2019

UNSUPERVISED LEARNING IN PHYLOGENOMIC ANALYSIS OVER THE SPACE OF PHYLOGENETIC TREES

Qiwen Kang
University of Kentucky, kangqw@hotmail.com
Digital Object Identifier: https://doi.org/10.13023/etd.2019.189

STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained needed written permission statement(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine) which will be submitted to UKnowledge as Additional File.

I hereby grant to The University of Kentucky and its agents the irrevocable, non-exclusive, and royalty-free license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless an embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student's advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student's thesis including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Qiwen Kang, Student

Dr. Ruriko Yoshida, Major Professor

Dr. Constance Wood, Director of Graduate Studies

UNSUPERVISED LEARNING IN PHYLOGENOMIC ANALYSIS OVER THE SPACE OF PHYLOGENETIC TREES

DISSERTATION

A dissertation submitted in partial
fulfillment of the requirements for
the degree of Doctor of Philosophy
in the College of Arts and Sciences
at the University of Kentucky

By
Qiwen Kang
Lexington, Kentucky

Director: Dr. Ruriko Yoshida, Professor of Statistics
Lexington, Kentucky
2019

ABSTRACT OF DISSERTATION

UNSUPERVISED LEARNING IN PHYLOGENOMIC ANALYSIS OVER THE SPACE OF PHYLOGENETIC TREES

A phylogenetic tree is a tree to represent an evolutionary history between species or other entities. Phylogenomics is a new field intersecting phylogenetics and genomics and it is well-known that we need statistical learning methods to handle and analyze a large amount of data which can be generated relatively cheaply with new technologies. Based on the existing Markov models, we introduce a new method, `CURatio`, to identify outliers in a given gene data set. This method, intrinsically an unsupervised method, can find outliers from thousands or even more genes. This ability of analyzing large amounts of genes (even with missing information) makes it unique in many parametric methods. At the same time, the exploration of statistical analysis in high-dimensional space of phylogenetic trees has never stopped, many tree metrics are proposed to statistical methodology. *Tropical metric* is one of them. We implement a MCMC sampling method to estimate the principal components in a tree space with tropical metric for achieving dimension reduction and visualizing the result in a 2-D tropical triangle.

KEYWORDS: Evolutionary models, Gene trees, Phylogenomics, MCMC, Tropical geometry

Author's signature: Qiwen Kang

Date: May 6, 2019

UNSUPERVISED LEARNING IN PHYLOGENOMIC ANALYSIS OVER THE SPACE OF PHYLOGENETIC TREES

By

Qiwen Kang

Director of Dissertation: Ruriko Yoshida

Director of Graduate Studies: Constance Wood

Date: May 6, 2019

ACKNOWLEDGMENTS

This dissertation would not have been finished without the support of Dr. Ruriko Yoshida. As my advisor and mentor, she has provided me professional and personal guidance and taught me a lot about a good scientist both in research and life.

I am grateful to each of my dissertation committee members: Dr. William Griffith, Dr. Arnold Stromberg, Dr. Christopher Schardl and Dr. Katherine Thompson. Their suggestions and insights encourage me to keep exploring and moving forward. In addition, I also would like to thank Dr. Neil Moore for his guidance of programming and research.

I would like to thank Dr. Arnold Stromberg and Dr. Richard Kryscio for financially supporting me during these years, especially thank Dr. Richard Kryscio for his patient guidance in statistical consultation and clinical trial knowledge.

I also appreciate my parents, Yunxing Kang and Ping Gao, and my girlfriend, Yiliang Xu, for their unbelievable support. They are the most important people for me in this world.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1 Introduction

1.1 Phylogenetic Tree

Phylogenetics

Phylogenetics is a study of development or evolutionary history and relationship of a group of organisms. The basic idea is to compare the characteristics of species and to consider that similar species are genetically similar. Usually this relationship between species with a common ancestor can be represented by a *phylogenetic tree*. In the past, biologists used morphological characters collected from living or fossilized organisms, to infer phylogenetic trees. Then, phylogenentic trees with stochastic processes and combinatorics are applied to genetic data, such as DNA/RNA, protein sequences. But in recent decades, with the accumulation of genetic sequence data and cost reduction of generating genome data, traditional methods can no longer meet the needs of analysis. Researchers need some new approaches to deal with hundreds of thousands of phylogenetics trees. In addition, it has been proved that likelihood-based tree reconstruction on a concatenation of alignments can be positively misleading [53].

Generic Notation

A *phylogentic tree* $T = (V, E)$ is an acyclic, directed graph, which consists of a finite set $V(T)$ of *nodes*, also called *vertices*), and a finite set $E(T)$ of distinct unordered pairs of distinct elements of $V(T)$ called *edges*. In a phylogenetic tree, a node represents the taxonomic unit. Specifically, nodes with degree ≥ 2 are *interior nodes* referred as extinct or hypothetical taxonomic units; nodes with degree $= 1$ appears at the tips of a tree, also called *leaves*. An edge connecting to a leaf is called an *external edge*; otherwise, it is called an *internal edge*.

A *rooted tree* is a tree in which a node has been designated as the root node, and thus the direction of ancestral relationships is determined. Correspondingly, an *unrooted tree* has no root. The fastest way to generate an unrooted tree is simply

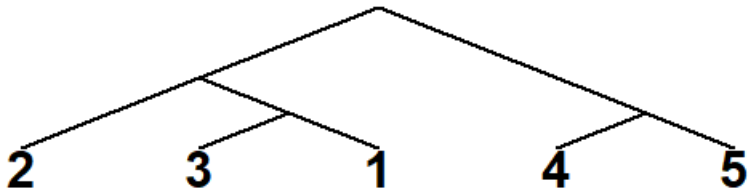

Figure 1.1: Binary tree

omitting the root of a rooted one. A *bifurcating tree* is a tree whose edges can be split into at most two edges (Figure 1.1). In contrast, a *multifurcating tree* may have more than two edges at interior vertices. If we assign distinct values to leaves, the tree *topology* (i.e. the configuration of tree shape and combination of leaves) will vary. For a labeled bifurcating tree, given the number of *leaves* N, the number of possible tree topologies is $(2N-3)!!$ for a rooted binary and $(2N-5)!!$ for an unrooted tree.

1.2 Evolutionary Model

In actual research, the raw data are sequences instead of the tree format data. This requires us to use certain algorithms to construct phylogenetic trees. The advantage of using trees is that in addition to the information contained in the DNA sequence, we explore the relationship between the sequences. *Distance based methods*, which is a family of phylogenetic reconstruction methods, was introduced in [10] and in [17]. The basic idea is to first define a distance between gene sequences and then construct a tree that fits the observed data as far as possible based on this distance. To some extent, such methods come from traditional clustering algorithms. This shift from nucleotide to distance naturally leads to a loss of genetic information during this transition. However, from the results of computer simulations, the losses among it are remarkably small [71].

There are many ways to define distance between two leaves. The simplest distance is *p distance*, also called *Hamming distance*, defined as the frequency of different base pairs between two sequences. In addition, the most popular method is to use an independent continuous Markov probability model to describe the changes between nucleotides. Here we mainly use the memoryless nature of the Markov chain and DNA sequences as an example: a nucleotide in a DNA sequence jumps from one state, which refers to nucleotide with different bases, to another depending only on the current state, regardless of where the current state comes from.

A continuous-time Markov chain with a finite discrete state space Σ can be used to model evolutionary history. The dynamic behavior under this evolutionary model is determined by the initial character state and a *transition rate matrix*, typically denoted Q, which describes the rates at which the different types of substitution occur. Many classes of transition rate matrices have been proposed, each of which makes different assumptions about the probabilistic nature of molecular evolution.

The General Time Reversible Model

Let $\pi_a,\ a \in \Sigma,\ \sum_a \pi_a = 1$, be the stationary distribution of the Markov chain and let $\theta_{ab} > 0,\ a, b \in \Sigma$ be parameters; for example, $\Sigma = \{A, G, C, T\}$. In Markov chain, the time-reversibility equation is:

$$\pi_i P(i|j,t) = \pi_j P(j|i,t), \qquad \text{for all pairs of states i, j} \tag{1.1}$$

where P is the transition matrix and $\mathbb{P}(j|i,t) = P[X_t = j|X_0 = i]$. This assumption is biologically unreasonable, but it will bring about easy mathematical calculations [71]. Then, the General Time Reversible (GTR) model has substitution rate matrix:

$$Q = \begin{bmatrix} \cdot & \theta_{AG}\pi_G & \theta_{AC}\pi_C & \theta_{AT}\pi_T \\ \theta_{AG}\pi_A & \cdot & \theta_{GC}\pi_C & \theta_{GT}\pi_T \\ \theta_{AC}\pi_A & \theta_{GC}\pi_G & \cdot & \theta_{CT}\pi_T \\ \theta_{AT}\pi_A & \theta_{GT}\pi_G & \theta_{CT}\pi_C & \cdot \end{bmatrix} \tag{1.2}$$

where the diagonal elements are such that each row sums to zero.

The behavior of a continuous-time Markov process on a state space Σ is governed by the transition rate matrix Q. The off-diagonal elements of Q represent the rates for the exponentially distributed variables that describe the amount of time that elapses before a particular type of base substitution occurs. The ij-th element of Q represents the rate at which characters in the i-th state are replaced with the j-th state. We use this rate matrix Q to compute a transition probability matrix $P(t)$ for evolutionary time $t > 0$. This probability matrix gives the probability that a character in state i at the present time will be in state j at an evolutionary time $t > 0$. Let X_t denote the state of a character site at time t from the present state; for any $i, j \in \Sigma$. The probability matrix is related to the rate matrix by the matrix exponential,

$$P(t) = \exp(tQ).$$

Since the model is reversible, we can use this matrix to look up the likelihood of observing a particular pair of characters in a sequence alignment, assuming the sequences are separated by time $t > 0$. In addition, we also assume that the sites in the alignment are independent; then the likelihood of the entire alignment is the product of the individual site likelihoods. The overall rate of substitution and the passage of time cannot be inferred separately without imposing additional assumptions and it is only possible to estimate their product. Thus, the *evolutionary branch length* $t > 0$ measures the mean number of substitution events expected to occur per site.

For the number of leaves $n = 2$, specifically $P(j|i, t)$ is the probability of a state i being substituted by a state j over an edge length t. Then the likelihood of two aligned sequences x_u^1, x_u^2, where u is the uth site in the alignment, is given by

$$L(t) = \prod_u P(x_u^2|x_u^1, t).$$

For the sake of numerical stability, it is more common to work with the log-likelihood,

$$l(t) = \log L(t) = \sum_u \log P(x_u^2|x_u^1, t). \tag{1.3}$$

Note that since $P(x_u^2|x_u^1, t)$ is an exponential family, $L(t)$ is also an exponential family. Thus, $l(t)$ in (1.3) is the sum of the exponential terms of $P(x_u^2|x_u^1, t)$, which is

proportional to the negative of the total branch length (this can be derived from formula (2) in [22]).

In general, for $n \geq 3$, let $\mathcal{T}_n$ be the space of all unrooted phylogenetic trees with n leaves. Let $\mathbf{t} = (t_i | i \in E(T))$ be a vector representation of branch lengths of a tree T. Let $x_u^1 \cdots x_u^n$ denote the residues at the uth site of n sequences (leaves) $\mathcal{D} = x^1 \cdots x^n$. The likelihood function of observing $\mathcal{D}$ given the tree topology τ and $\mathbf{t}$ is:

$$L(\mathcal{D}|\tau, \mathbf{t}) = L(x^1 \cdots x^n | \tau, \mathbf{t}) \tag{1.4}$$

$$= \prod_u P(x_u^1 \cdots x_u^n | \tau, \mathbf{t}) \tag{1.5}$$

$$= \prod_u \sum_{a^{n+1}, a^{n+2}, \cdots, a^{2n-1}} \pi_{a^{2n-1}} \times \prod_{i=n+1}^{2n-2} P(a^i | a^{\alpha(i)}, t_i) \prod_{i=1}^{n} P(x_u^i | a^{\alpha(i)}, t_i) \tag{1.6}$$

where $a^{\alpha(i)} \in \Sigma$, ranging over all extensions of the input data $\mathcal{D} = x^1 \cdots x^n$ to the internal nodes of T, denotes the assigned state at node $\alpha(i) \in T$; $\alpha(i)$ is defined as the node at the top of the edge i and π_a is the nucleotide equilibrium frequency implied by the evolutionary model. The sum in equation (1.6), which is a modification in [13], is over all possible states of residues a^k to internal nodes k where $k \in \mathbb{Z} : k \in [n+1, 2n-1]$. Similar to the case of $n = 2$, since P^i is an exponential family, $\prod_{i=n+1}^{2n-2} P(a^i | a^{\alpha(i)}, t_i) \prod_{i=1}^{n} P(x_u^i | a^{\alpha(i)}, t_i)$ is also an exponential family.

The Jukes-Cantor Model

Compared with the GTR model with high degree of freedom, the Jukes-Cantor (JC) model is a simplified version of it and it is also the simplest model of DNA evolution. In JC model, we assume that the mutation rates between all states are the same and that there are equal base frequencies ($\pi_A = \pi_G = \pi_C = \pi_T$). This assumption allows us to reduce the six parameters in the GTR model to one parameter, the mutation

rate μ. In this way, we have the JC model substitution rate matrix as follows:

$$Q = \begin{bmatrix} -3\lambda & \lambda & \lambda & \lambda \\ \lambda & -3\lambda & \lambda & \lambda \\ \lambda & \lambda & -3\lambda & \lambda \\ \lambda & \lambda & \lambda & -3\lambda \end{bmatrix}.$$

The probability matrix is related to the rate matrix by the matrix exponential:

$$P(t) = \begin{bmatrix} p_0(t) & p_1(t) & p_1(t) & p_1(t) \\ p_1(t) & p_0(t) & p_1(t) & p_1(t) \\ p_1(t) & p_1(t) & p_0(t) & p_1(t) \\ p_1(t) & p_1(t) & p_1(t) & p_0(t) \end{bmatrix}, with \begin{cases} p_0(t) = \frac{1}{4} + \frac{3}{4}e^{-4\lambda t} \\ p_1(t) = \frac{1}{4} - \frac{1}{4}e^{-4\lambda t} \end{cases}.$$

Specifically, the likelihood of two aligned sequences x_u^1, x_u^2 with length n, where u is the uth site in the alignment, based Jukes-Cantor model is given by

$$L(t) = \prod_u P(x_u^2 | x_u^1, t) \tag{1.7}$$

$$= p_0^m(t) p_1^{n-m}(t) \tag{1.8}$$

$$= (\frac{1}{4} + \frac{3}{4}e^{-4\lambda t})^m (\frac{1}{4} - \frac{1}{4}e^{-4\lambda t})^{n-m} \tag{1.9}$$

where m is the number of same states at the corresponding site of the two sequences.

1.3 Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) is a statistical sampling method for obtaining a sequence of random observations. It is a powerful and general method which allows us to sample from a probability distribution P when exact inference is intractable. In fact, this method originated from physics [44] and began to be significantly popular around 1990s.

Basically, in the *Metropolis* algorithm [43], the proposal distribution q must be symmetric; in other words, $q(x|y) = q(y|x)$ for all values of x and y. And then, if we can find a function f which is proportional to P, we can calculate the acceptance ratio $\alpha = \frac{f(y)}{f(x)}$ where y is sampled from the distribution $q(y|x)$ for next step. After

this, we generate a uniform random number u on $[0, 1]$ and accept the candidate y if $u \leq \alpha$; otherwise, we stay at x if $u > \alpha$.

The insights behind this method is to construct a Markov Chain whose equilibrium distribution equal to the probability distribution P. After many steps (probably hundreds or even thousands steps), the sampling distribution would converge to the stationary distribution and we will actually sample from the desired distribution P.

For more details, suppose we have the given probability distribution $\pi(x)$ and the transition matrix P ($p(i, j)$ is the probability of transition from state i to state j). Normally, we will have:

$$\pi(i)p(i, j) \neq \pi(j)p(j, i).$$

Thus, we would like to do some modification on this inequality to make it a equality. Here we add a accept rate $\alpha(i, j)$ into the inequality and we hope we could have:

$$\pi(i)p(i, j)\alpha(i, j) = \pi(j)p(j, i)\alpha(j, i). \tag{1.10}$$

To make it, a simple solution for the equality 1.10 is:

$$\alpha(i, j) = \pi(j)p(j, i) \tag{1.11}$$

$$\alpha(j, i) = \pi(i)p(i, j). \tag{1.12}$$

After we know $\alpha(i, j)$ and $\alpha(j, i)$, the equation 1.10 is established and we have:

$$\pi(i) \underbrace{p(i, j)\alpha(i, j)}_{\text{p'(i,j)}} = \pi(j) \underbrace{p(j, i)\alpha(j, i)}_{\text{p'(j,i)}}. \tag{1.13}$$

Finally, we construct a Markov chain from a random one with transition matrix P to another one with transition matrix P' which has the desired distribution as its equilibrium distribution.

1.4 Overview of Dissertation

The remainder of the dissertation is organized as the following. In Chapter 2, I introduce a new method, `CURatio`, for detecting outliers in a given gene data set. Under the relationship between species trees and gene trees, I defined a test statistic

by taking the ratio of total branch lengths of a tree in two scenarios caused by whether I constrained the tree topology using a consensus tree or not. I successively clustered gene trees into different groups with this test statistic using JC model as the measure of distance. Furthermore, I show our method has a better performance than `KDETREES` [68] in most cases using ROC to compare the simulation result. In addition, `CURatio` may have a more accurate result and a very wide application prospect since it is a parametric method compared to the non-parametric method `KDETREES`.

In Chapter 3, I implemented MCMC Metropolis-Hastings method with *tropical principal components* defined by Yoshida et. al. [73] to increase the efficiency and reduce running time of finding the tropical principal components. Since the proposal distribution at each step is uniform distribution, they are canceled by each other at the numerator and denominator such that the acceptance ratio will have same format as the ratio in *Metropolis* algorithm which leads to a slow convergence rate. To solve that issue, we did a small modification in the algorithm: remove one tree from the data set after each step. We also visualize the result and it is clear that the given tree data are clustered into different cells using tropical MCMC method. We have published a *R* package, *tropPCA* , includes all the programs about this tropical MCMC and some basic functions of tropical principal components on Github: https://github.com/QiwenKang/tropPCA.

Chapter 2 CURatio: Genome-wide Phylogenomic Analysis Method Using Ratios of Total Branch Lengths

Abstract

Evolutionary hypotheses provide important underpinnings of biological and medical sciences, and comprehensive, genome-wide understanding of evolutionary relationships among organisms are needed to test and refine such hypotheses. Theory and empirical evidence clearly indicate that phylogenies (trees) of different genes (loci) should not display precisely matching topologies. The main reason for such phylogenetic incongruence is reticulated evolutionary history of most species due to meiotic sexual recombination in eukaryotes, or horizontal transfers of genetic material in prokaryotes. Nevertheless, many genes should display topologically related phylogenies, and should group into one or more (for genetic hybrids) clusters in poly-dimensional "tree space". Unusual evolutionary histories or effects of selection may result in "outlier" genes with phylogenies that fall outside the main distribution(s) of trees in tree space. We present a new phylogenomic method, `CURatio`, which uses ratios of total branch lengths in gene trees to help identify phylogenetic outliers in a given set of ortholog groups from multiple genomes. An advantage of `CURatio` over other methods is that genes absent from and/or duplicated in some genomes can be included in the analysis. We conducted a simulation study under the coalescent model, and showed that, given sufficient species depth and topological difference, these ratios are significantly higher for the "outlier" gene phylogenies. Also, we applied `CURatio` to a set of annotated genomes of the fungal family, Clavicipitaceae, and identified alkaloid biosynthesis genes as outliers, probably due to a history of duplication and loss. The source code is available at `https://github.com/QiwenKang/CURatio`, and the empirical data set on Clavicipitaceae and simulated data set are available at Mendeley `https://data.mendeley.com/datasets/mrxts7wjrr/1`.

2.1 Introduction

In recent decades the field of phylogenetics has found applications in the analysis of genomic scale data (phylogenomics). In particular, it has been applied to analyze the relationships between species and populations, genome evolution, and the evolutionary processes of speciation and molecular evolution. However, today, we can generate genomic data so cheaply and quickly that we encounter a new problem: the sheer volume of genomic data and the lack of analytical tools for working with such quantities of data.

It is well-known that incomplete lineage sorting leads to differences in phylogenetic tree topologies among gene trees [39, 24, 67, 63]. Therefore, a key issue in systematic biology is to reconstruct the evolutionary history of populations and species from numerous gene trees with varying levels of discordance [6, 14].

Even though there has been much work in discordant phylogenetic relationships [48, 61, 39, 5], it is only recently that researchers have shifted away from single gene or concatenated gene estimates of phylogeny towards these multi-locus approaches, e.g., [8, 76, 3, 21, 64]. For example, researchers have begun to consider the effect of genetic drift in producing patterns of incomplete lineage sorting and gene tree/species tree discordance, largely using coalescent theory [54, 55, 12, 36, 30, 75, 65]. Other research has addressed the reconstruction of species trees from the distribution of estimated gene trees [40, 9, 15, 45, 56, 29, 70, 31, 23].

It is well-known that several processes can reduce the correlation among gene trees, including negative or balancing selection [62], meiotic sexual recombination in eukaryotes [50], and horizontal transfers of genetic material especially in prokaryotes [51]. Such processes can strongly influence phylogenetic/species tree reconstruction from the distribution of gene trees [50, 42, 14].

In this paper we propose a method to detect outlier genes from the distribution of gene trees based on likelihood functions. Here, we focus on the problem of *discordance* among gene trees, and the distribution of gene trees as a whole. We view "typical" gene trees as samples from some distribution f (e.g., a coalescent model)

that generates gene trees as independent samples. We also suppose that there may be "atypical" outlier gene trees that in effect are sampled from some other distribution f' very different from f. We are interested in estimating the distribution f for typical gene trees, and also identifying outlier gene trees that were probably not generated by f. Trees identified as outliers can be inspected for biologically interesting properties or evolutionary histories. Also, identifying and removing outliers that violate model assumptions can improve inferences made from collections of gene trees.

Here we propose the `CURatio` method based on likelihood ratios. A likelihood ratio test is a statistical test used to compare the goodness of fit of two models: the null model and an alternative model. In this paper, the null model is the evolutionary model constrained to a fixed species tree topology and the alternative is the evolutionary model unconstrained to any fixed species tree topology. If a gene tree follows the species tree, the likelihood ratio between these models should be close to one. If it does not, this ratio should be significantly greater than one. Here we demonstrate the method on simulated data sets, as well as an empirical set from 12 genomes in the fungal family Clavicipitaceae.

2.2 Methods

Test statistics

For each gene tree, we consider the following hypotheses:

$$H_0 : \text{A gene tree with the data } \mathcal{D} \text{ is congruent to the given tree topology } \tau.$$
$$H_1 : \text{A gene tree with the data } \mathcal{D} \text{ is not congruent to the given tree topology } \tau.$$

In this paper we are testing these hypotheses using the *ratio between the total branch lengths in the constrained and unconstrained trees.*

Under the maximum likelihood estimation (MLE), branch lengths in a tree are the expected number of mutations per site in certain time period. This means that the total branch length of a tree under the MLE is the expectation of the total number of mutations per site over the certain time period.

Our objective is to test how a gene tree fits a given species tree topology. If the tree topology τ is not the "best" tree topology for the observed dataset and for a given evolutionary model, then the expected number of mutations per site would increase to fit the data to the given tree topology τ. Thus the total branch length would increase if τ is not well-fitted to the given observed data under the given evolutionary model.

Therefore, with the given data set, we infer the MLE tree T_0 under the null hypothesis H_0 by constraining the tree to have topology τ under the given model, and we infer the MLE tree T_1 under the alternative hypothesis H_1 by not constraining the tree topology (i.e., finding the optimal tree topology under the model).

$$\Lambda' = \frac{\sum_{l \in E(\tau|\mathcal{D},\mathcal{M})} l}{\sum_{l \in E(\tau^*|\mathcal{D},\mathcal{M})} l},$$

where $E(\tau|\mathcal{D},\mathcal{M})$ defines the set of edges on τ given $\mathcal{D},\mathcal{M}$. $\mathcal{M}$ is an evolutionary model, τ is the constrained tree topology, and τ^* is the MLE tree topology.

Note that Λ, $\Lambda' \geq 0$ and $\Lambda \in [0,1]$; however, Λ' can be greater than one. Also note that the stronger the evidence against H_0, the smaller Λ becomes. On the other hand, the stronger the evidence against H_0, the greater Λ' becomes.

Note that the log ratio test statistic Λ' is standardized: i.e., like the Z statistic, it does not depend on the scale. In addition, we compute each Λ' independently from each $\mathcal{D}$, and since the Λ' are standardized, we can compare them even though each gene tree is reconstructed independently from each alignment. This is a significant difference from the Shimodaira-Hasegawa (SH) [59] and approximately unbiased (AU) tests [18]. SH and AU test whether the given trees are congruent to each other by comparing likelihood functions in the same given data set $\mathcal{D}$. However, our `CURatio` test compares test statistics that are independent of scale, therefore lacking the constraints of SH and AU.

The `CURatio` method operates in the following manner: Given a set of alignments $\{A_1, \ldots, A_g\}$ for g genes on n individuals and a tree topology T for the constraint tree, we reconstruct the MLE gene trees from each alignment both constrained or unconstrained by T. Next, we calculate the ratio of total branch length of the con-

strained and the unconstrained tree. The pseudocode in Algorithm 1 summarizes this process.

Algorithm 1: CURatio

Input: A set of alignments $\{A_1, \ldots, A_g\}$ for g genes on n individuals (species) and a tree topology T for the constraint tree.

Output: A sequence of ratios $(r_1, ..., r_g)$.

1. For $i = 1, \ldots, g$, do
   a) Reconstruct the MLE gene tree T_i from an alignment A_i for $i = 1, \ldots, g$ without any constraint.
   b) Reconstruct the MLE gene tree T_i' from an alignment A_i for $i = 1, \ldots, g$ with the constraint tree topology T.
   c) Compute the total branch length b_i of T_i.
   d) Compute the total branch length b_i' of T_i'.
   e) Compute $r_i = b_i'/b_i$.
2. Return the ratios $(r_1, ..., r_g)$.

Once we have all the ratios, we compute the cut-off value P as the 95th percentile of the collection $\{r_1, ..., r_g\}$. Finally, we select the genes with ratios which are greater than P. Figure. 2.1 depicts the `CURatio` algorithm as well.

The hypothesis test is performed as follows: We compute the test statistics r_i from the observed data (alignments) A_i. Then we estimate the distribution of r_i under the null hypothesis (if we know the asymptotic distribution of r_i then we use it, but this is still an approximation). If A_i yields r_i in the rejection region, for example above the 95th percentile of the estimated distribution, then A_i is considered an outlier. The performance of this test is shown in Figure 2.2 for varying P from 0 to 1.

In addition we can use this method for the significant test by computing the test statistic of a particular gene tree (i.e. the ratio) and count how many of the gene trees which have higher ratios than the ratio of the gene tree you want to test. The proportion of this number is the estimated cut-off value P.

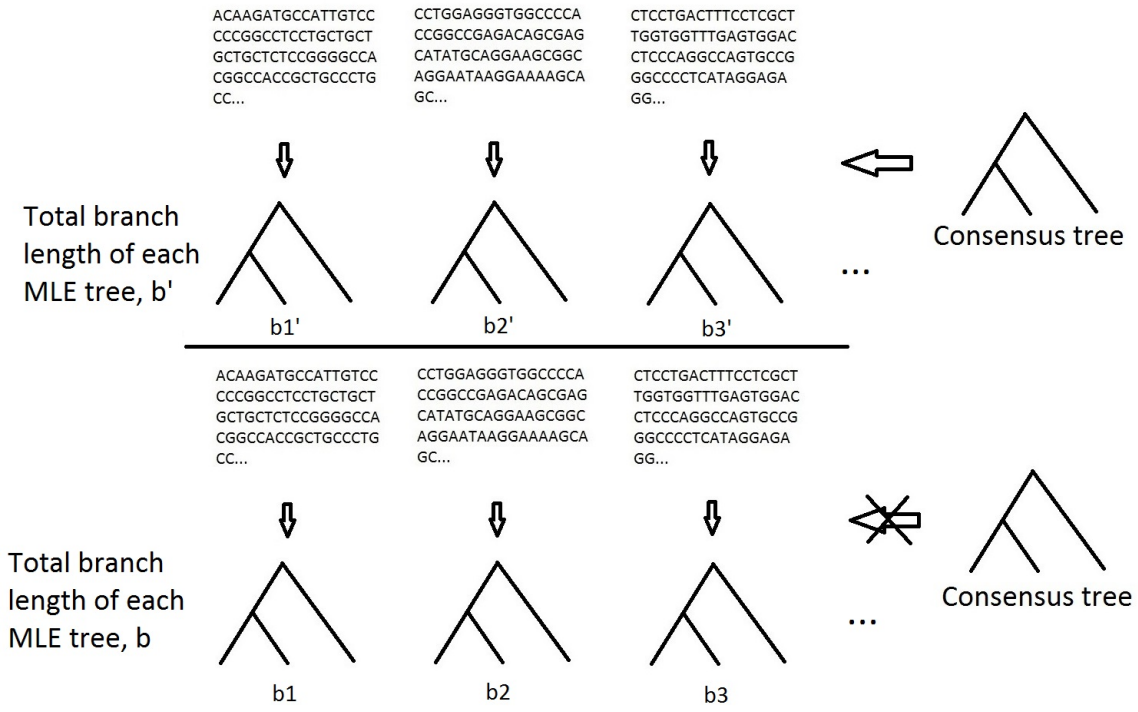

Figure 2.1: CURatio alogrithm

Empirical Data Set

Genome sequences determined for one isolate each of 12 species in the fungal family Clavicipitaceae were annotated with MAKER version 2.28[7]. The annotation of *Epichloë festucae* Fl1 (GenBank BioProject PRJNA51625) was manually refined based on cDNA and RNA-seq data sets, and the resulting gene models were included as evidence in the MAKER annotations of the other genomes. The other genomes in this study were from *Aciculosporium take* (PRJNA67241), *Atkinsonella texensis* B6155 (PRJNA274998), *Balansia obtecta* B249 (PRJNA221345), *Claviceps purpurea* 20.1 (PRJNA76493), *Epichloë amarillans* E4668 (PRJNA222148), *Epichloë inebrians* E818 (PRJNA174039), *Epichloë glyceriae* E277 (PRJNA67247), *Epichloë mollis* AL9923 (PRJNA215230), *Epichloë typhina* subsp. *poae* E5819 (PRJNA68441), *Metarhizium robertsii* ARSEF 23 (PRJNA38717) and *Periglandula ipomoeae* P4806 (PRJNA67303).

Gene models for the 12 genomes were subjected to OrthoMCL version 2.0.2 [33] to

classify ortholog groups, as described in the OrthoMCL algorithm document (`https://docs.google.com/document/d/1RB-SqCjBmcpNq-YbOYdFxotHGuU7RK_wqxqDAMjyP_w/pub`). Because OrthoMCL-derived ortholog groups may contain paralogs as well as orthologs [33], we used the refiner COCO-CL [26] to divide ortholog groups. To improve the reliability of the refinement process and the quality of generated alignments, we used a modified version of COCO-CL described in Protocol S2 of [57].

For each gene, the nucleotide sequence was identified from the start codon to the stop codon, including introns; all such gene sequences for each ortholog group were aligned by MAFFT version 6.864b [27, 28]. Finally, the ortholog groups were filtered to exclude those that had more than one representative from any genome, those that had fewer than five orthologs, and those for which the alignment had fewer than 50% non-gap characters for every gene sequence. The latter condition was imposed to filter out groups that included misannotated genes, although it also removed some ortholog groups that included pseudogenes. In total, 4266 out of 16995 ortholog groups passed the filters.

Phylogenies were determined by maximum likelihood estimation (MLE) implemented in the `R` package `ape` [49] under a Jukes-Cantor model. Those 3408 ortholog groups that had a representative from each of the 12 genomes were analyzed in a batch by CONSENSE in the `PHYLIP` version 3.2 package [16], and a 65% consensus tree was chosen as the constraint tree; this corresponded to a 70% consensus of the trees inferred under a GTR+Gamma model.

2.3 Results

Simulated data set

We conducted simulations to test `CURatio` on gene trees generated under the coalescent process,

$$Depth = Population\ Size \times C \tag{2.1}$$

where *Depth* is the depth of the species tree, *Population Size* is the effective population size (N_e) and C is a parameter, which we varied from 0.6 to 6.0 as in[19, 72].

For each value of C, we generated 2000 species trees with 10 leaves each under the Yule process, and calculated the Robinson-Foulds (RF) distance [52] for each pair of trees using the `R` package `phangorn` [58]. Then, for each RF distance 2, 4, 6, 8, 10, 12 and 14, we randomly selected ten pairs of species trees. For each selected pair we called one species tree "TreeOne" and the other "TreeTwo".

From each species tree, we generated 1000 gene trees with 10 leaves under the coalescent model using the software `Mesquite` [41], with the fixed "Population Size" equal to 10000 and the depth of the species tree determined by the parameter C (Equation 2.1). For each pair of species trees, we called the set of gene trees generated from TreeOne "GeneOne", and the set generated from TreeTwo "GeneTwo".

We then simulated DNA alignments based on these gene trees using `PAML` [69] under the Jukes-Cantor (JC) model, which is a special case of the GTR model with equal mutation rates $\frac{\mu}{4}$, where μ is the overall substitution rate.

Algorithm 2: Simulating Data Sets Process

```
for each C (from 0.6 to 6.0) do
    generate 2000 species trees randomly and calculate pairwise RF distance;
    for each RF distance (2, 4, 6, 8, 10, 12, 14) do
        randomly pick 10 pairs of species trees;
        for each pair of species trees(S1,S2) do
            generate 1000 gene trees G1 from S1;
            generate 1000 gene trees G2 from S2;
            generate 1000 alignments A1 from each tree in G1;
            generate 1000 alignments A2 from each tree in G2;
        end
    end
end
```

The first simulation produced ROC curves for comparing `CURatio` with `KDETREES`. `KDETREES` is a non-parametric method to estimate the distribution of trees and identify potential outlier gene trees which are probably not generated by this distribution; `CURatio`, on the other hand, is a parametric method. Note that `CURatio` does not fit a chi-squared distribution because it is not a traditional likelihood ratio test. Instead, potential outlier genes can be identified by those giving a value of r in a high percentile (e.g. 95th or 99th) of the distribution of r values of all the genes in the genome

for which phylogenies were determined. We used the set of alignments GeneOne and their corresponding trees as the non-outlier data set, and we used the set of alignments GeneTwo and their corresponding trees as the outlier data set. The constraint tree was the species tree corresponding to GeneOne. The process is summarized in Algorithm 2.

Algorithm 3: Summary of the simulation comparing `CURatio` and `KDETREES`. For our simulation, $m = 100$, $n = 1$

Input: A set of alignments $\{A_1, \ldots, A_g\}$ for g genes and their corresponding trees as the non-outlier data set. A set of alignments $\{B_1, \ldots, B_r\}$ for r genes and their corresponding trees as the outlier data set. A species tree, S, corresponding to the non-outlier trees,

Output: Average number of true and false outlier identifications for each method

for *each C (from 1.0 to 6.0)* **do**

- Randomly sample m alignments and their corresponding trees from the non-outlier data set;
- Randomly sample n alignments and their corresponding trees from the outlier data set;
- Detect outliers with both `CURatio` and `KDETREES`;
- Tally true and false outlier identifications for both methods;

end

We randomly selected a data set for each C value from our simulations regardless of RF distance. As shown in Figure 2.2, `CURatio` performed as well or better than `KDETREES` for C values up to 2. `KDETREES` performed better than `CURatio` at $C = 4$. For $C = 6$ the ROC curves for both methods passed close to the (0,1) point.

Our second simulation procedure is outlined in Algorithm 3. For each pair of species trees and the associated gene trees, we applied `CURatio` (Algorithm 1) four times, to obtain four sets of ratios: once on the set of alignments GeneOne against the corresponding species tree TreeOne; once on GeneOne against the other species tree; and likewise for the GeneTwo alignments. Then we used `R` to calculate Tukey's five number summary (minimum, lower-hinge, median, upper-hinge, maximum) of each of the four sets of ratios. We were particularly interested in the trend of the medians of GeneOne with TreeTwo, and GeneTwo with TreeOne, with increasing C and different RF distances between the species trees. Significant differences were

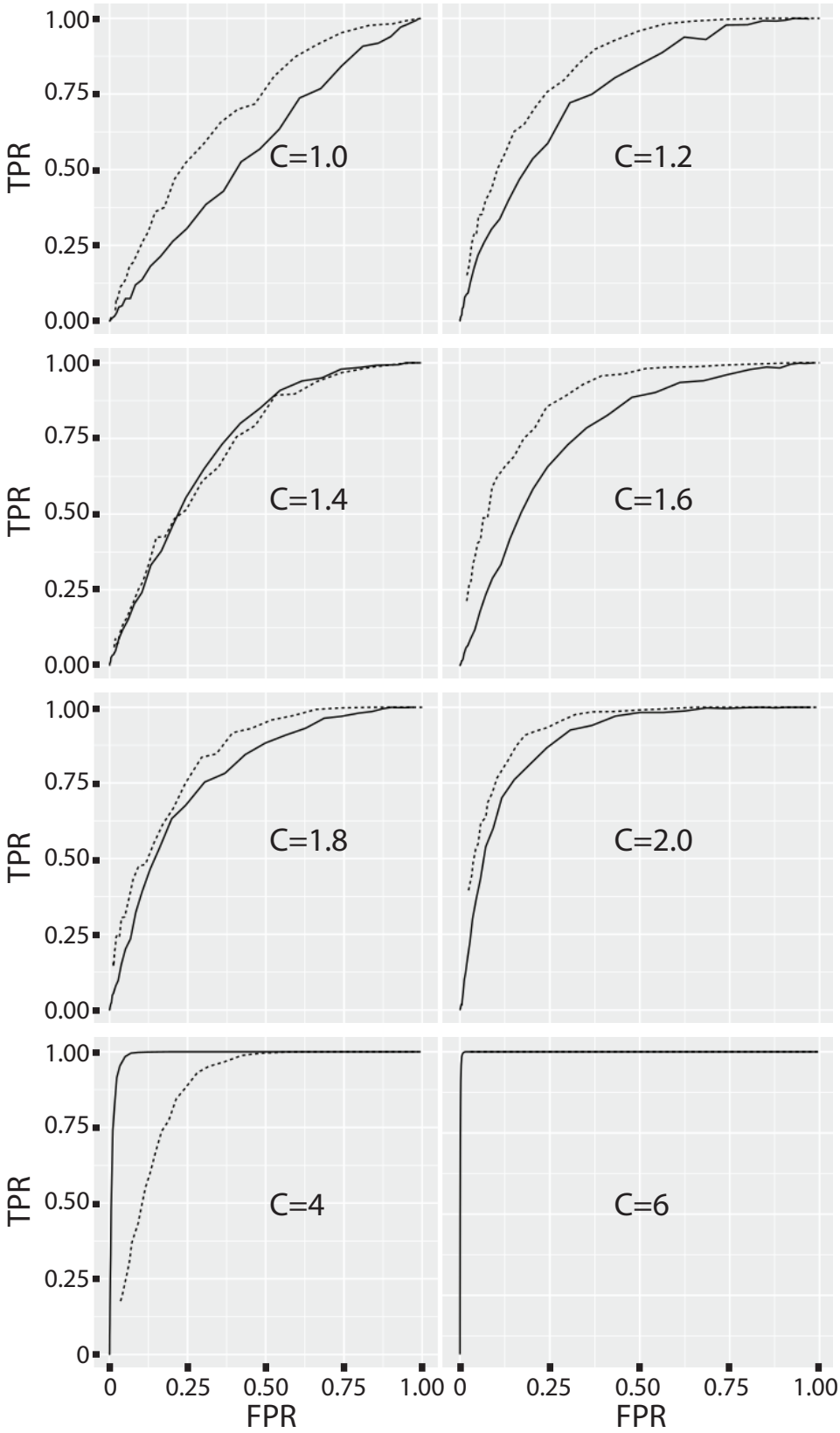

Figure 2.2: ROC curves comparing results of `CURatio` (dashed line) and `KDETREES` (solid line) as the C value is changed. TPR stands for true positive rate and FPR stands for false positive rate.

apparent at RF = 4 and high C values; at RF = 6 or higher, significant differences were also apparent for C values of 2 or less (see Figure 2.3).

Algorithm 4: LOESS Plot

Input: Two sets of alignments, A^1 and A^2, and their corresponding species trees, S_1 and S_2.
Output: The trend of medians
for *each RF distance (2, 4, 6, 8, 10, 12, 14)* **do**
  for *each combination of sets of alignments and species tree, (A^1,S_1)(A^1,S_2)(A^2,S_1)(A^2,S_2)* **do**
    Apply Algorithm 1;
    Calculate the medians.
  end
  Apply "LOESS" from R to fit a smooth curve.
end

For visualization, we applied "LOESS" from `R` on these medians, fitting a smooth curve through the points in Figure 2.3, where we can observe that both of the two ratios are greater than one. But if we use the species tree as the constraint tree, the ratio tends to be relatively close to one. Meanwhile, if we use a constraint tree with a different topology from the species tree, then the ratio tends to be greater than one.

When using the correct species tree as the constraint tree, larger values of C resulted in ratios approaching one. This was as expected because, as C gets larger, the species tree becomes taller and narrower relative to population size, so gene trees tend to follow the species tree topology more closely. Also as expected, such behavior was not apparent when the gene trees differed from the species trees, particularly at RF distances of six or greater.

An important feature of `CURatio` is that it is applicable to datasets that include ortholog groups where some taxa lack the gene, as well as ortholog groups with paralogs. For paralogs, in-paralogs arise from gene duplications on terminal branches and should not cause deviation from the constraint tree, whereas out-paralogs arise from gene duplications on internal branches and consequently differ from the constraint tree (see Figure 2.4). A simulation (see Figure 2.5) illustrates that out-paralogs can result in ratios significantly greater than one.

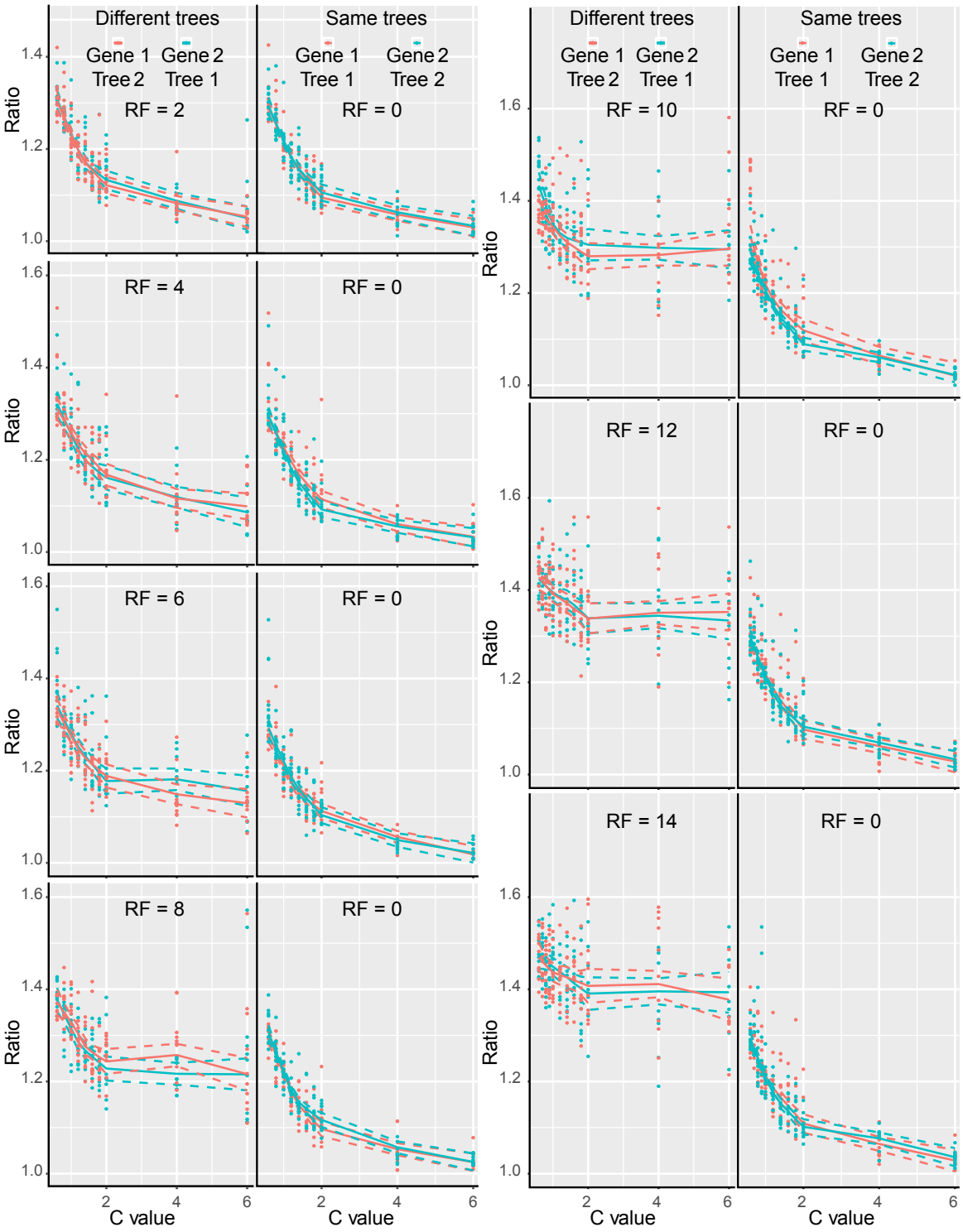

Figure 2.3: LOESS on medians of four sets of ratios fitting a smooth curve through the points. Each set contains 10 points for each C value. The area between the two dashed lines is the 95% confidence interval. When using a constraint tree with a different topology from the species tree ("Different trees" columns), the ratio tends to be greater than one. Significant differences were apparent at RF $= 4$ for high values of C; at RF ≥ 6, significant differences were also apparent for smaller C values ($C \leq 2$)

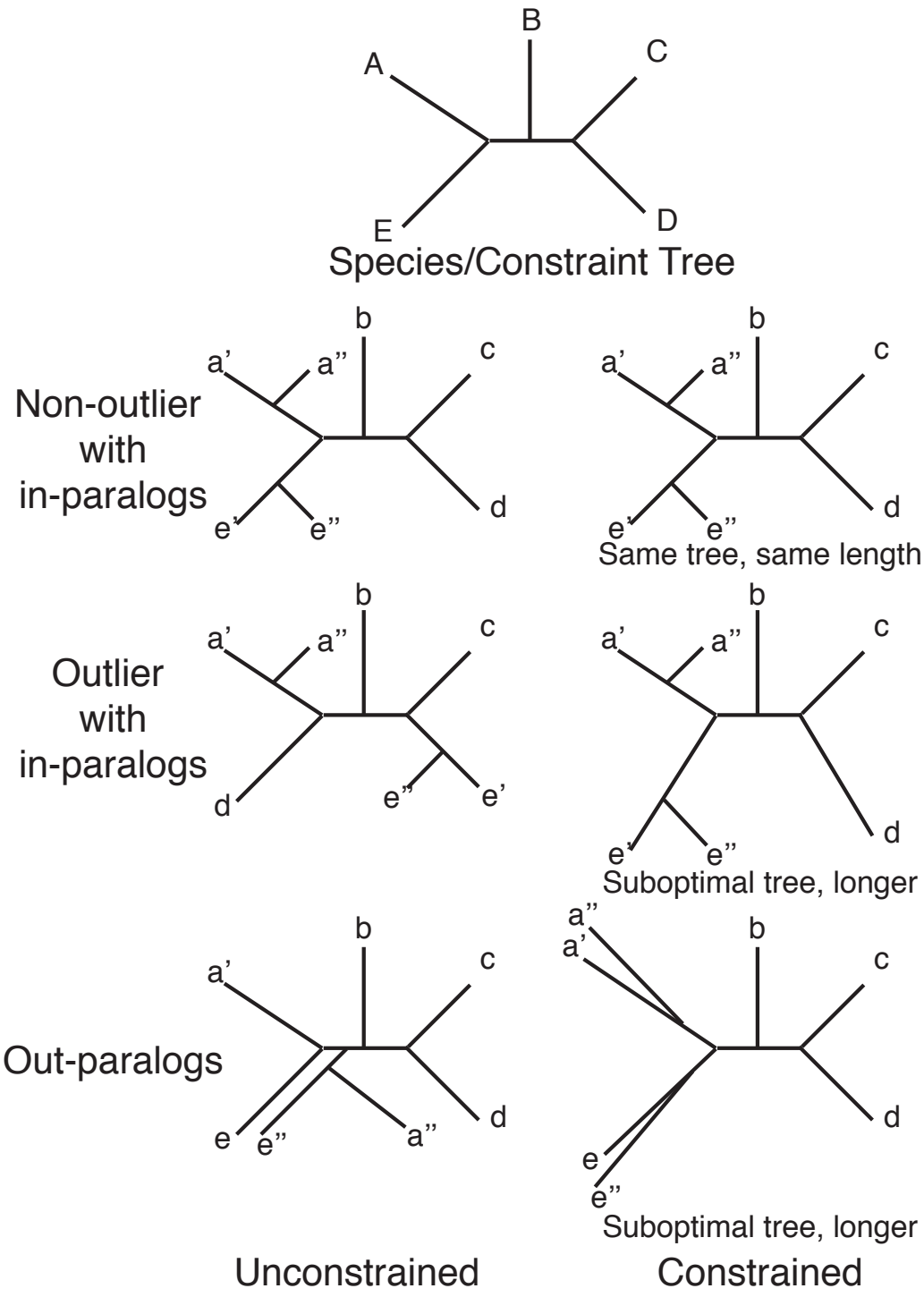

Figure 2.4: Examples of constrained and unconstrained tree configurations for ortholog groups with either in-paralogs or out-paralogs. In-paralogs arise from gene duplication on terminal branches, whereas out-paralogs arise from gene duplication in common ancestors of two or more species. For non-outlier trees the ratios of constrained to unconstrained tree lengths should be close to one, whereas for ortholog groups with outlier phylogenies and ortholog groups with out-paralogs the ratios should be greater than one.

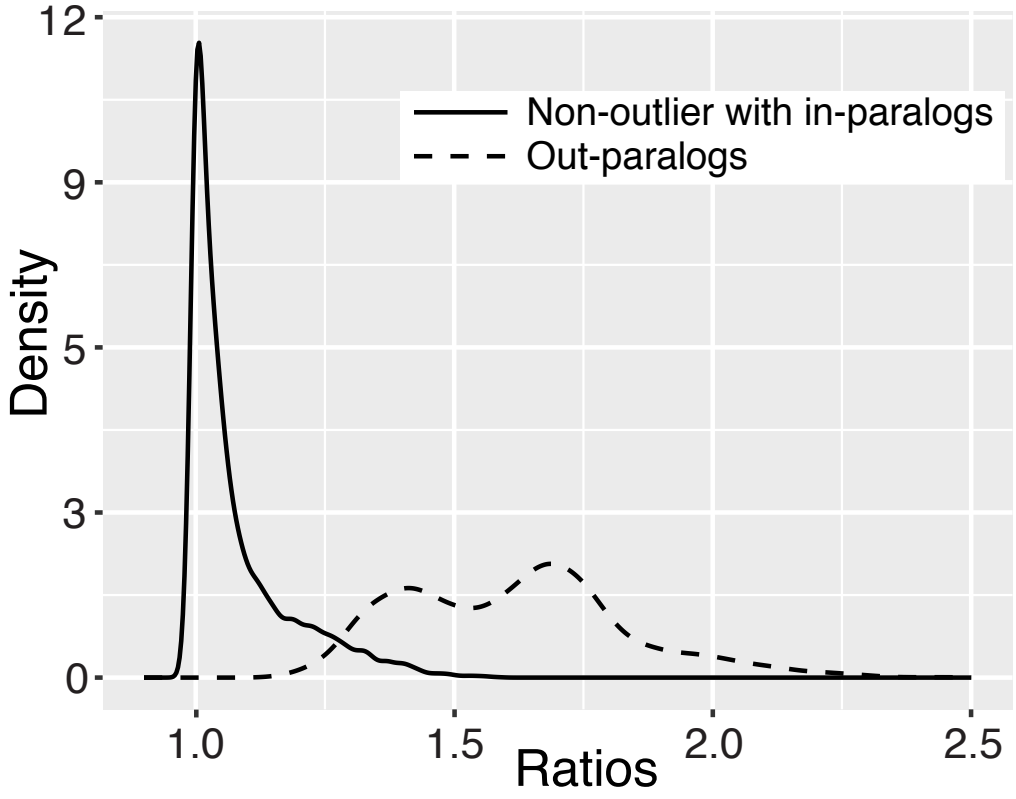

Figure 2.5: Density plots of ratios of constrained to unconstrained tree lengths for non-outlier ortholog groups with in-paralogs and ortholog groups with out-paralogs as disgrammed in Figure 2.4. The p-value of a two-sample t-test was 2.2×10^{-16}, indicating a statistically significant difference between non-outliers with in-paralog and out-paralog groups.

Analysis of an Empirical Data Set

`CURatio` was applied to ortholog groups from a set of 12 genomes of fungi in the family Clavicipitaceae; a histogram of ratios of constrained tree length to unconstrained tree length is presented in Figure 2.6. Although there was a negative trend between the ratios and the numbers of genomes containing orthologs in an ortholog group, the correlation coefficient was -0.433. Thus, there was not a strong general relationship between whether a gene was a core gene (present in all 12 genomes) or flexible gene (present in fewer than 12 genomes) and its conformity to the species tree.

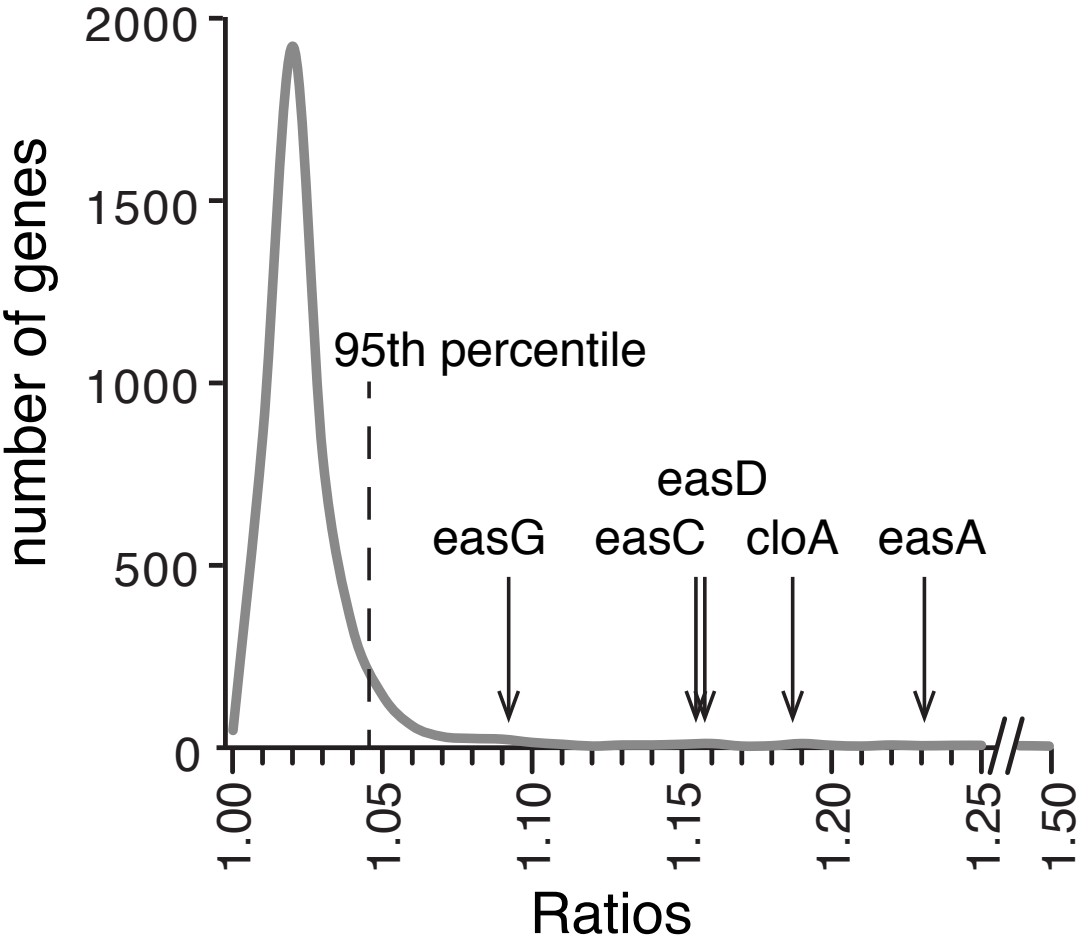

Figure 2.6: A histogram of log ratios of constrained tree length to unconstrained tree length based on the empirical data set of 4266 ortholog groups from 12 annotated fungal genomes. The lowest observed ratio was approximately 0.994. The ratios obtained for ergot alkaloid biosynthesis genes are indicated by arrows.

It has been noted previously that, in the Clavicipitaceae, phylogenies of ergot alkaloid biosynthesis (*EAS*) genes fail to match phylogenies of core housekeeping genes commonly used to infer species relationships [37, 74]. Ortholog groups for five *EAS* genes passed the filters (see Section 2.2) and were included in our analysis. All five *EAS* genes gave ratios exceeding 1.09, and were therefore considered significant outliers. This was in keeping with expectations for *EAS* genes (see Figure 2.6). Figure 2.7 compares ratios for nine core housekeeping genes and a mating type gene (*mtAC*) with those of the five *EAS* genes, *easG*, *easC*, *easD*, *cloA* and *easA*. If, instead of ratios, genes are ordered by RF values, the difference between *EAS* genes and housekeeping

genes is much less apparent. With RF = 5, *easG* is in the 52nd percentile, and with RF = 9, *easC*, *easD*, *cloA* and *easA* are in the 95th percentile. RF values for housekeeping genes ranged from 2 to 9, with *tefA*, *rpbB* and *actG* having $RF = 5$ (95th percentile), *tubP* RF = 7 (80th percentile), and *gapD* RF = 9 (95th percentile). In contrast, the housekeeping genes chosen for analysis (Figure 2.7) had ratios ranging from the 4th to the 73rd percentile, whereas the *EAS* genes all had ratios in the 99th percentile.

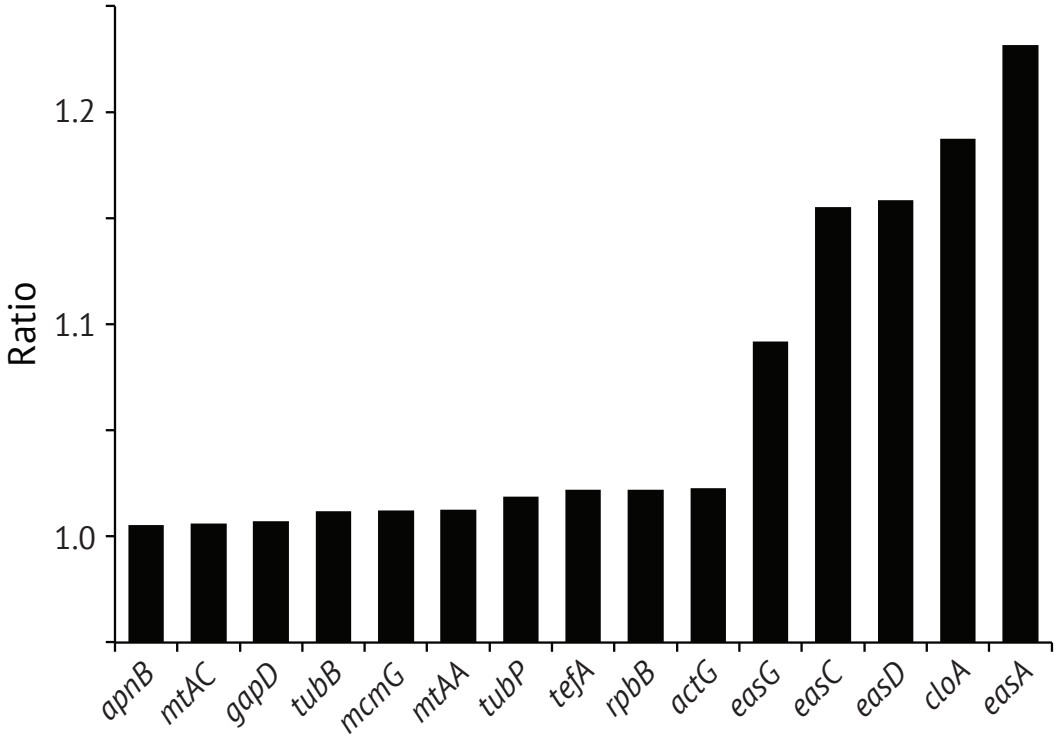

Figure 2.7: Ratios of constrained to unconstrained tree lengths for nine core housekeeping genes and a mating type gene (*mtAC*) with those of the five ergot alkaloid biosynthesis genes, *easG*, *easC*, *easD*, *cloA* and *easA*.

2.4 Discussion

Our objective was to develop a simple statistical approach to identify genes with evolutionary histories that significantly deviate from their corresponding species phylogeny, and particularly an approach that can accommodate genes that are missing or duplicated (paralogs) in some genomes. We have proposed a novel statistical method to detect outlying gene trees from a large set of gene trees, for example obtained by

whole genome analyses. For each set of orthologous genes we calculate the length of the MLE tree constrained to the postulated species tree, divided by the length of the unconstrained MLE tree to give the `CURatio` statistic. In this paper we approximate the distribution of the `CURatio` from the observations and we take ratios more than 95th percentile as outliers. Another phylogenomic use for this method is to explore relative deviations from the more common phylogenies, such as different r_i percentiles, to address questions such as whether some classes of genes tend to deviate more than others. Importantly, the `CURatio` method can be applied to gene sets in which some genes are lacking in some of the taxa, making it possible to compare such flexible genes with the species tree.

We applied the `CURatio` method to simulated data, with gene trees derived from the coalescent model, based on species trees differing by RF distances of 2 through 14, assuming $N_e = 10,000$ and various C values for population depth $= N_e \times C$. With these parameters, average ratios were significantly different for the same versus different species trees for $C \geq 0.6$ at moderate to high RF distances.

A set of genomes from Clavicipitaceae was chosen for an empirical test of `CURatio` because previous investigations of species and alkaloid gene phylogenies indicated different evolutionary histories [74]. Of the 12 genomes included, *EAS* genes were present in 10 of the genomes. The maximum number of *EAS* genes was 14, and nine *EAS* genes were shared among all 10 genomes. Despite sharing a similar topology, *easG* had a much lower RF (= 5) than the other *EAS* genes (RF = 9), simply because *easG* was not represented in all of the genomes that contained the other *EAS* genes. Nevertheless, the *EAS* genes all had ratios in the 99th percentile. Furthermore, the 10 housekeeping and mating type genes had a wide range of RF values (2 to 9), but all had ratios very close to 1.00. Given the overlap in RF values, *EAS* genes were not discoverable as outliers based on RF. In fact, RF did not correlate significantly with ratios ($R^2 = 0.0483$). The obvious reason is that RF is a purely topological measure, and some genes that gave high RF differed from the constraint tree only in short branches. Constraining such trees only slightly lengthened them.

For various reasons, only five of the 14 *EAS* genes passed the filter to be included in

the analysis (see Section 2.2). Of the excluded genes, three were present in fewer than five of the genomes, *dmaW* was duplicated in *C. purpurea* (in this run we excluded duplicated genes), the closely linked *easF* and *easE* genes were sometimes misannotated as a single gene, and the *lpsA*, *lpsB* and *lpsC* genes were not separated from other nonribosomal peptide synthetase genes by the `OrthoMCL/COCO-CL` pipeline. The stringency of the filter was deemed necessary to minimize cases of outliers originating from misannotations or incorrect inferences of orthology, but in future, consideration can be given to refining orthology searches and subsequent filters to capture a greater proportion of shared genes for the `CURatio` test. It seems likely that flexible genes were disproportionately excluded, so more inclusive representation may well affect the observed distribution of ratios. Additionally, although not included in our empirical analysis, the implementation of `CURatio` allows for inclusion of genes for which more than one ortholog (up to a user-set maximum) may occur in a taxon or genome.

It will be hard to give a specific running time function of the input size since `CURatio` is based on maximum likelihood with its NP-hardness. But in our data set, the size of each alignment would be around 40 KB. It just takes 3.1s to calculate the ratio. For the whole data set, it takes around 21 mins including 4266 alignments. All the code is running on a computer with processor Intel Core i7-6700 3.40GHz×8, memory 15.6 GB and OS type Ubuntu 17.10 64-bit.

In simulations in which the constraint tree differed from the species tree by RF distances of 2–14, we observed that tree-length ratios leveled out at 1.1–1.3 for $C = 2$ or greater. It would be of interest to derive an explicit formula for the expected ratio under a given model and number of leaves. Also of interest is the possibility of estimating population depths based on the distributions of ratios from empirical data sets.

In this paper, we choose JC model, which is a specific case of Markov models, as our evolutionary model. Markov models are popular in molecular evolution area. Its no memory feature is that the transition probabilities depends only upon the current state. This makes it natural to assume that the nucleotide sites in DNA sequence evolved independently of each other. However, such assumption is often inappropriate

in co-evolution [66]. We will discuss this situation and develop alternative models in future work.

Chapter 3 Tropical Principal Components Using Metropolis-Hastings Algorithm

Abstract

Principal component analysis is one of the most popular unsupervised learning methods for reducing the dimension of a given data set in a high-dimensional Euclidean space. However, computing principal components on a space of phylogenetic trees with fixed labels of leaves is a challenging task since a space of phylogenetic trees is not Euclidean. In 2017, Yoshida et. al. defined a notion of tropical principal component analysis and they have applied it to a space of phylogenetic trees. The challenge, however, they encountered was a long computational time for large data set.

In this paper we estimate tropical principal components in a space of phylogenetic trees using the Metropolis-Hasting algorithm. We have implemented an `R` software package to efficiently estimate tropical principal components and then we have applied it to African coelacanth genomes data set.

3.1 Introduction

Principal component analysis (PCA) is one of the most popular and robust unsupervised learning methods for reducing the dimension of a high-dimensional data set in Euclidean spaces. PCA is a statistical method that takes data points in a high dimensional Euclidean space into a lower dimensional plane which minimizes the sum of squares between each point in the data set and their orthogonal projection onto the plane. It has been used for clustering high dimensional data points for statistical analysis and it is one of the simplest and most robust ways of doing such dimension reduction in a Euclidean vector space. However, it assumes the properties of a Euclidean vector space while the space of rooted equidistant trees on n leaves, a polyhedral complex of dimension $n-2$, realized as the set of all ultrametrics is not Euclidean.

One classical way to conduct a statistical analysis on phylogenetic trees with n leaves is to map each tree to a vector in $\mathbb{R}^{\binom{n}{2}}$, for example using the *dissimilarity map*. Given any tree T of n leaves with branch length information, one may produce a corresponding *distance matrix*, $D(T)$. The distance matrix is an $n \times n$ symmetric matrix of non-negative real numbers, with elements corresponding to $d_{ij}(T)$, the sum of the branch lengths between pairs of leaves in the tree. To calculate $d_{ij}(T)$, one simply determines which edges of the tree form the path from a leaf i to a leaf j, and then sums the lengths of these branches. Since $D(T)$ is symmetric and has zeros on the diagonal, the upper-triangular portion of the matrix contains all of the unique information found in the matrix. We can vectorize T by enumerating this unique portion of the distance matrix,

$$v_d(T) := (d_{12}(T), d_{13}(T), \ldots, d_{23}(T), \ldots, d_{n-1n}(T))$$

which is called the *dissimilarity map* of a tree T and is a vector in $\mathbb{R}^{\binom{n}{2}}$. If it is clear we simply abbreviate $D(T)$ with D.

Let D be a distance matrix computed from a phylogenetic tree, that is, a non-negative symmetric $n \times n$-matrix $D = (d_{ij})$ with zero entries on the diagonal such that all triangle inequalities are satisfied:

$$d_{ik} \leq d_{ij} + d_{jk} \quad \text{for all } \ i, j, k \ \text{ in } \ [n] := \{1, 2, \ldots, n\}.$$

If a distance matrix D is computed from an equidistant tree, it is well-known that elements in D satisfy the following strengthening of the triangle inequalities

$$d_{ik} \leq \mathrm{m}ax(d_{ij}, d_{jk}) \quad \text{for all } \ i, j, k \in [n]. \tag{3.1}$$

If (3.1) holds then the metric D is called an *ultrametric*. The set of all ultrametrics contains the ray $\mathbb{R}_{\geq 0}\mathbf{1} = (a, a, \ldots, a)$, where $s \in \mathbb{R}$, spanned by the metric $\mathbf{1} = (1, 1, \ldots, 1)$, which is defined by $d_{ij} = 1$ for $1 \leq i < j \leq n$. The image of the set of ultrametrics in the quotient space $\mathbb{R}^{\binom{n}{2}}/\mathbb{R}\mathbf{1}$ is denoted $\mathcal{U}_n$ and called the *space of ultrametrics*. Therefore, we can consider the space of ultrametrics as a treespace for all possible equidistant phylogenetic trees with n leaves.

However, the space of phylogenetic trees with n leaves is not an Euclidean space. In fact, it is a union of lower dimensional polyhedral cones in $\mathbb{R}^{\binom{n}{2}}$. Therefore we cannot directly apply classical PCA to a set of gene trees. Nye showed an algorithm in [46] to compute the first order principal component over the space of phylogenetic trees of n leaves using the unique shortest connecting paths, or geodesics, defined by the $\mathrm{CAT}(0)$-metric introduced by Billera-Holmes-Vogtman (BHV) over the tree space of phylogenetic trees with fixed labeled leaves [4]. Nye in [46] used a convex hull of two points, i.e., the geodesic, on the tree space as the first order PCA. However, we could not generalize this idea for computing higher order principal components with the BHV metric because, in 2017, Lin et. al. showed that the convex hull of three points with the BHV metric over the tree space has an arbitrary dimension [35]. On the other hand the tropical metric in tree space defined by the tropical convexity in the max-plus algebra is well studied [38].

Now we turn to *tropical mathematics* [60]. This furnishes a metric and a convexity structure on the tree space which is radically different from BHV. Let $e = \binom{n}{2}$. Tropical geometry gives an alternative geometric structure on $\mathcal{U}_n$, via the graphic matroid of the complete graph [38, Example 4.2.14], i.e., $\mathcal{U}_n$ can be written as a tropical linear space under the max-plus algebra. We mostly use the max-plus algebra, so our convention is opposite to that of [38] and [47]. The connection between phylogenetic trees and tropical lines, identifying tree space with a tropical Grassmannian, has been explained in many sources, including [38, §4.3], [47, §3.5], and [60, Fact 6]. However, the restriction to ultrametrics [2, §4] offers a fresh perspective.

In 2017, Yoshida et. al. defined a notion of *tropical principal components* [73]: Tropical convex hull, i.e., tropical polytope, which minimizes the sum of squares between each point in the data set and their orthogonal projection onto the tropical polytope with the *tropical metric* d_{tr}. They have introduced a mathematical foundation on tropical principal components and they have applied it to computing tropical principal components in $\mathcal{U}_n$. However, it is not efficient to compute tropical principal components using their implementations even though the time complexity of computing tropical principal components is still unknown.

In this paper we have developed a method to estimate tropical principal components via Metropolis-Hasting algorithm and then we have applied it to coelacanths genome and transcriptome data from Liang et. al. [34]. This paper is organized as follows: In Section 3.2 we discuss the basics of tropical geometry and review the interpretation of the space of equidistant trees as a tropical linear space. Then we review the tropical principal components introduced by Yoshida et. al. In Section 3.3 we describe our algorithm and then in Section 3.5 we apply our method to the coelacanths genome data set.

3.2 Tropical Principal Components

In this section we review some basics of tropical geometry and then we review the tropical principal components developed by [73]. See [38] or [25] for more details.

In the *tropical semiring* $(\mathbb{R} \cup \{+\infty\}, \oplus, \odot)$, the basic arithmetic operations of addition and multiplication are redefined as follows:

$$a \oplus b := \min\{a, b\}, \qquad a \odot b := a + b \qquad \text{where } a, b \in \mathbb{R}.$$

The element $-\infty$ is the identity element for addition and 0 is the identity element for multiplication: for all $a \in \mathbb{R} \cup \{+\infty\}$, we have $a \oplus -\infty = a$ and $a \odot 0 = a$.

Similarly, there is another way to define addition and multiplication using maximum instead of minimum, which is called $max - plus\ semiring$ $(\mathbb{R} \cup \{-\infty\}, \oplus, \odot)$:

$$a \boxplus b := \max\{a, b\}, \qquad a \odot b := a + b \qquad \text{where } a, b \in \mathbb{R}.$$

With given scalars $a, b \in \mathbb{R} \cup \{-\infty\}$ and vectors $v = (v_1, \ldots, v_e), w = (w_1, \ldots, w_e) \in (\mathbb{R} \cup \infty)^e$, we can define tropical scalar multiplication and tropical vector addition as

$$a \odot v = (a + v_1, a + v_2, \ldots, a + v_e)$$

$$a \odot v \boxplus b \odot w = (\max\{a + v_1, b + w_1\}, \ldots, \max\{a + v_e, b + w_e\})$$

In tropical geometry we often work in the *tropical projective torus* $\mathbb{R}^e/\mathbb{R}\mathbf{1}$, where $\mathbf{1}$ denotes the all-ones vector. Given two points v, w in the tropical projective torus,

their *tropical distance* $d_{tr}(v, w)$ is defined as follows:

$$d_{tr}(v, w) \;=\; \max\big\{\, |v_i - w_i - v_j + w_j| \;:\; 1 \le i < j \le e \,\big\}, \tag{3.2}$$

where $v = (v_1, \ldots, v_e)$ and $w = (w_1, \ldots, w_e)$. This metric is also known as the *generalized Hilbert projective metric* [1, §2.2], [11, §3.3].

A subset $S \subset \mathbb{R}^e$ is said *tropically convex* if it contains the point $a \odot x \boxplus b \odot y$ for all $x, y \in S$ and all $a, b \in \mathbb{R}$. The *tropical convex hull* or *tropical polytope* $\mathrm{t}conv(V)$ of a given subset $V \subset \mathbb{R}^e$ is the smallest tropically convex subset containing $V \subset \mathbb{R}^e$. The tropical convex hull of V can be also written as the set of all tropical linear combinations

$$\mathrm{t}conv(V) = \{a_1 \odot v_1 \boxplus a_2 \odot v_2 \boxplus \cdots \boxplus a_r \odot v_r : v_1, \ldots, v_r \in V \text{ and } a_1, \ldots, a_r \in \mathbb{R}\}.$$

Any tropically convex subset S of $\mathbb{R}^e$ is closed under tropical scalar multiplication, $\mathbb{R} \odot S \subseteq S$.

Let $\mathcal{P}$ be a tropical polytope $\mathcal{P} = \mathrm{t}conv(D^{(1)}, D^{(2)}, \ldots, D^{(s)})$, where the $D^{(i)}$ are points in $\mathbb{R}^e/\mathbb{R}\mathbf{1}$. There is a projection map $\pi_{\mathcal{P}}$ sending any point D to a closest point in the tropical polytope $\mathcal{P}$ as

$$\pi_{\mathcal{P}}(D) \;=\; \lambda_1 \odot D^{(1)} \;\boxplus\; \lambda_2 \odot D^{(2)} \;\boxplus\; \cdots \;\boxplus\; \lambda_s \odot D^{(s)}, \tag{3.3}$$

where $\lambda_k = \mathrm{m}in(D - D^{(k)})$ for $k = 1, \ldots, s$. This formula appears as [38, Formula 5.2.3].

Now we review how tropical geometry connects to the space of phylogenetic trees. It is well known that all ultrametrics are tree metrics. In fact, all ultrametrics are derived from *equidistant trees*, where all leaves have the same distance to some distinguished root vertex. Furthermore, the tree metric of an equidistant tree is an ultrametric; hence ultrametrics and equidistant trees convey equivalent information.

Let L_n denote the subspace of $\mathbb{R}^e$ defined by the linear equations $x_{ij} - x_{ik} + x_{jk} = 0$ for $1 \le i < j < k \le n$. The tropicalization $\mathrm{Trop}(L_n) \subseteq \mathbb{R}^e/\mathbb{R}\mathbf{1}$ is the tropical linear space consisting of points $(v_{12}, v_{13}, \ldots, v_{n-1,n})$ such that $\max(v_{ij}, v_{ik}, v_{jk})$ is obtained at least twice for all triples $i, j, k \in [n]$.

Theorem 1 *[73] The image of $\mathcal{U}_n$ in the tropical projective torus $\mathbb{R}^e/\mathbb{R}\mathbf{1}$ coincides with Trop(L_n).*

A tropical principal component analysis defined in [73] is the tropical convex hull of s points in $\mathcal{U}_n$ minimizing the sum of distances between each point in the sample to its projection onto the convex hull. While we can generalize this to arbitrary s, here we focus on the second order principal components for simplification. The second order tropical principal components can be written as follows:

Problem 1 *We seek a solution for the following optimization problem:*

$$\min_{D^{(1)},D^{(2)},D^{(3)}\in\mathcal{U}_n} \sum_{i=1}^{n} d_{tr}(d_i, d_i')$$

where

$$d_i' = \lambda_1^i \odot D^{(1)} \oplus \lambda_2^i \odot D^{(2)} \oplus \lambda_3^i \odot D^{(3)}, \quad \text{where } \lambda_k^i = min(d_i - D^{(k)}), \tag{3.4}$$

and

$$d_{tr}(d_i, d_i') = \max\{|d_i(k) - d_i'(k) - d_i(l) + d_i'(l)| : 1 \le k < l \le e\} \tag{3.5}$$

with

$$d_i = (d_i(1), \ldots, d_i(e)) \text{ and } d_i' = (d_i'(1), \ldots, d_i'(e)). \tag{3.6}$$

Even though we do not know the time complexity to solve the optimization problem in Problem 1, the implementation by [73] was not efficient in general. Therefore in this paper we have applied the Metropolis-Hastings algorithm to approximate the optimal solution for Problem 1.

3.3 MCMC Algorithm

Let n be the number of principal components and N be the number of trees. $\Phi_{(w_1,w_2,w_3)}$ be the tropical triangle defined by w_1, w_2, w_3. Hence, we have $f(w_1, w_2, w_3) = \Pi_{\Phi_{u_1,u_2,u_3}}(S)$ is the sum of tropical distance in the tropical triangle defined by w_1, w_2, w_3 where $S = \{d_1, \ldots d_n\}$ is the sample of ultrametrics; g, which is the function which

project trees onto the tropical tree space. At first, we randomly select n trees from the whole tree data set T and we define the combination of these n trees as pcs. We give an initial value as large as possible to the sum of tropical distance $tropDist$. Then, we randomly select a tree a from pcs and replace it with a tree b randomly selected from the rest of trees S ($S = T \setminus pcs$) such that we can have a new tree combination $\hat{pcs} = pcs \setminus a \cap b$. Next, we calculate the ratio of the sum of tropical distance $r = f(pcs)/f(\hat{pcs})$ and compare it with a number u randomly selected from uniform(0, 1) distribution. Once $u \leq min(r, 1)$, we replace pcs with $\hat{pcs}$ and record the projected points $projPoints$, $tropDist$ and $\hat{pcs}$ if $f(pcs) < tropDist$. Finally, we remove b from S and repeat the whole process $N - n$ times. This process is summarized in Algorithm 5 as below:

Algorithm 5: Markov Chain Monte Carlo sampling

Input: Initial distance vectors D of trees T, the number of principal components n, the number of trees N.

Output: The combination of trees $comb$, the projected points $projPoints$ and the sum of tropical distances $tropDist$.

Let $pcs = n$ random trees selected from T, $S = T \setminus pcs$, $tropDist = 1000000$;
for $i = 1, ..., N - n$ **do**
    a = Select one tree randomly from pcs;
    b = Select one tree randomly from S;
    $\hat{pcs} = pcs \setminus a \cap b$;
    $r = f(pcs)/f(\hat{pcs})$;
    Randomly select a number u from uniform(0,1) distribution;
    if $u \leq min(r, 1)$ **then**
        $pcs = \hat{pcs}$;
        if $f(pcs) < tropDist$ **then**
            $projPoints = g(\hat{pcs})$, $tropDist = f(\hat{pcs})$, $comb = \hat{pcs}$
        end
    end
    $S = S \setminus b$;
end

We define B as the set of all possible combination of trees and $W_i = [w_1\, w_2 \ldots w_n] \in B$ as a specific combination of trees which is actually a set of principal components. Thus, the probability of sampling would be as follows:

$$P(W_i) = \frac{f(W_i)}{\sum_{i \in B} f(W_i)} \tag{3.7}$$

Since we would like to find the combination of trees which leads to minimum sum of tropical distances, we have $f(W_i)$ negatively corresponds to $P(W_i)$ such that the acceptance ratio in our case should be $f(pcs)/f(p\hat{c}s)$ instead of $f(p\hat{c}s)/f(pcs)$.

3.4 Factor of Explained Variance

Like traditional principle component analysis, we need a factor to evaluate the performance of tropical PCA, this factor is also called R^2 defined as the proportion of explained variance in terms of tropical geometric set up as below:

$$R^2 = 1 - \frac{\Pi_\Phi(S)}{\Pi_\Phi(S) + SS_{reg}} \tag{3.8}$$

where SS_{reg} is defined as the "explained sum of squared" which is:

$$SS_{reg} = \sum_{i=1}^{n} d_{tr}(\hat{u}_i, \bar{u})$$

where $\hat{u}_i$ is the tropical projection of an ultrametric u_i for a tree T_i on a tropical polytope and $\bar{u}$ is defined as

$$\bar{u} = \arg\max_{u} \sum_{i=1}^{n} d_{tr}(\hat{u}_i, \bar{u})$$

which is also called a Fermat Weber point of $\{\hat{u}_1, ..., \hat{u}_n\}$.

3.5 Application to Empirical Datasets

Apicomplexa data

The phylum Apicomplexa contains many important protozoan pathogens [32], including the mosquito-transmitted *Plasmodium* spp., the causative agents of malaria, *Toxoplasma gondii*, which is one of the most prevalent zoonotic pathogens worldwide, and the water-born pathogen *Cryptosporidium* spp. Several members of the Apicomplexa also cause significant morbidity and mortality in both wildlife and domestic animals. These include *Theileria* spp. and *Babesia* spp., which are tick-borne haemoprotozoan pathogens that infect and cause disease in ungulates, and several

species of *Eimeria*, which are enteric parasites that are particularly detrimental to the poultry industry. Due to their medical and veterinary importance, whole genome sequencing projects have been completed for multiple prominent members of the Apicomplexa. The second order tropical principal components computed from the Apicomplexa data set is shown in Figure. 3.1 and Figure. 3.2.

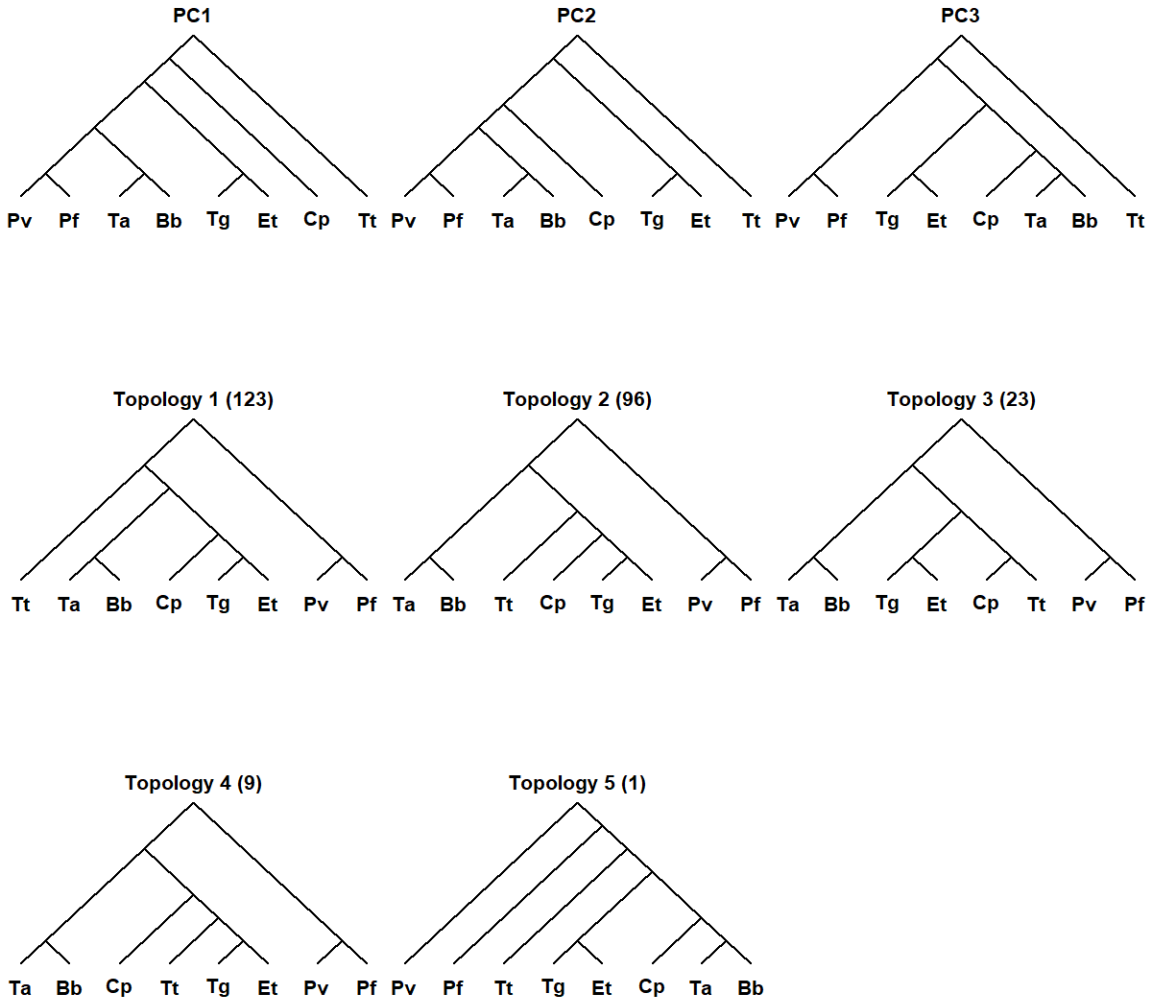

Figure 3.1: Projected topology frequencies from the Apicomplexa data set: parenthesized numbers give the frequencies of each topology.

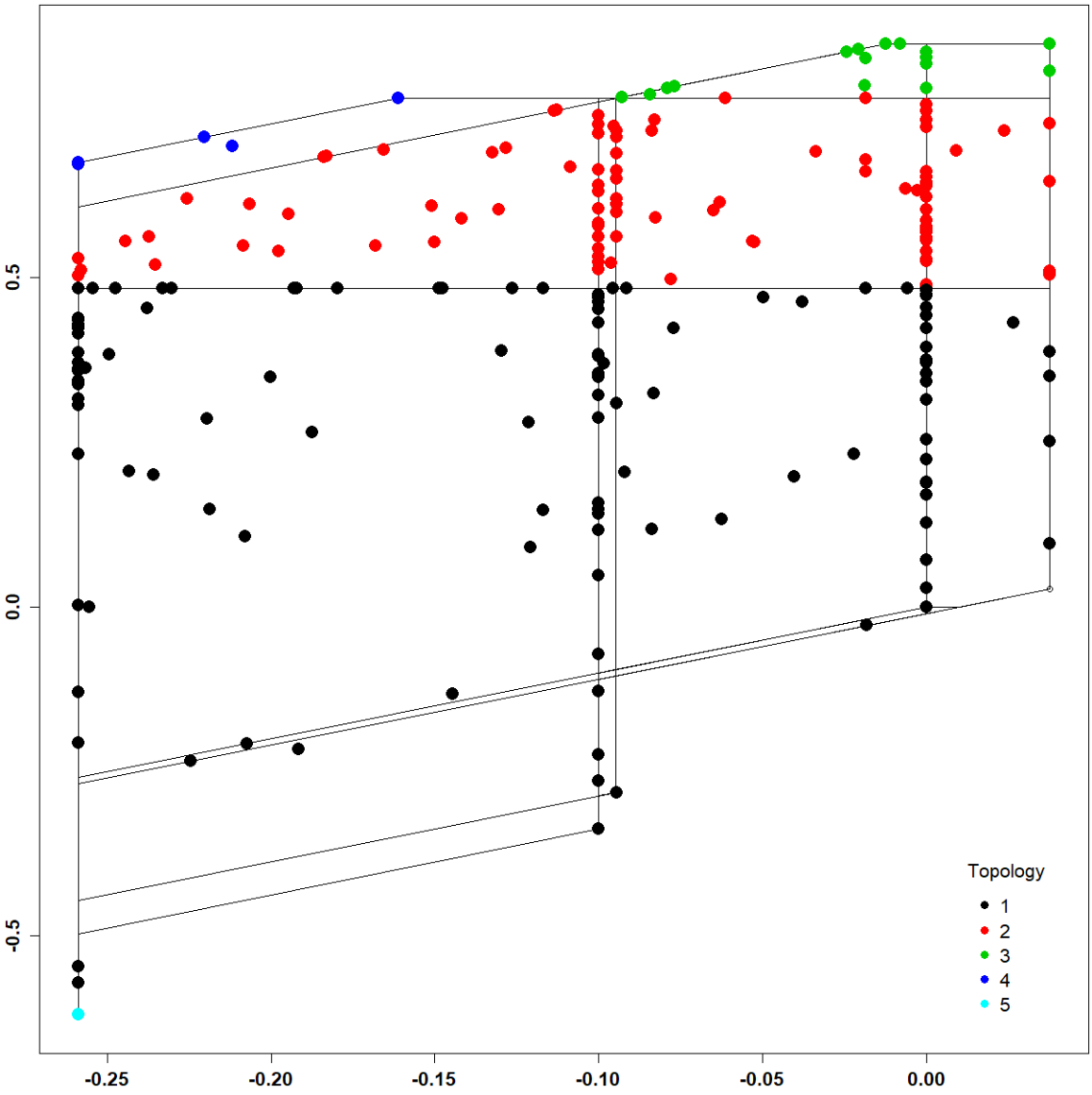

Figure 3.2: Projected points in the tropical polytope PCA of the Apicomplexa data set.

Coelacanths genome and transcriptome data

We have applied the clustering methods to the data set comprising 1,290 nuclear genes encoding 690,838 amino acid residues obtained from genome and transcriptome data by [34]. Over the last decades, the phylogenetic relations between coelacanths, lungfishes, and tetrapods have been controversial despite there has been much work on the data set [20]. After we clean the data set, it consisted of 1193 gene alignments for 10 species: lungfish, *Protopterus annectens*, and coelacanth, *Latimeria chalum-*

nae; three tetrapods, frog, *Xenopus tropicalis*, chicken, *Gallus gallus*, and human, *Homo sapiens*; two ray-finned fish, *Danio rerio* and *Takifugu rubripes*; and three cartilaginous fish included as an out-group, *Scyliorhinus canicula*, *Leucoraja erinacea* and *Callorhinchus milii*. The second order tropical principal components computed from the Coelacanths genome and transcriptome data set is shown in Figure. 3.3 and Figure. 3.4.

It takes around 6 mins to finish a round. The running time could be reduced if we consider parallel computing. All the code is running on a computer with processor Intel Core i7-6700 3.40GHz×8, memory 15.6 GB and OS type Ubuntu 18.04 64-bit.

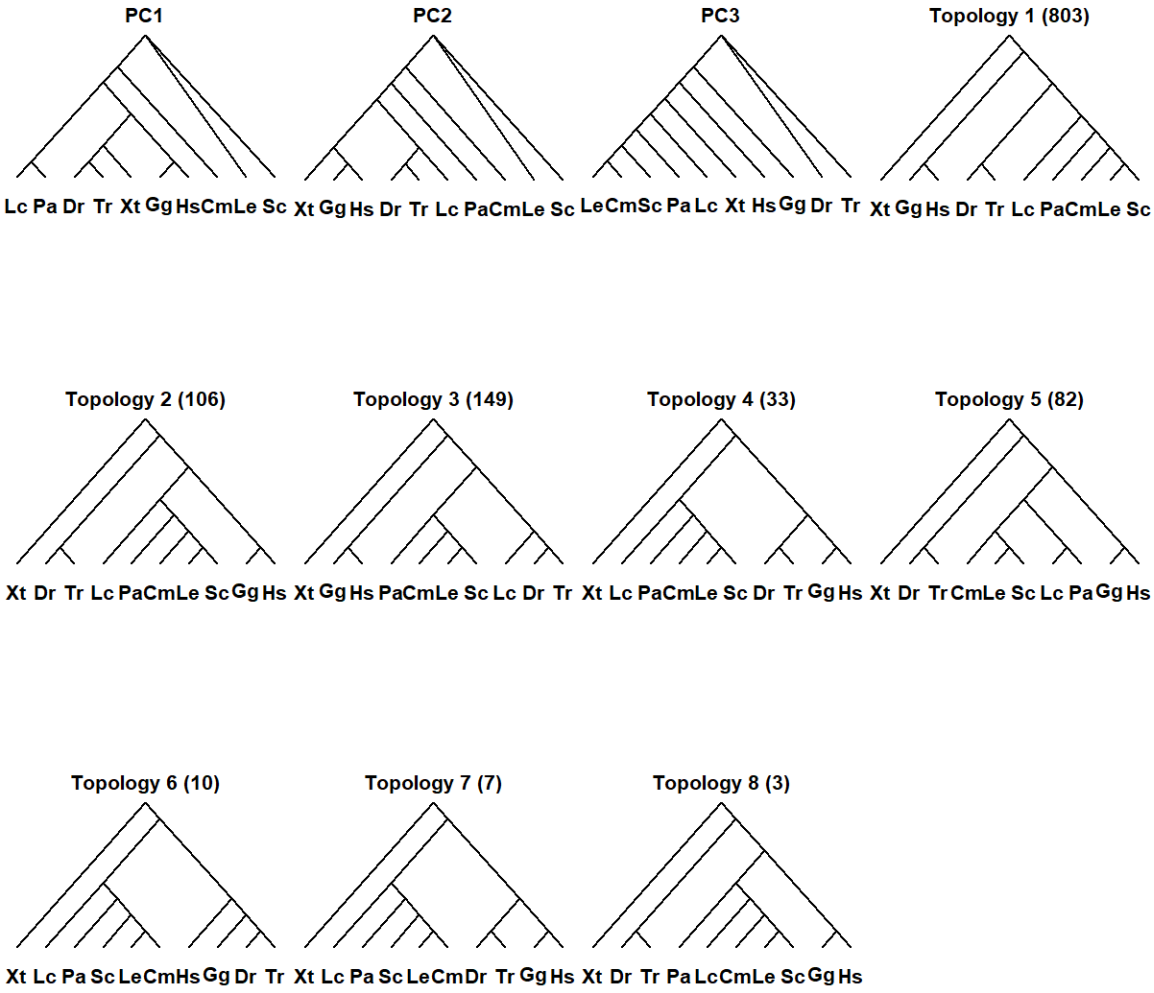

Figure 3.3: Projected topology frequencies from the Coelacanths genome data set: parenthesized numbers give the frequencies of each topology. Labels abbreviations are: Latimeria, Lc; Scyliorhin, Sc; Leucoraja, Le; Callorhinc, Cm; Takifugu, Tr; Danio, Dr; Lungfish, Pa; Homo, Hs; Gallus, Gg; Xenopus, Xt.

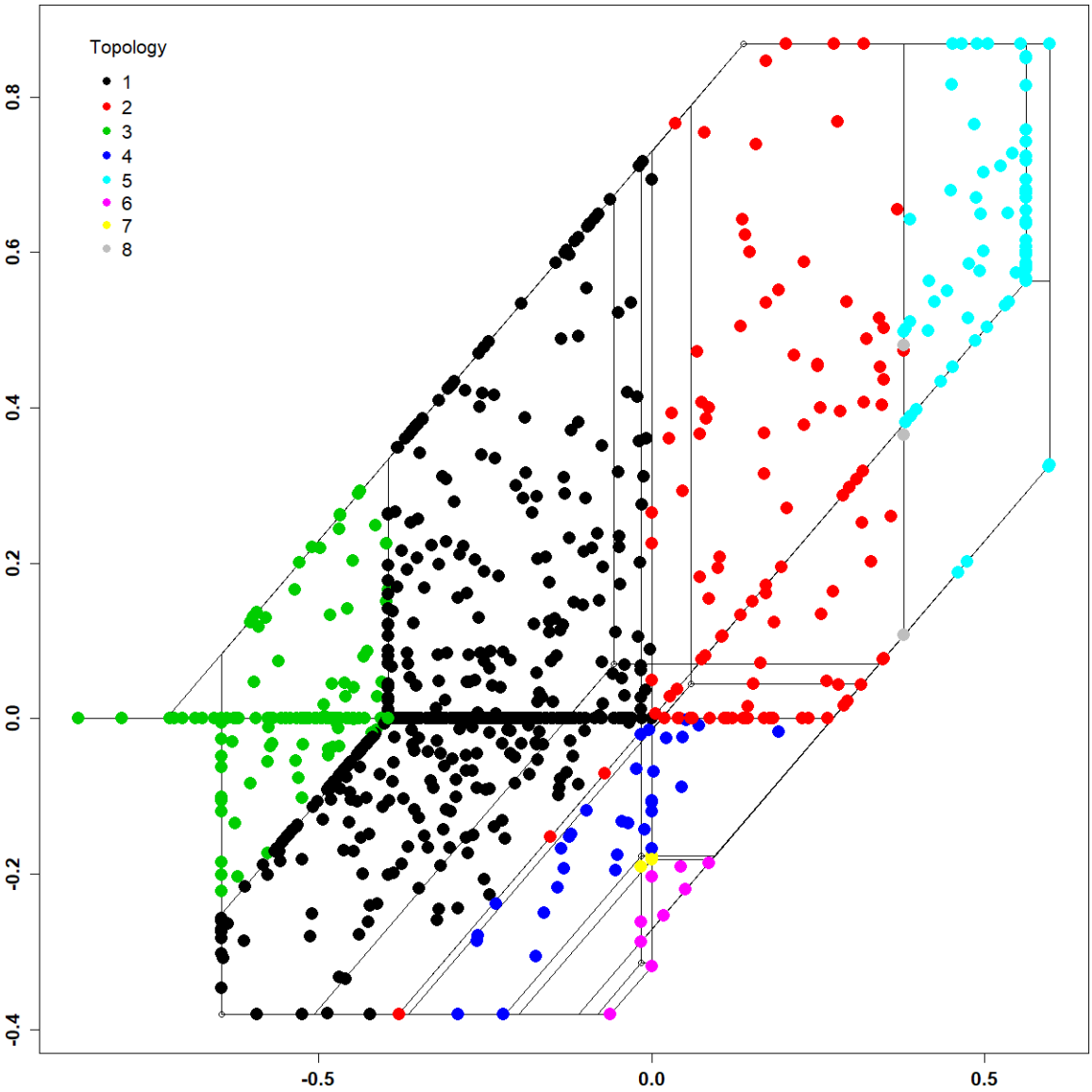

Figure 3.4: Projected points in the tropical polytope PCA of the Coelacanths genome and transcriptome data set.

Influenza data

Influenza is a respiratory illness caused by influenza virus, which is also a highly contagious and fast-transmitting disease. Especially, Influenza A often undergoes antigenic variation and is extremely prone to widespread epidemics. In this section, we analysis 1089 Influenza A H3N2 sequences collected in the United States between 1993 and 2016. Sequences are randomly selected from four or five consecutive seasons which corresponds to four or five leaves in a phylogeny tree. With these sequences, around

20,000 unrooted trees are built using Neighbor-joining method based on hamming distance for most years.

To achieve tree dimensionality reduction and visualize a fitted tropical polytope with three vertices, tropical principal component analysis is applied here. Due to the sensitivity of ordinary PCA for outliers, we remove these trees identified by *KDE-TREES* method as outliers from each data set. After that, we implemented tropical MCMC method to approximate principal components since the number of trees is large. We ran this tropical MCMC process 5 times and selected the tree combination with minimum sum of tropical distance. Each time was run in parallel on eighteen CPU cores, Intel(R) Xeon(R) W-2155 CPU @ 3.30GHZ 3.31GHz, and took almost 2 hours to finish.

In general, the projected topologies are congruent with our intuition: Sample 4 and Sample 3, Sample 5 and Sample 4 are grouped together since they are two consecutive years. Compared to the tree topology with BHV metric of this data set, they are quite similar, and seem to differ by the nearest neighbor interchange.

Figure 3.5 is a plot of the best-fit tropical polytope over the two-dimensional plane $\mathbb{R}^3/\mathbb{R}\mathbf{1}$ with its cells and projected points at year 2008 for trees with five leaves. The phylogenetic trees are clustered and divide the tropical polytope PCA into several different regions. In addition, the projected points are not equally distributed in cells; adjacent cells may correspond to single tree rearrangement.

Figure 3.6 is the second principal components computed with BHV metric from the Influenza A data set.

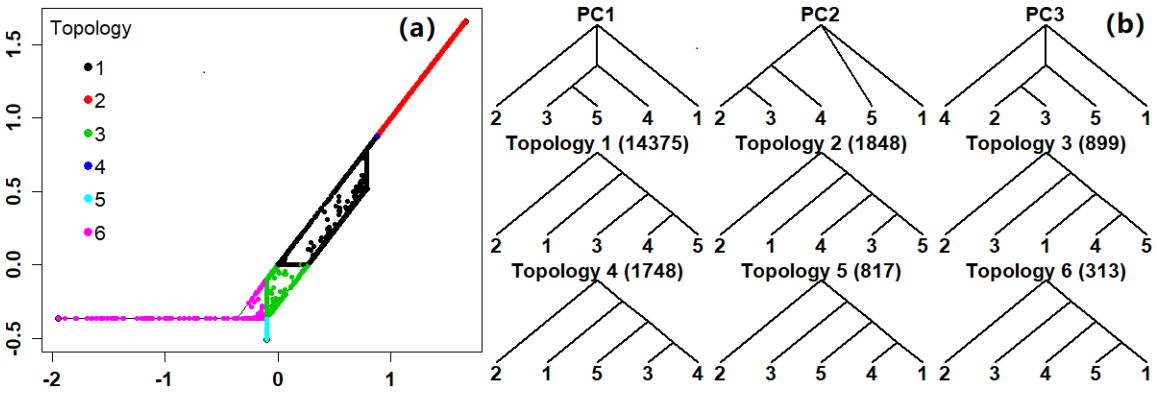

Figure 3.5: The second principal components with tropical metric from the Influenza A data set with five consecutive seasons: (a) The projected points in the tropical polytope PCA; (b) Threes second order principal components and projected tree topology. Labels abbreviations are: Sample1, 1; Sample2, 2; Sample3, 3; Sample4, 5.

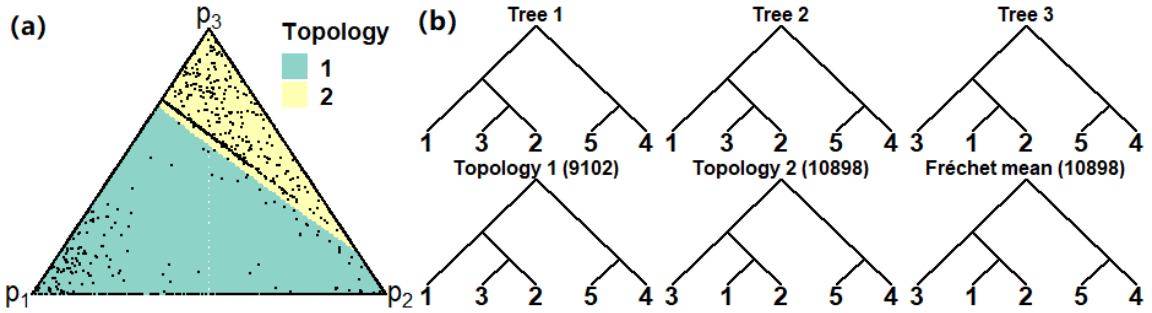

Figure 3.6: The second principal components computed with BHV metric from the Influenza A data set with five consecutive seasons: (a) The simplex shaded by the topology of the corresponding points on the affine subspace; (b): Tree 1, Tree 2, and Tree 3 correspond to three weighted Fréchet means. Topology 1 is the topology of trees on the affine subspace.

In terms of R^2, it is obvious from Table 3.1 that BHV metric has a better performance than tropical metric until year 2006. After that time point, tropical PCA could explain more variance than PCA using BHV metric. This result may due to various computational source of errors. It is obvious that tropical PCA has a better performance since its R^2 is greater than the R^2 of PCA using BHV metric except year 2000, 2001 for tree with 4 leaves and year 1999, 2001 for tree with 5 leaves.

Table 3.1: R Squares of Tropical PCA and BHV

Year	Tree with 4 leaves		Tree with 5 leaves	
	Tropical metric	BHV metric	Tropical metric	BHV metric
1993	0.9559	0.7099	0.7269	0.3019
1994	0.9426	0.4611	0.8505	0.4347
1995	0.8665	0.1900	0.9577	0.3151
1996	0.9821	0.2150	0.7482	0.5025
1997	0.9532	0.0069	0.8437	0.0505
1998	0.9395	0.0452	0.8790	0.6408
1999	0.9069	0.0038	0.8564	0.9524
2000	0.9132	0.9555	0.7942	0.0014
2001	0.9088	0.9402	0.8302	0.9488
2002	0.9863	0.0107	0.9525	0.8962
2003	0.9848	0.0972	0.8622	0.4927
2004	0.9505	0.4272	0.7931	0.3651
2005	0.9949	0.4628	0.8304	0.3634
2006	0.9643	0.0951	0.7300	0.2383
2007	0.9381	0.5562	0.6995	0.2727
2008	0.8813	0.4887	0.4637	0.0460
2009	0.8926	0.0763	0.6289	0.1563
2010	0.8886	0.0329	0.6665	0.1935
2011	0.9016	0.3592	0.5920	0.2771
2012	.	0.2756	0.5568	0.1998
2013	0.7935	0.3612	0.5624	0.1279
2014	.	0.1383	N/A	N/A

3.6 Discussion

We achieved the application of the MCMC method in the tropical tree space. It has been shown this method works well in the tropical tree space instead of traditional Euclidean space. The three PCs returned from the MCMC method are combination of trees selected from the original data set. It is, however, not the true PCs in the tropical tree space. In other words, the result of this approximation brings extra errors increased with the size of the original data set.

In order to find the true PCs, Yoshida proposed a new MCMC Metropolis-Hasting algorithm. This new method is supposed to bring greater efficiency and less errors. The parameter k in this method is used to control the convergence rate. It makes it possible for us to control the "temperature" of our method. The whole process is

listed in Algorithm 6, Algorithm 7 and Algorithm 8. These algorithms compute a proposal state, i.e., a set of proposed trees.

Algorithm 6: Finding the proposal set of trees

Input: Input: Set of equidistant trees $\{T_1, T_2, T_3\}$, $k \in [m]$.
Output: Output: Next set of equidistant trees $\{T_1', T_2', T_3'\}$.
for $i = 1, 2, 3$ **do**
  Set $T_i' = T_i$;
  Pick random numbers $(i_1, \ldots, i_k) \subset [m]$ without replacement;
  Permute the tree leaf labels $(i_1, \ldots, i_k) \subset [m]$ of T_i' with a random permutation σ in the symmetric group on $\{i_1, \ldots, i_k\}$;
  Pick a random internal branch b_1 in T_i';
  Let l_i be the branch length of the internal branch you picked and update $l_i := l_i + \epsilon \cdot c$ where $\epsilon \sim Unif\{+1, -1\}$, and $c \sim Unif[0, l_i/m]$;
  Pick another branch b_2 on the path from the root to the leaf where the branch b_1 is also on the path;
  Let l is the branch length of b_2. If $l - \epsilon \cdot c < 0$ then set $l := 0$ and $l_i := l_i + I - \epsilon \cdot c$. If not then set $l := l - \epsilon \cdot c$
end
Return $\{T_1', T_2', T_3'\}$.

Now, using Metropolis algorithm to decide whether the proposal state should be accepted or rejected.

Algorithm 7: Metropolis-Hastings algorithm

Input: Input: Current set of equidistant trees $\{T_1, T_2, T_3\}$ and the proposal state, $\{T_1', T_2', T_3'\}$. The sample of ultrametrics $S = \{d_1, \ldots d_n\}$.
Output: Output: Decision whether we should accept the proposal or not.
Compute ultrametrics u_1, u_2, u_3, from T_1, T_2, T_3, respectively;
Compute ultrametrics v_1, v_2, v_3, from T_1', T_2', T_3', respectively;
Compute $\Pi_{\Phi_{u_1,u_2,u_3}}(S)$ and $\Pi_{\Phi_{v_1,v_2,v_3}}(S)$;
Set $p = \min\{1, \Pi_{\Phi_{u_1,u_2,u_3}}(S)/\Pi_{\Phi_{v_1,v_2,v_3}}(S)\}$;
Accept a proposal $\{T_1', T_2', T_3'\}$ with probability p. If not then stay at the current state $\{T_1, T_2, T_3\}$.

With Algorithms 6 and 7, we have the following MCMC algorithm.

Algorithm 8: MCMC algorithm to estimate the second order principal components

Input: Input: Sample of equidistant trees $\{T_1, \ldots, T_n\}$. Constant positive integer $C > 0$.

Input: Output: Second order principal components $\{T_1^*, T_2^*, T_3^*\}$.

Set $S := \{d_1, \ldots, d_n\}$ where d_i is the ultrametrics computed from a tree T_i, for $i = 1, \ldots, n$;

Pick random trees $\{T_0^1, T_0^2, T_0^3\} \subset \{T_1, \ldots, T_n\}$;

Compute ultrametrics u_1^*, u_2^*, u_3^*, from T_0^1, T_0^2, T_0^3, respectively;

Set $k = m$, where m is the number of leaves;

Set $i = 1$;

while *not converge* **do**

  if *i mod C equals zero and $k > 0$* **then**

    Set $k = k - 1$.

  end

  Compute the proposal $\{T_1^1, T_1^2, T_1^3\}$ via Algorithm 6 with $\{T_0^1, T_0^2, T_0^3\}$ and k;

  Set ultrametrics u_1, u_2, u_3, from T_1^1, T_1^2, T_1^3, respectively;

  if *Algorithm 7 returns "accept"* **then**

    Set $T_0^1 = T_1^1$, $T_0^2 = T_1^2$, and $T_0^3 = T_1^3$

  end

  if $\Pi_{\Phi_{u_1,u_2,u_3}}(S) < \Pi_{\Phi_{u_1^*,u_2^*,u_3^*}}(S)$ **then**

    Set $u_1^* := u_1$, $u_2^* := u_2$, $u_3^* := u_3$

  end

  Set $i = i + 1$

end

Return the ultrametrics u_1^*, u_2^*, u_3^*.

Appendix A: Code for Chapter 2

```
# Description
-------------------------------------------------------------------------

# In this script, we just include our CURatio functions
# The first one "CURatio" is for calculating CURatios.
# The second one is for the case when we have multi-
    representatives
# of one gene. The "dupl" function could duplicate the a
    spacific gene
# in a given species tree.

CURatio <- function(stdTree){
  library(ape)
  library(phangorn)
  # In this part, we define some variables to save the value
  b <- c() # It is the sum of branch length of the tree
      without a consensus tree.
  # You can change the topology of the tree.
  B <- c() # It is the sum of branch length of the tree with
      a consensus tree.
  # You cannot change the topology of the tree.
  dist <- c() # It is the Robinson-Foulds distance
  output <- c() # This is the ratios
  fileListNew <- c() # This is the new name list to save the
      name we need.
  # Since we want to remove the
```

```
# Given a consensus tree, this stdTree is getted from the
    user.
# Since the tree have no branch lengths, so we calculate
    the branch lengths
# using Grafen's method (Grafen, 1989).
stan.tree <- compute.brlen(stdTree)
# The tip labels of the consensus tree
stan.name <- Tree$tip.label
# Reading data set from the current work station
fileList <- list.files(path='.',pattern='.fasta')

# From here, we begin our for loop, reading each DNA
    sequence into RAM
# and calculating the ratios.
for(i in 1:length(fileList)){
  # Reading the DNA sequence into RAM
  data.list <- read.phyDat(file=fileList[[i]],format='fasta
      ',type='DNA')
  # Since we want to change the name of the tip labels
  splitValue <- sapply(names(data.list)[1:length(data.list)
      ],function(x) strsplit(x,"|",fixed=T))
  nameValue <- lapply(splitValue,function(x) x[1])
  names(data.list) <- unlist(nameValue)

  # We separate the DNA sequence into 3 different cases:
      <5, 5~11, 12.
  if (5 <= length(data.list) && length(data.list) < 12){

    # STEP 1: Calculating the sum of branch lengths of the
        tree without consensus tree
```

```
# Computing pairwise distances for an object of class
    phyDat.
dm <- dist.hamming(data.list)
# Performing the neighbor-joining tree estimation of
    Saitou and Nei (1987).
treeNJ <- NJ(dm)
# Computing the likelihood of a phylogenetic tree given
     a sequence alignment and a model.
fit <- pml(treeNJ,data.list)
# Optimizing the different model parameters.
treeFit <- optim.pml(fit, optNni=TRUE, model="JC")
#
b[i] <- sum(treeFit$tree$edge.length)

# STEP 2: Calculating the sum of branch lengths of the
    tree with consensus tree
# If the number of tips is less than 12, we need to
    remove some of the missing
# tips from the consensus tree.
data.name <- attr(data.list,'names')
index <- which(stan.name%in%data.name)
remove.name <- stan.name[-index]
tree.new <- drop.tip(stan.tree,remove.name)

# Computing the likelihood of a phylogenetic tree given
     a sequence alignment and a model.
fit2 <- pml(tree.new,data.list)
# Optimizing the different model parameters.
treeFit2 <- optim.pml(fit2, optNni=FALSE, model="JC") #
   ## MLE tree under the JC model with constraint
```

```
  #
  B[i] <- sum(treeFit2$tree$edge.length)

  # We calculate the Robinson-Foulds distance to compare
      the two trees
  dist[i] <- RF.dist(treeFit$tree,treeFit2$tree)
  # We keep all the new names of the DNA sequences
  fileListNew[i] <- paste(unlist(strsplit(fileList[i],"-
      wg"))[1],unlist(strsplit(fileList[i],"-wg"))[2],sep=
      '')
  # Calculating the ratios
  output[i] <- B[i]/b[i]

} else if(length(data.list) == 12){
  # Here, we do the same work. But the case is when the
      number of
  # tree's tips is equal to 12.
  # STEP 1
  dm <- dist.hamming(data.list)
  treeNJ <- NJ(dm)
  fit <- pml(treeNJ,data.list)
  treeFit <- optim.pml(fit, optNni=TRUE, model="JC")
  b[i] <- sum(treeFit$tree$edge.length)

  # STEP 2

  fit2 <- pml(stan.tree,data.list)
  treeFit2 <- optim.pml(fit2, optNni=FALSE, model="JC") #
      ## MLE tree under the JC model with constraint
  B[i] <- sum(treeFit2$tree$edge.length)
```

```
        dist[i] <- RF.dist(treeFit$tree,treeFit2$tree)
        fileListNew[i] <- paste(unlist(strsplit(fileList[i],"-
            wg"))[1],unlist(strsplit(fileList[i],"-wg"))[2],sep=
            '')
        #OUTPUT
        output[i] <- B[i]/b[i]
      } else next

    }
    # Since the numbers of some of the trees's tips are less
        than 5, so we get some
    # NA value in our data set. We use "position" to mark the
        position.
    position <- is.na(output)
    # We remove the NA value from the data set.
    result.mix <- data.frame(fileListNew[!position],output[!
        position],dist[!position],stringsAsFactors=FALSE)
    # We return the final dataframe data set at last.
    return(result.mix)
}

dupl <- function(tip_label,consen_path){
    # tip_label: The tip lable you want to duplicate
    # consen_path: The directory of your consensus tree.
    if(require(ape)){
      dupl_text <- readLines(consen_path)
      if(grepl(tip_label, dupl_text)){
        new_pattern <- paste("(",tip_label,",",tip_label,")",
            sep = "")
```

```
      new_lines <- gsub(tip_label,new_pattern,dupl_text)
      conTree <- read.tree(text = new_lines)
      return(conTree)
    }
    else{
      warning("No tip label matched in the tree.")
    }
  }
  else{
    warning("This function requires 'ape' package.")
  }
}



##### Outlier group DNA generating
##### Qiwen Kang

# Functions
  ------------------------------------------------------------------------


# The first two functions are for simulating the DNA
    aligments
printf <- function(...) invisible(print(sprintf(...)))
mdk <- function(path){
  fileNames <- list.files(path)
  gene1 <- read.nexus(paste(path,fileNames[1],sep=""))

  for(i in 1:6000){
    gene1[[i]]$edge.length <- gene1[[i]]$edge.length/sum(
        gene1[[i]]$edge.length)
```

```
    wt1 <- printf(paste(path, "outlier%d.tre", sep = ""), i)
    write.tree(gene1[[i]], wt1, append = FALSE, digits = 4)
    comd <- printf("./run_paml_JC_%s_>_jnk2", wt1)
    system(comd)
  }
}

# These two functions are for calculating CURatio
sevenJC<-function(spTree,fileDir){
  #### The sum of branch length of each tree
  b <- rep(NA,6000)
  B <- rep(NA,6000)
  output <- rep(NA,6000)
  stan.tree <- compute.brlen(spTree)
  fileList <- list.files(path=fileDir,pattern = ".phylip")

  #### Calculating the ratio
  for(i in 1:6000){
    workDir<-paste(fileDir,fileList[[i]],sep="")
    # STEP 1
    # Reading the original tree, we also need to clean the
       name value, it
    # is kind of messy
    dataOri<-read.dna(workDir)
    data <- as.list(dataOri)
    data$aa <- as.list(dataOri)$a
    data$ee <- as.list(dataOri)$e
    data.list<-phyDat(data,type='DNA')
```

```
    # STEP 2
    # Reading the MLE tree
    dm<-dist.hamming(data.list)
    treeNJ<-NJ(dm)
    fit<- pml(treeNJ,data.list)
    treeJC <- optim.pml(fit, optNni=TRUE, model="JC")
    b[i]<-sum(treeJC$tree$edge.length)

    # STEP 3

    fit2<- pml(stan.tree,data.list)
    fit.opt<- optim.pml(fit2, optNni=FALSE, model="JC") ###
       MLE tree under the JC model with constraint
    B[i]<-sum(fit.opt$tree$edge.length)
    #OUTPUT
    output[i]<-B[i]/b[i]
  }
  return(output)
}

sevenJC_out<-function(spTree,fileDir){
        #### The sum of branch length of each tree
        b <- rep(NA,6000)
        B <- rep(NA,6000)
        output <- rep(NA,6000)
        stan.tree <- compute.brlen(spTree)
        fileList <- list.files(path=fileDir,pattern = ".
           phylip")

        #### Calculating the ratio
```

```
for(i in 1:6000){
        workDir<-paste(fileDir,fileList[[i]],sep="")
        # STEP 1
        # Reading the original tree, we also need to
            clean the name value, it
        # is kind of messy
        dataOri<-read.dna(workDir)
        data <- as.list(dataOri)
        data$aa <- as.list(dataOri)$e
        data$ee <- as.list(dataOri)$e
        data$e <- as.list(dataOri)$a
        data.list<-phyDat(data,type='DNA')

        # STEP 2
        # Reading the MLE tree
        dm<-dist.hamming(data.list)
        treeNJ<-NJ(dm)
        fit<- pml(treeNJ,data.list)
        treeJC <- optim.pml(fit, optNni=TRUE, model="
            JC")
        b[i]<-sum(treeJC$tree$edge.length)

        # STEP 3

        fit2<- pml(stan.tree,data.list)
        fit.opt<- optim.pml(fit2, optNni=FALSE, model
            ="JC") ### MLE tree under the JC model
            with constraint
        B[i]<-sum(fit.opt$tree$edge.length)
        #OUTPUT
```

```
        output[i]<-B[i]/b[i]
    }
    return(output)
}


# Setting up
    ------------------------------------------------------------------------


library(ape)
library(phangorn)
setwd("~/r/180116sevenLeaves/")

# Creating DNA alignment (Just running 1 time)
    ------------------------------------------------------------------------


path_non <- "./data2/non_outlier/"
path_out <- "./data2/out/"
# mdk(path_non)
# mdk(path_out)

# Calculating CURatio
    ------------------------------------------------------------------------
species <- unroot(read.nexus("./data2/sp30.nex")[[30]])

# Non_outlier with in-paralogs
output_non <- sevenJC(species,path_non)

fileList <- list.files(path=path_non,pattern = ".phylip")
```

```
result.mix1<-data.frame(fileList,output_non,stringsAsFactors=
    FALSE)
colnames(result.mix1)<-c('Names','Ratio')
write.table(result.mix1,paste(path_non,'Ratio_non.txt',sep=""
    ),sep='\t')

# out_paralogs
output_out <- sevenJC_out(species,path_non)

fileList2 <- list.files(path=path_non,pattern = ".phylip")
result.mix2<-data.frame(fileList2,output_out,stringsAsFactors
    =FALSE)
colnames(result.mix2)<-c('Names','Ratio')
write.table(result.mix2,paste(path_non,'Ratio_out.txt',sep=""
    ),sep='\t')

non <- read.table("./data2/non_outlier/Ratio_non.txt")
out <- read.table("./data2/non_outlier/Ratio_out.txt")
non[,"Group"] <- "Non_outlier"
out[,"Group"] <- "Out_paralog"
treeOne <- rbind(non,out)
require(ggplot2)
jpeg(filename = paste(workDir,'TreeOne',cValue,'.png',sep="")
    , width = 400, height = 300)
picOne <- ggplot(treeOne, aes(Ratio, fill = Group))+
          geom_histogram(data = subset(treeOne, Group == 'Non
              _outlier'), alpha = 0.2, binwidth = 0.05)+
          geom_histogram(data = subset(treeOne, Group == 'Out
              _paralog'), alpha = 0.2, binwidth = 0.05)+
          theme(plot.title = element_text(hjust = 0.5))+
```

```
            scale_fill_manual(values = c("red", "blue"))+
            xlim(c(0.9,2.5))+
            ylab("Count")+
            ggtitle('Histogram of Ratios')
print(picOne)
dev.off()
```

Appendix B: Code for Chapter 3

```
#' Tropical MCMC
#'
#' @param distVect_all
#' All of the distance vectors.
#' @param N
#'  Number of points in tropical space.
#' @param pcs
#' Number of pricinpal components
#' @param nr
#' Number of repeat times.
#' @param env
#' Parameter for changing environment.
#' @return
#' @export
#'
#' @examples
tropMCMC <- function(distVect_all, N, pcs, nr = 2, env = .
    GlobalEnv){
  env$sumValues <- rep(NA, nr)
  env$comb_list <- list()

  D_all <- matrix(unlist(distVec_all), ncol=N)
  for(j in 1:nr){
    sample_init <- sample(N,pcs)
    best <- 100000
    out <- c(1:N)[-sample_init]
```

```
    pc_base_init <- D_all[,sample_init]
    init_value <- tropDistSum(pc_base_init, distVec_all)
    while(length(out)!=0){
      change_ind <- sample(pcs,1)
      out_change <- sample(out, 1)
      comb_set <- c(sample_init[-change_ind], out_change)

      new_base <- D_all[,comb_set]

      update_value <- tropDistSum(new_base, distVec_all)
      r <- init_value/update_value

      if(runif(1) < min(r, 1)){
        sample_init <- comb_set

        best <- ifelse(update_value < best, update_value,
           best)
      }
      out <- out[-which(out==out_change)]
      init_value <- update_value

    }
    env$comb_list[[j]] <- sample_init
    env$sumValues[j] <- best
  }
  min_index <- which(sumValues==min(sumValues))
  return(comb_list[[min_index]])
}

#' Distance matrix
```

```
#'
#' @param trees
#' Phylogeny trees
#' @param tipOrder
#' The names of tip label. You cold give an order to them.
#' @description
#' This function is to get the distance vector of a phylogeny
    tree in tropical space
#' @return
#' @export
#'
#' @examples

.vec_fun<-function(x){
  m<-dim(x)[1]
  vecTreesVec<-rep(NA,choose(m,2))
  for(row.num in 1:(m-1)){
    for(col.num in (row.num+1):m){
      vecTreesVec[col.num-row.num+(m-1+(m-1-row.num+2))*(row.
          num-1)/2]<-x[row.num,col.num]
    }
  }
  vecTreesVec
}

distMat <- function(trees, tipOrder){ # Here trees should be
   a list
  if(class(trees)=="multiPhylo"){
    trees_root <- root(trees, outgroup = tipOrder[1],resolve.
        root=TRUE)
```

```
    chronotrees <- parLapply(cl, trees_root, chronos)
    dist_chrono <- parLapply(cl, chronotrees,cophenetic)

    dist_ordered <- parLapply(cl, dist_chrono, function(x) x[
       tipOrder, tipOrder])
    distVec_all <- parLapply(cl, dist_ordered, .vec_fun)

    # chronotrees <- lapply( trees_root, chronos)
    # dist_chrono <- lapply(chronotrees,cophenetic)
    #
    # dist_ordered <- lapply( dist_chrono, function(x) x[
       tipOrder, tipOrder])
    # distVec_all <- lapply( dist_ordered,vec_fun)

  }else {
    treeOne <- root(trees, outgroup = tipOrder[1],resolve.
       root=TRUE)
    chronoTree <- chronos(treeOne)
    dist_chrono_one <- cophenetic(chronoTree)

    dist_ordered_one <- dist_chrono_one[tipOrder, tipOrder]
    distVec_all <- vec_fun(dist_ordered_one)
  }

  return(distVec_all)
}

#' Porjected points
#'
```

```
#' @param D_s The distance matrix used to build tropical
    space
#' @param D Distance vector (matrix)
#'
#' @return
#' @export
#'
#' @examples
project_pi<-function(D_s,D){
  if(is.null(dim(D_s))){
    lambda <- min(D - D_s)
    pi_D <- c(t(lambda + t(D_s)))
  }else{
    lambda <- apply(D - D_s, 2, min)#D_s by row
    pi_D <- apply(t(lambda + t(D_s)),1,max)
  }

  # pi_D <- ifelse(rep(is.null(dim(D_s)),28), c(t(min(D - D_s
      ) + t(D_s))), apply(t(apply(D - D_s, 2, min) + t(D_s))
      ,1,max))

  # pi_D <- ifelse(rep(is.null(dim(D_s)),28), c(t(min(D[[1]]
      - D_s) + t(D_s))), apply(t(apply(D[[1]] - D_s, 2, min)
      + t(D_s)),1,max))

  return(pi_D)
}

#' The sum of tropical distance of multiple trees
#'
```

```
#' @param pc_base
#' Three disntance vetors used to build tropical space
#' @param distVec_all
#' All of the distance vectors.
#' @return
#' @export
#'
#' @examples
tropDistSum <- function(pc_base, distVec_all){

  proj_points <- parLapply(cl, distVec_all, project_pi , D_s
    = pc_base)
  tropical_dist_vec <- mapply(tropical_dist, distVec_all,
    proj_points)
  sum_dist <- sum(tropical_dist_vec)



  return(sum_dist)
}

#' Tropical distance between two trees.
#'
#'
#' @description
#' function to get tropical distance of two points
#' @export
#'
#' @examples

tropical_dist<-function(D_1,D_2){
```

```
  e <- length(D_1)
  t_dist <- 0
  for(i in 1:(e-1)){
    for(j in (i+1):e){
      if(abs(D_1[i]-D_2[i]-D_1[j]+D_2[j])>t_dist){
        t_dist<-abs(D_1[i]-D_2[i]-D_1[j]+D_2[j])
      }
    }
  }
  t_dist
}
```

Bibliography

[1] Marianne Akian, Stéphane Gaubert, Viorel Nitica, and Ivan Singer. Best approximation in max-plus semimodules. *arXiv preprint arXiv:1012.5492*, 2010.

[2] Federico Ardila and Carly Klivans. The bergman complex of a matroid and phylogenetic trees. *arXiv preprint math/0311370*, 2003.

[3] R. Betancur, C. Li, T.A. Munroe, J.A. Ballesteros, and G. Ortí. Addressing gene tree discordance and non-stationarity to resolve a multi-locus phylogeny of the flatfishes (Teleostei: Pleuronectiformes). *Systematic Biology*, page doi:10.1093/sysbio/syt039, 2013.

[4] Louis J Billera, Susan P Holmes, and Karen Vogtmann. Geometry of the space of phylogenetic trees. *Advances in Applied Mathematics*, 27(4):733–767, 2001.

[5] J.P. Bollback and J.P. Huelsenbeck. Parallel genetic evolution within and between bacteriophage species of varying degrees of divergence. *Genetics*, 181(1):225–234, 2009.

[6] P. Brito and S. Edwards. Multilocus phylogeography and phylogenetics using sequence-based markers. *Genetica*, 135:439–455, 2009.

[7] B.L. Cantarel, I. Korf, S.M.C. Robb, G. Parra, E. Ross, B. Moore, C. Holt, A.S. Alvarado, and M. Yandell. MAKER: an easy-to-use annotation pipeline designed for emerging model organism genomes. *Genome research*, 18(1):188–196, 2008.

[8] M. Carling and R. Brumfield. Integrating phylogenetic and population genetic analyses of multiple loci to test species divergence hypotheses in passerina buntings. *Genetics*, 178:363–377, 2008.

[9] B. C. Carstens and L. L. Knowles. Estimating species phylogeny from gene-tree probabilities despite incomplete lineage sorting: an example from Melanoplus grasshoppers. *Syst. Biol.*, 56:400–411, 2007.

[10] Luigi L Cavalli-Sforza and Anthony WF Edwards. Phylogenetic analysis: models and estimation procedures. *Evolution*, 21(3):550–570, 1967.

[11] Guy Cohen, Stéphane Gaubert, and Jean-Pierre Quadrat. Duality and separation theorems in idempotent semimodules. *arXiv preprint math/0212294*, 2002.

[12] J. H. Degnan and L. A. Salter. Gene tree distributions under the coalescent process. *Evolution*, 59:24–37, 2005.

[13] Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme Mitchison. *Probabilistic approaches to phylogeny*, pages 193–233. Cambridge University Press, 1998.

[14] S. Edwards. Is a new and general theory of molecular systematics emerging? *Evolution*, 63:1–19, 2009.

[15] S. V. Edwards, L. Liu, and D. K. Pearl. High-resolution species trees without concatenation. *Proc. Natl. Acad. Sci.*, 104:5936–5941, 2007.

[16] J. Felsenstein. PHYLIP – Phylogeny inference package (Version 3.2). *Cladistics*, 5:164–166, 1989.

[17] Walter M Fitch and Emanuel Margoliash. Construction of phylogenetic trees. *Science*, 155(3760):279–284, 1967.

[18] Shimodaira H. An approximately unbiased test of phylogenetic tree selection. *Syst Biol*, 51(3):492–508, 2002.

[19] D. Haws, P. Huggins, E. M. O'Neill, D. W. Weisrock, and R. Yoshida. A support vector machine based test for incongruence between sets of trees in tree space. *BMC Bioinformatics*, 13(210), 2012. doi:10.1186/1471-2105-13-210.

[20] S Blair Hedges. Vertebrates (vertebrata). *The timetree of life*, pages 309–314, 2009.

[21] J. Heled and A.J. Drummond. Bayesian inference of species trees from multilocus data. *Molecular Biology and Evolution*, 27(3):570–580, 2011.

[22] A. Hobolth and R. Yoshida. Maximum likelihood estimation of phylogenetic tree and substitution rates via generalized neighbor-joining and the EM algorithm. *Algebraic Biology 2005, Computer Algebra in Biology*, 1:41–50, 2005.

[23] R. Hovmoller, L. L. Knowles, and L. S. Kubatko. Effects of missing data on species tree estimation under the coalescent. *Molecular Phylogenetics and Evolution*, 69:1057–1062, 2013.

[24] D. H. Huson, T. Klopper, P. J. Lockhart, and M. A. Steel. *Reconstruction of reticulate networks from gene trees.* Research in Computational Molecular Biology, Proceedings. Springer-Verlag Berlin, Berlin, 2005.

[25] Michael Joswig. Essentials of tropical combinatorics. *Book in preparation*, 1:226, 2014.

[26] R. Jothi, E. Zotenko, A. Tasneem, and T.M. Przytycka. COCO-CL: hierarchical clustering of homology relations based on evolutionary correlations. *Bioinformatics*, 22:779–788, 2006. doi: 10.1093/bioinformatics/btl009.

[27] Kazutaka Katoh, Kazuharu Misawa, Kei-ichi Kuma, and Takashi Miyata. MAFFT: a novel method for rapid multiple sequence alignment based on fast fourier transform. *Nucleic Acids Research*, 30(14):3059–3066, 2002.

[28] Kazutaka Katoh and Hiroyuki Toh. Recent developments in the MAFFT multiple sequence alignment program. *Briefings in Bioinformatics*, 9(4):286–298, 2008.

[29] L.L. Knowles. Estimating species trees: Methods of phylogenetic analysis when there is incongruence across genes. *Systematic Biology*, 58(5):463–467, 2009.

[30] L.L. Knowles. Statistical phylogeography. *Annual Review of Ecology, Evolution, and Systematics*, 40:593–612, 2009.

[31] A.D. Leaché and B. Rannala. The accuracy of species tree estimation under simulation: A comparison of methods. *Systematic Biology*, 60(2):126–137, 2011.

[32] NORMAN D LEVINE. Progress in taxonomy of the apicomplexan protozoa. *Journal of Eukaryotic Microbiology*, 35(4):518–520, 1988.

[33] L. Li, C. J. Stoeckert, and D. S. Roos. OrthoMCL: Identification of ortholog groups for eukaryotic genomes. *Genome Res*, 13:2178–2189, 2003.

[34] Dan Liang, Xing Xing Shen, and Peng Zhang. One thousand two hundred ninety nuclear genes from a genome-wide survey support lungfishes as the sister group of tetrapods. *Molecular biology and evolution*, 30(8):1803–1807, 2013.

[35] Bo Lin, Bernd Sturmfels, Xiaoxian Tang, and Ruriko Yoshida. Convexity in tree spaces. *SIAM Journal on Discrete Mathematics*, 31(3):2015–2038, 2017.

[36] Liang Liu, Lili Yu, Laura Kubatko, Dennis K Pearl, and Scott V Edwards. Coalescent methods for estimating phylogenetic trees. *Molecular Phylogenetics and Evolution*, 53(1):320–328, 2009.

[37] M. Liu, D. G. Panaccione, and C. L. Schardl. Phylogenetic analyses reveal monophyletic origin of the ergot alkaloid gene dmaW in fungi. *Evolutionary Bioinformatics*, 5:15–30, 2009.

[38] Diane Maclagan and Bernd Sturmfels. *Introduction to tropical geometry*, volume 161. American Mathematical Soc., 2015.

[39] W. P. Maddison. Gene trees in species trees. *Systematic Biology*, 46(3):523–536, 1997.

[40] W. P. Maddison and L. L. Knowles. Inferring phylogeny despite incomplete lineage sorting. *Syst. Biol.*, 55:21–30, 2006.

[41] W. P. Maddison and D. R. Maddison. Mesquite: a modular system for evolutionary analysis. version 2.75, 2011.

[42] A. P. Martin and T. M. Burg. Perils of paralogy: Using HSP70 genes for inferring organismal phylogenies. *Systematic Biology*, 51:570–587, 2002.

[43] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[44] Nicholas Metropolis and Stanislaw Ulam. The monte carlo method. *Journal of the American statistical association*, 44(247):335–341, 1949.

[45] E. Mossel and S. Roch. Incomplete lineage sorting: consistent phylogeny estimation from multiple loci, 2007. arXiv q-bio.PE.

[46] Tom MW Nye et al. Principal components analysis in the space of phylogenetic trees. *The Annals of Statistics*, 39(5):2716–2739, 2011.

[47] Lior Pachter and Bernd Sturmfels. *Algebraic statistics for computational biology*, volume 13. Cambridge university press, 2005.

[48] P. Pamilo and M. Nei. Relationships between gene trees and species trees. *Mol. Biol. Evol.*, 5:568–583, 1988.

[49] E. Paradis, J. Claude, and K. Strimmer. APE: analyses of phylogenetics and evolution in R language. *Bioinformatics*, 20:289–290, 2004.

[50] D. Posada and K.A. Crandall. The effect of recombination on the accuracy of phylogeny reconstruction. *Journal of Molecular Evolution*, 54:396–402, 2002.

[51] M. C. Rivera, R. Jain, J. E. Moore, and J. A. Lake. Genomic evidence for two functionally distinct gene classes. *Proc Natl Acad Sci USA*, 95(11):6239–6244, 1998.

[52] D. F. Robinson and L. R. Foulds. Comparison of phylogenetic trees. *Math Biosci*, 53:131–147, 1981.

[53] Sebastien Roch and Mike Steel. Likelihood-based tree reconstruction on a concatenation of alignments can be positively misleading. *arXiv preprint arXiv:1409.2051*, 2014.

[54] N. Rosenberg. The probability of topological concordance of gene trees and species trees. *Theor. Popul. Biol.*, 61:225–247, 2002.

[55] N. A. Rosenberg. The shapes of neutral gene genealogies in two species: probabilities of monophyly, paraphyly, and polyphyly in a coalescent model. *Evolution*, 57:1465–1477, 2003.

[56] A. RoyChoudhury, J. Felsenstein, and E. A. Thompson. A two-stage pruning algorithm for likelihood computation for a population tree. *Genetics*, 180:1095–1105, 2008.

[57] CL Schardl, CA Young, U Hesse, SG Amyotte, K Andreeva, PJ Calie, DJ Fleetwood, DC Haws, N Moore, B Oeser, DG Panaccione, KK Schweri, CR Voisey, ML Farman, JW Jaromczyk, BA Roe, DM O'Sullivan, B Scott, P Tudzynski, Z An, EG Arnaoudova, CT Bullock, ND Charlton, L Chen, M Cox, RD Dinkins, S Florea, AE Glenn, A Gordon, U Güldener, DR Harris, W Hollin, J Jaromczyk, RD Johnson, AK Khan, E Leistner, A Leuchtmann, C Li, J Liu, J Liu, M Liu, W Mace, C Machado, P Nagabhyru, Pan J, J Schmid, K Sugawara, U Steiner, JE Takach, E Tanaka, JS Webb, EV Wilson, JL Wiseman, R Yoshida, and Z Zeng. Plant-symbiotic fungi as chemical engineers: multi-genome analysis of the Clavicipitaceae reveals dynamics of alkaloid loci. *PLoS Genet*, 9(2):e1003323. doi: 10.1371/journal.pgen.1003323, 2013.

[58] Klaus Peter Schliep. phangorn: phylogenetic analysis in R. *Bioinformatics*, 27(4):592–593, 2011.

[59] H. Shimodaira and M. Hasegawa. Multiple comparisons of log-likelihoods with applcations to phylogenetic inference. *Mol Biol Evol*, 16:1114 – 1116, 1999.

[60] David Speyer and Bernd Sturmfels. Tropical mathematics. *arXiv preprint math/0408099*, 2004.

[61] N. Takahata. Gene genealogy in three related populations: consistency probability between gene and population trees. *Genetics*, 122:957–966, 1989.

[62] N. Takahata and M. Nei. Allelic genealogy under overdominant and frequency-dependent selection and polymorphism of major histocompatibility complex loci. *Genetics*, 124:967–978, 1990.

[63] J. W. Taylor, D. J. Jacobson, S. Kroken, T. Kasuga, D. M. Geiser, D. S. Hibbett, and M. C. Fisher. Phylogenetic species recognition and species concepts in fungi. *Fungal Genetics and Biology*, 31:21 – 32, 2000.

[64] K.L. Thompson and L. Kubatko. Using ancestral information to detect and localize quantitative trait loci in genome-wide association studies. *BMC Bioinformatics*, 14:200, 2013.

[65] Y. Tian and L. Kubatko. Gene tree rooting methods give distributions that mimic the coalescent process. *Molecular Phylogenetics and Evolution*, 70:63–69, 2014.

[66] Tamir Tuller and Elchanan Mossel. Co-evolution is incompatible with the markov assumption in phylogenetics. *IEEE/ACM transactions on computational biology and bioinformatics*, 8(6):1667–1670, 2011.

[67] D. W. Weisrock, H. B. Shaffer, B. L. Storz, S. R. Storz, S. R. Storz, and S. R. Voss. Multiple nuclear gene sequences identify phylogenetic species boundaries in the rapidly radiating clade of mexican ambystomatid salamanders. *Molecular Ecology*, 15:2489–2503, 2006.

[68] G. Weyenberg, P.M. Huggins, C.L. Schardl, D.K. Howe, and R. Yoshida. KDETREES: non-parametric estimation of phylogenetic tree distributions. *Bioinformatics*, 30(16):2280–2287, 2014.

[69] Z Yang. PAML: A program package for phylogenetic analysis by maximum likelihood. *CABIOS*, 15:555–556, 1997.

[70] Z. Yang and B. Rannala. Bayesian species delimitation using multilocus sequence data. *PNAS*, 107(20):9264–9269, 2009.

[71] Ziheng Yang et al. *Computational molecular evolution*. Oxford University Press, 2006.

[72] R. Yoshida, K. Fukumizu, and C. Vogiatzis. Multi loci phylogenetic analysis with gene tree clustering. *Annals of Operations Research*, pages https://doi.org/10.1007/s10479–017–2456–9, 2017.

[73] Ruriko Yoshida, Leon Zhang, and Xu Zhang. Tropical principal component analysis and its application to phylogenetics. *arXiv preprint arXiv:1710.02682*, 2017.

[74] C. A. Young, C. L. Schardl, D. G. Panaccione, S. Florea, J. E. Takach, N. D. Charlton, N. Moore, J. S. Webb, and J. Jaromczyk. Genetics, genomics and evolution of ergot alkaloid diversity. *Toxins (Basel)*, 7:1273–1302, 2015.

[75] Y. Yu, J.H. Degnan C. Than, and L. Nakhieh. Coalescent histories on phylogenetic networks and detection of hybridization despite incomplete lineage sorting. *Systematic Biology*, 60(2):138–149, 2011.

[76] Y. Yu, T. Warnow, and L. Nakhleh. Algorithms for MDC-based multi-locus phylogeny inference: Beyond rooted binary gene trees on single alleles. *J Comput Biol*, 18(11):1543–1559, 2011.

Vita

Qiwen Kang

Education

- **M.S in Statistics** University of Kentucky, Lexington, KY, 2014 - 2016
- **B.S in Geographic Information Science** Jilin University, Changchun, China, 2010 - 2014
- **B.S in Computer Science** Jilin University, Changchun, China, 2010 - 2014

Experience

- **Research Assistant** Sanders-Brown Center on Aging, University of Kentucky, 2016 - 2019
- **Research Assistant** Applied Statistical Lab, University of Kentucky, 2015 - 2016
- **Teaching Assistant** Department of Statistics, University of Kentucky, 2014 - 2015

Publications

- Kang, Q., Schardl, C.W., Moore, N. and Yoshida, R., 2018. CURatio: Genome-wide phylogenomic analysis method using ratios of total branch lengths. IEEE/ACM transactions on computational biology and bioinformatics.
- Kang, Q. and Yoshida, R., 2018, July. Estimating Tropical Principal Components Using Metropolis Hasting Algorithm. In International Congress on Mathematical Software (pp. 272-279). Springer, Cham.