma's inequality to the tree martingale $\{Z_i, i \in \mathbb{N}\}$. First, set $Z_0 := \mathbb{E}[Z_n] = d_{KL}(\mathbb{P}_U, \mathbb{P}_Y)$. To apply Azuma's inequality, we have to verify $|Z_i - Z_{i-1}| \leq \mathscr{C}_U$ for all $i \in \mathbb{N}$.

- For $i = 1$, note that by definiton, we have

  $$Z_1 = \mathbb{E}[h_U(\mathcal{T})|Y_1] = \mathbb{E}[h_U(\mathcal{T})] = d_{KL}(\mathbb{P}_U, \mathbb{P}_Y) = Z_0$$

  so $|Z_1 - Z_0| = 0$.

- For $i \geq n$, note that $Z_i = \mathbb{E}[h_U(\mathcal{T})|\mathcal{T}] = h_U(\mathcal{T})$. So $|Z_i - Z_{i-1}| = 0$ for all $i > n$.

- Section (3.2.1) will establish $|Z_i - Z_{i-1}| \leq \ln n$ for $2 \leq i \leq n$.

With Corollary (3.1.8), we then have

$$\begin{aligned} \mathbb{P}[Z_n \leq 0] &\leq \exp\{-\frac{\mu_U^2}{2n(\ln n)^2}\} \\ &\sim \exp\{-\frac{c_U^2 n}{2(\ln n)^2}\} \to 0 \qquad \text{for } n \to \infty \end{aligned}$$

Note that $Z_n = h_U(\mathcal{T}) = \ln \frac{\mathbb{P}_U[\mathcal{T}]}{\mathbb{P}_Y[\mathcal{T}]}$. So for a tree $\mathcal{T}$ generated under the uniform model, the probability that $\mathbb{P}_U[\mathcal{T}]$ is smaller than $\mathbb{P}_Y[\mathcal{T}]$ tends to 0 quickly with n as the number of leaves tends to ∞. Therefore the Bayes factor $\frac{\mathbb{P}_U[\mathcal{T}]}{\mathbb{P}_Y[\mathcal{T}]}$ is a very good indicator as to whether a 'big' tree evolved under the uniform model or not.

3.2.1 Calculating a bound in the Azuma inequality

Let $\{Z_i, i \in \mathbb{N}\}$ be the tree martingale introduced above. We can transform Z_i into

$$\begin{aligned} Z_i &= \mathbb{E}[h_U|Y_i] \\ &= \sum_{\mathcal{T}\in RB(n)} h_U(\mathcal{T})\mathbb{P}[\mathcal{T}|Y_i] \\ &= \sum_{\mathcal{T}\in RB(n)} \ln \frac{\prod_{v\in\mathring{V}_\mathcal{T}} \lambda_v}{c_{n-1}} \mathbb{P}[\mathcal{T}|Y_i] \\ &= \sum_{\mathcal{T}\in RB(n)} \left(\sum_{v\in\mathring{V}_\mathcal{T}} \ln \lambda_v - \ln c_{n-1} \right) \mathbb{P}[\mathcal{T}|Y_i] \\ &= \left[\sum_{\mathcal{T}\in RB(n)} \left(\sum_{v\in\mathring{V}_\mathcal{T}} \ln \lambda_v \right) \mathbb{P}[\mathcal{T}|Y_i] \right] - \ln c_{n-1} \end{aligned}$$

The random variable Z_i therefore takes values

$$z_{i,t} = \left[\sum_{\mathcal{T}\in RB(n)} \left(\sum_{v\in\mathring{V}_\mathcal{T}} \ln \lambda_v \right) \mathbb{P}[\mathcal{T}|Y_i = t] \right] - \ln c_{n-1}$$

for all $t \in RB(i)$.

Assuming that $\mathcal{T}$ was generated under the uniform model, i.e.

$$\mathbb{P}[\mathcal{T}|Y_i = t] = \frac{\mathbb{P}[\mathcal{T}]}{\mathbb{P}[t]} = \frac{|RB(i)|}{|RB(n)|}$$

we get, for $t \in RB(i)$,

$$\begin{aligned} z_{i,t} &= \left[\sum_{\substack{\mathcal{T}\in RB(n) \\ \mathcal{T}|_{\{1,\ldots,i\}}=t}} \left(\sum_{v\in\mathring{V}_\mathcal{T}} \ln \lambda_v \right) \frac{|RB(i)|}{|RB(n)|} \right] - \ln c_{n-1} \\ &= \frac{|RB(i)|}{|RB(n)|} \left[\sum_{\substack{\mathcal{T}\in RB(n) \\ \mathcal{T}|_{\{1,\ldots,i\}}=t}} \sum_{v\in\mathring{V}_\mathcal{T}} \ln \lambda_v \right] - \ln c_{n-1}. \end{aligned}$$

Let $\mathcal{T}$ be a binary phylogenetic tree. For the subtree $\mathcal{T}|_{\{1,\ldots,i\}}$, we will write $\mathcal{T}(i)$. The set of all binary phylogenetic trees with leave set $\{1,\ldots,i-1,i+1,\ldots n\}$ shall be $RB(n,i)$. In the following, we will calculate an upper bound for $|Z_i - Z_{i-1}|$. Note that

$$|Z_i - Z_{i-1}| = \max_{t\in RB(i)} |z_{i,t} - z_{(i-1),t(i-1)}|.$$

The difference $|z_{i,t} - z_{(i-1),t(i-1)}|$ is

$$
\begin{aligned}
\Delta_{i,t} &= |z_{i,t} - z_{(i-1),t(i-1)}| \\
&= \left| \frac{|RB(i)|}{|RB(n)|} \sum_{\substack{\mathcal{T}\in RB(n)\\ \mathcal{T}(i)=t}} \sum_{v\in \mathring{V}_{\mathcal{T}}} \ln \lambda_v - \frac{|RB(i-1)|}{|RB(n)|} \sum_{\substack{\mathcal{T}\in RB(n)\\ \mathcal{T}(i-1)=t(i-1)}} \sum_{v\in \mathring{V}_{\mathcal{T}}} \ln \lambda_v \right| \\
&= \frac{|RB(i-1)|}{|RB(n)|} \left| (2i-3) \sum_{\substack{\mathcal{T}\in RB(n)\\ \mathcal{T}(i)=t}} \sum_{v\in \mathring{V}_{\mathcal{T}}} \ln \lambda_v - \sum_{\substack{t'\in RB(i)\\ t'(i-1)=t(i-1)}} \sum_{\substack{\mathcal{T}\in RB(n)\\ \mathcal{T}(i)=t'}} \sum_{v\in \mathring{V}_{\mathcal{T}}} \ln \lambda_v \right| \\
&= \frac{|RB(i-1)|}{|RB(n)|} \left| \sum_{\substack{t'\in RB(i)\\ t'(i-1)=t(i-1)}} \sum_{\substack{\mathcal{T}\in RB(n)\\ \mathcal{T}(i)=t}} \sum_{v\in \mathring{V}_{\mathcal{T}}} \ln \lambda_v - \sum_{\substack{t'\in RB(i)\\ t'(i-1)=t(i-1)}} \sum_{\substack{\mathcal{T}\in RB(n)\\ \mathcal{T}(i)=t'}} \sum_{v\in \mathring{V}_{\mathcal{T}}} \ln \lambda_v \right| \\
&= \frac{|RB(i-1)|}{|RB(n)|} \left| \sum_{\substack{t'\in RB(i)\\ t'(i-1)=t(i-1)}} \left(\sum_{\substack{\mathcal{T}\in RB(n)\\ \mathcal{T}(i)=t}} \sum_{v\in \mathring{V}_{\mathcal{T}}} \ln \lambda_v - \sum_{\substack{\mathcal{T}\in RB(n)\\ \mathcal{T}(i)=t'}} \sum_{v\in \mathring{V}_{\mathcal{T}}} \ln \lambda_v \right) \right| \\
&= \frac{|RB(i-1)|}{|RB(n)|} \left| \sum_{\substack{t'\in RB(i)\\ t'(i-1)=t(i-1)}} \sum_{\substack{\mathcal{T}'\in RB(n,i)\\ \mathcal{T}'(i-1)=t(i-1)}} \left(\sum_{\substack{\mathcal{T}\in RB(n)\\ \mathcal{T}\setminus i=\mathcal{T}'\\ \mathcal{T}(i)=t}} \sum_{v\in \mathring{V}_{\mathcal{T}}} \ln \lambda_v - \sum_{\substack{\mathcal{T}\in RB(n)\\ \mathcal{T}\setminus i=\mathcal{T}'\\ \mathcal{T}(i)=t'}} \sum_{v\in \mathring{V}_{\mathcal{T}}} \ln \lambda_v \right) \right| \\
&\le \frac{|RB(i-1)|}{|RB(n)|} \sum_{\substack{t'\in RB(i)\\ t'(i-1)=t(i-1)}} \sum_{\substack{\mathcal{T}'\in RB(n,i)\\ \mathcal{T}'(i-1)=t(i-1)}} \left| \sum_{\substack{\mathcal{T}\in RB(n)\\ \mathcal{T}\setminus i=\mathcal{T}'\\ \mathcal{T}(i)=t}} \sum_{v\in \mathring{V}_{\mathcal{T}}} \ln \lambda_v - \sum_{\substack{\mathcal{T}\in RB(n)\\ \mathcal{T}\setminus i=\mathcal{T}'\\ \mathcal{T}(i)=t'}} \sum_{v\in \mathring{V}_{\mathcal{T}}} \ln \lambda_v \right|
\end{aligned}
$$

Define

$$
s := \left| \sum_{\substack{\mathcal{T}\in RB(n)\\ \mathcal{T}\setminus i=\mathcal{T}'\\ \mathcal{T}(i)=t}} \sum_{v\in \mathring{V}_{\mathcal{T}}} \ln \lambda_v - \sum_{\substack{\mathcal{T}\in RB(n)\\ \mathcal{T}\setminus i=\mathcal{T}'\\ \mathcal{T}(i)=t'}} \sum_{v\in \mathring{V}_{\mathcal{T}}} \ln \lambda_v \right| .
$$

Consider the tree $\mathcal{T}$ in Fig. 3.1. Moving leaf i to a new position will change λ_v of a vertex v, if v is on the path P from v_i to v_i'. The change of λ_v, when $v <_{\mathcal{T}} v_i$, is $\lambda_v^{new} = \lambda_v - 1$. For the other vertices on that path, we have $\lambda_v^{new} = \lambda_v + 1$. So we

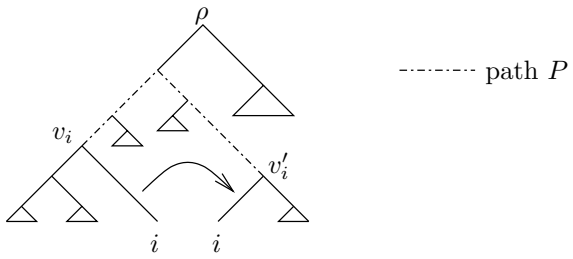


Figure 3.1: Tree $\mathcal{T}$ where leaf i is moved

get, with the property $\ln x - \ln y = \ln x/y$,

$$
\begin{aligned}
s &= \left| \sum_{\substack{\mathcal{T}\in RB(n)\\ \mathcal{T}\setminus i=\mathcal{T}'\\ \mathcal{T}(i)=t}} \left(\sum_{\substack{v\in \mathring{V}_{\mathcal{T}}\setminus v_i\\ v\in P\\ v<_{\mathcal{T}} v_i}} \left(\ln \frac{\lambda_v}{\lambda_v - 1}\right) + \sum_{\substack{v\in \mathring{V}_{\mathcal{T}}\setminus v_i\\ v\in P\\ v<_{\mathcal{T}} v_i'}} \left(\ln \frac{\lambda_v}{\lambda_v + 1}\right) + \ln \lambda_{v_i} - \ln \lambda'_{v_i} \right) \right| \\
&\leq \sum_{\substack{\mathcal{T}\in RB(n)\\ \mathcal{T}\setminus i=\mathcal{T}'\\ \mathcal{T}(i)=t}} \left| \sum_{\substack{v\in \mathring{V}_{\mathcal{T}}\setminus v_i\\ v\in P\\ v<_{\mathcal{T}} v_i}} \left(\ln \frac{\lambda_v}{\lambda_v - 1}\right) + \sum_{\substack{v\in \mathring{V}_{\mathcal{T}}\setminus v_i\\ v\in P\\ v<_{\mathcal{T}} v_i'}} \left(\ln \frac{\lambda_v}{\lambda_v + 1}\right) + \ln \lambda_{v_i} - \ln \lambda'_{v_i} \right| \\
&= \sum_{\substack{\mathcal{T}\in RB(n)\\ \mathcal{T}\setminus i=\mathcal{T}'\\ \mathcal{T}(i)=t}} \left| \sum_{\substack{v\in \mathring{V}_{\mathcal{T}}\setminus v_i\\ v\in P\\ v<_{\mathcal{T}} v_i}} \left(\ln \frac{\lambda_v}{\lambda_v - 1}\right) + \sum_{\substack{v\in \mathring{V}_{\mathcal{T}}\setminus v_i\\ v\in P\\ v<_{\mathcal{T}} v_i'}} \left(\ln \frac{\lambda_v}{\lambda_v + 1}\right) + s' \right|
\end{aligned}
$$

with

$$
s' = \begin{cases} \sum_{i=\lambda'_{v_i}+1}^{\lambda_{v_i}} \ln \frac{i}{i-1} & \text{if } \lambda'_{v_i} \leq \lambda_{v_i} \\ \sum_{i=\lambda_{v_i}+1}^{\lambda'_{v_i}} \ln \frac{i-1}{i} & \text{if } \lambda_{v_i} < \lambda'_{v_i} \end{cases}
$$

Note that for any $v, w \in P$ with $v, w <_{\mathcal{T}} v_i$ or $v, w <_{\mathcal{T}} v_i'$, we have $\lambda_v \neq \lambda_w$. That yields to

$$
s \leq \sum_{\substack{\mathcal{T}\in RB(n)\\ \mathcal{T}\setminus i=\mathcal{T}'\\ \mathcal{T}(i)=t}} \sum_{k=1}^{n-1} \ln \frac{k+1}{k}
$$

Overall, we get, with using the property $\ln(1+x) < x$ for $x > 0$,

$$\begin{aligned}
|z_{i,t} - z_{(i-1),t(i-1)}| &\leq \frac{|RB(i-1)|}{|RB(n)|} \sum_{\substack{t' \in RB(i) \\ t'(i-1)=t(i-1)}} \sum_{\substack{\mathcal{T}' \in RB(n,i) \\ \mathcal{T}'(i-1)=t(i-1)}} \sum_{\substack{\mathcal{T} \in RB(n) \\ \mathcal{T} \setminus i = \mathcal{T}' \\ \mathcal{T}(i)=t}} \sum_{k=1}^{n-1} \ln \frac{k+1}{k} \\
&= \frac{|RB(i)|}{|RB(n)|} \sum_{\substack{\mathcal{T}' \in RB(n,i) \\ \mathcal{T}'(i-1)=t(i-1)}} \sum_{\substack{\mathcal{T} \in RB(n) \\ \mathcal{T} \setminus i = \mathcal{T}' \\ \mathcal{T}(i)=t}} \sum_{k=1}^{n-1} \ln \left(1 + \frac{1}{k}\right) \\
&= \frac{|RB(i)|}{|RB(n)|} \sum_{\substack{\mathcal{T} \in RB(n) \\ \mathcal{T}(i)=t}} \sum_{k=1}^{n-1} \ln \left(1 + \frac{1}{k}\right) \\
&= \sum_{k=1}^{n-1} \ln \left(1 + \frac{1}{k}\right) \\
&< \sum_{k=1}^{n-1} \frac{1}{k} \\
&< \int_1^n \frac{1}{x} dx \\
&= \ln n.
\end{aligned}$$

Therefore,

$$|Z_i - Z_{i-1}| = \max_{t \in RB(i)} |z_{i,t} - z_{(i-1),t(i-1)}| \leq \ln n.$$

3.3 A martingale process on trees under the Yule model

In this section, we assume that a tree $\mathcal{T}$ evolved under the Yule model. Consider the following setting:

- Let $h_Y(\mathcal{T}) = -h_U(\mathcal{T}) = \ln \frac{\mathbb{P}_Y[\mathcal{T}]}{\mathbb{P}_U[\mathcal{T}]}$.
- For $j \in \{1, \ldots, n\}$, let $Y_j : RB(n) \to RB(j)$ with $Y_j(\mathcal{T}) = \mathcal{T}|_{\{1 \ldots j\}}$.
- For $j > n$, let $Y_j : RB(n) \to RB(n)$ with $Y_j(\mathcal{T}) = \mathcal{T}$.
- Let $\tilde{Z}_i = \mathbb{E}[h_Y | Y_1, \ldots Y_i]$.

Since $h_Y = -h_U$, the process $\{\tilde{Z}_i, i \in \mathbb{N}\}$ is a martingale with the same argumentation as in Section 3.2. Further, from Section 3.2, we get

$$\tilde{Z}_i = -\left[\sum_{\mathcal{T} \in RB(n)} \left(\sum_{v \in \mathring{V}_\mathcal{T}} \ln \lambda_v\right) \mathbb{P}[\mathcal{T}|Y_i]\right] + \ln c_{n-1}$$

and

$$\tilde{z}_{i,t} = -\left[\sum_{\mathcal{T}\in RB(n)}\left(\sum_{v\in\mathring{V}_{\mathcal{T}}}\ln\lambda_v\right)\mathbb{P}[\mathcal{T}|Y_i=t]\right]+\ln c_{n-1}$$

for all $t \in RB(i)$.

3.4 Hypothesis testing: Did $\mathcal{T}$ evolve under the Yule model?

In this section, the hypothesis that a given tree $\mathcal{T}$ evolved under the Yule model is tested against the uniform model.

In [10], a test between the Yule and the uniform model is developed by counting cherries. It is shown that the number of cherries in a tree is normally distributed with different expected values for the two models. The power of the test stated in [10] is above 0.90 for trees with more than 80 leaves. The power is only stated as an asymptotic result though.

We will give an analytic result for the power of the log-likelihood-ratio test for the Yule model against the uniform model.

First, we recall the basics about hypothesis testing. In a hypothesis test, we test for a given dataset x if a hypothesis H_0 is rejected in favor of a hypothesis H_1 or if H_0 is accepted. The hypothesis test is characterized by a decision rule, it decides if H_0 is accepted.

The Type I error of a hypothesis test is

$$\alpha = \mathbb{P}[H_0 \textit{ rejected } | H_0 \textit{ true}].$$

The Type II error of a hypothesis test is

$$\beta = \mathbb{P}[H_0 \textit{ retained } | H_1 \textit{ true}].$$

The power of the test is $1-\beta$.

The next Lemma, the Neyman-Pearson Lemma (see [13]), states that for a given Type I error, the likelihood-ratio test is the test with the smallest Type II error.

Lemma 3.4.1 (Neyman-Pearson Lemma)**.** *When performing a hypothesis test between two point hypotheses H_0 and H_1, then the likelihood-ratio test which rejects H_0 in favor of H_1 when*

$$\frac{\mathbb{P}[x|H_0 \textit{ true}]}{\mathbb{P}[x|H_1 \textit{ true}]} \leq k$$

with k being some positive constant, is the most powerful test of size α, where $\alpha = \mathbb{P}[\frac{\mathbb{P}[x|H_0 \textit{ true}]}{\mathbb{P}[x|H_1 \textit{ true}]} \leq k | H_0 \textit{ true}] = \mathbb{P}[H_0 \textit{ rejected} | H_0 \textit{ true}]$ as defined above.

Note that the log-likelihood-ratio test, i.e. rejecting H_0 if

$$\ln \frac{\mathbb{P}[x|H_0 \; true]}{\mathbb{P}[x|H_1 \; true]} \leq \ln k$$

is equivalent to the likelihood-ratio test. We will test the Yule model against the uniform model with the log-likelihood-ratio test to get the smallest Type II error.

Let H_0 and H_1 be the following hypotheses.

H_0: $\mathcal{T}$ evolved under the Yule model
H_1: $\mathcal{T}$ evolved under the uniform model

The decision rule for this test shall be:

- $\tilde{Z}_n = \ln \frac{\mathbb{P}_Y[\mathcal{T}]}{\mathbb{P}_U[\mathcal{T}]} > 0 \Rightarrow$ accept H_0.
- $\tilde{Z}_n = \ln \frac{\mathbb{P}_Y[\mathcal{T}]}{\mathbb{P}_U[\mathcal{T}]} \leq 0 \Rightarrow$ reject H_0.

The Type I and Type II error can be obtained with simulations, i.e. construct a lot of trees with n leaves under the Yule model and estimate α and β.

With the results from the previous sections, we can provide an analytical bound for the Type II error.

A bound for the Type II error of this test is, with Corollary (3.1.8) and Theorem (2.3.10),

$$\begin{aligned}
\beta = \mathbb{P}[H_0 \; retained \; | H_1 \; true] &= \mathbb{P}_U[\ln \frac{\mathbb{P}_Y[\mathcal{T}]}{\mathbb{P}_U[\mathcal{T}]} > 0] \\
&= \mathbb{P}_U[\ln \frac{\mathbb{P}_U[\mathcal{T}]}{\mathbb{P}_Y[\mathcal{T}]} < 0] \\
&\leq \exp\{-\frac{\mu_U^2}{2n\mathscr{C}_U^2}\} \\
&\leq \exp\{-\frac{\mu_U^2}{2n(\ln n)^2}\} \\
&= \exp\{-\frac{(nS_n - \ln c_{n-1})^2}{2n(\ln n)^2}\} \qquad (3.4)
\end{aligned}$$

with S_n and c_n as defined in Theorem (2.3.10). Asymptotically, we get, with Theorem (2.3.12),

$$\begin{aligned}
\beta &\sim \exp\{-\frac{(c_U n)^2}{2n(\ln n)^2}\} \\
&\leq \exp\{-\frac{((1.44 - \ln 4)n)^2}{2n(\ln n)^2}\} \\
&\approx \exp\{-0.00144\frac{n}{(\ln n)^2}\}
\end{aligned}$$

So the power of the test, $1 - \beta$, tends to 1 as n tends to ∞.

With the current bound, the power of the test, calculated by Equation (3.4), is bigger than 0.85 only for trees with more than 600 leaves. It is probably possible to improve the bound for the Azuma inequality though. If the current bound, $\ln n$, could be improved to $1/4 \ln n$, the power of the test would be bigger than 0.90 for trees with more than 50 leaves. A bound of $1/2 \ln n$ would result in a power bigger than 0.90 for trees with more than 170 leaves.

Chapter 4

The Rank Function

Consider the primate tree in Appendix C. Was speciation event with label 76 more likely to be an early event in the tree or a late event? What is the probability that 76 was the 6th speciation event? Was it more likely that speciation event 76 happened before speciation event 162 or 162 before 76? This chapter will provide an answer to those questions, under the assumption that each rank function is equally likely, which is, in particular, the case under the Yule model.

The algorithms RankProb, Compare and an algorithm for obtaining the expected rank and variance for a vertex were implemented in Python. The code is attached in Appendix B. This is joint work with Daniel Ford from Stanford University.

In Section 4.3, we will show how to estimate edge lengths in a tree by calculating the probability distribution of the rank of a vertex. This question was posed by Arne Mooers and Rutger Vos, who constructed the primate supertree and wanted to estimate the edge lengths for it (see [16]).

4.1 Probability distribution of the rank of a vertex

Let $\mathcal{T}$ be a binary phylogenetic tree. Specifying an order for the speciation events (i.e. the interior nodes) in $\mathcal{T}$ is equivalent to introducing a rank function on $\mathcal{T}$. In this chapter, we are interested in the distribution of the possible ranks for a certain vertex, i.e. we want to know the probability of $r(v) = i$ for a given $v \in \mathring{V}$. In other words, we want to calculate $\mathbb{P}[r(v) = i|\mathcal{T}]$, with $r \in r(\mathcal{T})$, $r(\mathcal{T})$ is the set of possible rank functions on the tree $\mathcal{T}$. If every rank function on a given tree is equally likely, we have

$$\mathbb{P}[r(v) = i|\mathcal{T}] = \frac{|\{r : r(v) = i, r \in r(\mathcal{T})\}|}{|r(\mathcal{T})|} \tag{4.1}$$

A formula for the denominator is given in Lemma (1.3.8). The enumerator will be calculated in polynomial time by algorithm RankCount.

Examples of stochastic models on phylogenetic trees where each rank function is equally likely:

- For the Yule model, we have seen in Theorem (2.2.4), that $\mathbb{P}[r|\mathcal{T}]$ is the uniform distribution.
- As we have seen in Remark (2.2.8), the coalescent model has the same probability distribution on rooted binary ranked trees as the Yule model. So $\mathbb{P}[r|\mathcal{T}]$ is the uniform distribution.
- In the uniform model no rank function is induced when a tree is generated. We can assume though that for a given tree $\mathcal{T}$, each rank function is equally likely. Then, Equation (4.1) holds as well.

Definition 4.1.1. Let $\mathcal{T}$ be a rooted phylogenetic tree. Define

$$\alpha_{\mathcal{T},v}(i) := |\{r : r(v) = i,\ r \in r(\mathcal{T})\}|$$

for $v \in \mathring{V}, i \in 1, \ldots, |\mathring{V}|$. In other words, $\alpha_{\mathcal{T},v}(i)$ denotes the number of rank functions r for $\mathcal{T}$ in which v comes in the i-th position.

The following results will be needed in the next sections.

Lemma 4.1.2. *Let*

$$\begin{aligned} x^1 &= \{x_1^1, x_2^1 \ldots x_{n_1}^1\} \\ x^2 &= \{x_1^2, x_2^2 \ldots x_{n2}^2\} \\ &\vdots \\ x^d &= \{x_1^d, x_2^d \ldots x_{n_d}^d\} \end{aligned}$$

be d disjoint sets with the linear order $x_1^i < x_2^i < \ldots < x_{n_i}^i$ for each $i \in \{1, \ldots, d\}$. The number $\mathscr{L}$ of possible linear orders on the set $x^1 \cup x^2 \cup \ldots \cup x^d$, with the linear order of each original set x^i being preserved, is

$$\mathscr{L} = \frac{\left(\sum_{i=1}^{d} n_i\right)!}{\prod_{i=1}^{d} n_i!}$$

Proof. The number $\tilde{\mathscr{L}}$ of linear orders of the $\sum_{i=1}^{d} n_i$ elements of $x^1 \cup x^2 \cup \ldots \cup x^d$, allowing any order on x^i, is $\tilde{\mathscr{L}} = \left(\sum_{i=1}^{d} n_i\right)!$. The number $\tilde{\mathscr{L}}_i$ of linear orders of the n_i elements of x^i is $(n_i)!$. Since for $\mathscr{L}$, we only allow the linear order $x_1^i < x_2^i < \ldots < x_{n_i}^i$ on x^i, it holds

$$\mathscr{L} = \frac{\tilde{\mathscr{L}}}{\prod_{i=1}^{d} \tilde{\mathscr{L}}_i} = \frac{\left(\sum_{i=1}^{d} n_i\right)!}{\prod_{i=1}^{d} n_i!}$$

□

Corollary 4.1.3. *For $d = 2$ in Lemma (4.1.2), we have*

$$\mathscr{L} = \binom{n_1 + n_2}{n_1}$$

possible linear orders on $x^1 \cup x^2$, preserving the linear order on x^1 and x^2.

Proof. From Lemma (4.1.2) follows

$$\mathscr{L} = \frac{\left(\sum_{i=1}^{2} n_i\right)!}{\prod_{i=1}^{2} n_i!} = \frac{(n_1 + n_2)!}{(n_1)!(n_2)!} = \binom{n_1 + n_2}{n_1}$$

□

Remark 4.1.4. The values $\binom{n}{k}$ for all $n, k \leq N$ $(n, k, N \in \mathbb{N})$ can be calculated in $O(N^2)$, *cf.* Pascal's Triangle. In Appendix B, a dynamic programming version for calculating $\binom{n}{k}$ is implemented. Thus, after $O(N^2)$ calculations, any value $\binom{n}{k}$ with $n, k \leq N$ can be obtained in constant time in an algorithm.

4.1.1 Polynomial-time algorithms

In the following, we give a polynomial algorithm to determine $\alpha_{\mathcal{T},v}(i)$ for $v \in \mathring{V}$ and $i = 1, \ldots, |\mathring{V}|$ in a binary phylogenetic tree $\mathcal{T}$.

Algorithm: RankCount$(\mathcal{T}, v)$
Input: A rooted binary phylogenetic tree $\mathcal{T}$ and an interior vertex v.
Output: The values of $\alpha_{\mathcal{T},v}(i)$ for $i = 1, \ldots, |\mathring{V}|$.

1: Denote the vertices of the path from v to root ρ with $(v = x_1, x_2, \ldots, x_n = \rho)$.
2: Denote the subtree of $\mathcal{T}$, consisting of root x_m and all its descendants, by $\mathcal{T}_m$ for $m = 1, \ldots, n$. (*cf.* Figure 4.1).
3: **for** $m = 1, \ldots, n$ **do**
4: **for** $i = 1, \ldots, |\mathring{V}_{\mathcal{T}}|$ **do**
5: $\alpha_{\mathcal{T}_m,v}(i) := 0$
6: **end for**
7: **end for**
8: $\alpha_{\mathcal{T}_1,v}(1) := \dfrac{|\mathring{V}_{\mathcal{T}_1}|!}{\prod_{v \in \mathring{V}_{\mathcal{T}_1}} \lambda_v}$
9: **for** $m = 2, \ldots, n$ **do**
10: $\mathcal{T}'_{m-1} := \mathcal{T}_m|_{L_{\mathcal{T}_m} \setminus L_{\mathcal{T}_{m-1}}}$ (*cf.* Figure 4.2)
11: $R_{\mathcal{T}'_{m-1}} := \dfrac{|\mathring{V}_{\mathcal{T}'_{m-1}}|!}{\prod_{v \in \mathring{V}_{\mathcal{T}'_{m-1}}} \lambda_v}$

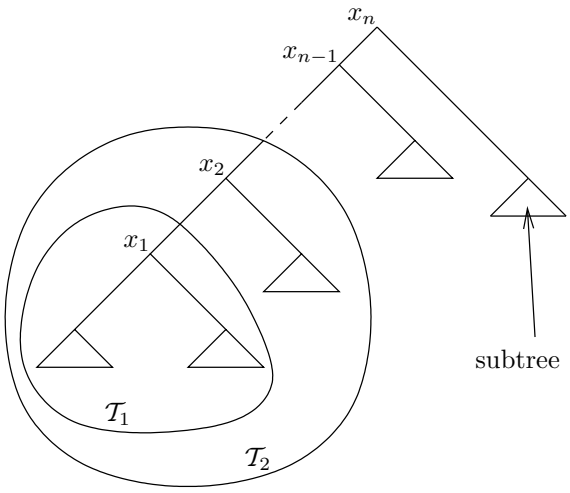


Figure 4.1: Labeling the tree for RankCount

12: **for** $i = m, \ldots, |\mathring{V}_{\mathcal{T}_m}|$ **do**
13: $M := \min\{|\mathring{V}_{\mathcal{T}'_{m-1}}|, i-2\}$
14: $\alpha_{\mathcal{T}_m,v}(i) :=$
$\sum_{j=0}^{M} \alpha_{\mathcal{T}_{m-1},v}(i-j-1) R_{\mathcal{T}'_{m-1}} \binom{|\mathring{V}_{\mathcal{T}_{m-1}}|+|\mathring{V}_{\mathcal{T}'_{m-1}}|-(i-1)}{|\mathring{V}_{\mathcal{T}'_{m-1}}|-j} \binom{i-2}{j}$ $(*)$
15: **end for**
16: **end for**
17: RETURN $\alpha_{\mathcal{T},v} := \alpha_{\mathcal{T}_n,v}$

Theorem 4.1.5. RankCount *returns the quantities*

$$\alpha_{\mathcal{T},v}(i) = |\{r : r(v) = i, r \in r(\mathcal{T})\}|$$

for each given $v \in \mathring{V}$ *and all* $i \in 1, \ldots, |\mathring{V}|$.

Proof. We have to show that all the $\alpha_{\mathcal{T}_m,v}(i)$ produced by RankCount equal the $\alpha_{\mathcal{T}_m,v}(i)$ defined in (4.1.1). In the following, we denote the values $\alpha_{\mathcal{T}_m,v}(i)$ produced by the algorithm with $\alpha^{Alg}_{\mathcal{T}_m,v}(i)$ and $\alpha_{\mathcal{T}_m,v}(i)$ shall denote the number of rank functions with $r(v) = i$ as defined in (4.1.1). We will show $\alpha_{\mathcal{T}_m,v}(i) = \alpha^{Alg}_{\mathcal{T}_m,v}(i)$ for $m = 1, \ldots, n,\ i = 1, \ldots, |\mathring{V}_{\mathcal{T}}|$. This is done by induction over m.
For $m = 1$, $\alpha_{\mathcal{T}_1,v}(1) = \alpha^{Alg}_{\mathcal{T}_1,v}(1)$ since (1.3.8) holds. Vertex v is the root of $\mathcal{T}_1$, so $\alpha_{\mathcal{T}_1,v}(i) = 0$ for all $i > 1$.
Let $m = k$ and $\alpha_{\mathcal{T}_m,v}(i) = \alpha^{Alg}_{\mathcal{T}_m,v}(i)$ holds for all $m < k$. $\alpha_{\mathcal{T}_k,v}(i) = 0$ clearly holds for all $i > |\mathring{V}_{\mathcal{T}_k}|$ since $r_{\mathcal{T}_k} : v \to \{1, \ldots, |\mathring{V}_{\mathcal{T}_k}|\}$. So it is left to verify that the term $(*)$ returns the right values for $\alpha_{\mathcal{T}_k,v}(i)$. Assume that the vertex v is in the $(i-j-1)$-th position in $\mathcal{T}_{k-1}$ (with $i-j-1 > 0$) for some rank function $r_{\mathcal{T}_{k-1}}$ and v shall be in the i-th position in $\mathcal{T}_k$. We want to combine the linear order in the tree $\mathcal{T}_{k-1}$ induced by $r_{\mathcal{T}_{k-1}}$ with a linear order in $\mathcal{T}'_{k-1}$ induced by $r_{\mathcal{T}'_{k-1}}$ to get

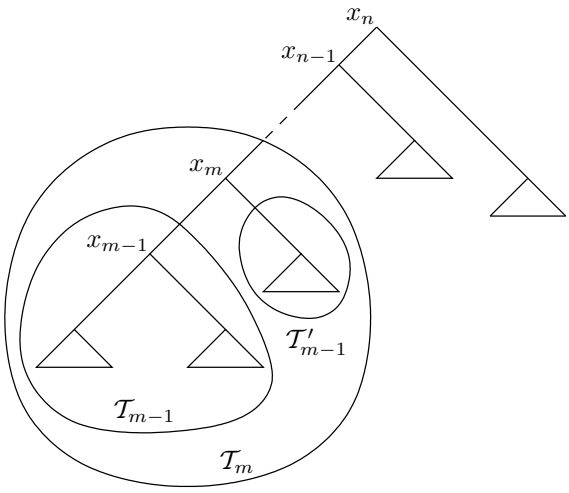


Figure 4.2: Labeling the tree for recursion in RankCount

a linear order on $\mathcal{T}_k$. The first j vertices of $\mathcal{T}'_{k-1}$ must be inserted between vertices of $\mathcal{T}_{k-1}$ with lower rank than v so that v ends up to be in the i-th position of the tree $\mathcal{T}_k$. We will count the number of possibilities to do so. The tree $\mathcal{T}'_{k-1}$ has

$$R_{\mathcal{T}'_{k-1}} = \frac{|\mathring{V}_{\mathcal{T}'_{k-1}}|!}{\prod\limits_{v \in \mathring{V}_{\mathcal{T}'_{k-1}}} \lambda_v}$$

possible rank functions. Combining a rank function $r_{\mathcal{T}_{k-1}}$ with a rank function $r_{\mathcal{T}'_{k-1}}$ for getting a rank function $r_{\mathcal{T}_k}$ with $r_{\mathcal{T}_k}(v) = i$ means inserting the first j vertices of $\mathcal{T}'_{k-1}$ anywhere between the first $(i-j-2)$ vertices of $\mathcal{T}_{k-1}$. There are

$$\binom{(i-j-2)+j}{j} = \binom{i-2}{j}$$

possibilities according to Corollary 4.1.3. For combining the $|\mathring{V}_{\mathcal{T}_{k-1}}| - (i-j-1)$ vertices of rank larger than v in $\mathcal{T}_{k-1}$ with the remaining $|\mathring{V}_{\mathcal{T}'_{k-1}}| - j$ vertices in $\mathcal{T}'_{k-1}$, we have

$$\binom{|\mathring{V}_{\mathcal{T}_{k-1}}| - (i-j-1) + |\mathring{V}_{\mathcal{T}'_{k-1}}| - j}{|\mathring{V}_{\mathcal{T}'_{k-1}}| - j} = \binom{|\mathring{V}_{\mathcal{T}_{k-1}}| + |\mathring{V}_{\mathcal{T}'_{k-1}}| - (i-1)}{|\mathring{V}_{\mathcal{T}'_{k-1}}| - j}$$

possibilities. This follows again from Corollary 4.1.3. The number of rank functions $r_{\mathcal{T}_{k-1}}$ with $r_{\mathcal{T}_{k-1}}(v) = i-j-1$ is $\alpha_{\mathcal{T}_{k-1},v}(i-j-1)$ by the induction assumption. Multiplying all those possibilities gives

$$\alpha_{\mathcal{T}_{k-1},v}(i-j-1) R_{\mathcal{T}'_{k-1}} \binom{|\mathring{V}_{\mathcal{T}_{k-1}}| + |\mathring{V}_{\mathcal{T}'_{k-1}}| - (i-1)}{|\mathring{V}_{\mathcal{T}'_{k-1}}| - j} \binom{i-2}{j}$$

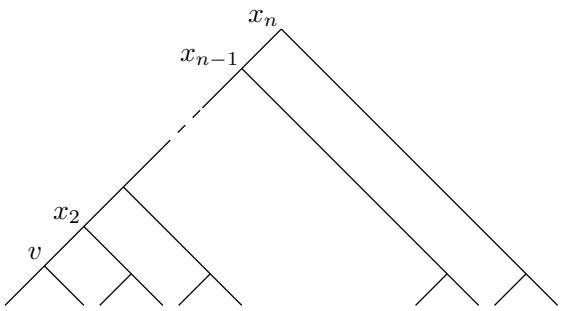


Figure 4.3: Illustration for runtime of RankCount

$\alpha_{\mathcal{T}_k,v}(i)$ is then the sum over all possible j which is equal to the term $(*)$ for $\alpha^{Alg}_{\mathcal{T}_k,v}(i)$. This establishes the theorem. □

Theorem 4.1.6. *The runtime of* RankCount *is* $O(|\mathring{V}|^2)$.

Proof. Note that the number of rank functions $R_{\mathcal{T}} = \frac{|\mathring{V}_{\mathcal{T}}|!}{\prod_{v\in\mathring{V}_{\mathcal{T}}}\lambda_v}$ on a tree $\mathcal{T}$ with $\mathring{V}$ interior vertices can be calculated in $O(|\mathring{V}|)$, i.e. in linear time.

Further, note that the combinatorial factors $\binom{n}{k}$ for all $n, k \leq |\mathring{V}|$ can be calculated in advance in quadratic time, see Remark (4.1.4). In the algorithm, those factors can then be obtained in constant time.

Contributions to the runtime from each line in RankCount (the runtime is always w.r.t. $|\mathring{V}|$):
Line 1–2: linear time
Line 3–7: quadratic time
Line 8: linear time
Line 9–16: quadratic time since:

Line 11: $R_{\mathcal{T}'_{m-1}}$ can be calculated in $O(|\mathring{V}|)$. This has to be done for $m = 1,\dots,n$, so overall the runtime for calculating all $R_{\mathcal{T}'_{m-1}}$ is no more than $O(|\mathring{V}|^2)$ since $n \leq |\mathring{V}|$.

Line 14: We add up all calculations needed for obtaining $\alpha_{\mathcal{T}_m,v}(i)$, $m = 1,\dots,n$, $i = 1,\dots,|\mathring{V}_{\mathcal{T}_m}|$:

$$\sum_{m=2}^{n} |\mathring{V}_{\mathcal{T}_m}||\mathring{V}_{\mathcal{T}'_{m-1}}| \leq \sum_{m=2}^{n} |\mathring{V}||\mathring{V}_{\mathcal{T}'_{m-1}}| = |\mathring{V}| \sum_{m=2}^{n} |\mathring{V}_{\mathcal{T}'_{m-1}}| \leq |\mathring{V}|^2$$

The last inequality holds since the vertices of the $\mathcal{T}'_m$, $m = 1,\dots,n-1$, are distinct. Therefore, line 14 contributes a quadratic runtime.
Line 17: constant time

So overall, the runtime is no more than $O(|\mathring{V}|^2)$. Figure 4.3 shows a tree for which the runtime of RankCount is actually quadratic. Counting all the calcu-

lations for term $(*)$ in the algorithm for the tree in 4.3 yields to

$$\begin{aligned}\sum_{m=2}^{n}\sum_{i=m}^{|\mathring{V}_{\mathcal{T}_m}|}|\mathring{V}_{\mathcal{T}'_{m-1}}|+1 &= \sum_{m=2}^{n}\sum_{i=m}^{|\mathring{V}_{\mathcal{T}_m}|}2\\ &= \sum_{m=2}^{n}2(|\mathring{V}_{\mathcal{T}_m}|-(m-1))\\ &= \sum_{m=2}^{n}2((2m-1)-(m-1))\\ &= \sum_{m=2}^{n}2m\\ &= n(n+1)-2\end{aligned}$$

Since $n=(|\mathring{V}|+1)/2$, we have a quadratic runtime.

□

Corollary 4.1.7. *The probability $\mathbb{P}[r(v)=i|\mathcal{T}]$ can be calculated in $O(|V|^2)$. We have*

$$\mathbb{P}[r(v)=i|\mathcal{T}]=\frac{\alpha_{\mathcal{T},v}(i)}{\sum_{i=1}^{|\mathring{V}|}\alpha_{\mathcal{T},v}(i)}=\frac{\alpha_{\mathcal{T},v}(i)\prod_{v\in\mathring{V}}\lambda_v}{|\mathring{V}|!}. \tag{4.2}$$

Proof. The first equality in (4.2) follows from basic probability theory. The second equality holds since $\frac{|\mathring{V}|!}{\prod_{v\in\mathring{V}}\lambda_v}=\sum_i\alpha_{\mathcal{T},v}(i)$ by (1.3.8). The complexity of the runtime follows from (4.1.6). □

Remark 4.1.8. We will write $\mathbb{P}[r(v)=i]$ instead of $\mathbb{P}[r(v)=i|\mathcal{T}]$ in the following. With $\mathbb{P}[r(v)=i]$ from Corollary (4.1.7), the expected value $\mu_{r(v)}$ and the variance $\sigma^2_{r(v)}$ for $r(v)$ can be calculated by

$$\begin{aligned}\mu_{r(v)} &= \sum_{i=1}^{|\mathring{V}|}i\mathbb{P}[r(v)=i]\\ \sigma^2_{r(v)} &= \sum_{i=1}^{|\mathring{V}|}i^2\mathbb{P}[r(v)=i]-\mu^2_{r(v)}\end{aligned}$$

Example 4.1.9. We will illustrate the algorithm RankCount for the tree in Figure 4.4. We get the following values:

$\alpha_{\mathcal{T}_1,v}(1)=\frac{2!}{2\cdot1}=1$

$\alpha_{\mathcal{T}_2,v}(2)=\alpha_{\mathcal{T}_{m-1},v}(1)1\binom{2+1-1}{1}\binom{0}{0}=2$
$\alpha_{\mathcal{T}_2,v}(3)=\alpha_{\mathcal{T}_{m-1},v}(1)1\binom{2+1-2}{1}\binom{1}{0}=1$
$\alpha_{\mathcal{T}_2,v}(4)=0$

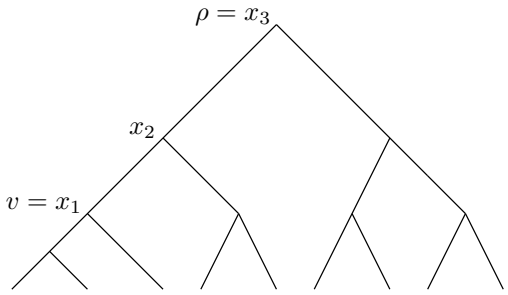


Figure 4.4: Tree to illustrate the algorithm RankCount

$\alpha_{\mathcal{T}_3,v}(3) = \alpha_{\mathcal{T}_{m-1},v}(2)2\binom{4+3-2}{3}\binom{1}{0} + \alpha_{\mathcal{T}_{m-1},v}(1)2\binom{4+3-2}{2}\binom{1}{1} = 40 + 0 = 40$
$\alpha_{\mathcal{T}_3,v}(4) = \alpha_{\mathcal{T}_{m-1},v}(3)2\binom{4+3-3}{3}\binom{2}{0} + \alpha_{\mathcal{T}_{m-1},v}(2)2\binom{4+3-3}{2}\binom{2}{1} = 8 + 48 = 56$
$\alpha_{\mathcal{T}_3,v}(5) = \alpha_{\mathcal{T}_{m-1},v}(3)2\binom{4+3-4}{2}\binom{3}{1} + \alpha_{\mathcal{T}_{m-1},v}(2)2\binom{4+3-4}{1}\binom{3}{2} = 18 + 36 = 54$
$\alpha_{\mathcal{T}_3,v}(6) = \alpha_{\mathcal{T}_{m-1},v}(3)2\binom{4+3-5}{1}\binom{4}{2} + \alpha_{\mathcal{T}_{m-1},v}(2)2\binom{4+3-5}{0}\binom{4}{3} = 24 + 16 = 40$
$\alpha_{\mathcal{T}_3,v}(7) = \alpha_{\mathcal{T}_{m-1},v}(3)2\binom{4+3-6}{0}\binom{5}{3} = 20$
$\alpha_{\mathcal{T}_3,v}(8) = 0$

With $\alpha_{\mathcal{T}_3,v} = \alpha_{\mathcal{T},v}$, we get

$$\begin{aligned}
\mathbb{P}[r(v) = 1] &= 0 \\
\mathbb{P}[r(v) = 2] &= 0 \\
\mathbb{P}[r(v) = 3] &= \frac{40}{40 + 56 + 54 + 40 + 20} = \frac{40}{210} = \frac{20}{105} \\
\mathbb{P}[r(v) = 4] &= \frac{28}{105} \\
\mathbb{P}[r(v) = 5] &= \frac{27}{105} \\
\mathbb{P}[r(v) = 6] &= \frac{20}{105} \\
\mathbb{P}[r(v) = 7] &= \frac{10}{105} \\
\mathbb{P}[r(v) = 8] &= 0
\end{aligned}$$

Therefore, the expected value $\mu_{r(v)}$ is

$$\mu_{r(v)} = \sum_{i=1}^{8} i\mathbb{P}[r(v) = i) = \frac{497}{105} \approx 4.73$$

and the variance $\sigma^2_{r(v)}$ is

$$\sigma^2_{r(v)} = \sum_{i=1}^{8} i^2\mathbb{P}[r(v) = i) - \mu^2_{r(v)} = \frac{2513}{105} - \frac{497^2}{105^2} = \frac{344}{225} \approx 1.53$$

Remark 4.1.10. Note that $\mathbb{P}[r(v) = i] = \frac{\alpha_{\mathcal{T},v}(i)}{\sum_j \alpha_{\mathcal{T},v}(j)}$. Common factors in all $\alpha_{\mathcal{T},v}(i), i = 1, \ldots, |\mathring{V}_{\mathcal{T}_v}|$ will therefore cancel out.

The next algorithm, RankProb, is a modification of RankCount such that common factors of $\alpha_{\mathcal{T},v}(i), i = 1, \ldots, |\mathring{V}_{\mathcal{T}_v}|$, will not be included. Therefore, the numbers we have to deal with in the algorithm stay smaller and the number of calculations is reduced.

Algorithm: RankProb($\mathcal{T}, v$)
Input: A rooted binary phylogenetic tree $\mathcal{T}$ and an interior vertex v.
Output: The probabilities $\mathbb{P}[r(v) = i]$ for $i = 1, \ldots, |\mathring{V}|$.

1: Denote the vertices of the path from v to root ρ with $(v = x_1, x_2, \ldots, x_n = \rho)$.
2: Denote the subtree of $\mathcal{T}$, consisting of root x_m and all its descendants, by $\mathcal{T}_m$ for $m = 1, \ldots, n$. (*cf.* Figure 4.1).
3: **for** $m = 1, \ldots, n$ **do**
4: **for** $i = 1, \ldots, |\mathring{V}_{\mathcal{T}}|$ **do**
5: $\tilde{\alpha}_{\mathcal{T}_m,v}(i) := 0$
6: **end for**
7: **end for**
8: $\tilde{\alpha}_{\mathcal{T}_1,v}(1) := 1$
9: **for** $m = 2, \ldots, n$ **do**
10: $\mathcal{T}'_{m-1} := \mathcal{T}_m|_{L_{\mathcal{T}_m} \setminus L_{\mathcal{T}_{m-1}}}$ (*cf.* Figure 4.2)
11: **for** $i = m, \ldots, |\mathring{V}_{\mathcal{T}_m}|$ **do**
12: $M := \min\{|\mathring{V}_{\mathcal{T}'_{m-1}}|, i-2\}$
13: $\tilde{\alpha}_{\mathcal{T}_m,v}(i) := \sum_{j=0}^{M} \tilde{\alpha}_{\mathcal{T}_{m-1},v}(i-j-1) \binom{|\mathring{V}_{\mathcal{T}_{m-1}}| + |\mathring{V}_{\mathcal{T}'_{m-1}}| - (i-1)}{|\mathring{V}_{\mathcal{T}'_{m-1}}| - j} \binom{i-2}{j}$
14: **end for**
15: **end for**
16: **for** $i = 1, \ldots, |\mathring{V}_{\mathcal{T}}|$ **do**
17: $\mathbb{P}[r(v) = i] := \frac{\tilde{\alpha}_{\mathcal{T}_n,v}(i)}{\sum_j \tilde{\alpha}_{\mathcal{T}_n,v}(j)}$
18: **end for**
19: RETURN $\mathbb{P}[r(v) = i], i = 1, \ldots, |\mathring{V}|$.

Theorem 4.1.11. RankProb *returns the quantities*

$$\mathbb{P}[r(v) = i]$$

for each given $v \in \mathring{V}$ *and all* $i \in 1, \ldots, |\mathring{V}|$. *The runtime is* $O(|\mathring{V}|^2)$.

Proof. Note that the structure of RankProb is the same as the structure of RankCount. The only difference is that common factors to $\alpha_{\mathcal{T}_m,v}(i)$ for all i are not included. Those common factors do not change the probabilities since they cancel out once calculating the probabilities. Therefore, since RankCount works correct, also RankProb works correct.

It is left to verify the runtime. The only time consuming step in RankProb is line 13. This line is of the same complexity as line 14 in RankCount. Line 14 in RankCount contributed a quadratic time. Therefore, the runtime of RankProb is quadratic as well. □

4.1.2 Non-binary trees and ranks

Let $\mathcal{T}$ be a non-binary phylogenetic tree. Assume that any possible rank function on $\mathcal{T}$ is equally likely. With that assumption, we have

$$\mathbb{P}[r(v) = i] = \frac{\alpha_{\mathcal{T},v}(i)}{|r(\mathcal{T})|}.$$

To calculate these probabilities, the algorithm RankProb can be generalized to non-binary trees. We call the generalized algorithm RankProbGen.

Algorithm RankProbGen $(\mathcal{T}, v)$
Input: A rooted phylogenetic tree $\mathcal{T}$ and an interior vertex v.
Output: The probabilities $\mathbb{P}[r(v) = i]$ for $i = 1, \ldots, |\mathring{V}|$.

1: Denote the vertices of the path from v to root ρ with $(v = x_1, x_2, \ldots, x_n = \rho)$.
2: Denote the subtree of $\mathcal{T}$, consisting of root x_m and all its descendants, by $\mathcal{T}_m$ for $m = 1, \ldots, n$.
3: **for** $m = 1, \ldots, n$ **do**
4: **for** $i = 1, \ldots, |\mathring{V}_{\mathcal{T}}|$ **do**
5: $\tilde{\alpha}_{\mathcal{T}_m,v}(i) = 0$
6: **end for**
7: **end for**
8: $\tilde{\alpha}_{\mathcal{T}_1,v}(1) = 1$
9: **for** $m = 2, \ldots, n$ **do**
10: Label the subtree $\mathcal{T}_m \setminus \mathcal{T}_{m-1}$ by $\mathcal{T}'_{m-1}$ (*cf.* Figure 4.5)
11: $M = \min\{|\mathring{V}_{\mathcal{T}'_{m-1}}| - 1, i - 2\}$
12: **for** $i = m, \ldots, |\mathring{V}_{\mathcal{T}_m}|$ **do**
13: $$\tilde{\alpha}_{\mathcal{T}_m,v}(i) := \sum_{j=0}^{M} \tilde{\alpha}_{\mathcal{T}_{m-1},v}(i-j-1) \binom{|\mathring{V}_{\mathcal{T}_{m-1}}| + |\mathring{V}_{\mathcal{T}'_{m-1}}| - 1 - (i-1)}{|\mathring{V}_{\mathcal{T}'_{m-1}}| - 1 - j} \binom{i-2}{j}$$
14: **end for**
15: **end for**
16: **for** $i = 1, \ldots, |\mathring{V}_{\mathcal{T}}|$ **do**
17: $\mathbb{P}[r(v) = i] = \frac{\tilde{\alpha}_{\mathcal{T}_n,v}(i)}{\sum_j \tilde{\alpha}_{\mathcal{T}_n,v}(j)}$
18: **end for**
19: RETURN $\mathbb{P}[r(v) = i], i = 1, \ldots, |\mathring{V}|$.

Theorem 4.1.12. RankProbGen *returns the probabilities*

$$\mathbb{P}[r(v) = i]$$

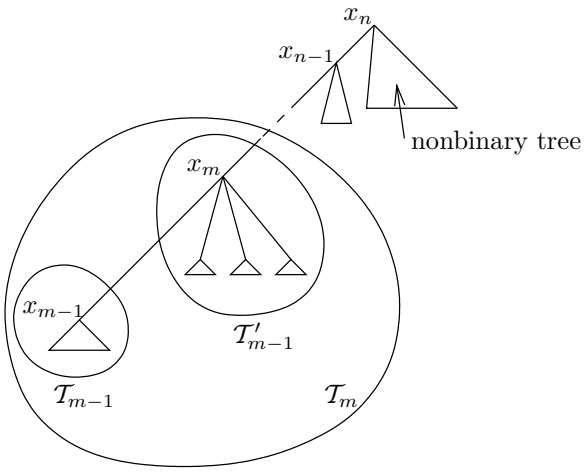


Figure 4.5: Labelling the tree for algorithm RankProbGen.

for each given $v \in \mathring{V}$ *and all* $i \in 1, \ldots, |\mathring{V}|$. *The runtime is* $O(|\mathring{V}|^2)$.

Proof. The algorithm is the same as RankProb. The only difference is that in each step, we define $\mathcal{T}'_{m-1} := \mathcal{T}_m \setminus \mathcal{T}_{m-1}$, i.e. the root of $\mathcal{T}'_m$ is x_m. For any rank function on $\mathcal{T}'_m$, we now insert the first j elements (excluding the root x_m) before the vertex v. The number of ways to insert these vertices is counted analogously to the proof of Theorem (4.1.5). The number of possible rank functions on $\mathcal{T}'_m$ does not have to be calculated, since these factors cancel out when calculating the probabilities.

Since we do the same iterations as in RankProb, the algorithm RankProbGen has quadratic runtime as well. □

4.2 Comparing two interior vertices

Assume again that every rank function on a binary phylogenetic tree $\mathcal{T}$ is equally likely. We want to compare two interior vertices u and v of $\mathcal{T}$. Was u more likely before v or v before u (*cf.* Fig. 4.6)? In other words, we want to know the probability

$$\mathbb{P}_{u<v} := \mathbb{P}[r(u) < r(v)|\mathcal{T}]$$

where $r(T)$ is the set of all possible rank functions on $\mathcal{T}$. This probability is, by Theorem (2.2.4), equivalent to counting all the possible rank functions on $\mathcal{T}$ in which u has lower rank than v and divide that number by all possible rank functions on $\mathcal{T}$. The algorithm Compare will solve this problem in quadratic time.

Algorithm Compare $(\mathcal{T}, u, v)$
Input: A rooted phylogenetic tree $\mathcal{T}$ and two distinct interior vertices u and v.
Output: The probability $\mathbb{P}_{u<v} := \mathbb{P}[r(u) < r(v)|\mathcal{T}]$.

1: Denote the most recent common ancestor of u and v by ρ_1.

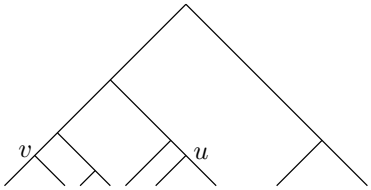


Figure 4.6: What is the probability that vertex u has smaller rank than vertex v?

2: **if** $\rho_1 = v$ **then**
3: RETURN $\mathbb{P}_{u<v} = 0$.
4: **end if**
5: **if** $\rho_1 = u$ **then**
6: RETURN $\mathbb{P}_{u<v} = 1$.
7: **end if**
8: Let $\mathcal{T}_{\rho_1}$ be the subtree of $\mathcal{T}$ which is induced by ρ_1.
9: Delete the vertex ρ_1 from $\mathcal{T}_{\rho_1}$. The two evolving subtrees are labeled $\mathcal{T}_u$ and $\mathcal{T}_v$ with $u \in \mathcal{T}_u$ and $v \in \mathcal{T}_v$.
10: Run RankProb$(\mathcal{T}_u, u)$ and RankProb$(\mathcal{T}_v, v)$ to get $\mathbb{P}[r(u) = i]$ on $\mathcal{T}_u$ and $\mathbb{P}[r(v) = i]$ on $\mathcal{T}_v$ for all possible i.
11: **for** $i = 1, \ldots, |\mathring{V}_{\mathcal{T}_u}|$ **do**
12: $ucum(i) := \sum_{k=1}^{i} \mathbb{P}[r(u) = i]$
13: **end for**
14: $\mathbb{P}_{u<v} := 0$
15: **for** $i = 1, \ldots, |\mathring{V}_{\mathcal{T}_v}|$ **do**
16: **for** $j = 1, \ldots |\mathring{V}_{\mathcal{T}_u}|$ **do**
17: $p := \mathbb{P}[r(v) = i] \cdot \binom{i-1+j}{j} \cdot \binom{|\mathring{V}_{\mathcal{T}_v}|-i+|\mathring{V}_{\mathcal{T}_u}|-j}{|\mathring{V}_{\mathcal{T}_u}|-j} \cdot ucum(j)$ $\quad (*)$
18: $\mathbb{P}_{u<v} := \mathbb{P}_{u<v} + p$
19: **end for**
20: **end for**
21: $tot := \binom{|\mathring{V}_{\mathcal{T}_u}|+|\mathring{V}_{\mathcal{T}_v}|}{|\mathring{V}_{\mathcal{T}_v}|}$
22: $\mathbb{P}_{u<v} := \mathbb{P}_{u<v}/tot$
23: RETURN $\mathbb{P}_{u<v}$

Theorem 4.2.1. *The algorithm* Compare *returns the value*

$$\mathbb{P}_{u<v} = \mathbb{P}[r(u) < r(v)|\mathcal{T}].$$

Proof. Note that the probability of u having smaller rank than v in tree $\mathcal{T}_{\rho_1}$ equals the probability of u having smaller rank than v in tree $\mathcal{T}$, since for any rank function on $\mathcal{T}_{\rho_1}$, there is the same number of linear extensions to get a rank function on the tree $\mathcal{T}$.

So it is sufficient to calculate the probability $\mathbb{P}_{u<v}$ in $\mathcal{T}_{\rho_1}$. If $\rho_1 = u$, u is before v in $\mathcal{T}$ and we return $\mathbb{P}_{u<v} = 1$. If $\rho_1 = v$, v is before u in $\mathcal{T}$ and we return $\mathbb{P}_{u<v} = 0$.

In the following, let $\rho_1 \neq u, \rho_1 \neq v$. The run of RankProb gives us the probability $\mathbb{P}[r(u) = i]$ in the tree $\mathcal{T}_u$ and $\mathbb{P}[r(v) = i]$ in $\mathcal{T}_v$ for all i. We want to combine these two linear orders. Assume that $r(v) = i$ and we insert j vertices of $\mathcal{T}_u$ before v. Inserting j vertices of $\mathcal{T}_u$ into the linear order of $\mathcal{T}_v$ before v is possible in $\binom{i-1+j}{j}$ ways (see Corollary 4.1.3). Putting the remaining vertices in a linear order is possible in $\binom{|\mathring{V}_{\mathcal{T}_v}|-i+|\mathring{V}_{\mathcal{T}_u}|-j}{|\mathring{V}_{\mathcal{T}_u}|-j}$ ways. The probability that the vertex u is among the j vertices which have smaller rank than v is $\mathbb{P}[r(u) \leq j] = ucum(j)$. There are $|r(\mathcal{T}_u)|$ possible linear orders on $\mathcal{T}_u$ and $|r(\mathcal{T}_v)|$ possible linear orders on $\mathcal{T}_v$. The number of linear orders where vertex v has rank i in $\mathcal{T}_v$, v has rank $i+j$ in $\mathcal{T}_{\rho_1}$ and $r(u) < i + j$ therefore equals

$$p'_{i,j} = \mathbb{P}[r(v) = i] \cdot |r(\mathcal{T}_v)| \cdot \binom{i-1+j}{j} \cdot \binom{|\mathring{V}_{\mathcal{T}_v}| - i + |\mathring{V}_{\mathcal{T}_u}| - j}{|\mathring{V}_{\mathcal{T}_u}| - j} \cdot ucum(j) \cdot |r(\mathcal{T}_u)|$$

Adding up the p' for each i and j gives us the number of linear orders where u is earlier than v.

Combining a linear order on $\mathcal{T}_v$ with a linear order on $\mathcal{T}_u$ is possible in

$$tot := \binom{|\mathring{V}_{\mathcal{T}_u}| + |\mathring{V}_{\mathcal{T}_v}|}{|\mathring{V}_{\mathcal{T}_v}|}$$

different ways (see Corollary 4.1.3). There are $|r(\mathcal{T}_u)|$ linear orders on $\mathcal{T}_u$ and $|r(\mathcal{T}_v)|$ linear orders on $\mathcal{T}_v$, so on $\mathcal{T}_{\rho_1}$, we have

$$tot' := \binom{|\mathring{V}_{\mathcal{T}_u}| + |\mathring{V}_{\mathcal{T}_v}|}{|\mathring{V}_{\mathcal{T}_v}|} |r(\mathcal{T}_v)||r(\mathcal{T}_v)|$$

linear orders. Therefore we get

$$\mathbb{P}_{u<v} = \frac{\sum_{i,j} p'_{i,j}}{tot'} = \frac{\sum_{i,j} p_{i,j}}{tot}$$

with $p_{i,j} = \mathbb{P}[r(v) = i] \cdot \binom{i-1+j}{j} \cdot \binom{|\mathring{V}_{\mathcal{T}_v}|-i+|\mathring{V}_{\mathcal{T}_u}|-j}{|\mathring{V}_{\mathcal{T}_u}|-j} \cdot ucum(j)$. This shows that Compare works correct. □

Theorem 4.2.2. *The runtime of* Compare *is* $O(|\mathring{V}|^2)$.

Proof. Again, note that the combinatorial factors $\binom{n}{k}$ for all $n, k \leq |\mathring{V}|$ can be calculated in advance in quadratic time, see Remark (4.1.4). In the algorithm, those factors can then be obtained in constant time.

Contributions to the runtime from each line in Compare (the runtime is always w.r.t. $|\mathring{V}|$):
Line 1: linear time
Line 2–7: constant time
Line 8: linear time
Line 9: constant time
Line 10: quadratic time, since RankProb has quadratic runtime

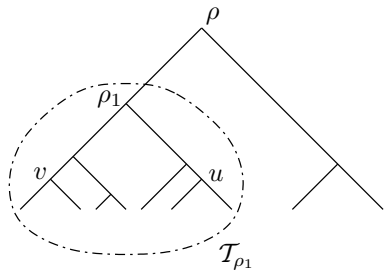


Figure 4.7: Example for COMPARE: Calculate the probability of $u < v$ in the displayed tree $\mathcal{T}$.

Line 11–13: linear time
Line 14: constant time
Line 15–20: quadratic time since $(*)$ has to be evaluated $|\mathring{V}_{\mathcal{T}_u}| \cdot |\mathring{V}_{\mathcal{T}_u}| \leq |\mathring{V}_{\mathcal{T}}|^2$ times
Line 21–23: constant time

Therefore, the overall runtime of COMPARE is $O(|\mathring{V}|^2)$. □

Example 4.2.3. Fig. 4.7 displays the tree $\mathcal{T}$. We want to calculate the probability $\mathbb{P}_{u<v}$, i.e. the probability of vertex u having a smaller rank than vertex v.

A run of the Python code attached in Appendix B with input $(\mathcal{T}, u, v)$ returns $\mathbb{P}_{u<v} = \frac{9}{20}$.

4.3 Application of RANKPROB - Estimating edge lengths in a Yule tree

In [16], a primate supertree on 218 species was constructed with the MRP method (Matrix Representation using Parsimony analysis, see [2, 12]). The resulting supertree is shown in Appendix C. This tree has only 210 interior vertices. There are six 'soft' polytomies in the supertree, i.e. six vertices have more than two direct descendants because the exact resolution is unclear (i.e. the supertree is non-binary).

Since for most of the interior vertices, no molecular estimates were available, the edge lengths for the tree were estimated. Here, the length of an edge represents the time between two speciation events.

A very common stochastic model for trees with edge lengths is the continuous-time Yule model. As in the discrete-time Yule model, at every point in time, each species is equally likely to split and give birth to two new species. The expected waiting time for the next speciation event in a tree with n leaves is $1/n$. That is, each species at any given time has a constant speciation rate (normalized so that 1 is the expected time until it next speciates).

It was assumed that the primate tree $\mathcal{T}_p$ evolved under the continuous-time Yule model. In [16], 10^6 rank functions on $\mathcal{T}_p$ were drawn uniformly at random.

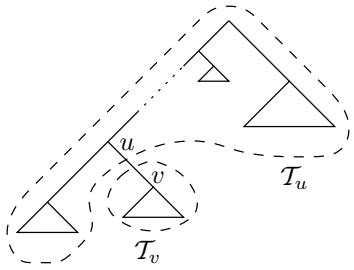


Figure 4.8: Labeling the tree for estimating the edge lengths.

For each of those rank functions, the expected time intervals, i.e. the edge lengths, between vertices were considered (the expected waiting time after the $(n-1)$th event until the nth event is $1/n$).

The authors of [16] concluded their paper by asking for an analytical approach to the estimation of the edge length, and we provide this now.

4.3.1 Analytical estimation of the edge length

Let (u, v) be an interior edge in $\mathcal{T}$ with $u <_{\mathcal{T}} v$. Let X be the random variable 'length of the edge (u, v)' given that $\mathcal{T}$ is generated according to the continuous-time Yule model.

The expected length $\mathbb{E}[X]$ of the edge (u, v) is given by

$$\mathbb{E}[X] = \sum_{i,j} \mathbb{E}[X|r(u) = i, r(v) = j]\mathbb{P}[r(u) = i, r(v) = j].$$

Since under the continuous-time Yule model, the expected waiting time for the next event is $1/n$, we have

$$\mathbb{E}[X|r(u) = i, r(v) = j] = \sum_{k=1}^{j-i} \frac{1}{i+k}.$$

It remains to calculate the probability $\mathbb{P}[r(u) = i, r(v) = j]$. We count all the possible rank functions where $r(u) = i$ and $r(v) = j$. The subtree $\mathcal{T}_v$ consists of v and all its descendants. The tree $\mathcal{T}_u$ evolves from $\mathcal{T}$ when we replace the subtree $\mathcal{T}_v$ by a leaf, see Fig. 4.8.

Note that $\mathbb{P}[r(u) = i, r(v) = j] = 0$ if $|\mathring{V}_{\mathcal{T}_u}| < j - 1$. Therefore, assume $|\mathring{V}_{\mathcal{T}_u}| \geq j - 1$ in the following.

The number of rank functions in $\mathcal{T}_u$ is denoted by $R_{\mathcal{T}_u}$. The probability $\mathbb{P}[r(u) = i]$ can be calculated with RankProb($\mathcal{T}_u$, u). So the number of rank functions in $\mathcal{T}_u$ with $\mathbb{P}[r(u) = i]$ is $\mathbb{P}[r(u) = i] \cdot R_{\mathcal{T}_u}$.

The number of rank functions in $\mathcal{T}_v$ is denoted by $R_{\mathcal{T}_v}$. Let any linear order on the tree $\mathcal{T}_u$ and $\mathcal{T}_v$ be given. Combining those two linear orders to an order on $\mathcal{T}$, where $r(v) = j$ holds, means, that the vertices with rank $1, 2, \ldots, j-1$ in $\mathcal{T}_u$ keep their rank. Vertex v gets rank j. The remaining $|\mathring{V}_{\mathcal{T}_u}| - (j-1)$ vertices in $\mathcal{T}_u$

and $|\mathring{V}_{\mathcal{T}_v}| - 1$ vertices in $\mathcal{T}_v$ have to be shuffled together. According to Corollary (4.1.3), this can be done in

$$\binom{|\mathring{V}_{\mathcal{T}_u}| - (j-1) + |\mathring{V}_{\mathcal{T}_v}| - 1}{|\mathring{V}_{\mathcal{T}_v}| - 1} = \binom{|\mathring{V}_{\mathcal{T}_u}| + |\mathring{V}_{\mathcal{T}_v}| - j}{|\mathring{V}_{\mathcal{T}_v}| - 1}$$

different ways. Overall, we have

$$\mathbb{P}[r(u) = i] \cdot R_{\mathcal{T}_u} \cdot R_{\mathcal{T}_v} \cdot \binom{|\mathring{V}_{\mathcal{T}_u}| + |\mathring{V}_{\mathcal{T}_v}| - j}{|\mathring{V}_{\mathcal{T}_v}| - 1}$$

different rank functions on $\mathcal{T}$ with $r(u) = i$ and $r(v) = j$. For the probability $\mathbb{P}[r(u) = i, r(v) = j]$, we get

$$\mathbb{P}[r(u) = i, r(v) = j] = \frac{\mathbb{P}[r(u) = i] \cdot R_{\mathcal{T}_u} \cdot R_{\mathcal{T}_v} \cdot \binom{|\mathring{V}_{\mathcal{T}_u}| + |\mathring{V}_{\mathcal{T}_v}| - j}{|\mathring{V}_{\mathcal{T}_v}| - 1}}{\sum_{i,j} \mathbb{P}[r(u) = i] \cdot R_{\mathcal{T}_u} \cdot R_{\mathcal{T}_v} \cdot \binom{|\mathring{V}_{\mathcal{T}_u}| + |\mathring{V}_{\mathcal{T}_v}| - j}{|\mathring{V}_{\mathcal{T}_v}| - 1}}$$

Since $R_{\mathcal{T}_u}$ and $R_{\mathcal{T}_v}$ are independent of i and j, those factors cancel out, and we get

$$\mathbb{P}[r(u) = i, r(v) = j] = \frac{\mathbb{P}[r(u) = i] \cdot \binom{|\mathring{V}_{\mathcal{T}_u}| + |\mathring{V}_{\mathcal{T}_v}| - j}{|\mathring{V}_{\mathcal{T}_v}| - 1}}{\sum_{i,j} \mathbb{P}[r(u) = i] \cdot \binom{|\mathring{V}_{\mathcal{T}_u}| + |\mathring{V}_{\mathcal{T}_v}| - j}{|\mathring{V}_{\mathcal{T}_v}| - 1}}$$

Further, we note that

$$\binom{|\mathring{V}_{\mathcal{T}_u}| + |\mathring{V}_{\mathcal{T}_v}| - j}{|\mathring{V}_{\mathcal{T}_v}| - 1} = \frac{(|\mathring{V}_{\mathcal{T}}| - j)!}{(|\mathring{V}_{\mathcal{T}_v}| - 1)!(|\mathring{V}_{\mathcal{T}}| - j - (|\mathring{V}_{\mathcal{T}_v}| - 1))!}$$

Again, since $(|\mathring{V}_{\mathcal{T}_v}| - 1)!$ is independent of i and j, this factor cancels out, and we are left with

$$\mathbb{P}[r(u) = i, r(v) = j] = \frac{\mathbb{P}[r(u) = i] \cdot \prod_{k=0}^{|\mathring{V}_{\mathcal{T}_v}|-2} (|\mathring{V}_{\mathcal{T}}| - j - k)}{\sum_{i,j} \mathbb{P}[r(u) = i] \cdot \prod_{k=0}^{|\mathring{V}_{\mathcal{T}_v}|-2} (|\mathring{V}_{\mathcal{T}}| - j - k)}$$

Let $\Omega = \{(i,j) : i < j, i, j \in \{1, \ldots, |\mathring{V}|\}, |\mathring{V}_{\mathcal{T}_u}| \geq j - 1\}$. With that notation, the expected edge length $\mathbb{E}[X]$ is

$$\begin{aligned}
\mathbb{E}[X] &= \sum_{(i,j)\in\Omega} \mathbb{E}[X | r(u) = i, r(v) = j] \mathbb{P}[r(u) = i, r(v) = j] \\
&= \sum_{(i,j)\in\Omega} \left[\left(\sum_{k=1}^{j-i} \frac{1}{i+k} \right) \frac{\mathbb{P}[r(u) = i] \cdot \prod_{k=0}^{|\mathring{V}_{\mathcal{T}_v}|-2} (|\mathring{V}_{\mathcal{T}}| - j - k)}{\sum_{(i,j)\in\Omega} \left[\mathbb{P}[r(u) = i] \cdot \prod_{k=0}^{|\mathring{V}_{\mathcal{T}_v}|-2} (|\mathring{V}_{\mathcal{T}}| - j - k) \right]} \right] \\
&= \frac{\sum_{(i,j)\in\Omega} \left[\left(\sum_{k=1}^{j-i} \frac{1}{i+k} \right) \cdot \mathbb{P}[r(u) = i] \cdot \prod_{k=0}^{|\mathring{V}_{\mathcal{T}_v}|-2} (|\mathring{V}_{\mathcal{T}}| - j - k) \right]}{\sum_{(i,j)\in\Omega} \left[\mathbb{P}[r(u) = i] \cdot \prod_{k=0}^{|\mathring{V}_{\mathcal{T}_v}|-2} (|\mathring{V}_{\mathcal{T}}| - j - k) \right]}
\end{aligned} \tag{4.3}$$

Remark 4.3.1. With Equation (4.3), we can estimate the length of all the interior edges. For the pendant edges, the approach above gives us no estimate though. All we know is that the time from the latest interior vertex, which has rank $n-1$, until the presence is expected to be at most $1/n$ where n is the number of leaves.

Remark 4.3.2. In a supertree, we can have interior vertices which are not fully resolved, i.e. an interior vertex can have more than two descendants, because the exact resolution is unclear. Our calculation for the expected edge length assumes a binary tree though.

However, we can calculate the expected edge length for each possible binary resolution of the supertree. Assume the supertree $\mathcal{T}$ has the possible binary resolutions $\mathcal{T}_1, \ldots, \mathcal{T}_m$. For an edge (u, v) in $\mathcal{T}$ where $u <_{\mathcal{T}} v$, the expected edge length is calculated in the trees $\mathcal{T}_i$ for $i = 1, \ldots, m$. The expected edge length in $\mathcal{T}_i$ is denoted by e_i for $i = 1, \ldots, m$.

We calculate the expected edge length $\mathbb{E}[X]$ of (u, v) in the supertree $\mathcal{T}$ by

$$\mathbb{E}[X] = \frac{\sum_i e_i \mathbb{P}[\mathcal{T}_i]}{\sum_i \mathbb{P}[\mathcal{T}_i]} \tag{4.4}$$

where the probability $\mathbb{P}[\mathcal{T}_i]$ is calculated according to Corollary (2.2.5).

Note that if u is a vertex with more than two descendants in $\mathcal{T}$, v is in general not a direct descendant of u in $\mathcal{T}_i$. The value e_i in resolution $\mathcal{T}_i$ is then the sum of all expected edge lengths on the path from u to v in $\mathcal{T}_i$.

Remark 4.3.3. In the primate supertree in Appendix C, there are six interior vertices with more than two descendants (vertex labels $48, 63, 148, 153, 157$ and 200). For the vertices labeled with 63 and 200, only one resolution is possible (up to the labeling).

The interior vertices with label 48, 153 and 157 have three descendants each. So there are 3^3 possible binary resolutions. The interior vertex 148 has four leaf-descendants. There are two possible binary resolutions (up to the labeling). To calculate the expected edge lengths for the primate supertree, we therefore have to calculate the expected edge lengths on $3^3 \cdot 2$ binary trees and then calculate the weighted sum from Equation (4.4).

Chapter 5

Speciation Rates

This chapter was motivated by Craig Moritz and Andrew Hugall, biologists from Berkeley and Adelaide. They looked at a tree showing the relationships between a set of snails. Each of those snails lives either in rain forest or open forest. The tree has edge lengths assigned. Moritz and Hugall asked if the rate of speciation is different for rain forest snails and open forest snails.

Mathematically, determining the rate of speciation is the following problem. The leaves are divided into two classes, α and β (e.g. rain forest and open forest snails). Given the rate that a species belonging to class α changes to a species belonging to class β (and vice versa), we calculate the expected length of an edge between two species of group α (resp. β). This expected length is an estimate for the inverse of the rate of speciation and is calculated in linear time.

5.1 Some notation

Definition 5.1.1. Let X' be a non-empty subset of X. Let C be a non-empty set. A *character* on X is a function $\chi : X' \to C$. C is the *character state set* of χ. If $X' = X$, we say χ is a *full character*. If $|C| = 2$, we say χ is a *binary character*.

Definition 5.1.2. Let $\mathcal{T}$ be a rooted phylogenetic X-tree with vertex set V and leaf set $L \subset V$. Let χ be a full binary character on $\mathcal{T}$, $\chi : X \to \{\alpha, \beta\}$. Define

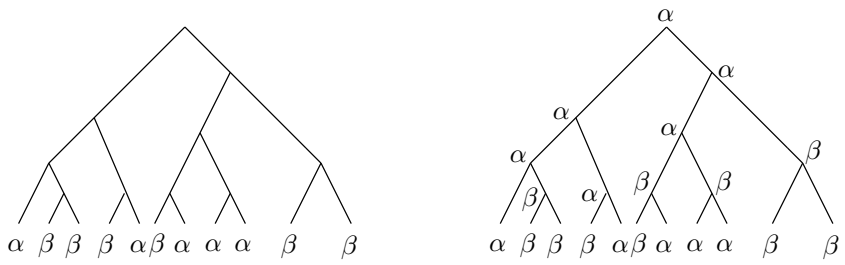


Figure 5.1: A phylogenetic tree with a full character on the left and a phylogenetic state tree on the right (without the leaf labels).

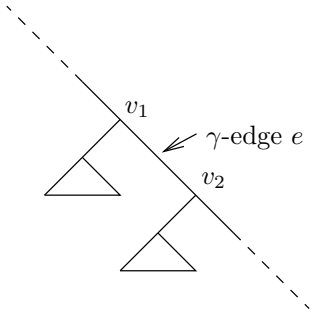


Figure 5.2: With $s(v_1) = \gamma_1$ and $s(v_2) = \gamma_2$, the edge $e = (v_1, v_2)$ is a γ-edge.

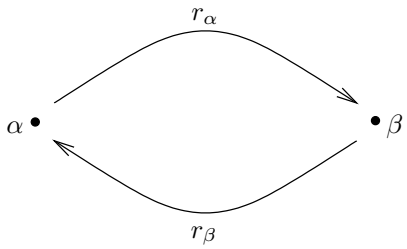


Figure 5.3: Rate of the state change for a binary character

$s : V \to \{\alpha, \beta\}$ with $s|_L = \chi \circ \phi^{-1}$. $(\mathcal{T}, s)$ is called a *phylogenetic state tree*, s a *state function*.

In the following, the phylogenetic state tree $(\mathcal{T}, s)$ shall have assigned a function $l : E \to \mathbb{R}^+$. l shall denote the edge lengths of $\mathcal{T}$. Let $\eta \in \{\alpha, \beta\}$ throughout this chapter. Let v be any node in $(\mathcal{T}, s)$ with $s(v) = \eta$. We then say that the *state of* v is η. Let $\gamma \in \{\alpha, \beta\} \times \{\alpha, \beta\}$ throughout the chapter, i.e. $\gamma = (\gamma_1, \gamma_2)$ with $\gamma_1, \gamma_2 \in \{\alpha, \beta\}$. An edge $e = (v_1, v_2)$ of $(\mathcal{T}, s)$ where $v_1 <_\mathcal{T} v_2$ and $s(v_1) = \gamma_1$, $s(v_2) = \gamma_2$ is called a $\gamma - edge$.

5.2 Markov Chain Model

Throughout evolution, assume that state α changes to state β with rate r_α and state β changes to state α with rate r_β, so the rates only depend upon the state of the last vertex (see Fig. 5.3). This means that the state change follows a Markov Chain model, and for that model, we want to calculate the transision matrix

$$P(l(e)) = \begin{pmatrix} p_{\alpha\alpha}(l(e)) & p_{\alpha\beta}(l(e)) \\ p_{\beta\alpha}(l(e)) & p_{\beta\beta}(l(e)) \end{pmatrix}$$

where $p_{\gamma_1\gamma_2}(l(e)) = \mathbb{P}\left[(s(v_2) = \gamma_2)|(s(v_1) = \gamma_1)\right]$ with $e = (v_1, v_2)$ and $v_1 <_\mathcal{T} v_2$.

The rate matrix R is defined as

$$R = \begin{pmatrix} -r_\alpha & r_\alpha \\ r_\beta & -r_\beta \end{pmatrix}$$

Diagonalization of R yields

$$R = \begin{pmatrix} -r_\alpha & r_\alpha \\ r_\beta & -r_\beta \end{pmatrix} = S \begin{pmatrix} 0 & 0 \\ 0 & -(r_\alpha + r_\beta) \end{pmatrix} S^{-1}$$

with

$$S = \begin{pmatrix} 1 & r_\alpha \\ 1 & -r_\beta \end{pmatrix}$$

From stochastic processes, we know that the connection between the rate matrix and the transition matrix is

$$P'(l(e)) = RP(l(e))$$

Solving this differential equation yields

$$P(l(e)) = P(0)e^{R(l(e))}$$

with $P(0) = Id$ since $l(e) = 0$ means staying in the vertex. Therefore $P(l(e))$ can be rewritten as

$$\begin{aligned} P(l(e)) &= e^{R(l(e))} \\ &= \exp\{S \begin{pmatrix} 0 & 0 \\ 0 & -(r_\alpha + r_\beta) \end{pmatrix} S^{-1} l(e)\} \\ &= S \exp\{\begin{pmatrix} 0 & 0 \\ 0 & -(r_\alpha + r_\beta) \end{pmatrix} l(e)\} S^{-1} \\ &= S \begin{pmatrix} 1 & 0 \\ 0 & e^{-(r_\alpha + r_\beta)l(e)} \end{pmatrix} S^{-1} \\ &= \begin{pmatrix} \frac{1}{r_\alpha + r_\beta}\left(r_\beta + r_\alpha e^{-(r_\alpha + r_\beta)l(e)}\right) & \frac{r_\alpha}{r_\alpha + r_\beta}\left(1 - e^{-(r_\alpha + r_\beta)l(e)}\right) \\ \frac{r_\beta}{r_\alpha + r_\beta}\left(1 - e^{-(r_\alpha + r_\beta)l(e)}\right) & \frac{1}{r_\alpha + r_\beta}\left(r_\alpha + r_\beta e^{-(r_\alpha + r_\beta)l(e)}\right) \end{pmatrix} \end{aligned}$$

The initial probability of vertex v being in state η shall be π_η, $\eta \in \{\alpha, \beta\}$. It holds

$$\begin{pmatrix} \pi_\alpha & \pi_\beta \end{pmatrix} R = \begin{pmatrix} \pi_\alpha & \pi_\beta \end{pmatrix} \begin{pmatrix} -r_\alpha & r_\alpha \\ r_\beta & -r_\beta \end{pmatrix} = 0$$

so

$$\pi = \begin{pmatrix} \pi_\alpha & \pi_\beta \end{pmatrix} = \begin{pmatrix} \frac{r_\beta}{r_\alpha + r_\beta} & \frac{r_\alpha}{r_\alpha + r_\beta} \end{pmatrix}$$

Therefore, for any given phylogenetic tree $\mathcal{T}$ with edge lengths $l(e)$, the probability of its vertices being in states according to a state function s is

$$\mathbb{P}[s] = \pi_{s(\rho)} \prod_{\substack{e \in E \\ e=(v_1,v_2) \\ v_1 <_{\mathcal{T}} v_2}} p_{s(v_1),s(v_2)} \tag{5.1}$$

Furthermore, it holds for any $e \in E$ with $e = (v_1, v_2)$

$$p_{s(v_1),s(v_2)}(l(e)) = \frac{r_{s(v_1)}}{r_{s(v_2)}} p_{s(v_2),s(v_1)}(l(e)) \tag{5.2}$$

5.3 Expected length of a γ-edge

Given a phylogenetic tree $\mathcal{T}$ with character χ, edge length $l(e)$ and rate matrix R, we want to calculate the expected average length of a γ-edge over all $(\mathcal{T}, s)$. The inverse of this length is an estimate for the rate of speciation.

Calculating the expected average length of a γ-edge over all $(\mathcal{T}, s)$ means calculating

$$\mathbb{E}_\chi \left[\frac{\sum\limits_{e \in E,\ e\ \gamma-edge} l(e)}{\#\ of\ \gamma - edges} \right]$$

where $\mathbb{E}_\chi$ denotes the expected value over all s given $s|_L = \chi$. Trying to calculate this expected value turns out to give us very nasty recursion formulas.
So we change the problem slightly and try to calculate instead

$$\Psi_\gamma = \frac{\mathbb{E}_\chi \left[\sum\limits_{e \in E,\ e\ \gamma-edge} l(e) \right]}{\mathbb{E}_\chi \left[\#\ of\ \gamma - edges \right]}$$

Define the random variable

$$X_\gamma(e) := \begin{cases} 1 & \text{if } e \text{ is } \gamma\text{-edge} \\ 0 & \text{else} \end{cases}$$

With that, we get

$$\begin{aligned} \Psi_\gamma &= \frac{\mathbb{E}_\chi \left[\sum\limits_{e \in E,\ e\ \gamma-edge} l(e) \right]}{\mathbb{E}_\chi \left[\#\ of\ \gamma - edges \right]} \\ &= \frac{\mathbb{E}_\chi \left[\sum\limits_{e \in E} l(e) X_\gamma(e) \right]}{\mathbb{E}_\chi \left[\sum\limits_{e \in E} X_\gamma(e) \right]} \\ &= \frac{\sum\limits_{e \in E} l(e) \mathbb{P}\left[(X_\gamma(e) = 1) | \chi \right]}{\sum\limits_{e \in E} \mathbb{P}\left[(X_\gamma(e) = 1) | \chi \right]} \end{aligned} \tag{5.3}$$

where $\mathbb{P}\left[(X_\gamma(e) = 1) | \chi \right]$ denotes the probability of e being a γ-edge given $s|_L = \chi$. So it is basically left to calculate $\mathbb{P}\left[(X_\gamma(e) = 1) | \chi \right]$. To do so, we first define two subtrees of $\mathcal{T}$ (see also Fig. 5.4). Denote the end vertices of e by ρ_1 and ρ_2 with $\rho_1 <_\mathcal{T} \rho_2$. By deleting the γ-edge e in $\mathcal{T}$, we get two new trees $\mathcal{T}_1$ and $\mathcal{T}_2$, $\mathcal{T}_1$ with $\rho_1 \in \mathcal{T}_1$ and character $\chi_1 = \chi|_{\phi^{-1}(L_{\mathcal{T}_1})}$, and $\mathcal{T}_2$ with $\rho_2 \in \mathcal{T}_2$ and character

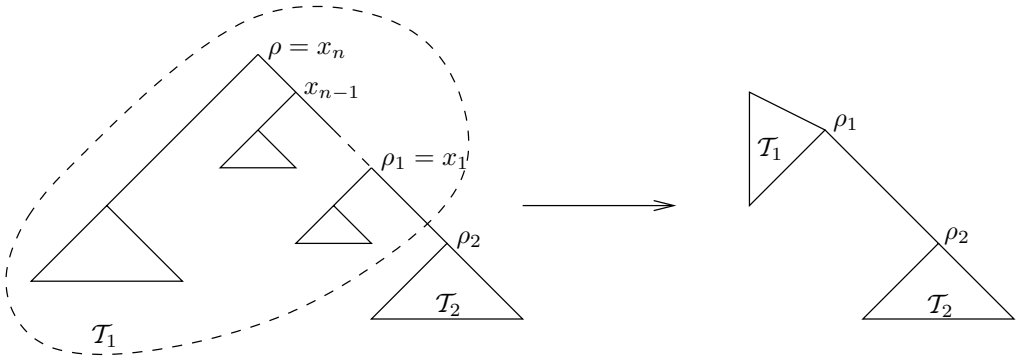


Figure 5.4: Calculating the expected edge length: Defining $\mathcal{T}_1$ and $\mathcal{T}_2$

$\chi_2 = \chi|_{\phi^{-1}(L_{\mathcal{T}_2})}$ where $L_{\mathcal{T}_i}$ denotes the set of leaves of $\mathcal{T}_i$, $i \in \{1, 2\}$. The root in $\mathcal{T}_i$ shall be ρ_i, so ρ becomes an ordinary vertex in $\mathcal{T}_1$.

$\mathbb{P}\left[\chi_i | (s(\rho_i) = \gamma_i)\right]$ shall denote the probability of the character χ_i on the tree $\mathcal{T}_i$ given $s(\rho_i) = \gamma_i$. $\mathbb{P}\left[\chi_{\mathcal{T}\setminus\mathcal{T}_2} | (s(\rho_1) = \gamma_1)\right]$ shall denote the probability of the character $\chi_{\mathcal{T}\setminus\mathcal{T}_2}$ on the tree $\mathcal{T} \setminus \mathcal{T}_2$ given $s(\rho_1) = \gamma_1$. $\mathbb{P}[\chi_{\mathcal{T}}, s]$ shall denote the probability of the character χ and the state function s on the tree $\mathcal{T}$. We denote the vertices on the path from ρ_1 to ρ by $\rho_1 = x_1, x_2, \ldots, x_{n-1}, x_n = \rho$. With (5.1) and (5.2), it holds

$$\begin{aligned}
\mathbb{P}\left[\chi_1, s\right] &= \frac{\pi_{s(\rho_1)} \prod_{i=1}^{n-1} p_{s(x_i),s(x_{i+1})}}{\pi_{s(\rho)} \prod_{i=1}^{n-1} p_{s(x_{i+1}),s(x_i)}} \mathbb{P}\left[\chi_{\mathcal{T}\setminus\mathcal{T}_2}, s\right] \\
&= \frac{\pi_{s(\rho_1)} \prod_{i=1}^{n-1} p_{s(x_i),s(x_{i+1})}}{\pi_{s(\rho)} \prod_{i=1}^{n-1} \frac{r_{s(x_{i+1})}}{r_{s(x_i)}} p_{s(x_i),s(x_{i+1})}} \mathbb{P}\left[\chi_{\mathcal{T}\setminus\mathcal{T}_2}, s\right] \\
&= \frac{\pi_{s(\rho_1)} r_{s(x_1)}}{\pi_{s(\rho)} r_{s(x_n)}} \mathbb{P}\left[\chi_{\mathcal{T}\setminus\mathcal{T}_2}, s\right] \\
&= \frac{\frac{r_\alpha r_\beta}{r_\alpha + r_\beta}}{\frac{r_\alpha r_\beta}{r_\alpha + r_\beta}} \mathbb{P}\left[\chi_{\mathcal{T}\setminus\mathcal{T}_2}, s\right] \\
&= \mathbb{P}\left[\chi_{\mathcal{T}\setminus\mathcal{T}_2}, s\right]
\end{aligned}$$

This yields

$$\begin{aligned}
\mathbb{P}\left[\chi_1 | (s(\rho_1) = \gamma_1)\right] &= \sum_{s : s(\rho_1) = \gamma_1} \mathbb{P}\left[\chi_1, s\right] \\
&= \sum_{s : s(\rho_1) = \gamma_1} \mathbb{P}\left[\chi_{\mathcal{T}\setminus\mathcal{T}_2}, s\right] \\
&= \mathbb{P}\left[\chi_{\mathcal{T}\setminus\mathcal{T}_2} | (s(\rho_1) = \gamma_1)\right]
\end{aligned}$$

With that result, we get

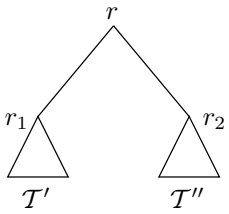


Figure 5.5: Calculating the expected edge length: Defining $\tilde{\mathcal{T}}$

$$\begin{aligned}
&\mathbb{P}\left[(X_\gamma(e)=1)|\chi\right] = \\
&= \frac{\mathbb{P}\left[(X_\gamma(e)=1)\right]\mathbb{P}\left[\chi|(X_\gamma(e)=1)\right]}{\mathbb{P}\left[\chi\right]} \\
&= \frac{\pi_{\gamma_1}p_{\gamma_1\gamma_2}(l(e))\mathbb{P}\left[\chi_{\mathcal{T}\setminus\mathcal{T}_2}|(s(\rho_1)=\gamma_1)\right]\mathbb{P}\left[\chi_2|(s(\rho_2)=\gamma_2)\right]}{\sum_{\gamma=(\gamma_1,\gamma_2)}\pi_{\gamma_1}p_{\gamma_1\gamma_2}(l(e))\mathbb{P}\left[\chi_{\mathcal{T}_1}|(s(\rho_1)=\gamma_1)\right]\mathbb{P}\left[\chi_{\mathcal{T}_2}|(s(\rho_2)=\gamma_2)\right]} \\
&= \frac{\pi_{\gamma_1}p_{\gamma_1\gamma_2}(l(e))\mathbb{P}\left[\chi_1|(s(\rho_1)=\gamma_1)\right]\mathbb{P}\left[\chi_2|(s(\rho_2)=\gamma_2)\right]}{\sum_{\gamma=(\gamma_1,\gamma_2)}\pi_{\gamma_1}p_{\gamma_1\gamma_2}(l(e))\mathbb{P}\left[\chi_{\mathcal{T}_1}|(s(\rho_1)=\gamma_1)\right]\mathbb{P}\left[\chi_{\mathcal{T}_2}|(s(\rho_2)=\gamma_2)\right]} \qquad (5.4)
\end{aligned}$$

$\mathbb{P}\left[\chi_i|(s(\rho_i)=\gamma_i)\right]$ is calculated in a recursive way, starting from the bottom of the tree.

Suppose we have the subtree $\tilde{\mathcal{T}}$ as in Fig. 5.5 and either r_1, r_2 are leaves or we know $\mathbb{P}\left[\chi_{\mathcal{T}'}|(s(r_1)=\eta)\right]$ on tree $\mathcal{T}'$, $\mathbb{P}\left[\chi_{\mathcal{T}''}|(s(r_2)=\eta)\right]$ on tree $\mathcal{T}''$, for $\eta\in\{\alpha,\beta\}$. With that, we get the following *recursive formulas* for the probabilities on tree $\tilde{\mathcal{T}}$.

- For r_1 and r_2 leaves:

$$\mathbb{P}\left[\chi_{\tilde{\mathcal{T}}}|(s(r)=\eta)\right] = \frac{p_{\eta\chi(r_1)}p_{\eta\chi(r_2)}}{\sum_{\eta_1,\eta_2\in\{\alpha,\beta\}}p_{\eta\eta_1}p_{\eta\eta_2}}$$

- For r_1 leave, r_2 interior node:

$$\mathbb{P}\left[\chi_{\tilde{\mathcal{T}}}|(s(r)=\eta)\right] = \frac{\sum_{\eta_1\in\{\alpha,\beta\}}\mathbb{P}\left[\chi_{\mathcal{T}'}|(s(r_1)=\eta_1)\right]p_{\eta\chi(r_2)}p_{\eta\eta_1}}{\sum_{\eta_1,\eta_2\in\{\alpha,\beta\}}\mathbb{P}\left[\chi_{\mathcal{T}'}|(s(r_1)=\eta_1)\right]p_{\eta\eta_2}p_{\eta\eta_1}}$$

- For r_1 and r_2 interior nodes:

$$\begin{aligned}
&\mathbb{P}\left[\chi_{\tilde{\mathcal{T}}}|(s(r)=\eta)\right] = \\
&\sum_{\eta_1,\eta_2\in\{\alpha,\beta\}}\mathbb{P}\left[\chi_{\mathcal{T}'}|(s(r_1)=\eta_1)\right]\mathbb{P}\left[\chi_{\mathcal{T}''}|(s(r_2)=\eta_2)\right]p_{\eta\eta_1}p_{\eta\eta_2}
\end{aligned}$$

Algorithm EdgeLength $(\mathcal{T}, \chi)$
Input: A rooted binary phylogenetic tree $\mathcal{T}$ and a character χ on $\mathcal{T}$ with state change rates r_α and r_β
Output: The values Ψ_γ for $\gamma \in \{\alpha, \beta\} \times \{\alpha, \beta\}$ (*cf.* Equation (5.3))

- Define the subtrees $\mathcal{T}_1$ and $\mathcal{T}_2$ of $\mathcal{T}$ as described above.
- Calculate $\mathbb{P}\left[\chi_{\mathcal{T}_i} | (s(\rho_i) = \gamma_j)\right]$ for $i \in \{1, 2\}$, $j \in \{1, 2\}$, with the recursive formulas from above.
- Evaluate $\mathbb{P}\left[(X_\gamma(e) = 1) | \chi\right]$ according to (5.4) for all $\gamma \in \{\alpha, \beta\} \times \{\alpha, \beta\}$.
- Evaluate Ψ_γ according to (5.3) for all $\gamma \in \{\alpha, \beta\} \times \{\alpha, \beta\}$.

Theorem 5.3.1. EdgeLength *works correct, i.e. it returns*

$$\Psi_\gamma = \frac{\mathbb{E}_\chi \left[\sum_{e \in E,\ e\ \gamma-edge} (l(e)) \right]}{\mathbb{E}_\chi \left[\#\ of\ \gamma - edges \right]}$$

The complexity is $O(|V|)$, so it is linear.

Proof. The correctness of the algorithm follows from the construction above. It is left to verify the runtime.
Calculating the probabilities $\mathbb{P}\left[(\chi_{\mathcal{T}_i} | (s(\rho_i) = \gamma_j)\right]$ for $i \in \{1, 2\}$, $j \in \{1, 2\}$ with the recursive formulas requires $O(|V|)$ calculations since we have to evaluate one recursion formula for each vertex. For each edge e, $\mathbb{P}\left[(X_\gamma(e) = 1) | \chi\right]$ can then be calculated according to (5.4) with a constant number of calculations. So obtaining $\mathbb{P}\left[(X_\gamma(e) = 1) | \chi\right]$ for all e requires $O(|E|) = O(|V|)$ calculations. Calculating Ψ_γ according to (5.3) requires again $O(|E|)$ calculations. Therefore, the complexity is linear. □

Outlook

There are several topics in the thesis which suggest further work.

In Chapter 3, we conclude with the log-likelihood-ratio test for deciding if a tree evolved under Yule. The given bound for the power of the test, Equation (3.4), depends on the bound for the Azuma inequality. The bound $\ln n$ for the Azuma inequality was obtained in 3.2.1 by a lot of rough estimations. So we are very confident that there can be found a better bound $c \ln n$, with $c \ll 1$ being a constant. This would lead to an improved bound for the power of the log-likelihood-ratio test (i.e. one could show analytically that the log-likelihood-ratio test is very good even on trees with a small number of leaves).

The edge lengths estimation in Section 4.3 will be implemented by Rutger Vos in Perl for his library and in Java for Mesquite (Mesquite is a tree manipulation software suite). Once implemented, the algorithm can finally be applied to real data. One can then estimate the edge lengths of a constructed supertree.

Section 5 provides an algorithm for calculating $\Psi_{\alpha,\alpha}$ and $\Psi_{\beta,\beta}$ which estimate the average edge lengths. Let ψ_α be the speciation rate for species of class α and let ψ_β be the speciation rate for species of class β. One could test the hypothesis $\psi_{\alpha,\alpha} = \psi_{\beta,\beta}$ against $\psi_{\alpha,\alpha} \neq \psi_{\beta,\beta}$ with the statistic $\frac{\Psi_{\alpha,\alpha}}{\Psi_{\beta,\beta}}$. For evaluating this test, i.e. obtaining the Type I and Type II error, one can use simulations.

Further, in Section 5, we assumed that the transition rates r_α and r_β are given. An interesting open question is how to handle the problem without having these transition rates in advance.

Appendix A

List of Symbols

Symbol	*Meaning*	*page*
$\leq_T$	partial order on the vertices of a tree T	6
$\leq_{\mathcal{T}}$	partial order on the vertices of $\mathcal{T}$	6
$(2n-1)!!$	$(2n-1) \times (2n-3) \ldots 3 \times 1$	10
$(\mathcal{T}, s)$	phylogenetic state tree	6
$(\mathcal{T}, r)$	ranked phylogenetic tree $\mathcal{T}$ with rank function r	6
$\alpha_{\mathcal{T},v}(i)$	$\lvert\{r : r(v) = i, r \in r(\mathcal{T})\}\rvert$	41
χ	character on a phylogenetic tree	57
$\delta(v)$	degree of vertex v	5
λ_v	number of elements of $\mathring{V}$ that are descendants of v	7
π	initial probability distribution of Markov chain	59
ρ	root of a tree	5
ϕ	labelling function of a phylogenetic tree $\mathcal{T}$	5
Ψ_γ	estimated length of a γ-edge	60
$\mathcal{T}$	phylogenetic X-tree	5
$\mathcal{T}_p$	Primate supertree constructed in [16]	73
$\mathcal{T}_v$	phylogenetic subtree of $\mathcal{T}$ induced by vertex v	6
$\mathcal{T}_{X'}$	phylogenetic subtree of $\mathcal{T}$ with label set X'	6
$\mathbb{J}_p$	Entropy of the probability distribution p	19
$\mathbb{P}_{u<v}$	Probability $\mathbb{P}[r(u) < r(v)\lvert\mathcal{T}]$	50
$\mathbb{P}_U$	Uniform distribution on $RB(X)$	18
$\mathbb{P}_U[\mathcal{T}]$	Probability of $\mathcal{T}$ under the uniform model	18
$\mathbb{P}_Y$	Yule distribution on $RB(X)$	18
$\mathbb{P}_Y[\mathcal{T}]$	Probability of $\mathcal{T}$ under the Yule model	18
c_n	Catalan number	10
C	set of character states	57
$d(v)$	number of direct descendants of vertex v	6

$d_{KL}(p,q)$	Kullbach-Liebler distance between p and q	20
$E, E_{\mathcal{T}}$	Edges of a phylogenetic tree $\mathcal{T}$	5
$l(e)$	Length of edge e in $\mathcal{T}$	58
$L, L_{\mathcal{T}}$	Leaf set of a (phylogenetic) tree	5
p_{γ_1,γ_2}	probability of state change from γ_1 to γ_2	58
$P(l(e))$	transition matrix of Markov chain, dependent on edge length	58
$r_\alpha(r_\beta)$	rate of change from state α to β (β to α)	58
$r, r_{\mathcal{T}}$	rank function of phylogenetic tree $\mathcal{T}$	6
$r(\mathcal{T})$	Set of rank functions on $\mathcal{T}$	6
$rRB(n)$	Set of ranked binary phylogenetic X-trees with $X = \{1, 2, \ldots n\}$	9
$rRB(X)$	Set of ranked binary phylogenetic X-trees	8
R	rate matrix of a Markov chain	58
$RB(n)$	Set of binary phylogenetic X-trees with $X = \{1, 2, \ldots n\}$	9
$RB(X)$	Set of binary phylogenetic X-trees	8
s	state function	58
$V, V_{\mathcal{T}}$	Set of vertices of a (phylogenetic) tree	5
$\mathring{V}, \mathring{V}_{\mathcal{T}}$	Set of interior vertices of a (phylogenetic) tree	5

Appendix B

Algorithms coded in Python

```
# Rank functions
# Daniel Ford, Tanja Gernhard 2006
#
# Functions:
#
# rankprob(t,u) - returns the probability distribution
#       of the rank of vertex "u" in tree "t"
# expectedrank(t,u) returns the expected rank
#       of vertex "u" and the variance
# compare(t,u,v) - returns the probability that "u"
#       is below "v" in tree "t"



import random

# How we store the trees:
# The interior vertices of a tree with n leaves are
#       labeled by 1...n-1
# Example input tree for all the algorithms below:
# The tree "t" below has n=9 leaves and the inner nodes have
#       label 1...8
t1 = ((), (), {'leaves_below': 2, 'label': 4}), (),
    {'leaves_below': 3, 'label': 3})
t2 = ((), (), {'leaves_below': 2, 'label': 7}), ((), (),
    {'leaves_below': 2, 'label': 8}),
    {'leaves_below': 4, 'label': 6})
t3 = ((), (), {'leaves_below': 2, 'label': 5})
t4 = (t1,t3,{'leaves_below': 5, 'label': 2})
t = (t2,t4,{'leaves_below': 9, 'label': 1})
```

```
# Calculation of n choose j
# This version saves partial results for use later
nc_matrix = [] #stores the values of nchoose(n,j)
# -- note: order of indices is reversed
def nchoose_static(n,j,nc_matrix):
    if j>n:
        return 0
    if len(nc_matrix)<j+1:
        for i in range(len(nc_matrix),j+1):
            nc_matrix += [[]]
    if len(nc_matrix[j])<n+1:
        for i in range(len(nc_matrix[j]),j):
           nc_matrix[j]+=[0]
        if len(nc_matrix[j])==j:
           nc_matrix[j]+=[1]
        for i in range(len(nc_matrix[j]),n+1):
           nc_matrix[j]+=[nc_matrix[j][i-1]*i/(i-j)]
    return nc_matrix[j][n]

# dynamic programming verion
def nchoose(n,j):
    return nchoose_static(n,j,nc_matrix)
     #nc_matrix acts as a static variable


# get the number of descendants of u and of all vertices on the
# path to the root (subroutine for rankprob(t,u))
def numDescendants(t,u):
    if t == ():
        return [False,False]
    if t[2]["label"]==u:
        return [True,[t[2]["leaves_below"]-1]]
    x = numDescendants(t[0],u)
    if x[0] == True:
        if t[1]==():
            n = 0
        else:
            n = t[1][2]["leaves_below"]-1
        return [True,x[1]+[n]]
    y = numDescendants(t[1],u)
    if y[0] == True:
        if t[0]==():
            n = 0
        else:
            n = t[0][2]["leaves_below"]-1
```

```
        return [True,y[1]+[n]]
    else:
        return [False,False]


# A version of rankprob which uses the function numDescendants
def rankprob(t,u):
    x = numDescendants(t,u)
    x = x[1]
    lhsm = x[0]
    k = len(x)
    start = 1
    end = 1
    rp = [0,1]
    step = 1
    while step < k:
        rhsm = x[step]
        newstart = start+1
        newend = end+rhsm+1
        rp2 = []
        for i in range(0,newend+1):
            rp2+=[0]
        for i in range(newstart,newend+1):
            q = max(0,i-1-end)
            for j in range(q,min(rhsm,i-2)+1):
                a = rp[i-j-1]*nchoose(lhsm + rhsm - (i-1),rhsm-j)
                 *nchoose(i-2,j)
                rp2[i]+=a
        rp = rp2
        start = newstart
        end = newend
        lhsm = lhsm+rhsm+1
        step+=1
    tot = float(sum(rp))
    for i in range(0,len(rp)):
        rp[i] = rp[i]/tot
    return rp


# For tree "t" and vertex "u" calculate the
# expected rank and variance
def expectedrank(t,u):
    rp = rankprob(t,u)
    mu = 0
    sigma = 0
```

```
    for i in range(0,len(rp)):
        mu += i*rp[i]
        sigma += i*i*rp[i]
    return (mu,sigma-mu*mu)


# GCD - assumes positive integers as input
# (subroutine for compare(t,u,v))
def gcd(n,m):
    if n==m:
        return n
    if m>n:
        [n,m]=[m,n]
    i = n/m
    n = n-m*i
    if n==0:
        return m
    return gcd(m,n)


# Takes two large integers and attempts to divide them and give
# the float answer without overflowing
# (subroutine for compare(t,u,v))
# does this by first taking out the gcd
def gcd_divide(n,m):
    x = gcd(n,m)
    n = n/x
    m = m/x
    return n/float(m)


# returns the subtree rooted at the common ancestor of u and v
# (subroutine for compare(t,u,v))
# return
# True/False - have we found u yet
# True/False - have we found v yet
# the subtree - if we have found u and v
# the u half of the subtree
# the v half of the subtree
def subtree(t,u,v):
    if t == ():
        return [False,False,False,False,False]
    [a,b,c,x1,x2]=subtree(t[0],u,v)
    [d,e,f,y1,y2]=subtree(t[1],u,v)
    if (a and b):
```

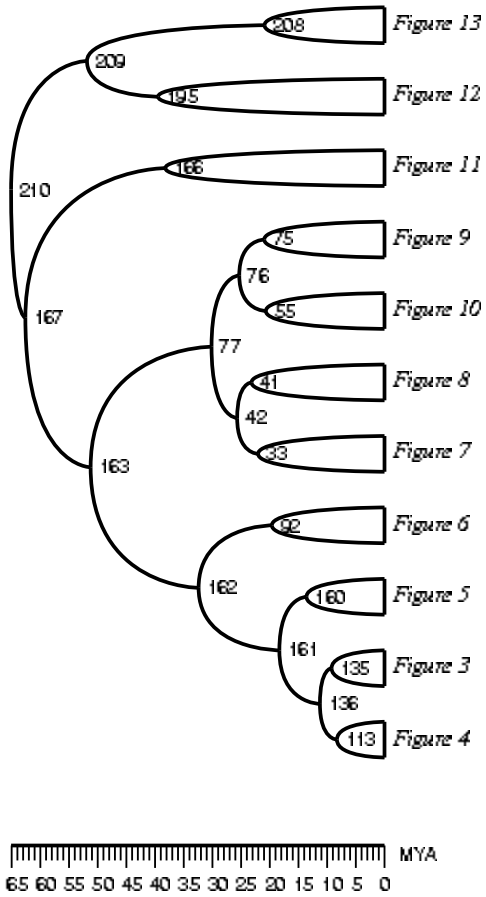
```
        return [a,b,c,x1,x2]
    if (d and e):
        return [d,e,f,y1,y2]
    #
    x = (a or d or t[2]["label"]==u)
    y = (b or e or t[2]["label"]==v)
    #
    t1 = False
    t2 = False
    #
    if a:
      t1 = x1
    if b:
        t2 = x2
    if d:
        t1 = y1
    if e:
        t2 = y2
    #
    if x and (not y):
      t1 = t
    elif y and (not x):
        t2 = t
    #
    if t[2]["label"]==u:
        t1 = t
    if t[2]["label"]==v:
        t2 = t
    return [x,y,t,t1,t2]


# Gives the probability that vertex labeled v is
# below vertex labeled u
def compare(t,u,v):
    [a,b,c,d,e] = subtree(t,u,v)
    if not (a and b):
        print "This tree does not have those vertices!"
        return 0
    if (c[2]["label"]==u):
        return 1.0
    if (c[2]["label"]==v):
        return 0.0
    tu = d
    tv = e
    usize = d[2]["leaves_below"]-1
```

```
    vsize = e[2]["leaves_below"]-1
    x = rankprob(tu,u)
    y = rankprob(tv,v)
    for i in range(len(x),usize+2):
         x+=[0]
    xcumulative = [0]
    for i in range(1,len(x)):
        xcumulative+=[xcumulative[i-1]+x[i]]
    rp = [0]
    for i in range(1,len(y)):
        rp+=[0]
        for j in range(1,usize+1):
            a = y[i]*nchoose(i-1+j,j)*nchoose(vsize-i+usize-j,
              usize-j)*xcumulative[j]
         rp[i]+=a
    tot = nchoose(usize+vsize,vsize)
    return sum(rp)/float(tot)
```

Appendix C

Primate Supertree



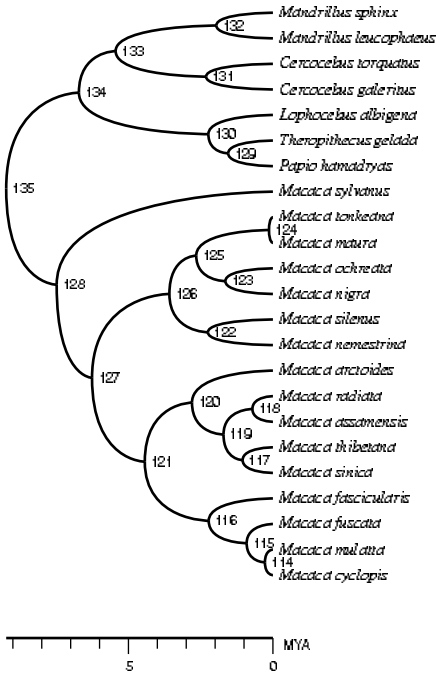


Figure C.1: Primate Supertree - Figure 3

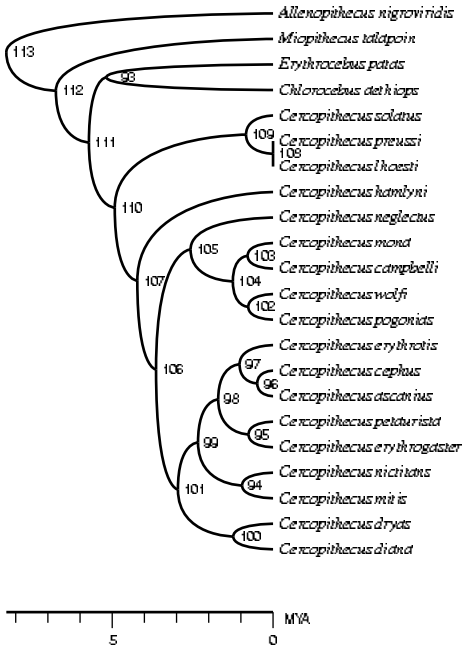


Figure C.2: Primate Supertree - Figure 4

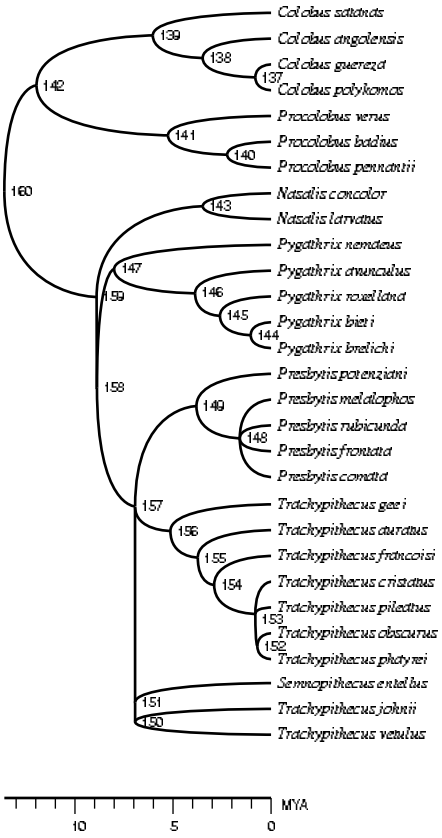


Figure C.3: Primate Supertree - Figure 5

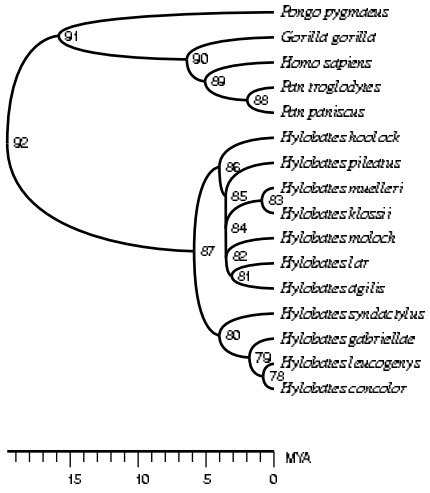


Figure C.4: Primate Supertree - Figure 6

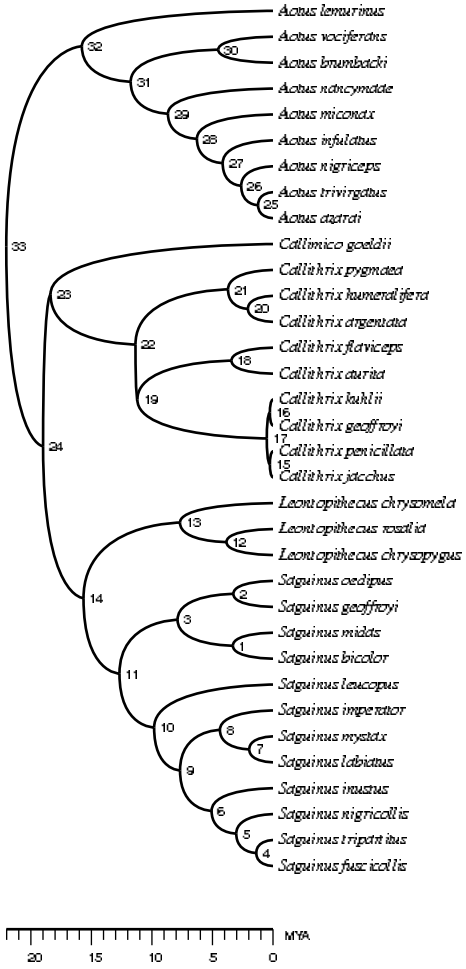


Figure C.5: Primate Supertree - Figure 7

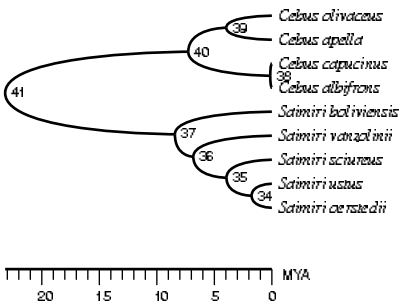


Figure C.6: Primate Supertree - Figure 8

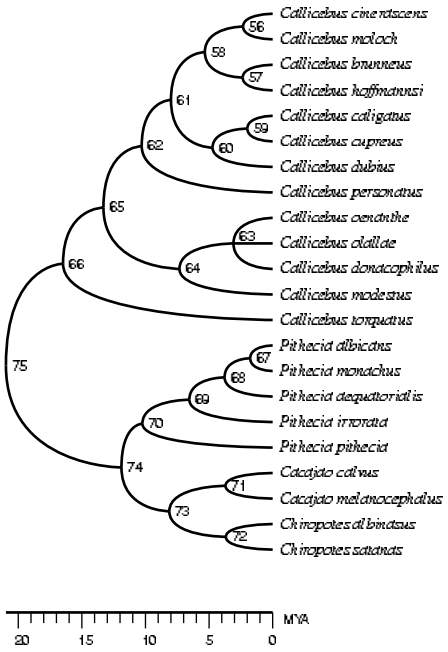


Figure C.7: Primate Supertree - Figure 9

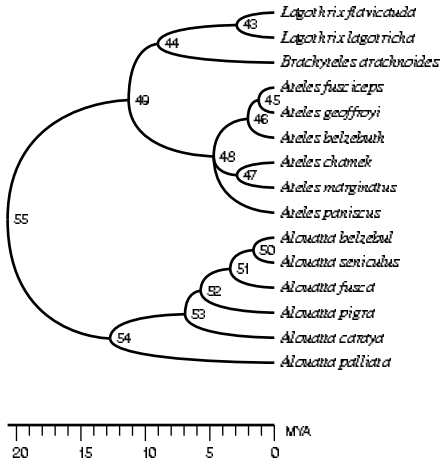


Figure C.8: Primate Supertree - Figure 10

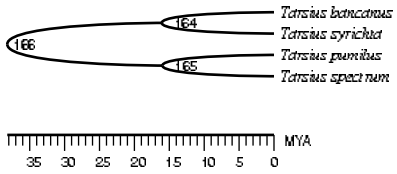


Figure C.9: Primate Supertree - Figure 11

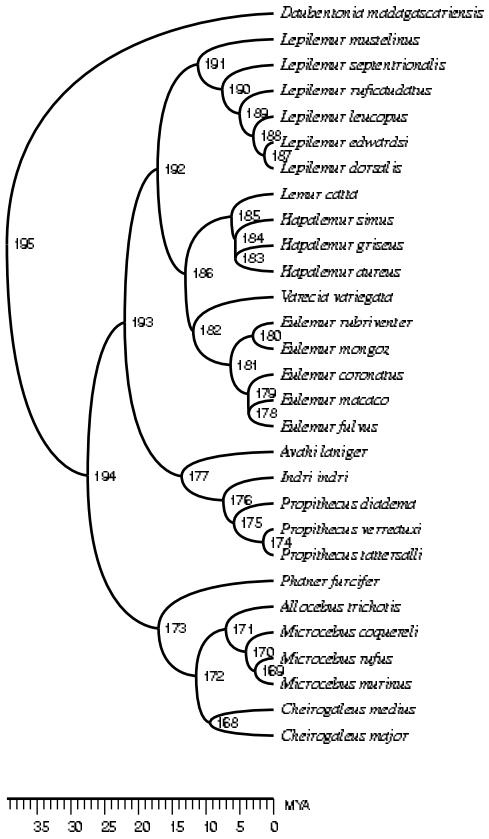


Figure C.10: Primate Supertree - Figure 12

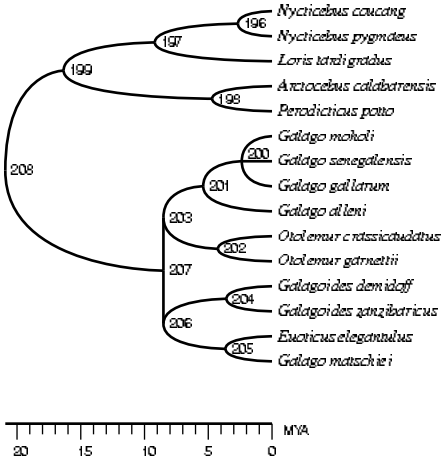


Figure C.11: Primate Supertree - Figure 13

Bibliography

[1] D. Aldous and R. Pemantle, editors. *Random discrete structures*, volume 76 of *The IMA Volumes in Mathematics and its Applications*. Springer-Verlag, New York, 1996. Papers from the workshop held in Minneapolis, Minnesota, November 15–19, 1993.

[2] B. R. Baum. Combining trees as a way of combining data sets for phylogenetic inference, and the desirability of combining gene trees. *Taxon*, 41(1):3–10, 1992.

[3] I. N. Bronstein, K. A. Semendjajew, G. Musiol, and H. Mühlig. *Taschenbuch der Mathematik*. Verlag Harri Deutsch, Thun, expanded edition, 2001.

[4] J. K. M. Brown. Probabilities of evolutionary trees. *Syst. Biol.*, 43(1):78–91, 1994.

[5] A. W. F. Edwards. Estimation of the branch points of a branching diffusion process. (With discussion.). *J. Roy. Statist. Soc. Ser. B*, 32:155–174, 1970.

[6] B. S. Everitt. *The Cambridge dictionary of statistics*. Cambridge University Press, Cambridge, 1998.

[7] D. J. Ford. Probabilities on cladograms: introduction to the alpha model. *Manuscript*, 2005.

[8] E. F. Harding. The probabilities of rooted tree-shapes generated by random bifurcation. *Advances in Appl. Probability*, 3:44–77, 1971.

[9] A. McKenzie. *Stochastic Speciation Models for Evolutionary Trees*. PhD thesis, University of Canterbury, 2000.

[10] A. McKenzie and M. Steel. Distributions of cherries for two models of trees. *Math. Biosci.*, 164(1):81–92, 2000.

[11] I. Pinelis. Evolutionary models of phylogenetic trees. *Roy. Soc. Lond. Proc. Ser. Biol. Sci.*, 270(1522):1425–1431+15, 2003. With an electronic appendix [DOI 10. 1098 spb. 2003. 2374].

[12] M. Ragan. Phylogenetic inference based on matrix representation of trees. *Mol. Phylogenet. Evol.*, 1:53–58, 1992.

[13] S. M. Ross. *Stochastic processes*. Wiley Series in Probability and Statistics: Probability and Statistics. John Wiley & Sons Inc., New York, second edition, 1996.

[14] C. Semple and M. Steel. *Phylogenetics*, volume 24 of *Oxford Lecture Series in Mathematics and its Applications*. Oxford University Press, Oxford, 2003.

[15] M. Steel and A. McKenzie. Properties of phylogenetic trees generated by Yule-type speciation models. *Math. Biosci.*, 170(1):91–112, 2001.

[16] R. A. Vos and A. O. Mooers. A dated MRP supertree for the order primates. *Manuscript*.

[17] Wikipedia. http://en.wikipedia.org/wiki/.

[18] G. U. Yule. A mathematical theory of evolution: based on the conclusions of Dr. J.C. Willis. *Philos. Trans. Roy. Soc. London Ser. B*, 213:21–87, 1924.

[19] D. Zwillinger, S. G. Krantz, and K. H. Rosen, editors. *CRC standard mathematical tables and formulae*. CRC Press, Boca Raton, FL, 30th edition, 1996.

Index