Phylogenetic reconstruction accuracy in the face of heterogeneity, recombination, and reticulate evolution

By

Kun-Chieh Wang

A dissertation submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

(Mathematics)

at the

UNIVERSITY OF WISCONSIN – MADISON

2017

Date of final oral examination: April 26, 2017

The dissertation is approved by the following members of the Final Oral Committee:

Professor S. Roch, Professor, Mathematics

Professor C. Ané, Professor, Statistics

Professor D. Anderson, Associate Professor, Mathematics

Professor G. Craciun, Professor, Mathematics

Professor S. Angenent, Professor, Mathematics

ABSTRACT

Unearthing the evolutionary relationships between different but related taxa, both living and extinct, in order to understand the tree of life is one of the main goals for both biologists and mathematicians. From the time of *On the Origin of Species* by Charles Darwin around 1859, numerous models, theories and algorithms have been raised and improved. The purpose of this thesis is to further extend some existing theories and introduce a new network algorithm in phylogenetic reconstruction.

The purpose of my first project is to estimate the accuracy of the Fitch algorithm. The Fitch algorithm is used to solve ancestral state reconstruction problem, whose goal is to find the state of the common ancestors from the state of multiple species. As a parsimony method, the Fitch algorithm reconstruct the states which minimize number of state changes along all possible mutations. My results indicate that in the Cavender-Farris-Neyman model, both on a deterministic tree with branching number larger than $\frac{3}{2}$ and on the Yule model with parameter larger than 6, the accuracy is strictly better than a random guessing.

The second project focuses on comparing a few multi-locus reconstruction methods

which deal with incomplete lineage sorting (ILS), under the assumption that the loci involve both ILS and recombination in the three-taxon case. The results show the decay rate of the accuracy of these multi-locus reconstruction methods, and I compare the accuracies via simulation.

In the third project, I introduce a fast-converging split network reconstruction algorithm and prove the consistency of the algorithm. Like the idea of the short quartet method (SQM), the algorithm relies on short distances only. By shortening the radius of the trusted region, the algorithm eliminates the disadvantage of Neighbor-Net and the split decomposition method, the most popular network reconstruction methods in the last couple of decades.

ACKNOWLEDGEMENTS

First and foremost, I would like to deeply thank my advisor, Professor Sebastien Roch for his passion and his endless assistance. Without him, my research results and this thesis would not have been possible. I learned a lot from him, in both research ideas and life thoughts.

I am thankful to Professor Cécilé Ané, Professor David Anderson, Professor Gheorghe Craciun, and Professor Sigurd Angenent for being my committee members. I would like to thank them for their time, questions, and comments.

Last but not least, I would like to thank my parents and my friends, especially Li-Hsiang Kuo, Kuan-Ying Lee, Meng-Che Ho, Jie-Ren Shih, for their endless support and unconditional love.

CONTENTS

CHAPTER 1

INTRODUCTION

Unearthing the evolutionary relationships between different but related taxa, both living and extinct, in order to understand the tree of life is one of the main goals for both biologists and mathematicians. From the time of *On the Origin of Species* by Charles Darwin around 1859, numerous models, theories and algorithms have been raised and improved. Even now, there is still a huge amount of research being investigated to solve this mystery (see, e.g. [31]).

Phylogenetic trees, which are widely used in this sort of analysis, are a graphical tree structure with the leaves representing the taxa we are interested in, the internal nodes representing the ancestral species that are usually unobservable, and every edge indicating an evolution event. There are many different models of evolution to simulate mutations along its edges. Some models used nowaday are the r-state Poisson model [31], the Kimura 2-parameter or 3-parameter model [32], and the generalised time reversible (or GTR in short) model [60]. These models have been applied in many phylogenetic

studies.

As we mentioned above, the internal nodes of the phylogenetic tree are used to represent those species that are unobservable, so one of our major goals is to uncover important information, like DNA sequences, of those species. This is defined as the *ancestral sequence reconstruction problem*. A significant amount of studies has been involved in solving the ancestral sequence reconstruction problem. Some main approaches can be classified into: maximum parsimony (see, e.g., [23]), maximum likelihood (see, e.g., [47]), and Bayesian inference (see, e.g., [48]).

One of the main categories to solve the ancestral reconstruction problem is maximum parsimony. First introduced by Walter M. Fitch in 1971 [23], the maximum parsimony method reconstructs the state of all interior nodes of a phylogenetic tree in a way that minimizes the number of state changes along all the edges. Even though it is not consistent under some conditions like long-branch attraction [21], the maximum parsimony method has been widely accepted and studied (see, e.g., [21, 58, 39, 64, 56, 11, 55, 35, 24, 65]).

The Fitch algorithm [23] is the most popular algorithm that implements the maximum parsimony method. Despite its efficiency, there are only few studies to support the accuracy of the Fitch algorithm. (see, e.g., [58, 39, 64, 55, 35, 65]). Mike Steel found the accuracy of the Fitch algorithm in the CavenderFarrisNeyman model when the tree is a complete binary tree with equal branch lengths in [58], and Li *et al.* broadened the result to ultrametric trees in [65].

Our first project, as shown in Section 2, further generalizes the results of [58, 65] by estimating the accuracy of the Fitch algorithm on a general phylogenetic binary tree. Our result indicates that on both a deterministic tree and the Yule model, under certain

conditions, the accuracy of the Fitch algorithm is strictly better than random guessing in the CavenderFarrisNeyman model.

Nevertheless, there are still some phenomena that can not be fully explained. The rapid increases of sequence data from multiple loci, thanks to the development of technology, has drawn biologists and mathematicians to notice that the evolutionary histories of individual genes can be different from their underlying species tree. These phenomena are caused by, for example, incomplete lineage sorting, gene gain and loss, or horizontal gene transfer. The relationship between species trees and gene trees has been a focus in recent studies. (see, e.g., [22, 49, 40, 44]).

One of the main reasons for the incompatibility between gene trees and species tree is incomplete lineage sorting (ILS), when two lineages fail to coalesce in a population, resulting in the lineage merging with another less related lineage first and causing a discord between gene trees and species trees. Numerous algorithms have been raised to reconstruct the species tree from multiple loci to account for ILS, for example, the Maximum Likelihood (ML) method [37], the R^* consensus method [5], and the Species Trees Estimation Using Average Coalescent Time (STEAC) method [38]. As shown in [53], Sebastien Roch compares these algorithms in the three-taxon case. He showed that under the three-taxon case assumption, the algorithms can be separated into three categories: ML/GLASS/MT [36, 19, 37], R^*/STAR/MDC [40, 5, 13, 38, 61], and STEAC/SC [38]. Furthermore, he showed that in accuracy ML/GLASS/MT dominates the other two categories in this case, and he computed the decay rate of each category.

The purpose of our second project is, as shown in Section 3, to mimic Roch's work and extend the result by considering recombination, another important factor that causes the disagreement between species trees and gene trees. Recombination causes different

parts on the same gene have different evolutionary histories, resulting in the appearance that the gene might demonstrates a totally different topology from its real history. In our work, we indicate the decay rate of each categories of algorithms for the three-taxon case, under the assumption that the loci we collect from the species tree involve both incomplete lineage sorting and recombination.

Despite the wide use of phylogenetic trees, phylogenetic networks are more adequate to represent data sets involving reticulate events caused by, for example, hybridization, horizontal gene transfer, and gene duplication and loss [31, 15, 16, 25, 52, 57, 59]. A split network, first introduced by Bandelt and Dress in [3], is a representation of an unrooted phylogenetic network. Since then, an abundance of algorithms have been introduced for reconstructing a split network from some certain information on species. Based on the information the algorithms need, we can classify the algorithms into a few categories [29, 31]: Neighbor-Net [7] and the split decomposition method [3] take the pairwise distances between taxa as input, while the consensus network method [27] and the super network method [30, 62] use a set of phylogenetic trees for reconstruction, and the median-joining method [4] and the quasi median-joining method [2] require condensed characters.

The distance-based reconstruction methods, Neighbor-Net algorithm and the split decomposition method, are the most popular algorithms. Because of their consistency if the error of estimated distances is bounded [3, 43, 34], they have been widely accepted and used. Nevertheless, both methods require that the estimation between every pair of taxa have a bounded error, and as a result, they require impractically large sample length to ensure the consistency if the diameter of the network is large.

Our third project, as shown in Section 4, is to introduce a fast-converging split

network algorithm by relying on only short distances, similar to the idea of the short quartet method (SQM) [20]. The algorithm is analogous to the algorithm introduced in [12], with the fact that every edge in a phylogenetic tree corresponds to a split in a split network. By shortening the radius of the trusted region from the diameter of a split network to a linear combination of the depth and the incompatibility of a split network, the algorithm removes the disadvantage of Neighbor-Net and the split decomposition method.

CHAPTER 2

ACCURACY OF FITCH ALGORITHM

Contribution I lead all aspects of the work and wrote the first draft of this section.

2.1 Introduction

Ancestral sequence reconstruction is an important technique in the study of molecular evolution, which incorporates sequences of modern species or taxa, like proteins or genes, into an evolution model in order to comprehend sequences of their common ancestor. The concept was first introduced by Pauling and Zuckerkandl in 1963 [50], and it, accompanied with the rapid accumulation of biomolecular sequence data, has become an essential approach to understanding the evolutionary history of proteins and genes (see, for example [23, 47, 48]).

The study of ancestral sequence reconstruction usually uses phylogenetic trees, which graphically have a tree structure, with the leaves representing the extant taxa and the

edges indicating evolution events. The state of any character in the sequence changes along the edges of the phylogenetic tree, and the sequence of a descendant might be different from the sequence of its ancestor. With the assumption that every character in the sequence evolves identically and independently, ancestral sequence reconstruction can be broken into multiple ancestral state reconstruction questions, whose goal is to find the state of the root given a phylogenetic tree and the state of all leaves.

The Fitch algorithm, first introduced by Walter M. Fitch in 1971 [23], is a technique for ancestral state reconstruction on a binary tree. As a parsimony method, it estimates the state of the root by minimizing the total changes on the edges. Despite its efficiency, there are only a small number of studies to support its accuracy [65, 39, 35, 55, 64, 58]. In [58], Steel found the accuracy of the Fitch algorithm for the Cavender-Farris-Neyman model when the tree is a complete binary tree with equal branch lengths. And in [65], they complete Steel's work and broaden the result to include the case when the tree is an ultrametric binary tree.

Our goal in this section is to generalize the result in [58, 65]. First, we studied the accuracy of the Fitch algorithm for the Cavender-Farris-Neyman model on a general binary tree. Then, we expand the result by considering the Yule model. We prove that in both cases, with certain restrictions, the accuracy of the Fitch method is strictly better than random guessing in a symmetric two-state model.

2.2 Background

We start with some basic definitions. See [31] for an in-depth exposition.

Definition 2.1 (Phylogenetic Tree [31])**.** *We say* $\mathcal{T} = (\mathcal{X}, V, E, L)$ *is a phylogenetic*

tree on $\mathcal{X}$ if (V, E) is a tree, and $L : \mathcal{X} \to V$ is the taxon labeling that assigns exactly one taxon to every leaf and none to any internal node. We additionally say $\mathcal{T} = (\mathcal{X}, V, E, \rho, L)$ is rooted if (V, E, ρ) is a rooted tree with root ρ. Moreover, we say $\mathcal{T}$ is edge-weighted or weighted if there is a mapping $w : E \to (0, \infty)$ that assigns a positive weight to every edge of the tree.

The weight of an edge usually correlates to the evolutionary time. In this section, we use the r-state Poisson Model as the evolution model on a phylogenetic tree.

Definition 2.2 (r-state Poisson Model [31])**.** *Suppose $\mathcal{T}$ is an edge-weighted, rooted phylogenetic tree with edge-weight function w. The r-state Poisson Model makes the following assumptions:*

1. *The possible state set S contains all the possible states for each site, and $|S| = r$. Usually $r = 4$ ($S = \{A, T, C, G\}$) (the Jukes-Cantor model), or $r = 2$ ($S = \{0, 1\}$) (the Cavender-Farris-Neyman model).*

2. *The sequence length is an input parameter, and the state of each site of the initial sequence at the root is drawn independently and uniformly at random from the set of all possible states S.*

3. *The sites evolve identically and independently along the edges of $\mathcal{T}$ at a fixed rate μ.*

4. *With each edge e of $\mathcal{T}$, the expected number of mutations per site along e is given by $\mu w(e)$. The probabilities of change from one state to another are equal.*

In this model, as described in [31], there is an explicit formula for the probability of

the events that two adjacent nodes have equal or different states at a specific site in the sequence.

Lemma 2.3 (Changing and preserving probability of the r-state Poisson Model)**.** *Suppose $e = (z, x) \in E$ is an edge of the phylogenetic tree $\mathcal{T} = (\mathcal{X}, V, E, L, w)$, then for any specific corresponding site s_z and s_x, and any state $s_1, s_2 \in S$, $s_1 \neq s_2$, we have:*

$$\begin{aligned} \mathbb{P}[s_x = s_1 | s_z = s_1] &= \frac{1}{r}(1 + (r-1)e^{\frac{r}{r-1}\mu w(e)}) \\ \mathbb{P}[s_x = s_2 | s_z = s_1] &= \frac{1}{r}(1 - e^{\frac{r}{r-1}\mu w(e)}) \end{aligned}$$

Because the r-state Poisson model assumes that all sites in the sequence evolve identically and independently, we consider only one particular position at a time. As a result, ancestral sequence reconstruction can be broken into multiple ancestral state reconstructions, which are described as follows.

Definition 2.4 (Ancestral state reconstruction problem)**.** *Given an edge-weighted phylogenetic tree $\mathcal{T} = (\mathcal{X}, V, E, \rho, L, w)$, and the state of every leaf , uncover the state of all interior nodes of the tree $\mathcal{T}$, in particular the root ρ.*

The Fitch algorithm is a method for the ancestral state reconstruction problem. As a parsimony method, it reconstructs the state of all interior nodes of a phylogenetic tree in a way that minimizes the number of state changes along all the edges. In detail, the Fitch algorithm consists of two traversals of the tree: First, bottom-up to determine the possible state set of each node, then top-down to assign the ancestral state of each node:

Algorithm 2.5 (Fitch Algorithm [23])**.** *Given a weighted phylogenetic tree $\mathcal{T} = (\mathcal{X}, V, E, \rho, L, w)$ and the state s_x for every leaf $x \in L$, the Fitch algorithm is composed of the following two parts:*

1. *Bottom-Up traversal: The goal of this part is to construct the possible state set* $\tilde{S}_z$ *for each node* z*:*

   (a) *If* $z \in V$ *is a leaf, then* $\tilde{S}_z = \{s_z\}$.

   (b) *If* $z \in V$ *is an interior node with children* x *and* y*, then*

$$\tilde{S}_z = \begin{cases} \tilde{S}_x \cap \tilde{S}_y, & \text{if } \tilde{S}_x \cap \tilde{S}_y \neq \emptyset \\ \tilde{S}_x \cup \tilde{S}_y, & \text{otherwise} \end{cases}$$

2. *Top-Down: The goal of this part is to assign the state* $\tilde{s}_z$ *to every node* z *that minimizes the total number of state changes in the whole tree:*

   (a) *For the root* ρ *of the tree, assign any state in* $\tilde{S}_\rho$ *as* $\tilde{s}_\rho$.

   (b) *For any* $x \in V - \{\rho\}$*, suppose* $z \in V$ *is its parent, then*

   - *if* $\tilde{s}_z \in \tilde{S}_x$*, then* $\tilde{s}_x = \tilde{s}_z$.
   - *if* $\tilde{s}_z \notin \tilde{S}_x$*, then assign any state in* $\tilde{S}_x$ *as* $\tilde{s}_x$.

Let S be the set of all possible states. As explained in [65], in order to understand the accuracy of the Fitch algorithm, we need to know the distribution of $\tilde{S}_z$ given the original state $s_z = s \in S$. Since $\tilde{S}_z$ only depends on the states of the leaves below z, we can compute the exact conditional probability:

$$\mathbb{P}_z[S'|s] = \mathbb{P}[\tilde{S}_z = S'|s_z = s] = \sum_D \mathbb{P}[D|s_z = s]\mathbb{P}[\tilde{S}_z = S'|D]$$

where D_z denotes a set of states for the leaves below z. With this notation, the probability that the Fitch algorithm correctly reconstructs the root state, denoted as $RA_F(\mathcal{T})$, can be written as:

$$RA_F(\mathcal{T}) = \sum_{s,S'} \mathbb{P}[s_\rho = s]\mathbb{P}_\rho[S'|s]\mathbb{P}[s \text{ is selected from } S']$$

In the rest of this section, we assume that the number of states is 2, with $S = \{0, 1\}$. In other words, we consider the Cavender-Farris-Neyman model. Then there are only three possibilities for S': $\{0\}$, $\{1\}$, and $\{0, 1\}$. By symmetry, for any node $z \in V$, we make the following notations:

$$\begin{aligned} \alpha_z &= \mathbb{P}_z[\{0\}|0] = \mathbb{P}_z[\{1\}|1] \\ \beta_z &= \mathbb{P}_z[\{1\}|0] = \mathbb{P}_z[\{0\}|1] \end{aligned}$$

and so

$$\mathbb{P}_z[\{0, 1\}|0] = \mathbb{P}_z[\{0, 1\}|1] = 1 - \alpha_z - \beta_z$$

Then we are able to express $RA_F(\mathcal{T})$ in terms of α_ρ and β_ρ [65].

$$\begin{aligned} RA_F(\mathcal{T}) &= \sum_{s,S'} \mathbb{P}[s_\rho = s]\mathbb{P}_\rho[S'|s]\mathbb{P}[s \text{ is selected from } S'] \\ &= \sum_{s} \mathbb{P}[s_\rho = s](\alpha_\rho \cdot 1 + (1 - \alpha_\rho - \beta_\rho) \cdot \frac{1}{2}) \\ &= \alpha_\rho + (1 - \alpha_\rho - \beta_\rho) \cdot \frac{1}{2} \\ &= \frac{1}{2} + \frac{1}{2}(\alpha_\rho - \beta_\rho) \end{aligned} \tag{2.1}$$

In the Cavender-Farris-Neyman model, states are more likely to be preserved than be changed along the edges, and so $\alpha_\rho - \beta_\rho$ needs to be non-negative. The goal of estimating the accuracy of the Fitch algorithm in the Cavender-Farris-Neyman model is evaluating how far away $\alpha_\rho - \beta_\rho$ can be from 0. Nevertheless, it is far from trivial to estimate $\alpha_\rho - \beta_\rho$ from the states of leaves. Some useful recursive relations are needed for further analysis.

Suppose an interior node $z \in V$ has children x, y. Let p_x be the probability that the state of x is different from z, and $q_x = 1 - p_x$ is the probability that the state is

preserved. Define p_y, q_y in a similar way. The following recursive relation is introduced in [39], which uses the fact that $\tilde{S}_z = \{1\}$ if and only if one of $\tilde{S}_x$ and $\tilde{S}_y$ is $\{1\}$, and the other is $\{1\}$ or $\{0, 1\}$:

$$\begin{aligned} \alpha_z &= (q_x\alpha_x + p_x\beta_x)(q_y\alpha_y + p_y\beta_y) \\ &\quad +(q_x\alpha_x + p_x\beta_x)(1 - \alpha_y - \beta_y) \\ &\quad +(1 - \alpha_x - \beta_x)(q_y\alpha_y + p_y\beta_y) \end{aligned} \tag{2.2}$$

Similarly,

$$\begin{aligned} \beta_z &= (p_x\alpha_x + q_x\beta_x)(p_y\alpha_y + q_y\beta_y) \\ &\quad +(p_x\alpha_x + q_x\beta_x)(1 - \alpha_y - \beta_y) \\ &\quad +(1 - \alpha_x - \beta_x)(p_y\alpha_y + q_y\beta_y) \end{aligned} \tag{2.3}$$

The recursive relations are complicated for theoretical analysis. In [65], they derive the following relations, with assuming $C_z = 1 - \alpha_z - \beta_z$ and $D_z = \alpha_z - \beta_z$ for any node $z \in V$: (for further properties of C_z and D_z, see Lemma 2.17)

$$C_z = \frac{1}{2} \cdot [1 - C_x - C_y + 3C_xC_y - (1 - 2p_x)(1 - 2p_y)D_xD_y] \tag{2.4}$$

$$D_z = \frac{1}{2}(1 - 2p_x)(1 + C_y)D_x + \frac{1}{2}(1 - 2p_y)(1 + C_x)D_y \tag{2.5}$$

With this recursion, the following lemma has been shown in [58, 65].

Lemma 2.6 (Accuracy of the Fitch Algorithm on complete binary tree with equal edge length [58, 65])**.** *Let $\mathcal{T}_n$ be the complete binary tree of 2^n leaves in which the changing probability is p along all the edges. Then in the Cavender-Farris-Neyman model, the accuracy of the Fitch algorithm for reconstructing the root state in $\mathcal{T}_n$, $RA_F(\mathcal{T}_n)$, converges*

as n goes to infinity to:

$$\lim_{n\to\infty} RA_F(\mathcal{T}_n) = \begin{cases} \frac{1}{2} + \frac{\sqrt{(1-8p)(1-4p)}}{2(1-2p)^2} & , \text{ if } p \in [0, \frac{1}{8}] \\ \frac{1}{2} & , \text{ if } p \in [\frac{1}{8}, \frac{1}{2}] \end{cases}$$

In order to further generalize the result, we further simplify the recursive relation. We make the following notations:

Notation 2.7. *For any $z \in V$, we let $F_z = 1 - 3C_z = 3(\alpha_z + \beta_z) - 2$ and $G_z = D_z = \alpha_z - \beta_z$. Moreover, for any $x \in V - \{\rho\}$, we let $\theta_x = 1 - 2p_x$.*

Notation 2.8. *Suppose $z, x, y \in V$ in phylogenetic tree $\mathcal{T}$, such that z is the parent of x, y, we denote $s(x)$ as the sibling of x, that is, y, and similarly, $s(y) = x$.*

With these notations, we can re-write equation 2.4 and 2.5 into the following equations: (for further properties of F_z and G_z, see Lemma 2.18)

$$F_z = \frac{3}{2}\theta_x\theta_y G_x G_y - \frac{1}{2}F_x F_y \tag{2.6}$$

$$\begin{aligned} G_z &= (\frac{4 - F_y}{6})\theta_x G_x + (\frac{4 - F_x}{6})\theta_y G_y \\ &= (\frac{4 - F_{s(x)}}{6})\theta_x G_x + (\frac{4 - F_{s(y)}}{6})\theta_y G_y \end{aligned} \tag{2.7}$$

We can express equation 2.1 as

$$RA_F(\mathcal{T}) = \frac{1}{2} + \frac{1}{2}(\alpha_\rho - \beta_\rho) = \frac{1}{2} + \frac{1}{2}D_\rho = \frac{1}{2} + \frac{1}{2}G_\rho \tag{2.8}$$

Therefore, to understand the accuracy of the Fitch algorithm, we need to study the behaviour of D_ρ or G_ρ. Based on equation 2.7, we can express G_ρ as:

$$G_\rho = \sum_{x\in L}[\prod_{z\in a(x)-\{\rho\}}(\frac{4 - F^\pi_{s(z)}}{6})\theta_z]G_x = \sum_{x\in L}[\prod_{z\in a(x)-\{\rho\}}(\frac{4 - F^\pi_{s(z)}}{6})\theta_z] \tag{2.9}$$

where $a(x)$ is the set of all the ancestral nodes of x in the phylogenetic tree $\mathcal{T}$, including x itself. The last equal sign uses the fact that, as we will show in Lemma 2.18, $G_x = 1$ if $x \in L$ is a leaf. This is an important equation which inspires all our work in this section because we have a more explicit formula for G_ρ to study.

Besides the improved recursive relation, in order to understand the accuracy of the Fitch algorithm as the tree gets larger, we need a definition for a "growing" tree. One way to accomplish this is using the concept of **infinite tree**, which is a cycle-free graph with a countable number of vertices, and equipped it with **minimum cutset**:

Definition 2.9 (infinite phylogenetic tree)**.** *We say $\mathcal{T}_\infty = (V, E, \rho, w)$ is a weighted (rooted) infinite phylogenetic tree if (V, E, ρ) is an infinite tree. Again in this section we assume that all nodes, except ρ, must have degree $= 3$*

We can apply the Fitch algorithm on arbitrary minimum cutset of an infinite phylogenetic tree.

Definition 2.10 (minimum cutset [51])**.** *We say π is a minimum cutset of a weighted infinite phylogenetic tree $\mathcal{T}_\infty = (V, E, \rho, w)$ if $\pi \in V$, $|\pi|$ finite, and any infinite self-avoiding path from root ρ must pass through one and only one vertex in π.*

For any minimum cutset π of a weighted infinite phylogenetic tree $\mathcal{T}_\infty$, we can regard π as the set of leaves and apply the Fitch algorithm to reconstruct the state of the root ρ. What we want to know now is what conditions $\mathcal{T}_\infty$ needs in order to guarantee that, for any minimum cutset π of $\mathcal{T}_\infty$, the accuracy of the Fitch algorithm is always strictly larger than $1/2$ (because we assume that the number of state is 2 and we want the accuracy is better than a random guessing). We need the following notations:

Notation 2.11. *Suppose $\mathcal{T}_\infty = (V, E, \rho, w)$ is a weighted infinite phylogenetic tree, π is a minimum cutset of $\mathcal{T}_\infty$, and $\mathcal{X}^\pi$ is a set of taxa equipped with the labeling function $L^\pi : \mathcal{X}^\pi \to \pi$. We use $\mathcal{T}_\infty^\pi = (\mathcal{X}^\pi, V^\pi, E^\pi, \rho, L^\pi, w)$ to denote the weighted phylogenetic tree which is generated by the connected component of ρ after separating $\mathcal{T}_\infty$ by π. For every $z \in V^\pi$, we denote F_z^π and G_z^π as the F_z and G_z value defined in recursive equation 2.6 and 2.7 generated by applying the Fitch algorithm on $\mathcal{T}_\infty^\pi$.*

With this notation, we can re-write equation 2.9 into:

$$G_\rho^\pi = \sum_{x\in\pi}[\prod_{z\in a(x)-\{\rho\}}(\frac{4-F_{s(z)}^\pi}{6})\theta_z]G_x^\pi = \sum_{x\in\pi}[\prod_{z\in a(x)-\{\rho\}}(\frac{4-F_{s(z)}^\pi}{6})\theta_z] \tag{2.10}$$

again where $a(x)$ is the set of all the ancestral nodes of x in $\mathcal{T}_\infty$, including x itself. This inspires us to use the **branching number** for further analysis:

Definition 2.12 (branching number [51])**.** *We define the branching number of an infinite weighted phylogenetic tree $\mathcal{T}_\infty = (V, E, \rho, w)$ as:*

$$br(\mathcal{T}_\infty) = \sup\{\kappa : \inf_{cutset\ \pi}\sum_{x\in\pi}[\prod_{z\in a(x)-\{\rho\}}\kappa^{-1}\theta_z] > 0\} \tag{2.11}$$

Our first main theorem is:

Theorem 2.13 (Accuracy of the Fitch Algorithm on an infinite phylogenetic tree)**.** *Given an infinite weighted phylogenetic tree $\mathcal{T}_\infty = (V, E, \rho, w)$. If $br(\mathcal{T}_\infty) > \frac{3}{2}$ and there exists $\tau > 0$ such that $\theta_x \geq \tau$ for all $x \in V - \{\rho\}$, then there exists $\epsilon > 0$ such that for any cutset π of $\mathcal{T}_\infty$, $G_\rho^\pi > \epsilon$.*

The theorem is more flexible than the previous research because of the following two reasons. First, we do not require that all the edges have the same conservation probability. Second, and more importantly, we allow the conservation probability to be close to $\frac{1}{2}$ if other edges can support it.

Example 2.14 (On the infinite phylogenetic tree with bounded edge length)**.** *Suppose we have an infinite phylogenetic tree* $\mathcal{T}_\infty = (V, E, \rho, w)$ *such that the weight function* w *satisfies that*

$$\sup_{e\in E} w(e) > \frac{1}{2}\ln\frac{4}{3}\ , \ or \ \sup_{x\in V-\{\rho\}} p_x = p_0 < \frac{1}{8}$$

which implies that

$$\inf_{x\in V-\{\rho\}} \theta_x = \inf_{x\in V-\{\rho\}}(1-2p_x) = 1-2p_0 > \frac{3}{4} > 0$$

then for any minimum cutset π*,*

$$\sum_{x\in\pi}[\prod_{z\in a(x)-\{\rho\}}(\theta_z \times \frac{1}{2(1-2p_0)})] \geq \sum_{x\in\pi}[\prod_{z\in a(x)-\{\rho\}}(\frac{1}{2})] = 1 > 0$$

Hence, $br(\mathcal{T}_\infty) \geq 2(1-2p_0) > \frac{3}{2}$*. And, by our first theorem, there exists* $\epsilon > 0$ *such that* $G^\pi_\rho > \epsilon$ *for all minimum cutset* π*. In particular, if* $\mathcal{T}_\infty$ *satisfies* $p_x = p < \frac{1}{8}$ *for all* $x \in V - \{\rho\}$*, then there exists* $\epsilon > 0$ *such that* $G^\pi_\rho > \epsilon$ *for all minimum cutset* π*. This shows part of result in Lemma 2.6: If* $p \in [0, \frac{1}{8})$*, then there is an* $\epsilon > 0$ *such that* $RA_F(\mathcal{T}_n) > \frac{1}{2} + \epsilon$ *for all* n*.*

The weight function w, or the edge length, usually reflects the time difference between the parent and children. The simplest time-based phylogenetic model is the pure-birth Yule model introduced by Yule in 1924 [63], which assumes that each extant species randomly separates into two species at some constant rate λ:

Definition 2.15 (Yule Model, [63])**.** *An infinite phylogenetic tree* $\mathcal{T}_\infty = (V, E, \rho, w)$ *is a Yule model with rate* λ *if for all* $e \in E$*,* $w(e)$ *is an independent exponential distributed random variable with rate* λ*. We use* $\mathcal{T}_{Y:\lambda}$ *to denote the Yule model with rate* λ*.*

We can compute the expectation of θ_x for all $x \in V - \{\rho\}$ in a Yule model $\mathcal{T}_{Y:\lambda}$:

$$\begin{aligned}
\mathbb{E}\theta_x &= \int (1 - 2p_x)dp \\
&= \int_0^\infty [1 - 2 \cdot (\frac{1}{2}(1 - e^{-2t}))](\lambda e^{-\lambda t})dt \\
&= \int_0^\infty e^{-2t} \cdot (\lambda e^{-\lambda t})dt \\
&= \frac{\lambda}{\lambda + 2}
\end{aligned} \tag{2.12}$$

As we will show later,

$$br(\mathcal{T}_{Y:\lambda}) = 2\mathbb{E}\theta_* \; a.s. \tag{2.13}$$

And the computation above will lead us to our second main theorem :

Theorem 2.16 (Accuracy of Fitch Algorithm on Yule Model)**.** *Suppose* $\lambda > 6$*. For any* $\delta > 0$*, there is an* $\epsilon > 0$ *such that for any minimum cutset* π*,*

$$\mathbb{P}[\mathcal{T}_{Y:\lambda} : G_\rho^\pi > \epsilon] > 1 - \delta$$

The remaining section is organized in the following way: We prove our recursive relation in 2.3. Then in 2.4, we show the proof of our first main theorem, Theorem 2.13. Next in 2.5, we show our claim of equation 2.13. Last we will prove our second main theorem, Theorem 2.16, in 2.6.

2.3 Recursive relations

In this section, we assume $\mathcal{T} = (\mathcal{X}, V, E, \rho, L, w)$ is an edge-weighted phylogenetic tree. We will prove our recursive relation 2.6 and 2.7, and all the properties of F_z and G_z for any $z \in V$. The following recursive relations are introduced in [65], and lead us to our recursive relations.

Lemma 2.17 (Properties of C_z and D_z [65]). *For any $z \in V$, let $C_z = 1 - \alpha_z - \beta_z$ and $D_z = \alpha_z - \beta_z$, then*

1. *If $z \in L$ is a leaf, then $C_z = 0$ and $D_z = 1$.*

2. *C_* and D_* satisfy the following recursive relation: If z is an interior node with children x, y, then*

$$\begin{aligned} C_z &= \frac{1}{2} \cdot [1 - C_x - C_y + 3C_xC_y - (1-2p_x)(1-2p_y)D_xD_y] \\ D_z &= \frac{1}{2}(1-2p_x)(1+C_y)D_x + \frac{1}{2}(1-2p_y)(1+C_x)D_y \end{aligned}$$

3. *$p_x < \frac{1}{2}$ for all $x \in V - \{\rho\}$. As a result, $0 \le C_z \le \frac{1}{2}$ and $0 \le D_z \le 1$ for all $z \in V$.*

We will use the properties above to show the following lemma:

Lemma 2.18 (Properties of F_z and G_z). *For any $z \in V$, let $F_z = 1 - 3C_z = 1 - 3 \cdot (1 - \alpha_z - \beta_z) = 3(\alpha_z + \beta_z) - 2$ and $G_z = D_z = \alpha_z - \beta_z$, and assume $\theta_x = 1 - 2p_x$ for any $x \in V - \{\rho\}$, then*

1. *If $z \in L$ is a leaf, then $F_z = 1$ and $G_z = 0$.*

2. *F_* and G_* satisfy the following recursive relation: If z is an interior node with children x, y, then*

$$\begin{aligned} F_z &= \frac{3}{2}\theta_x\theta_y G_x G_y - \frac{1}{2}F_xF_y \\ G_z &= (\frac{4-F_y}{6})\theta_x G_x + (\frac{4-F_x}{6})\theta_y G_y \end{aligned}$$

3. *$p_x < \frac{1}{2}$ for all $x \in V - \{\rho\}$. As a result, $\theta_z > 0$ for all $x \in V - \{\rho\}$, and $-\frac{1}{2} \le F_z \le 1$ and $0 \le G_z \le 1$ for all $z \in V$.*

Proof. The proof is based on the property of C_* and D_* from Lemma 2.17:

1. If z is a leaf, we have $C_z = 0$ and $D_z = 1$, and thus $F_z = 1 - 3C_z = 1$ and $G_z = D_z = 1$.

2. By the recursive relation (2.4) and (2.5):

$$\begin{aligned}
F_z &= 1 - 3C_z \\
&= 1 - \frac{3}{2}[1 - C_x - C_y + 3C_xC_y - (1-2p_x)(1-2p_y)D_xD_y] \\
&= 1 - \frac{3}{2}[1 - C_x - C_y + 3C_xC_y] + \frac{3}{2}\theta_x\theta_y G_xG_y \\
&= 1 - \frac{1}{2}[3 - 3C_x - 3C_y + 9C_xC_y] + \frac{3}{2}\theta_x\theta_y G_xG_y \\
&= 1 - \frac{1}{2}[2 + (1-3C_x)(1-3C_y)] + \frac{3}{2}\theta_x\theta_y G_xG_y \\
&= 1 - 1 - \frac{1}{2}F_xF_y + \frac{3}{2}\theta_x\theta_y G_xG_y = \frac{3}{2}\theta_x\theta_y G_xG_y - \frac{1}{2}F_xF_y \\
G_z &= D_z \\
&= \frac{1}{2}(1-2p_x)(1+C_y)D_x + \frac{1}{2}(1-2p_y)(1+C_x)D_y \\
&= \frac{1}{2}(1+C_y)\theta_x G_x + \frac{1}{2}(1+C_x)\theta_y D_y \\
&= \frac{1}{2}(1+\frac{1-F_y}{3})\theta_x G_x + \frac{1}{2}(1+\frac{1-F_x}{3})\theta_y D_y \\
&= (\frac{4-F_y}{6})\theta_x G_x + (\frac{4-F_x}{6})\theta_y D_y
\end{aligned}$$

3. $\theta_x = 1 - 2p_x > 0$ can be proved right from $p_x < \frac{1}{2}$ for all $x \in V - \{\rho\}$. Moreover, because $0 \le C_z \le \frac{1}{2}$ and $0 \le D_z \le 1$, we have $-\frac{1}{2} \le F_z = 1 - 3C_z \le 1$ and $0 \le G_z = D_z \le 1$ for all $z \in V$.

□

As we mentioned in equation 2.8, higher G_z or D_z value means that the Fitch algorithm has a higher probability of predicting the state of z. A straightforward claim

is, if the child has a larger G value, which means we have a higher chance of predicting the state for the child, then we should have a higher chance of predicting the state for the parent, that is, the parent should also have a larger G value. The statement is, unfortunately, not always true. Figure 2.3 is a counterexample. In graphs, $p_* = 0$ implies $\theta_* = 1 - 2 \cdot 0 = 1$, and $p_* = 0.1$ implies $\theta_* = 1 - 2 \cdot 0.1 = 0.8$. According to Lemma 2.18, we have in both graphs:

$$\begin{aligned} F_y &= \frac{3}{2} \cdot 1 \cdot 1 \cdot 1 \cdot 1 - \frac{1}{2} \cdot 1 \cdot 1 = 1 \\ G_y &= \frac{4-1}{6} \cdot 1 \cdot 1 + \frac{4-1}{6} \cdot 1 \cdot 1 = 1 \end{aligned}$$

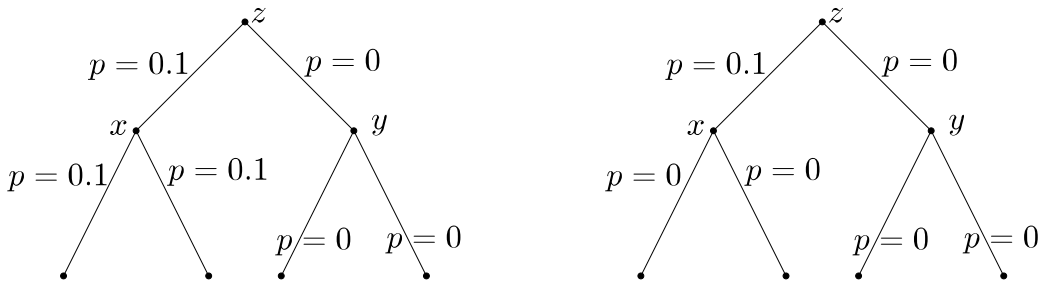
And for z and x, in graph 1(a),

$$\begin{aligned} F_x &= \frac{3}{2} \cdot 0.8 \cdot 0.8 \cdot 1 \cdot 1 - \frac{1}{2} \cdot 1 \cdot 1 = 0.46 \\ G_x &= \frac{4-1}{6} \cdot 0.8 \cdot 1 + \frac{4-1}{6} \cdot 0.8 \cdot 1 = 0.8 \\ G_z &= \frac{4-1}{6} \cdot 0.8 \cdot 0.8 + \frac{4-0.46}{6} \cdot 1 \cdot 1 = 0.91 \end{aligned}$$

while in graph 1(b),

$$\begin{aligned} F_x &= \frac{3}{2} \cdot 1 \cdot 1 \cdot 1 \cdot 1 - \frac{1}{2} \cdot 1 \cdot 1 = 1 \\ G_x &= \frac{4-1}{6} \cdot 1 \cdot 1 + \frac{4-1}{6} \cdot 1 \cdot 1 = 1 \\ G_z &= \frac{4-1}{6} \cdot 0.8 \cdot 1 + \frac{4-1}{6} \cdot 1 \cdot 1 = 0.9 \end{aligned}$$

As we can see, graph 1(b) has larger G_x but smaller G_z than graph 1(b), which means graph 1(b) has a higher chance of reconstructing the state of x, but a lower chance of reconstructing the state of z.



(a) Figure A

(b) Figure B

Figure 1: An example shows that higher probability of getting the right children state does not imply higher probability of getting the right parent state. Given the node y are equal in both graph, graph (b) has larger G_x value than graph (a) but has smaller G_z value than graph (a).

2.4 On an infinite phylogenetic tree

In this section, we are going to prove our first theorem:

Theorem 2.13 (Accuracy of the Fitch Algorithm on an infinite phylogenetic tree)**.** *Given an infinite weighted phylogenetic tree $\mathcal{T}_\infty = (V, E, \rho, w)$. If $br(\mathcal{T}_\infty) > \frac{3}{2}$ and there exists $\tau > 0$ such that $\theta_x \geq \tau$ for all $x \in V - \{\rho\}$, then there exists $\epsilon > 0$ such that for any cutset π of $\mathcal{T}_\infty$, $G^\pi_\rho > \epsilon$.*

We use the following notation for the proof:

Notation 2.19. *Suppose π is a minimum cutset of $\mathcal{T}_\infty$. For any $z \in V^\pi$, we use $\mathcal{L}^\pi_{z,N} \subset V^\pi$ to represent the set consisting of all the descendants of z which are exactly N-levels lower from z, and use $\mathcal{M}^\pi_{z,N} = \cup^N_{n=1}\mathcal{L}^\pi_{z,n} \subset V^\pi$ to denote the set which contains all the descendants of z up to N-level lower from z.*

We use proof by contradiction, and the sketch of the proof is as follows. If there is a

minimum cutset π such that G_ρ^π is too small, then, as we will show later, for any $x \in V^\pi$ that is not too far away from the root ρ, G_x^π needs to be small as well. In other words, there is a minimum cutset $\pi' \subset V^\pi$ such that the sub-tree $\mathcal{T}_\infty^{\pi'}$ of $\mathcal{T}_\infty^\pi$ satisfies that $G_{x'}^\pi$ is small for any $x' \in V^{\pi'}$.

Next, for any node $x' \in V^{\pi'}$, because $G_{x'}^\pi$ is small, any descendant of x' which is not too far away from x' needs to have a small G value as well. From recursive formula 2.6 we know that

$$F_z^\pi \approx -\frac{1}{2} F_x^\pi F_y^\pi$$

which should go to 0 pretty fast. Therefore, any node $x' \in V^{\pi'}$ should have small $F_{x'}^\pi$ as well. If we choose the value properly, the boundary of $F_{x'}^\pi$ can be small enough such that the coefficient term in recursive formula 2.7, $\frac{4-F_{s(x')}^\pi}{6}$, should be large enough to exceed the branching number of $\mathcal{T}_\infty$. This phenomenon along with the definition of the branching number limits the lower bound of G_ρ^π, which causes a contradiction and ends the proof.

To make concrete the proof, we need the following two lemmas. The corollary of the first lemma shows that if G_z^π is small and a descendant $x \in V^\pi$ is not too far away from the z, then G_x^π needs to be small as well.

Lemma 2.20 (Upper bound of G for a child)**.** *Given an infinite phylogenetic tree $\mathcal{T}_\infty = (V, E, \rho, w)$ and a minimum cutset π of $\mathcal{T}_\infty$. If $x \in V^\pi$ is a child of $z \in V^\pi$ and $\theta_x \geq \tau > 0$, then*

$$G_x^\pi \leq (\frac{2}{\tau}) G_z^\pi \tag{2.14}$$

Proof. Suppose $y \in V^\pi$ is the other child of z. From the recursion formula 2.7 and the

fact that $F_y^\pi \leq 1$ from lemma 2.18, we have:

$$G_z^\pi \geq (\frac{4 - F_y^\pi}{6})\theta_x G_x^\pi \geq (\frac{4-1}{6})\tau G_x^\pi = \frac{\tau}{2} G_x^\pi$$

and the statement follows. □

The following corollary is obtained immediately by recursively applying lemma 2.20:

Corollary 2.21 (Upper bound of G for a descendent)**.** *Given an infinite phylogenetic tree $\mathcal{T}_\infty = (V, E, \rho, w)$ and a minimum cutset π of $\mathcal{T}_\infty$. If $x \in \mathcal{L}_{z,N}^\pi$ for some $z \in V^\pi$ and positive integer N, and $\theta_{x'} \geq \tau > 0$ for all x' on the path from z to x, or $a(x) - a(z)$, then*

$$G_x^\pi \leq (\frac{2}{\tau})^N G_z^\pi$$

The previous corollary proves an important claim in the sketch. Moreover, we will use it to prove another important claim in the sketch, that if $G_{x'}^\pi$ is small, then $F_{x'}^\pi$ is small.

Lemma 2.22 (Lemma: FA-TIP-Upper bound of F)**.** *Define function $g_{\epsilon',\tau} : \mathbb{Z}^+ \to \mathbb{R}^+$:*

$$g_{\epsilon',\tau}(n) = (\frac{2}{\tau})^n \epsilon'$$

Given an infinite phylogenetic tree $\mathcal{T}_\infty = (V, E, \rho, w)$ and a minimum cutset π of $\mathcal{T}_\infty$. Suppose vertex $z \in V^\pi$, constants $1 > \epsilon' > 0$, $1 \geq \tau > 0$, $\phi > 0$, and positive integer N satisfy:

1. $[g_{\epsilon',\tau}(N)]^2 \leq \min\{\frac{1}{3}, \frac{2}{3}(4\phi - 2^{1-2^N})\}$
2. $G_z^\pi \leq \epsilon'$
3. *For any $x \in \mathcal{M}_{z,N}^\pi$, $\theta_x \geq \tau$*

then $|F_z^\pi| \le 4\phi$.

Proof. First, notice that the first condition leads to the following two inequalities:

$$\frac{3}{2}\cdot(\frac{\tau}{2})^2+\frac{9}{8}\cdot[g_{\epsilon',\tau}(N)]^2+\frac{3}{4} \le \frac{3}{2}\cdot(\frac{1}{2})^2+\frac{9}{8}\cdot\frac{1}{3}+\frac{3}{4}=\frac{3}{2} \tag{2.15}$$

$$\frac{3}{2}\cdot[g_{\epsilon',\tau}(N)]^2+2^{1-2^N} \le \frac{3}{2}\cdot[\frac{2}{3}(4\phi-2^{1-2^N})]+2^{1-2^N}=4\phi \tag{2.16}$$

The idea of this proof is using induction. We will show that for any integer n from $N-1$ to 0, F_x^π has an upper bound for every $x \in \mathcal{L}_{z,n}^\pi$ (For convenience, we assume $\mathcal{L}_{z,0}^\pi = \{z\}$). The recursive formula that we are using is definitely equation 2.6, and so we need to make sure for every $x \in \mathcal{M}_{z,N}^\pi \cup \{z\}$, x is not a leaf of $\mathcal{T}_\infty^\pi$ and so it has children. $z \notin \pi$ since $G_z^\pi \le \epsilon' < 1$, and the proof of $\mathcal{M}_{z,N}^\pi \cap \pi = \emptyset$ is from Corollary 2.21: For any $x \in \mathcal{M}_{z,N}^\pi$,

$$G_x^\pi \le (\frac{2}{\tau})^N\epsilon' = g_{\epsilon',\tau}(N) \le \sqrt{\frac{1}{3}} < 1$$

Because $G_x^\pi \ne 1$, so $x \notin \pi$ by Lemma 2.18. Therefore, we can apply equation 2.6 to any $x \in \mathcal{M}_{z,N}^\pi \cup \{z\}$.

For $0 \le n \le N$, let

$$F_{z,n}^\pi = \sup_{x\in\mathcal{L}_{z,n}^\pi}|F_x^\pi|, \quad G_{z,n}^\pi = \sup_{x\in\mathcal{L}_{z,n}^\pi}|G_x^\pi|$$

By Corollary 2.21, we have an upper bound for $G_{z,n}^\pi$ for any $1 \le n \le N$:

$$G_{z,n}^\pi \le (\frac{2}{\tau})^n G_z^\pi \le (\frac{2}{\tau})^n\epsilon' = g_{\epsilon',\tau}(n)$$

Hence, by recursive relation 2.6, for any $1 \le n \le N$,

$$F_{z,n-1}^\pi \le \frac{3}{2}(G_{z,n}^\pi)^2+\frac{1}{2}(F_{z,n}^\pi)^2 \le \frac{3}{2}[g_{\epsilon',\tau}(n)]^2+\frac{1}{2}(F_{z,n}^\pi)^2$$

We will use the inequality above to show that for all $0 \le n \le N-1$

$$F_{z,n}^\pi \le \frac{3}{2}[g_{\epsilon',\tau}(N)]^2+2^{1-2^{N-n}}$$

by induction from $n = N-1$ toward $n = 0$. The base case is when $n = N-1$. Recall that $F^{\pi}_{z,N}$ has a natural upper bound 1 from Lemma 2.18, so

$$F^{\pi}_{z,N-1} \leq \frac{3}{2}[g_{\epsilon',\tau}(N)]^2 + \frac{1}{2} = \frac{3}{2}[g_{\epsilon',\tau}(N)]^2 + 2^{1-2^{N-(N-1)}}$$

and the inequality holds. Now suppose the inequality is true in the $n = k$ case for some $1 \leq k \leq N-1$, then for $n = k-1$:

$$\begin{aligned}
F^{\pi}_{z,k-1} &\leq \frac{3}{2}[g_{\epsilon',\tau}(k)]^2 + \frac{1}{2}(F^{\pi}_{z,k})^2 \\
&\leq \frac{3}{2}[g_{\epsilon',\tau}(N-1)]^2 + \frac{1}{2}\cdot(\frac{3}{2}([g_{\epsilon',\tau}(N)]^2 + 2^{1-2^{N-k}})^2 \\
&= \frac{3}{2}\cdot(\frac{\tau}{2}[g_{\epsilon',\tau}(N)])^2 + \frac{1}{2}\cdot(\frac{3}{2}([g_{\epsilon',\tau}(N)]^2)^2 \\
&\quad + \frac{1}{2}\cdot 2\cdot(\frac{3}{2}([g_{\epsilon',\tau}(N)]^2))(2^{1-2^{N-k}}) + \frac{1}{2}\cdot(2^{1-2^{N-k}})^2 \\
&= \frac{3}{2}\cdot(\frac{\tau}{2})^2\cdot[g_{\epsilon',\tau}(N)]^2 + \frac{1}{2}\cdot\frac{9}{4}\cdot[g_{\epsilon',\tau}(N)]^4 \\
&\quad + \frac{3}{2}\cdot(2^{1-2^{N-k}})\cdot[g_{\epsilon',\tau}(N)]^2 + 2^{-1}\cdot 2^{2-2^{N-k+1}} \\
&\leq [g_{\epsilon',\tau}(N)]^2\cdot\{\frac{3}{2}\cdot(\frac{\tau}{2})^2 + \frac{9}{8}\cdot[g_{\epsilon',\tau}(N)]^2 + \frac{3}{4}\} + 2^{1-2^{N-k+1}} \\
&\leq [g_{\epsilon',\tau}(N)]^2\cdot\frac{3}{2} + 2^{1-2^{N-k+1}}
\end{aligned}$$

where the last inequality comes from inequality 2.15. Thus, the inequality holds for all $0 \leq n \leq N-1$. In particular, for the $n = 0$ case, along with inequality 2.16, we have

$$|F^{\pi}_{z}| = F^{\pi}_{z,0} \leq \frac{3}{2}[g_{\epsilon',\tau}(N)]^2 + 2^{1-2^{N-0}} \leq 4\phi$$

and this finishes the proof of the lemma. □

With these two lemmas, we are able to prove our first main theorem:

Theorem 2.13 (Accuracy of the Fitch Algorithm on an infinite phylogenetic tree)**.** *Given an infinite weighted phylogenetic tree $\mathcal{T}_\infty = (V, E, \rho, w)$. If $br(\mathcal{T}_\infty) > \frac{3}{2}$ and there*

exists $\tau > 0$ such that $\theta_x \geq \tau$ for all $x \in V - \{\rho\}$, then there exists $\epsilon > 0$ such that for any cutset π of $\mathcal{T}_\infty$, $G_\rho^\pi > \epsilon$.

Proof. Because $br(\mathcal{T}_\infty) > \frac{3}{2}$, we can find an $\phi > 0$ such that $\frac{3}{2} \cdot \frac{1}{1-\phi} < br(\mathcal{T}_\infty)$. By definition of the branching number, there exists $\zeta > 0$ such that for any minimum cutset π of $\mathcal{T}_\infty$:

$$\sum_{x \in \pi} (\prod_{z \in a(x) - \{\rho\}} \frac{2}{3}(1-\phi)\theta_z) > \zeta$$

Because $\phi > 0$, there exists a positive integer N so that $2^{1-2^N} < 4\phi$, and so we can choose $1 > \epsilon' > 0$ such that

$$[g_{\epsilon',\tau}(N)]^2 = ((\frac{2}{\tau})^N \epsilon')^2 < \min\{\frac{1}{3}, \frac{2}{3}(4\phi - 2^{1-2^N})\}$$

Since ζ, τ, and ϵ' are all positive, there exists $\epsilon > 0$ such that

$$\epsilon < \min\{\zeta, 1\} \cdot (\frac{\tau}{2}\epsilon')$$

We will show that $G_\rho^\pi > \epsilon$ for any minimum cutset π of $\mathcal{T}_\infty$.

The proof is based on contradiction. If there is a minimum cutset π of $\mathcal{T}_\infty$ such that $G_\rho^\pi \leq \epsilon$, we construct another minimum cutset π' of $\mathcal{T}_\infty$ in the following way: π' consists of every first vertex x' on the path from ρ to some $x \in \pi$ such that $G_{x'}^\pi > \frac{\tau}{2}\epsilon'$. That is, we define

$$\pi' = \{x' \in a(x) : x \in \pi, G_{x'}^\pi > \frac{\tau}{2}\epsilon_1, \text{ and } \forall z' \in a(x') - \{x'\}, G_{z'}^\pi \leq \frac{\tau}{2}\epsilon_1\}$$

We need to show π' is a minimum cutset of $\mathcal{T}_\infty$. Because π is a minimum cutset, any infinite path from ρ will visit at least one vertex $x \in \pi$, and hence it will pass at least one vertex $x' \in \pi'$. Moreover, because we only put the first vertex x' whose $G_{x'}^\pi$ exceeds

$\frac{\tau}{2}\epsilon'$, any infinite path from ρ will pass through only one vertex in π'. Therefore, π' is a minimum cutset.

For every vertex $x' \in V^{\pi'} - \{\rho\}$, the parent of x', say z', satisfies $G^{\pi}_{z'} \leq \frac{\tau}{2}\epsilon'$ from the definition of π'. and by Lemma 2.20, we have $G^{\pi}_{x'} \leq (\frac{2}{\tau})(\frac{\tau}{2}\epsilon') = \epsilon'$. In other words, for any $x' \in V^{\pi'} - \{\rho\}$, we have

$$G^{\pi}_{x'} \leq \epsilon' \text{ with } [g_{\epsilon',\tau}(N)]^2 < \min\{\frac{1}{3}, \frac{2}{3}(4\phi - 2^{1-2^N})\}$$

Hence, by Lemma 2.22, we know that $F^{\pi}_{x'} \leq 4\phi$, and this shows $F^{\pi}_{s(x')} \leq 4\phi$ for every $x' \in V^{\pi'} - \{\rho\}$ as well. Therefore, by the definition of branching number:

$$\begin{aligned}
\epsilon \geq G^{\pi}_{\rho} &= \sum_{x' \in \pi'} \Big(\prod_{z' \in a(x') - \{\rho\}} \frac{4 - F^{\pi}_{s(z')}}{6} \theta_{z'} \Big) G^{\pi}_{x'} \\
&> \frac{\tau}{2}\epsilon' \sum_{x' \in \pi'} \Big(\prod_{z' \in a(x') - \{\rho\}} \frac{4 - F^{\pi}_{s(z')}}{6} \theta_{z'} \Big) \\
&\geq \frac{\tau}{2}\epsilon' \sum_{x' \in \pi'} \Big(\prod_{z' \in a(x') - \{\rho\}} \frac{4 - 4\phi}{6} \theta_{z'} \Big) \\
&= \frac{\tau}{2}\epsilon' \sum_{x' \in \pi'} \Big(\prod_{z' \in a(x') - \{\rho\}} \frac{2}{3}(1 - \phi) \theta_{z'} \Big) \\
&> \frac{\tau}{2}\epsilon'\zeta \geq \epsilon
\end{aligned}$$

This causes a contradiction. Hence, $G^{\pi}_{\rho} > \epsilon$. □

2.5 Branching Number of Yule Model

In this section, we will prove the following proposition, which is essential in the proof of our second theorem.

Proposition 2.23 (Branching Number of Yule Model)**.** *If $\lambda > 1$, then*

$$br(\mathcal{T}_{Y:\lambda}) = \frac{2\lambda}{\lambda + 2} \; a.s.$$

For the remainder of this section, we use $\mathcal{T}_{Y:\lambda} = (V, E, \rho, w)$ to denote a pure-birth Yule Model with rate λ. And for every $x \in V - \{\rho\}$, we use p_x to denote the probability that the state of x is different from its parent in the Cavender-Farris-Neyman model, and $\theta_x = 1 - 2p_x$. That is,

$$\begin{aligned} p_x &= \frac{1}{2}(1 - e^{-2w(x)}) \\ \theta_x &= e^{-2w(x)} \end{aligned}$$

as we shown in equation 2.12, $\mathbb{E}\theta_* = \frac{\lambda}{\lambda+2}$. Hence, our goal is to prove

$$br(\mathcal{T}_{Y:\lambda}) = 2\mathbb{E}\theta_* \; a.s.$$

The way we prove Proposition 2.23 is by showing

$$\mathbb{P}[br(\mathcal{T}_{Y:\lambda}) = \kappa] = 1 \text{ for some } \kappa > 0,$$

and both

$$\kappa \leq 2\mathbb{E}\theta_* \text{ and } \kappa \geq 2\mathbb{E}\theta_*$$

We will use the following definitions and lemmas, which are similar to those in [51]. The main tool is the concept of **inherited**:

Definition 2.24 (Inherited [51])**.** *We say a property A of $\mathcal{T}_{Y:\lambda}$ is inherited if whenever $\mathcal{T}_{Y:\lambda}$ has A, both immediate descendant subtrees have property A.*

An important and useful lemma of inherited is:

Lemma 2.25 (0-1 law of inherited event)**.** *If property A is inherited, then*

$$\mathbb{P}[\mathcal{T}_{Y:\lambda} \in A] \in \{0, 1\}$$

Proof. Let $\mathcal{T}_1$, $\mathcal{T}_2$ be the immediate descendant subtrees of $\mathcal{T}_{Y:\lambda}$. Since A is inherited:

$$\mathbb{P}[\mathcal{T}_{Y:\lambda} \in A] \leq \mathbb{P}[\mathcal{T}_1, T_2 \in A] = \mathbb{P}[T_1 \in A]\mathbb{P}[T_2 \in A]$$

Because $\mathbb{P}[\mathcal{T}_{Y:\lambda} \in A] = \mathbb{P}[T_1 \in A] = \mathbb{P}[T_2 \in A]$, we have:

$$\mathbb{P}[\mathcal{T}_{Y:\lambda} \in A] \leq (\mathbb{P}[\mathcal{T}_{Y:\lambda} \in A])^2$$

which shows that $\mathbb{P}[\mathcal{T}_{Y:\lambda} \in A] \in \{0, 1\}$ □

On the other hand, there is an inherited property of branching number.

Lemma 2.26 (Inherited property of branching number)**.** *For any $\kappa > 0$, the event*

$$\{br(\mathcal{T}_{Y:\lambda}) \leq \kappa\}$$

is inherited.

Proof. Let $\mathcal{T}_1$, $\mathcal{T}_2$ be the immediate descendant subtrees of $\mathcal{T}_{Y:\lambda}$, and ρ_1, ρ_2 be the root of T_1 and T_2. Assume that $br(\mathcal{T}_{Y:\lambda}) \leq \kappa$. If $br(\mathcal{T}_1) = \kappa_0 > \kappa$, then by definition of branching number, with $\kappa < \kappa_1 = \kappa + \frac{(\kappa_0 - \kappa)}{2} < \kappa_0$, there exists $\zeta > 0$ such that for any minimum cutset π_1 of $\mathcal{T}_1$, we have

$$\sum_{x \in \pi_1} \Big(\prod_{z \in a(x) - \{\rho\}} \kappa_1^{-1} \theta_z \Big) > \zeta$$

On the other hand, for any minimum cutset π of $\mathcal{T}_{Y:\lambda}$, let π_1, π_2 be the corresponding minimum cutset of $\mathcal{T}_1$ and T_2, then

$$\begin{aligned} & \sum_{x \in \pi} \Big(\prod_{w \in a(x) - \{\rho\}} \kappa_1^{-1} \theta_z \Big) \\ = \; & \kappa_1^{-1} \theta_{\rho_1} \sum_{x \in \pi_1} \Big(\prod_{w \in a(x) - \{\rho\}} \kappa_1^{-1} \theta_z \Big) + \kappa_1^{-1} \theta_{\rho_2} \sum_{x \in \pi_2} \Big(\prod_{w \in a(x) - \{\rho\}} \kappa_1^{-1} \theta_z \Big) \\ = \; & \kappa_1^{-1} \theta_{\rho_1} \zeta + 0 > 0 \end{aligned}$$

and thus $br(\mathcal{T}_{Y:\lambda}) \geq \kappa_1 > \kappa$, which causes a contradiction. Hence, $br(\mathcal{T}_1) \leq \kappa$. Similarly, $br(\mathcal{T}_2) \leq \kappa$, and so the property $\{br(\mathcal{T}_{Y:\lambda}) \leq \kappa\}$ is inherited. □

Combining the previous two lemmas, we know there exists κ such that $\mathbb{P}[br(\mathcal{T}_{Y:\lambda}) = \kappa] = 1$. The remaining goal is to prove that $\kappa = 2\mathbb{E}\theta_*$. The proof uses the definition of Bernoulli(p) percolation, cluster and critical probability of any weighted infinite tree.

Definition 2.27 (Bernoulli(p) percolation [51])**.** *Given $p \in [0,1]$ and a weighted Yule model $\mathcal{T}_{Y:\lambda} = (V, E, \rho, w)$, the Bernoulli(p) percolation on $\mathcal{T}_{Y:\lambda}$ is a random subgraph of $\mathcal{T}_{Y:\lambda}$ obtained by independently including each original edge $e = (z, x) \in E$ (z is the parent of x) with probability $p \cdot \theta_x$, and discarding it with probability $1 - p \cdot \theta_x$. We call the former case an open edge and the latter one a closed edge, and we use $\mathbb{P}_{p,\mathcal{T}_{Y:\lambda}}$ to denote the probability measure of the Bernoulli(p) percolation measure on $\mathcal{T}_{Y:\lambda}$.*

Definition 2.28 (Cluster [51])**.** *The connected components of the open edges in percolation are called clusters. We use $C(z)$ to denote the cluster which contains z for any node z.*

Definition 2.29 (Critical probability [51])**.** *The critical probability of $\mathcal{T}_{Y:\lambda} = (V, E, \rho, w)$ is :*

$$p_c(\mathcal{T}_{Y:\lambda}) = \inf\{p : \mathbb{P}_{p,\mathcal{T}_{Y:\lambda}}[|C(\rho)| = \infty] > 0\}$$

Lemma 2.30 (The event that there is an infinite path from ρ is inherited)**.** *Suppose $p > 0$. The event*

$$\{\mathbb{P}_{p,\mathcal{T}_{Y:\lambda}}[|C(\rho) = \infty|] = 0\}$$

is inherited.

Proof. Suppose ρ_1, ρ_2 are children of root ρ, and so they are the root of immediately descendant subtrees. Because in $\mathbb{P}_{p,\mathcal{T}_{Y:\lambda}}$, almost surely both edges (ρ, ρ_1) and (ρ, ρ_2) have a positive probability to be open, we know that $\mathbb{P}_{p,\mathcal{T}_{Y:\lambda}}[|C(\rho)| = \infty] = 0$ implies both $\mathbb{P}_{p,\mathcal{T}_{Y:\lambda}}[|C(\rho_1)| = \infty] = 0$ and $\mathbb{P}_{p,\mathcal{T}_{Y:\lambda}}[|C(\rho_2)| = \infty] = 0$, and the statement follows. □

The following lemma describes the relation between percolation and branching number.

Lemma 2.31 (Relation between branching number and critical probability)**.** *For any* $\mathcal{T}_{Y:\lambda}$, $br(\mathcal{T}_{Y:\lambda}) \geq (p_c(\mathcal{T}_{Y:\lambda}))^{-1}$

Proof. We denote the event that vertex x, y are connected by a path after performing the percolation by $\{x \leftrightarrow y\}$. For any minimum cutset π, after performing the percolation, if $|C(\rho)| = \infty$, then ρ needs to connect to at least one of the vertex x in π. Notice $\{\rho \leftrightarrow x\}$ if and only if the path connecting these two points is made up of all open edges. Therefore,

$$\mathbb{P}_{p,\mathcal{T}_{Y:\lambda}}[|C(\rho)| = \infty] \leq \sum_{x\in\pi} \mathbb{P}_{p,\mathcal{T}_{Y:\lambda}}[\rho \leftrightarrow x] = \sum_{x\in\pi} \prod_{z\in a(x)-\{\rho\}} p \cdot \theta_x$$

For any $p_0 \leq (br(\mathcal{T}_{Y:\lambda}))^{-1}$, by definition of the branching number, there exists a sequence of minimum cutset $\{\pi_n\}$ such that as $n \to \infty$,

$$\sum_{x\in\pi_n} \prod_{z\in a(x)-\{\rho\}} p_0 \cdot \theta_x \to 0$$

which means $\mathbb{P}_{p_0,\mathcal{T}_{Y:\lambda}}[|C(\rho)| = \infty] = 0$, and so $p_0 \leq p_c(\mathcal{T}_{Y:\lambda})$. Thus, $(br(\mathcal{T}_{Y:\lambda}))^{-1} \leq p_c(\mathcal{T}_{Y:\lambda})$ and the lemma follows. □

Now we have all the tools to prove our proposition.

Proposition 2.23 (Branching Number of Yule Model)**.** *If $\lambda > 1$, then*

$$br(\mathcal{T}_{Y:\lambda}) = \frac{2\lambda}{\lambda + 2}\ a.s.$$

Proof. By Lemma 2.25 and Lemma 2.26, there exists κ such that $\mathbb{P}[br(\mathcal{T}_{Y:\lambda}) = \kappa] = 1$. We will now show both $\kappa \leq 2\mathbb{E}\theta_*$ and $\kappa \geq 2\mathbb{E}\theta_*$ to prove that $\kappa = 2\mathbb{E}\theta_*$.

1. First we will prove that $\kappa \leq 2\mathbb{E}\theta_*$. We show it by contradiction. If $\kappa = 2\xi\mathbb{E}\theta_*$ for some $\xi > 1$, then there exists $\epsilon > 0$ and $\delta > 0$ such that

$$\mathbb{P}[\mathcal{T}_{Y:\lambda} : \inf_{\pi} \sum_{x\in\pi} \prod_{z\in a(x)-\{\rho\}} (2\xi\mathbb{E}\theta_*)^{-1}\theta_z > \epsilon] > \delta \tag{2.17}$$

   and so for any minimum cutset π,

$$\mathbb{E}[\sum_{x\in\pi} \prod_{z\in a(x)-\{\rho\}} (2\xi\mathbb{E}\theta_*)^{-1}\theta_z] > \epsilon\delta \tag{2.18}$$

   However, consider the following minimum cutset sequence $\{\pi_n\}$: π_n consists of all vertices which are n-level down from the root ρ (so $\mathcal{T}_{Y:\lambda}^{\pi_n}$ is a perfect tree), then:

$$\begin{aligned}
&\mathbb{E}[\sum_{x\in\pi_n} \prod_{z\in a(x)-\{\rho\}} (2\xi\mathbb{E}\theta_*)^{-1}\theta_z] \\
=\ &\sum_{x\in\pi_n} \prod_{z\in a(x)-\{\rho\}} (2\xi\mathbb{E}\theta_*)^{-1}(\mathbb{E}\theta_z) \\
=\ &\sum_{x\in\pi_n} \prod_{z\in a(x)-\{\rho\}} (2\xi)^{-1} \\
=\ &\sum_{x\in\pi_n} (2\xi)^{-n} = 2^n(2\xi)^{-n} = (\xi)^{-n}
\end{aligned}$$

   which goes to 0 as $n \to \infty$ because $\xi > 1$. This causes a contradiction, and so $\kappa \leq 2\mathbb{E}\theta_*$.

2. Next, we will prove that $\kappa \geq 2\mathbb{E}\theta_*$. By the assumption that $\lambda > 1$, we have $(2\mathbb{E}\theta)^{-1} < 1$, and therefore there is some p such that $(2\mathbb{E}\theta)^{-1} < p \leq 1$. For any p

satisfying $(2\mathbb{E}\theta)^{-1} < p \leq 1$, perform the Bernoulli(p) percolation on $\mathcal{T}_{Y:\lambda}$ and let $\tilde{\mathcal{T}_{Y:\lambda}}$ be the connected component of the root ρ. Since the number of children for each vertex after percolation is $2 \cdot \mathbb{E}\theta_* \cdot p > 1$,

$$\mathbb{P}[|\tilde{\mathcal{T}_{Y:\lambda}}| = \infty] > 0$$

Notice

$$\mathbb{P}[|\tilde{\mathcal{T}_{Y:\lambda}}| = \infty] = \int \mathbb{P}_{p,\mathcal{T}_{Y:\lambda}}[|C(\rho)| = \infty] d\mathbb{P}(\mathcal{T}_{Y:\lambda})$$

and this shows

$$\mathbb{P}[\mathcal{T}_{Y:\lambda} : \mathbb{P}_{p,\mathcal{T}_{Y:\lambda}}[|C(\rho)| = \infty] > 0] > 0$$

On the other hand, by Lemma 2.30 and Lemma 2.25, we know that

$$\mathbb{P}[\mathcal{T}_{Y:\lambda} : \mathbb{P}_{p,\mathcal{T}_{Y:\lambda}}[|C(\rho)| = \infty] = 0] \in \{0, 1\}$$

Combining the above two properties, we know that

$$\mathbb{P}[\mathcal{T}_{Y:\lambda} : \mathbb{P}_{p,\mathcal{T}_{Y:\lambda}}[|C(\rho)| = \infty] > 0] = 1$$

This shows that $\mathbb{P}[\mathcal{T}_{Y:\lambda} : p > p_c(\mathcal{T}_{Y:\lambda})] = 1$. Now Lemma 2.31, we know that $p_c(\mathcal{T}_{Y:\lambda}) \geq (br(\mathcal{T}_{Y:\lambda}))^{-1}$, and so $\mathbb{P}[\mathcal{T}_{Y:\lambda} : p > (br(\mathcal{T}_{Y:\lambda}))^{-1}] = 1$. This holds for any $p > (2\mathbb{E}\theta_*)^{-1}$, so $(2\mathbb{E}\theta_*)^{-1} \geq \kappa^{-1}$ or $\kappa \geq 2\mathbb{E}\theta_*$

Combining both points above, we have $br(\mathcal{T}_{Y:\lambda}) = 2\mathbb{E}\theta_* = \frac{2\lambda}{\lambda+2}$ almost surely. □

2.6 On a Yule Model

The goal in this section is to prove our second main theorem.

Theorem 2.16 (Accuracy of Fitch Algorithm on Yule Model)**.** *Suppose* $\lambda > 6$*. For any* $\delta > 0$*, there is an* $\epsilon > 0$ *such that for any minimum cutset* π*,*

$$\mathbb{P}[\mathcal{T}_{Y:\lambda} : G^{\pi}_{\rho} > \epsilon] > 1 - \delta$$

We will use a similar notation as we used in Section 2.4, that is,

Notation 2.32. *Given an Yule model* $\mathcal{T}_{Y:\lambda} = (V, E, \rho, w)$ *and a minimum cutset* π*, for any* $z \in V^{\pi}$*, we use* $\mathcal{L}^{\pi}_{z,N} \subset V^{\pi}$ *to represent the set consisting of all the descendant of* z *which is exactly* N*-levels lower from* z*, and use* $\mathcal{M}^{\pi}_{z,N} = \cup_{n=1}^{N} \mathcal{L}^{\pi}_{z,n}$ *as the set which contains all the descendant of* z *up to* N*-level away from* z*.*

The idea of this proof comes from the proof of the deterministic tree case, Theorem 2.13. We would like to apply the same argument on the Yule model, but there are three obstacles that we need to conquer:

1. The most important part in the proof of deterministic tree case is the construction of π', which consists of all the first vertex x' on the path from ρ to the minimum cutset π such that $G^{\pi}_{x'}$ is larger than some specific value (See the proof of Theorem 2.13 in Section 2.4 for detailed definition of π'). However, in the Yule model, it complicates the analysis if π' is not fixed. What we will do here is choose $\pi' = \mathcal{L}^{\pi}_{\rho,N}$ for some N. Similar to the reason why we construct π' in the proof of the deterministic tree case, we need

$$\sum_{x \in \mathcal{L}^{\pi}_{\rho,N}} \Big(\prod_{z \in a(x) - \{\rho\}} \frac{2}{3}(1 - \phi)\theta_z \Big) G^{\pi}_{x} > \epsilon \tag{2.19}$$

   for some $\phi > 0$ such that $br(\mathcal{T}_{Y:\lambda}) > \frac{3}{2} \cdot \frac{1}{1-\phi}$. We can use induction to assume that $\mathbb{P}[G^{\pi}_{x} > \epsilon] > 1 - \delta$ for any $x \in \mathcal{L}^{\pi}_{\rho,N}$, and apply the coupling to put G^{π}_{x} aside when

using the definition of branching number. Chebyshev's inequality will be the last puzzle of the proof.

2. Next, in the deterministic tree case, we assume that there exists $\tau > 0$ such that $\theta_x \geq \pi$ for all vertex $x \in V$, which is not realistic in the Yule model. Nevertheless, what we need in the proof of Theorem 2.13 is that $\theta_x \geq \pi$ for $x \in \cup_{z \in \pi'} \mathcal{M}^{\pi}_{z,N'}$ for some N'. By the previous point, we have decided to fix π' as $\mathcal{L}^{\pi}_{\rho,N}$, and if we simply choose $N' = N$, what we need is $\theta_x \geq \tau$ for only those $x \in \mathcal{M}^{\pi}_{\rho,2N}$.

3. The previous two points seem persuasive to finish the proof. Nevertheless, the number does not match after some computation, and we put some effort into equation 2.19 to fix the problem. We consider two variables $\phi > \psi > 0$ such that, for any minimum cutset π and some $\zeta > 0$, both equations below have a high probability to hold.

$$\sum_{x \in \pi} \Big(\prod_{z \in a(x) - \{\rho\}} \frac{2}{3}(1-\phi)\theta_z \Big) > \zeta \tag{2.20}$$

$$\sum_{x \in \mathcal{L}^{\pi}_{\rho,N}} \Big(\prod_{z \in a(x) - \{\rho\}} \frac{2}{3}(1-\psi)\theta_z \Big) G^{\pi}_x > \epsilon \tag{2.21}$$

Equation 2.20 has a high probability by the definition of branching number. For equation 2.21, we can get a good bound now using Chebyshev's inequality, as we desired, because there is an extra term $(\frac{1-\psi}{1-\phi})^N$ when computing the expectation.

Hence, for any minimum cutset π, we will consider the following three events:

$$\begin{aligned}
A &: \{\forall x \in \mathcal{M}^{\pi}_{\rho,2N}, \theta_x \geq \tau\} \\
B &: \{\sup\{\zeta : \inf_{cutset\ \pi} \sum_{x \in \pi} \Big(\prod_{z \in a(x) - \{\rho\}} \frac{2}{3}(1-\phi) \cdot \theta_z \Big) > \zeta\} \geq \zeta_\delta\} \\
C &: \{ \sum_{x \in \mathcal{L}^{\pi}_{\rho,N}} \Big(\prod_{z \in a(x) - \{\rho\}} \frac{2}{3}(1-\psi)\theta_z \Big) G^{\pi}_x > \epsilon\}
\end{aligned}$$

then

$$\begin{aligned}
\mathbb{P}[G_\rho^\pi \le \epsilon] &= \mathbb{P}[(G_\rho^\pi \le \epsilon) \cap (A \cap B \cap C)] + \mathbb{P}[(G_\rho^\pi \le \epsilon) \cap (A \cap B \cap C)^c] \\
&\le \mathbb{P}[(G_\rho^\pi \le \epsilon) \cap (A \cap B \cap C)] + \mathbb{P}[(A \cap B \cap C)^c] \\
&\le \mathbb{P}[(G_\rho^\pi \le \epsilon) \cap (A \cap B \cap C)] + \mathbb{P}[A^c] + \mathbb{P}[(B \cap C)^c] \\
&= \mathbb{P}[(G_\rho^\pi \le \epsilon) \cap (A \cap B \cap C)] + \mathbb{P}[A^c] + \mathbb{P}[B^c] + \mathbb{P}[B \cap C^c]
\end{aligned}$$

Our goal is to show that the first term is 0, which is similar to the proof of the deterministic tree case), and all the other three terms are less than $\frac{\delta}{3}$ after choosing all the variables properly.

Proof. By Proposition 2.23, we know that $br(\mathcal{T}_{Y:\lambda}) = \frac{2\lambda}{\lambda+2} > \frac{3}{2}$ almost surely. Therefore, there exists $1 > \phi > 0$ such that almost surely

$$br(\mathcal{T}_{Y:\lambda}) > \frac{3}{2} \cdot \frac{1}{1-\phi} \tag{2.22}$$

By definition of branching number, the random variable on $\mathcal{T}_{Y:\lambda}$:

$$X_{Y:\lambda} = \sup\{\zeta : \inf_{cutset\ \pi} \sum_{x \in \pi} (\prod_{z \in a(x) - \{\rho\}} \frac{2}{3}(1-\phi) \cdot \theta_z) > \zeta\}$$

is positive almost surely. Hence, there exists a ζ_δ such that:

$$\mathbb{P}[X_{Y:\lambda} < \zeta_\delta] < \frac{\delta}{3}$$

On the other hand, since $\phi > 0$, there exists ψ such that $\phi > \psi > 0$, and as a consequence,

$$\frac{1-\psi}{1-\phi} > 1$$

After choosing ϕ, ζ_δ and ψ, we can find a large enough natural number N such that all the three following conditions hold.

1. $\eta = (\frac{1-\psi}{1-\phi})^N(1-\delta)\zeta_\delta > 2$, so $\eta > 1$ and $\eta < 2(\eta - 1)$
2. $\frac{\delta}{4\zeta_\delta(1-\delta)}(\frac{2}{3}(1-\phi))^N < \frac{\delta}{3}$
3. $4\psi - 2^{1-2^N} > 0$

Next, there exists a small enough $\tau > 0$ such that

$$(2^{2N+1} - 2)\tau^\lambda < \frac{\delta}{3}$$

and so there exists $1 > \epsilon > 0$ such that

$$((\frac{2}{\tau})^{2N}\epsilon)^2 < \min\{\frac{1}{3}, \frac{2}{3}(4\psi - 2^{1-2^N})\}$$

We will show that:

$$\mathbb{P}[\mathcal{T}_{Y:\lambda} : G^\pi_\rho > \epsilon] > 1 - \delta$$

We will prove the statement with induction. First, consider the basic case that if $\mathcal{M}^\pi_{\rho,2N}$ contains any vertex in π, say $x \in \pi \cap \mathcal{M}^\pi_{\rho,2N}$. There are at most $2N$ ancestral of x, and if all of these ancestor have $\theta_* \geq \tau$, then by Corollary 2.21, we have

$$1 = G^\pi_x \leq (\frac{2}{\tau})^{2N} G^\pi_\rho < \frac{1}{\sqrt{3}\epsilon} G^\pi_\rho \Rightarrow G^\pi_\rho > \sqrt{3}\epsilon > \epsilon$$

where the second inequality is from the assumption that $((\frac{2}{\tau})^{2N}\epsilon)^2 < \frac{1}{3}$. This event has probability

$$\begin{aligned}
& (\mathbb{P}[\theta_* \geq \tau])^{2N} \\
= \; & (\mathbb{P}[w \leq -\ln(\tau)])^{2N} \\
= \; & (1 - e^{-\lambda \ln \tau})^{2N} \\
= \; & (1 - \tau^\lambda)^{2N} \\
\geq \; & 1 - 2N\tau^\lambda \geq 1 - \frac{\delta}{3} > 1 - \delta
\end{aligned}$$

Therefore, $\mathbb{P}[\mathcal{T}_{Y:\lambda} : G_\rho^\pi > \epsilon] > 1 - \delta$.

Now suppose $\mathcal{M}_{\rho,2N}^\pi$ contains no vertices in π. By induction hypothesis, for any node $x \in \mathcal{L}_{\rho,N}^\pi$, $\mathbb{P}[G_x^\pi > \epsilon] > 1 - \delta$. Consider the following three events:

$$\begin{aligned} A &: \{\forall x \in \mathcal{M}_{\rho,2N}^\pi, \theta_x \geq \tau\} \\ B &: \{\sup\{\zeta : \inf_{cutset\ \pi} \sum_{x\in\pi} (\prod_{z\in a(x)-\{\rho\}} \frac{2}{3}(1-\phi)\cdot\theta_z) > \zeta\} \geq \zeta_\delta\} \\ C &: \{\sum_{x\in\mathcal{L}_{\rho,N}^\pi} (\prod_{z\in a(x)-\{\rho\}} \frac{2}{3}(1-\psi)\theta_z) G_x^\pi > \epsilon\} \end{aligned}$$

then

$$\begin{aligned} \mathbb{P}[G_\rho^\pi \leq \epsilon] &= \mathbb{P}[(G_\rho^\pi \leq \epsilon) \cap (A \cap B \cap C)] + \mathbb{P}[(G_\rho^\pi \leq \epsilon) \cap (A \cap B \cap C)^c] \\ &\leq \mathbb{P}[(G_\rho^\pi \leq \epsilon) \cap (A \cap B \cap C)] + \mathbb{P}[(A \cap B \cap C)^c] \\ &\leq \mathbb{P}[(G_\rho^\pi \leq \epsilon) \cap (A \cap B \cap C)] + \mathbb{P}[A^c] + \mathbb{P}[(B \cap C)^c] \\ &= \mathbb{P}[(G_\rho^\pi \leq \epsilon) \cap (A \cap B \cap C)] + \mathbb{P}[A^c] + \mathbb{P}[B^c] + \mathbb{P}[B \cap C^c] \end{aligned}$$

We will show that the first term is 0, and the later three terms are less or equal $\frac{\delta}{3}$.

For the first term, we follow the proof of Theorem 2.13. If $G_\rho^\pi \leq \epsilon$ and $\theta_x \geq \tau$ for all $x \in \mathcal{M}_{\rho,2N}^\pi$, then by Corollary 2.21:

$$\forall x \in \mathcal{M}_{\rho,N}^\pi, G_x^\pi \leq (\frac{2}{\tau})^N \epsilon$$

Next we show F_x^π is small as well by Lemma 2.22. Let $\epsilon' = (\frac{2}{\tau})^N \epsilon$, then for any $x \in \mathcal{M}_{\rho,N}^\pi$:

1. $((\frac{2}{\tau})^N \epsilon')^2 = ((\frac{2}{\tau})^{2N} \epsilon)^2 < \min\{\frac{1}{3}, \frac{2}{3}(4\psi - 2^{1-2^N})\}$
2. $G_x^\pi \leq (\frac{2}{\tau})^N \epsilon = \epsilon'$
3. For all $x' \in \mathcal{M}_{x,N}^\pi \subset \mathcal{M}_{\rho,2N}^\pi$, $\theta_{x'} \geq \tau$

Hence, by Lemma 2.22,

$$|F_x^\pi| \leq 4\psi$$

Consequently,

$$\frac{4 - F_x^\pi}{6} \geq \frac{2}{3}(1 - \psi)$$

Therefore,

$$\begin{aligned} \epsilon \geq G_\rho^\pi &= \sum_{x \in \mathcal{L}_{\rho,N}^\pi} \Big(\prod_{z \in a(x) - \{\rho\}} \frac{4 - F_{s(z)}^\pi}{6} \theta_z \Big) G_x^\pi \\ &\geq \sum_{x \in \mathcal{L}_{\rho,N}^\pi} \Big(\prod_{z \in a(x) - \{\rho\}} \frac{2}{3}(1 - \psi)\theta_z \Big) G_x^\pi > \epsilon \end{aligned}$$

where the last inequality is based on the assumption that event C occurs. This causes a contradiction. As a result, $\mathbb{P}[(G_\rho^\pi \leq \epsilon) \cap (A \cap B \cap C)] = 0$.

Next, we will show both $\mathbb{P}[A^c]$ and $\mathbb{P}[B^c]$ is less than $\frac{\delta}{3}$. The claim that $\mathbb{P}[B^c] = \mathbb{P}[X_{Y:\lambda} < \zeta_\delta] < \frac{\delta}{3}$ is from the choice of ζ_δ. For the claim that $\mathbb{P}[A^c] < \frac{\delta}{3}$, there are $2^{2N+1} - 2$ vertices in $\mathcal{M}_{\rho,2N}^\pi$, and for every $x \in \mathcal{M}_{\rho,2N}^\pi$,

$$\mathbb{P}[\theta_x \geq \tau] = \mathbb{P}[p_x \leq \frac{1 - \tau}{2}] = \mathbb{P}[t_x \leq -\ln(\tau)] = 1 - e^{-\lambda \ln \tau} = 1 - \tau^\lambda$$

The length of each edge is independent, so by assumption of τ,

$$\mathbb{P}[A^c] = 1 - (1 - \tau^\lambda)^{2^{2N+1}-2} \leq 1 - (1 - (2^{2N+1} - 2)\tau^\lambda) = (2^{2N+1} - 2)\tau^\lambda < \frac{\delta}{3}$$

Our final goal is proving $\mathbb{P}[B \cap C^c] \leq \frac{\delta}{3}$. Here we use our induction hypothesis that for any node $x \in \mathcal{L}_{\rho,N}^\pi$, $\mathbb{P}[G_x^\pi > \epsilon] > 1 - \delta$. Consider the coupling $\tilde{G}_x^\pi$ of G_x^π on $x \in \mathcal{L}_{\rho,N}^\pi$:

1. If $G_x^\pi \leq \epsilon$, then $\tilde{G}_x^\pi = 0$

2. If $G_x^\pi > \epsilon$, then randomly assign $\tilde{G}_x^\pi$ with the following values:

$$\mathbb{P}[\tilde{G}_x^\pi = 0] = \frac{\delta - \mathbb{P}[G_x^\pi \leq \epsilon]}{1 - \mathbb{P}[G_x^\pi \leq \epsilon]} \quad \text{and} \quad \mathbb{P}[\tilde{G}_x^\pi = \epsilon] = 1 - \frac{\delta - \mathbb{P}[G_x^\pi \leq \epsilon]}{1 - \mathbb{P}[G_x^\pi \leq \epsilon]}$$

It is well-defined since $\mathbb{P}[G_x^\pi \le \epsilon] \le \delta$. Then for any $x \in \mathcal{L}_{\rho,N}^\pi$, $G_x^\pi \ge \tilde{G}_x^\pi$. Moreover,

$$\begin{aligned}\mathbb{P}[\tilde{G}_x^\pi = 0] &= \mathbb{P}[G_x^\pi \le \epsilon] + (1 - \mathbb{P}[G_x^\pi \le \epsilon]) \cdot \frac{\delta - \mathbb{P}[G_x^\pi \le \epsilon]}{1 - \mathbb{P}[G_x^\pi \le \epsilon]} = \delta \\ \mathbb{P}[\tilde{G}_x^\pi = \epsilon] &= 1 - \mathbb{P}[\tilde{G}_x^\pi = 0] = 1 - \delta\end{aligned}$$

A little computation reveals that $\mathbb{E}\tilde{G}_x^\pi = \epsilon(1-\delta)$ and $\mathrm{Var}(\tilde{G}_x^\pi) = \epsilon^2\delta(1-\delta)$

Define a new random variable $\tilde{H}$:

$$\tilde{H} = \sum_{x \in \mathcal{L}_{\rho,N}^\pi} \Big(\prod_{z \in a(x) - \{\rho\}} \frac{2}{3}(1-\psi)\theta_z \Big) \tilde{G}_x^\pi \tag{2.23}$$

and consider the following two events $\tilde{B}$ and $\tilde{C}$:

$$\begin{aligned}\tilde{B} &: \Big\{ \sum_{x \in \mathcal{L}_{\rho,N}^\pi} \Big(\prod_{z \in a(x) - \{\rho\}} \frac{2}{3}(1-\phi) \cdot \theta_z \Big) > \zeta \Big\} \\ \tilde{C} &: \Big\{ \tilde{H} = \sum_{x \in \mathcal{L}_{\rho,N}^\pi} \Big(\prod_{z \in a(x) - \{\rho\}} \frac{2}{3}(1-\psi)\theta_z \Big) \tilde{G}_x^\pi > \epsilon \Big\}\end{aligned}$$

We have $B \subset \tilde{B}$ from the definition, and $\tilde{C} \subset C$ from the fact that $G_x^\pi \ge \tilde{G}_x^\pi$ almost surely. As a consequence,

$$\mathbb{P}[B \cap C^c] \le \mathbb{P}[\tilde{B} \cap \tilde{C}^c] \le \mathbb{P}[\tilde{C}^c | \tilde{B}]$$

We will now show $\mathbb{P}[\tilde{C}^c|\tilde{B}] \le \frac{\delta}{3}$, then $\mathbb{P}[B \cap C^c] \le \frac{\delta}{3}$ follows immediately.

For any $x \in \mathcal{L}_{\rho,N}^\pi$ and $z \in \mathcal{M}_{\rho,N}^\pi$, $\tilde{G}_x^\pi$ and θ_z are independent, and as a result, $\sigma(\{\tilde{G}_x^\pi : x \in \mathcal{L}_{\rho,N}^\pi\}$ is independent to $\sigma(\{\theta_z : z \in \mathcal{M}_{\rho,N}^\pi\})$. Notice $\tilde{B} \in \sigma(\{\theta_z : z \in \mathcal{M}_{\rho,N}^\pi\})$. Thus,

for any $\omega \in \sigma(\{\theta_z : z \in \mathcal{M}^{\pi}_{\rho,N}\})$ such that $\omega \in \tilde{B}$:

$$\begin{aligned}
\mathbb{E}\tilde{H}(\omega) &= \sum_{x\in\mathcal{L}^{\pi}_{\rho,N}} \Big(\prod_{z\in a(x)-\{\rho\}} \frac{2}{3}(1-\psi)\theta_z)\mathbb{E}\tilde{G}^{\pi}_x \\
&= \sum_{x\in\mathcal{L}^{\pi}_{\rho,N}} \Big(\prod_{z\in a(x)-\{\rho\}} \frac{2}{3}(1-\psi)\theta_z)\epsilon(1-\delta) \\
&= \epsilon(1-\delta) \sum_{x\in\mathcal{L}^{\pi}_{\rho,N}} \Big(\prod_{z\in a(x)-\{\rho\}} \frac{2}{3}(1-\phi)\frac{1-\psi}{1-\phi}\theta_z) \\
&= \epsilon(1-\delta)(\frac{1-\psi}{1-\phi})^N \sum_{x\in\mathcal{L}^{\pi}_{\rho,N}} \Big(\prod_{z\in a(x)-\{\rho\}} \frac{2}{3}(1-\phi)\theta_z) \\
&> \epsilon(1-\delta)(\frac{1-\psi}{1-\phi})^N \zeta_\delta = \eta\epsilon > \epsilon
\end{aligned}$$

and

$$\begin{aligned}
\mathrm{Var}(\tilde{H})(\omega) &= \sum_{x\in\mathcal{L}^{\pi}_{\rho,N}} \Big(\prod_{z\in a(x)-\{\rho\}} \frac{2}{3}(1-\psi)\theta_z)^2 \cdot \mathrm{Var}\tilde{G}^{\pi}_x \\
&= \sum_{x\in\mathcal{L}^{\pi}_{\rho,N}} \Big(\prod_{z\in a(x)-\{\rho\}} \frac{2}{3}(1-\psi)\theta_z)^2 \cdot \epsilon^2\delta(1-\delta) \\
&\leq \sum_{x\in\mathcal{L}^{\pi}_{\rho,N}} \Big(\prod_{z\in a(x)-\{\rho\}} (\frac{2}{3}(1-\psi))^2\theta_z) \cdot \epsilon^2\delta(1-\delta) \\
&= \epsilon\delta \cdot (\frac{2}{3}(1-\psi))^N \cdot \sum_{x\in\mathcal{L}^{\pi}_{\rho,N}} \Big(\prod_{z\in a(x)-\{\rho\}} \frac{2}{3}(1-\psi)\theta_z)\epsilon(1-\delta) \\
&= \epsilon\delta \cdot (\frac{2}{3}(1-\psi))^N \cdot \mathbb{E}\tilde{H}(\omega)
\end{aligned}$$

Now by Chebyshev's inequality

$$\begin{aligned}
\mathbb{P}[\tilde{H} \leq \epsilon](\omega) &\leq \frac{\mathrm{Var}(\tilde{H})(\omega)}{(\mathbb{E}\tilde{H}(\omega) - \epsilon)^2} \\
&\leq \frac{\mathrm{Var}(\tilde{H})(\omega)}{(\mathbb{E}\tilde{H}(\omega) - (1/\eta)\mathbb{E}\tilde{H}(\omega))^2} \\
&= (\frac{\eta}{\eta - 1})^2 \cdot \frac{\mathrm{Var}(\tilde{H}(\omega))}{(\mathbb{E}\tilde{H}(\omega))^2} \\
&\leq (\frac{\eta}{\eta - 1})^2 \cdot \frac{\epsilon\delta \cdot [\frac{2}{3}(1-\psi)]^N \cdot \mathbb{E}\tilde{H}(\omega)}{(\mathbb{E}\tilde{H}(\omega))^2} \\
&= (\frac{\eta}{\eta - 1})^2 \cdot \epsilon\delta \cdot [\frac{2}{3}(1-\psi)]^N \cdot \frac{1}{\mathbb{E}\tilde{H}(\omega)} \\
&\leq (\frac{\eta}{\eta - 1})^2 \cdot \epsilon\delta \cdot [\frac{2}{3}(1-\psi)]^N \cdot \frac{1}{\eta\epsilon} \\
&= \delta \cdot \frac{\eta}{(\eta-1)^2} \cdot [\frac{2}{3}(1-\psi)]^N \\
&\leq \delta \cdot \frac{\eta}{4\eta^2} \cdot [\frac{2}{3}(1-\psi)]^N \\
&= \frac{\delta}{4} \cdot [\frac{1}{\zeta_\delta(1-\delta)}(\frac{1-\phi}{1-\psi})^N] \cdot [\frac{2}{3}(1-\psi)]^N \\
&= \frac{\delta}{4\zeta_\delta(1-\delta)} \cdot [\frac{2}{3}(1-\phi)]^N < \frac{\delta}{3}
\end{aligned}$$

This holds for any $\omega \in \tilde{B}$. Hence,

$$\mathbb{P}[\tilde{C}^c|\tilde{B}] = \mathbb{P}[\tilde{H} < \epsilon|\tilde{B}] < \frac{\delta}{3}$$

Combining all the information above, we have

$$\begin{aligned}
\mathbb{P}[G_\rho^\pi \leq \epsilon] &\leq \mathbb{P}[(G_\rho^\pi \leq \epsilon) \cap (A \cap B \cap C)] + \mathbb{P}[A^c] + \mathbb{P}[B^c] + \mathbb{P}[B \cap C^c] \\
&< 0 + \frac{\delta}{3} + \frac{\delta}{3} + \frac{\delta}{3} = \delta
\end{aligned}$$

and this concludes the proof. □

CHAPTER 3

INCOMPLETE LINEAGE SORTING AND RECOMBINATION: THE THREE-TAXON CASE

Contribution I lead the theoretical part and wrote the first draft of this section. . For the simulation, I refined the first version of program, which is mentored by my advisor Sebastien Roch and implemented by two undergraduate students Zonglin Han and Calvin Kosmatka.

3.1 Introduction

Incomplete lineage sorting (ILS) is a phenomenon which causes a discord between a species tree and a gene tree [40, 14]. ILS occurs when two lineages fail to coalescence in a population, providing the opportunity for one lineage to merge with another less related lineage first. It has been an obstacle to conquer in the study of reconstructing a species tree from a set of gene trees, and a large amount of work has been invested to

explore methods to mitigate this problem [36, 19, 37]. In [53], Roch compares seven of these methods in the three-taxon case. Under this assumption, as shown in [53], these methods can be roughly separated into three categories: ML/GLASS/MT [36, 19, 37], R^*/STAR/MDC [5, 13, 38, 40, 61], and STEAC/SC [38]. In the paper, Roch shows that ML/GLASS/MT dominates the other two categories in accuracy. In addition, the decay rate of each category has been computed, which allows us to have a basic understanding of the accuracy in each category.

Our goal in this section is to extend the result in [53] when additionally taking recombination into account. Recombination is a phenomenon where different parts of the same gene might have different evolutionary histories, leading to different topologies [28, 26]. Because of the computational difficulty of the recombination process, some authors have provided some substitute algorithms to simulate recombination [42, 41]. Though they are approximate recombination process, simulations show that the simulated data from the algorithms is close to those from the full recombination process, and the approximate algorithms reduce much of computational complexity [17]. In this section, we will use one of the improved algorithms, sequentially Markov coalescent (SMC)[42], in our analysis.

This work is different from [33], which also discusses the accuracy of species-tree reconstruction with recombination. In [33], the authors fixed one reconstruction method (ML), and discussed the relationship between the accuracy of species-tree reconstruction and other possible factors, like the number of taxa, the number of loci, and the recombination rate. In our work, we fixed the number of taxa, the number of loci, and focus on the relationship between the accuracy of reconstruction algorithms and the recombination rate.

3.2 Background

Our goal is to compare methods that use multiple gene trees to estimates the tree for three taxa, under the assumption that the gene trees are affected by recombination. The subsections are organized in the following way: we will introduce the definitions and the notations in subsection 3.2.1. The multi-locus methods are explained in subsection 3.2.2, and the concept of recombination and sequential Markov coalescent are introduced in subsection 3.2.3. Last, we discuss the consistency of these method under the SMC model in subsection 3.2.4, and the large-deviation approach is explained in subsection 3.2.5.

3.2.1 Definition and Notation

We say a weighted rooted tree is an ultrametric tree if all the leaves are at the same distance from the root. For a three-leaf ultrametric tree with leaves a, b, and c, we denote by $ab|c$ the topology where a, b are closer to each other than to c, and similar notation for $ac|b$ and $bc|a$.

In the whole chapter, we assume the species tree S is ultrametric with three taxa A, B and C, and the topology of S is $AB|C$. We use τ_{AB} to denote the time difference between modern time and the time where A, B diverged. Similarly, we use τ_{ABC} to denote the time difference between modern time and the time when C separated from the ancestor of A and B. By assumption, we have $\tau_{ABC} \geq \tau_{AB}$ and we define

$$t = \tau_{ABC} - \tau_{AB} \tag{3.1}$$

Also, for convenience, we define $s = \tau_{AB}$ and so $\tau_{ABC} = s + t$.

Suppose we have L loci $l = 1, \ldots, L$ to estimate the topology of the species tree S. For each locus l, because we are analyzing the distances between DNA, we define the

distance between A and B as the weighted average of the Jukes-Cantor distance, that is,

$$d_{AB}^{(l)} = \sum_g w_g^{(l)} \cdot [\frac{1}{4}(1 - e^{-\frac{4}{3}d_{AB}^g})] = \frac{1}{4}(1 - \sum_g w_g^{(l)} e^{-\frac{4}{3}d_{AB}^g}) \tag{3.2}$$

where $w_g^{(l)}$ is the portion of gene tree g in locus l and d_{AB}^g is the distance between species A and B in g. Similar definition for $d_{AC}^{(l)}$ and $d_{BC}^{(l)}$. Notice, under our assumption, for any gene tree g, $d_{AB}^g \geq \tau_{AB} = s$, and so

$$d_{AB}^{(l)} = \frac{1}{4}(1 - \sum_g w_g^{(l)} e^{-\frac{4}{3}d_{AB}^g}) \geq \frac{1}{4}(1 - \sum_g w_g^{(l)} e^{-\frac{4}{3}s}) = \frac{1}{4}(1 - e^{-\frac{4}{3}s}) \tag{3.3}$$

Similarly,

$$d_{AC}^{(l)},\ d_{BC}^{(l)} \geq \frac{1}{4}(1 - e^{-\frac{4}{3}(s+t)}) \tag{3.4}$$

3.2.2 Multi-locus Methods

We briefly introduce the multi-loci methods that we are going to analyze in this subsection. As explained in previous work in [53], in the three-taxon setting, several of these methods are in fact equivalent and we follow the grouping given in [53]. For more details, see e.g. [37, 53].

1. **ML/GLASS/MT**:

   The Maximum Likelihood (ML) method selects the topology and the divergence times that maximize the likelihood of observing the gene trees. The result in [37] indicates that in the constant-population case, maximum likelihood will choose the largest possible divergence times. Therefore, this algorithm is equivalent to the Global Latest Split (GLASS) method which is studied in [46], and the Maximum

Tree (MT) method which is studied in [36, 19, 37] in the three-taxon case. In detail, the topology of the species tree is regarded as successfully reconstructed if

$$\min_l\{d_{AB}^{(l)}\} < \min\{\min_l\{d_{AC}^{(l)}\}, \min_l\{d_{BC}^{(l)}\}\} \tag{3.5}$$

2. R^***/STAR/MDC**:

The R^* consensus method [5, 13] first finds the topology with the highest frequency for any three-taxon set, then reconstructs a species tree that is compatible with the most three-taxon topologies. In the three-taxon setting, the topology of the species tree is determined by the frequency of the topologies acquired from loci. That is, the topology of the species tree is regarded as successfully reconstructed if

$$\begin{aligned}&\sum_l \mathbb{1}(d_{AB}^{(l)} < \min\{d_{AC}^{(l)}, d_{BC}^{(l)}\}) > \\ &\max\{\sum_l \mathbb{1}(d_{AC}^{(l)} < \min\{d_{AB}^{(l)}, d_{BC}^{(l)}\}), \sum_l \mathbb{1}(d_{BC}^{(l)} < \min\{d_{AB}^{(l)}, d_{AC}^{(l)}\})\}\end{aligned} \tag{3.6}$$

The other two methods in this group are the Species Tree Estimation Using Average Ranks of Coalescence Time (STAR) method and the Minimizing Deep Coalescences (MDC) method. The STAR method [38] estimates the species tree using the average ranks of gene coalescence times across the loci as the distance between taxa, where the rank of the root is equal to the number of taxa and the rank is decreased by 1 as the node goes from the root towards to the leaves. The MDC method [40, 61] outputs the species phylogeny that has the smallest lineage that fail to coalesce in a branch. As explained in [53], these two methods are equivalent to the R^* consensus method in the three-taxon setting.

3. **STEAC/SC**:

   The Species Tree Estimation Using Average Coalescent Time (STEAC) method [38] and the Shallowest Coalescences (SC) method both assign the average coalescent time as the distance between species and reconstruct the species tree based on this distance matrix. The only difference between these two methods is how they handle multiple alleles per populations, and are equivalent in our single-allele setting. In detail, the topology of the species tree is regarded as successfully reconstructed if

$$\sum_l d_{AB}^{(l)} < \min\{\sum_l d_{AC}^{(l)}, \sum_l d_{BC}^{(l)}\} \tag{3.7}$$

3.2.3 Recombination and Sequential Markov Coalescent

Recombination is a natural phenomenon. The phenomenon may occur, for example, when the DNA strands of a pair of homologous chromosomes break and rejoin during the process of meiosis in eukaryotes. Another example of recombination is in the process of DNA repair in bacteria. It is an obstacle for biologists and mathematicians when studying the evolutionary history between species.

Recombination results in different regions of a locus possibly having different underlying trees [28, 26]. However, estimating the likelihood of recombination is restricted by severe computing challenges [42]. The Sequential Markov Coalescent (SMC) is introduced in [42] in order to reduce the computing complexity. As shown in [42], the three main differences between the sequential Markov coalescent model from the original recombination process are:

1. The state space of the ancestral recombination graph is much reduced.

2. The SMC model tends to have many fewer recombination events.

3. The SMC process has a Markovian structure in the sequential generation of genealogies along a chromosome.

All these facts reduce the difficulty of simulating the recombination process. Though the process is not exactly the same as the original recombination process, it has been shown by simulations that SMC produces simulated data that for many statistics is close to that resulting from those of the original coalescent with recombination [17, 42]. See [42] for more details. The algorithm to simulate the SMC model is introduced below:

Algorithm 3.1 (Sequential Markov Coalescent [42])**.** *We can follow the following steps to simulate a unit locus under the SMC model:*

1. *Simulate a standard coalescent history at point 0 (i.e. without recombination). The resulting tree is g_0 and it has a total branch length of $w(g_0)$.*

2. *The distance along the unit interval until the first recombination event is exponentially distributed with rate $\rho w(g_0)$ for some rate $\rho > 0$. If the point at which the recombination event occurs is less than one, the position at which the recombination occurs on the marginal genealogy is drawn uniformly and the older portion of the branch on which the event occurred is erased, resulting in a "floating" lineage.*

3. *The floating lineage coalesces with the remaining genealogy at rate proportional to the number of ancestral lineages present.*

4. *The previous genealogy is discarded and the process repeated with the new genealogy until the next recombination event occurs beyond the unit interval.*

In this section, we assume that there is no rate variation for recombination.

3.2.4 Consistency of multi-locus methods

Before we analyze the exponential decay rate of each category of methods, we need to first discuss the consistency of these methods in our setting. That is, as the number of available unit loci in the SMC model goes to infinity, the methods should guarantee to output the correct topology. Here by unit loci we means genes, and we assume that different genes are independent. The consistency of ML/GLASS/MT is straightforward. Given enough unit loci in the SMC model, there should be at least one locus l' such that there is no recombination event along the locus, and it represents a gene tree on which the distance between species A and B is smaller than τ_{ABC}. When this event occurs, we have

$$\begin{aligned} \min_l\{d_{AB}^{(l)}\} < d_{AB}^{(l')} &= \frac{1}{4}(1-e^{-\frac{4}{3}(s+t)}) \\ &\leq \min\{\min_l\{d_{AC}^{(l)}\}, \min_l\{d_{BC}^{(l)}\}\} \end{aligned}$$

where the last inequality is from equation 3.4.

On the other hand, for the consistency of STEAC/SC method, as we will show later, the initial distribution is the stationary distribution in the SMC model for the three-taxon case. Therefore, at any point in any locus, the distribution of the underlying gene tree is the stationary distribution. Consequently, as the number of loci increases,

$$\frac{1}{|l|}\sum_l d_{AB}^{(l)} \to \mathbb{E}[d_{AB}^g], \quad \frac{1}{|l|}\sum_l d_{AC}^{(l)} \to \mathbb{E}[d_{AC}^g]$$

where d_{AB}^g and d_{AC}^g are the Jukes-Cantor distance between A, B and A,C in gene tree g, respectively. A little computation will show that $\mathbb{E}[d_{AB}^g] < \mathbb{E}[d_{AC}^g]$ and the consistency of STEAC/SC method follows.

Nevertheless, I haven't found a way to prove the consistency of the R^*/STAR/MDC method, and so I will leave it as a conjecture.

3.2.5 Large-deviations Approach

Similar to Roch's work in [53], in order to compare these groups of methods, we derive the rate of exponential decay of the probability of failure. Let S be a three-taxon species tree with internal branch length t, and assume we have L loci $l = 1, \ldots, L$ that are generated from S, on which the sequential Markov coalescent (SMC) with rate ρ takes place. As $L \to \infty$, large-deviation theory (see e.g. [18]) gives a characterization of the **(exponential) decay rate**

$$\alpha_M(t, \rho) = -\lim_{L\to\infty} \frac{1}{L} \ln \mathbb{P}[\text{Method } M \text{ fails given } L \text{ loci with SMC rate } \rho \text{ from } S] \quad (3.8)$$

That is, for large enough L,

$$\mathbb{P}[\text{Method } M \text{ fails given } L \text{ loci with SMC rate } \rho \text{ from } S] \approx e^{-L\alpha_M(t,\rho)} \quad (3.9)$$

3.3 Results

3.3.1 Stationary Distribution of SMC

For convenience, we use the following notations.

Notation 3.2. *For any gene tree g, we use d^g_{AB} to denote the distance between A and B in the gene tree. We use similar definition for d^g_{AC} and d^g_{BC}.*

Notation 3.3. *We denote by $\mathcal{T}_0(x, z)$ ($0 \le x \le t$, $0 \le z \le \infty$) the ultrametric gene tree satisfying $d^{\mathcal{T}_0(x,z)}_{AB} = s + x$ and $d^{\mathcal{T}_0(x,z)}_{AC} = s + t + z$ (and so $d^{\mathcal{T}_0(x,z)}_{BC} = s + t + z$ as well).*

Also, we denote by $\mathcal{T}_C(y,z)$ $(0 \le y \le z \le \infty)$ the ultrametric gene tree satisfied that $d_{AB}^{\mathcal{T}_C(x,z)} = s+t+y$ and $d_{AC}^{\mathcal{T}_C(x,z)} = s+t+z$ (and so $d_{BC}^{\mathcal{T}_C(x,z)} = s+t+z$). We use similar definition for $\mathcal{T}_A(y,z)$ and $\mathcal{T}_B(y,z)$.

Our first claim is

Proposition 3.4 (Stationary distribution)**.** *For the SMC model on a species tree S with internal branch length t, the Markov process has a stationary distribution π which satisfies*

$$\pi(T_0(x,z)) = e^{-x} \cdot e^{-z} \tag{3.10}$$

and for $I = A, B, C$

$$\pi(T_I(y,z)) = e^{-t} \cdot e^{-2y} \cdot e^{-z} \tag{3.11}$$

And therefore, the initial distribution is the stationary distribution.

The proof is basically by brute force.

3.3.2 Decay Rate of ML/GLASS/MT

Proposition 3.5 (ML/GLASS/MT)**.** *The decay rate of ML/GLASS/MT on S is*

$$\alpha_{ML}(t,\rho) = -\ln \mathbb{P}[d_{AB}^{(l)} > \frac{1}{4}(1 - e^{-\frac{4}{3}(s+t)})] \tag{3.12}$$

Proof. For any locus l, we let $\mathcal{E}^{(l)}$ be the event

$$\mathcal{E}^{(l)} = \{d_{AB}^{(l)} > \frac{1}{4}(1 - e^{-\frac{4}{3}(s+t)})\}$$

Given L loci, if there is any locus l such that $(\mathcal{E}^{(l)})^C$ occurs, then ML/GLASS/MT always succeeds. Therefore, ML/GLASS/MT fails with probability at most $(\mathbb{P}[\mathcal{E}^{(l)}])^L$.

And as a result,

$$\begin{aligned}
\alpha_{ML} &= \lim_{L\to\infty} -\frac{1}{L}\ln \mathbb{P}[\text{ML fail given } L \text{ loci}] \\
&\geq \lim_{L\to\infty} -\frac{1}{L}\ln(\mathbb{P}[\mathcal{E}^{(l)}])^L \\
&= \lim_{L\to\infty} -\ln \mathbb{P}[\mathcal{E}^{(l)}] \\
&= -\ln \mathbb{P}[\mathcal{E}^{(l)}]
\end{aligned}$$

On the other hand, for any $\epsilon > 0$, we denote by $\mathcal{E}_\epsilon^{(l)}$ the event

$$\mathcal{E}_\epsilon^{(l)} = \{d_{AB}^{(l)} \geq \frac{1}{4}(1 - e^{-\frac{4}{3}(s+t+\epsilon)})\}$$

Given L loci and $\epsilon > 0$, if both of the following hold,

- The first locus satisfies $\frac{1}{4}(1 - e^{-\frac{4}{3}(s+t+\frac{\epsilon}{2})}) = \min\{d_{AC}^{(l)}, d_{BC}^{(l)}\} < d_{AB}^{(l)}$, an event we denote by $\bar{\mathcal{E}}_\epsilon^{(l)}$.
- For all other $L-1$ loci, $\mathcal{E}_\epsilon^{(l)}$ occurs.

then ML/GLASS/MT will fail. As a consequence, ML/GLASS/MT fail with probability at least $\mathbb{P}[\bar{\mathcal{E}}_\epsilon^{(l)}]\mathbb{P}[\mathcal{E}_\epsilon^{(l)}]^{L-1}$. Notice $\bar{\mathcal{E}}_\epsilon^{(l)}$ occurs when we start with a $T_A(s+t+\frac{\epsilon}{2}, z)$ or $T_B(s+t+\frac{\epsilon}{2}, z)$ gene tree with $z > s+t+\frac{\epsilon}{2}$, and there is no recombination along the locus. Therefore,

$$\begin{aligned}
\mathbb{P}[\bar{\mathcal{E}}_\epsilon^{(l)}] &\geq 2\int_\epsilon^\infty e^{-\frac{\rho}{2}(3s+3t+\frac{\epsilon}{2}+2z)} \cdot e^{-\epsilon} e^{-z} dz \\
&= \frac{2}{\rho+1} e^{-\frac{\rho}{2}(3s+3t+\frac{\epsilon}{2})} e^{-\frac{3}{2}\epsilon} > 0
\end{aligned}$$

which shows that $\bar{\mathcal{E}}_\epsilon^{(l)}$ always have a positive probability to occur. Hence,

$$\begin{aligned}
\alpha_{ML} &= \lim_{L\to\infty} -\frac{1}{L}\ln \mathbb{P}[\text{ML fail given } L \text{ loci}] \\
&\leq \lim_{L\to\infty} -\frac{1}{L}\ln(\mathbb{P}[\bar{\mathcal{E}}_\epsilon^{(l)}]\mathbb{P}[\mathcal{E}_\epsilon^{(l)}]^{L-1}) \\
&= \lim_{L\to\infty} (-\frac{1}{L}\ln \mathbb{P}[\bar{\mathcal{E}}_\epsilon^{(l)}] - \frac{L-1}{L}\ln \mathbb{P}[\mathcal{E}_\epsilon^{(l)}]) \\
&= -\ln \mathbb{P}[\mathcal{E}_\epsilon^{(l)}]
\end{aligned}$$

This inequality holds for any $\epsilon > 0$. Since $\lim_{\epsilon\to 0}\mathbb{P}[\mathcal{E}_\epsilon^{(l)}] = \mathbb{P}[\mathcal{E}^{(l)}]$, we have $\alpha_{ML} \leq -\ln \mathbb{P}[\mathcal{E}_\epsilon^{(l)}]$. Therefore,

$$\alpha_{ML} = -\ln \mathbb{P}[\mathcal{E}^{(l)}] = -\ln \mathbb{P}[d_{AB}^{(l)} > \frac{1}{4}(1 - e^{-\frac{4}{3}(s+t)})]$$

as desired. □

3.3.3 Decay Rate of R*/STAR/MDC

Proposition 3.6. *The decay rate of R*/STAR/MDC on S is*

$$\alpha_{R^*}(t,\rho) = -\ln(\sqrt{2\beta(1-\beta)} + \frac{1}{2}(1-\beta)) \tag{3.13}$$

where $\beta = \mathbb{P}[d_{AB}^{(l)} < \min\{d_{AC}^{(l)}, d_{BC}^{(l)}\}]$.

Proof. We use the same definition and mimic the arguments in [53]. For a locus l, we let $Z_{AB}^{(l)}$ be 1 if the event $\{d_{AB}^{(l)} \leq \min\{d_{AC}^{(l)}, d_{BC}^{(l)}\}\}$ occurs, and 0 otherwise. We let

$$\mathcal{Z}_{AB} = \sum_{l=1}^{L} Z_{AB}^{(l)}$$

Similarly, we define $Z_{AC}^{(l)}$, $Z_{BC}^{(l)}$, $\mathcal{Z}_{AC}$ and $\mathcal{Z}_{BC}$. Then R^*/STAR/MDC fails if

$$\max\{\mathcal{Z}_{AC}, \mathcal{Z}_{BC}\} > \mathcal{Z}_{AB} = L - (\mathcal{Z}_{AC} + \mathcal{Z}_{BC})$$

an event we denote by $\mathcal{E}$. Re-write $\mathcal{E}$ as

$$2\max\{\mathcal{Z}_{AC}, \mathcal{Z}_{BC}\} + \min\{\mathcal{Z}_{AC}, \mathcal{Z}_{BC}\} > L$$

As shown in [53], with the auxiliary events

$$\mathcal{E}' = \{2\mathcal{Z}_{AC} + \mathcal{Z}_{BC} > L\}, \mathcal{E}'' = \{2\mathcal{Z}_{BC} + \mathcal{Z}_{AC} > L\}$$

we have

$$\mathbb{P}[\mathcal{E}'] \leq \mathbb{P}[\mathcal{E}] \leq \mathbb{P}[\mathcal{E}' \cup \mathcal{E}''] \leq 2\mathbb{P}[\mathcal{E}']$$

Hence

$$-\frac{1}{L}\ln \mathbb{P}[\mathcal{E}'] \geq -\frac{1}{L}\ln \mathbb{P}[\mathcal{E}] \geq -\frac{1}{L}\ln 2\mathbb{P}[\mathcal{E}'] = -\frac{1}{L}\ln \mathbb{P}[\mathcal{E}'] - \frac{1}{L}\ln 2$$

and consequently, as $L \to \infty$

$$\alpha_{R^*} = \lim_{L\to\infty} -\frac{1}{L}\ln \mathbb{P}[\mathcal{E}] = \lim_{L\to\infty} -\frac{1}{L}\ln \mathbb{P}[\mathcal{E}']$$

provided the limit exists.

Similar to [53], to compute the limit, we consider the moment-generating function

$$\phi(\psi) = \mathbb{E}[\exp(\psi[2Z_{AC}^{(l)} + Z_{BC}^{(l)}])]$$

We assume

$$\beta = \mathbb{P}[d_{AB}^{(l)} \leq \min\{d_{AC}^{(l)}, d_{BC}^{(l)}\}]$$

then by symmetry

$$\mathbb{P}[d_{AC}^{(l)} \leq \min\{d_{AB}^{(l)}, d_{BC}^{(l)}\}] = \mathbb{P}[d_{BC}^{(l)} \leq \min\{d_{AB}^{(l)}, d_{AC}^{(l)}\}] = \frac{1}{2}(1-\beta)$$

Therefore,

$$\phi(\psi) = \beta + \frac{1}{2}(1-\beta)(e^{\psi} + e^{2\psi}) < \infty$$

for all $\psi \in \mathbb{R}$. Let

$$W_\beta = \frac{1}{2}(1-\beta)$$

then the derivative of $\phi(\psi)$ is

$$\phi'(\psi) = W_\beta(e^\psi + 2e^{2\psi})$$

By large-deviation theory, our goal is to find a solution of the equation

$$1 = \frac{\phi'(\psi)}{\phi(\psi)}$$

We can re-write the previous equation as

$$\beta + W_\beta(e^\psi + e^{2\psi}) = W_\beta(e^\psi + 2e^{2\psi})$$

or

$$\beta = W_\beta e^{2\psi}$$

Hence, the solution ψ_* satisfies

$$e^{\psi_*} = \sqrt{\frac{\beta}{W_\beta}}$$

and

$$\phi(\psi_*) = \beta + W_\beta(e^{\psi_*} + e^{2\psi_*}) = 2\beta + \sqrt{\beta W_\beta}$$

Therefore, the large-derivation theory tells us

$$\begin{aligned}
\alpha_{R^*} = \lim_{L\to\infty} -\frac{1}{L}\ln \mathbb{P}[\mathcal{E}'] &= \psi_* - \ln\phi(\psi_*) \\
&= \ln\sqrt{\frac{\beta}{W_\beta}} - \ln(2\beta + \sqrt{\beta W_\beta}) \\
&= -\ln(\frac{2\beta + \sqrt{\beta W_\beta}}{\sqrt{\beta/W_\beta}}) \\
&= -\ln(2\sqrt{\beta W_\beta} + W_\beta) \\
&= -\ln(\sqrt{2\beta(1-\beta)} + \frac{1}{2}(1-\beta))
\end{aligned}$$

as desired. □

3.3.4 Decay Rate of STEAC/SC

Proposition 3.7. *The decay rate of STEAC/SC on S is*

$$\alpha_{STEAC}(t,\rho) = -\ln \phi(\psi_*) \tag{3.14}$$

where $\phi(\psi) = \mathbb{E}[\exp(\psi[d_{AB}^{(l)} - d_{AC}^{(l)}])]$ and ψ_* is a solution of $0 = \frac{\phi'(\psi)}{\phi(\psi)}$, if it exists.

Proof. Again we use the same definition and mimic the arguments in [53]. For a locus l, we let

$$\mathcal{D}_{AB} = \sum_{l=1}^{L} d_{AB}^{(l)}$$

and similar definition for $\mathcal{D}_{AC}, \mathcal{D}_{BC}$. Then STEAC/SC fails if

$$\mathcal{D}_{AB} > \min\{\mathcal{D}_{AC}, \mathcal{D}_{BC}\}$$

an event we denote by $\mathcal{E}$. Re-write $\mathcal{E}$ as

$$\mathcal{D}_{AB} - \min\{\mathcal{D}_{AC}, \mathcal{D}_{BC}\} > 0$$

Again as shown in [53], with the auxiliary events

$$\mathcal{E}' = \{\mathcal{D}_{AB} - \mathcal{D}_{AC} > 0\}, \mathcal{E}'' = \{\mathcal{D}_{AB} - \mathcal{D}_{BC} > 0\}$$

we have

$$\mathbb{P}[\mathcal{E}'] \leq \mathbb{P}[\mathcal{E}] \leq \mathbb{P}[\mathcal{E}' \cup \mathcal{E}''] \leq 2\mathbb{P}[\mathcal{E}']$$

Hence

$$-\frac{1}{L}\ln \mathbb{P}[\mathcal{E}'] \geq -\frac{1}{L}\ln \mathbb{P}[\mathcal{E}] \geq -\frac{1}{L}\ln 2\mathbb{P}[\mathcal{E}'] = -\frac{1}{L}\ln \mathbb{P}[\mathcal{E}'] - \frac{1}{L}\ln 2$$

Consequently, as $L \to \infty$

$$\alpha_{STEAC} = \lim_{L\to\infty} -\frac{1}{L}\ln \mathbb{P}[\mathcal{E}] = \lim_{L\to\infty} -\frac{1}{L}\ln \mathbb{P}[\mathcal{E}']$$

provided the limit exists.

Similar to [53], to compute the limit, we consider the moment-generating function

$$\phi(\psi) = \mathbb{E}[\exp(\psi[d_{AB}^{(l)} - d_{AC}^{(l)}])]$$

which is finite for finite ψ since $0 \leq d_{AB}^{(l)}, d_{AC}^{(l)} \leq \frac{1}{4}$. By large-deviation theory, we are looking for a solution ψ_* of the equation

$$0 = \frac{\phi'(\psi)}{\phi(\psi)}$$

Then

$$\begin{aligned}
\alpha_{STEAC} &= \lim_{L\to\infty} -\frac{1}{L}\ln \mathbb{P}[\mathcal{E}'] \\
&= \lim_{L\to\infty} -\frac{1}{L}\ln \mathbb{P}[\mathcal{D}_{AB} - \mathcal{D}_{AC} > 0] \\
&= -\ln \phi(\psi_*)
\end{aligned}$$

as desired. □

3.4 Simulation result

In [53], Roch showed that in the three-taxon case, without considering recombination, the decay rates of each categories when $t \to 0$ are

$$\begin{aligned}
\alpha_{ML}(t) &= t \\
\alpha_{R*}(t) &= \frac{3}{4}t^2 + O(t^3)
\end{aligned}$$

Also, he proved that the decay rate of ML/GLASS/MT always dominates the decay rate of R^*/STAR/MDC.

Our results indicate that, with recombination, the decay rate can be simplified by only relying on the information of one locus. In detail, from equation 3.12 and equation 3.13,

$$\begin{aligned} \alpha_{ML}(t,\rho) &= -\ln \mathbb{P}[d_{AB}^{(l)} > \frac{1}{4}(1-e^{-\frac{4}{3}(s+t)})] \\ \alpha_{R*}(t,\rho) &= -\ln(\sqrt{2\beta(1-\beta)} + \frac{1}{2}(1-\beta)), \\ & \quad \text{where } \beta = \mathbb{P}[d_{AB}^{(l)} < \min\{d_{AC}^{(l)}, d_{BC}^{(l)}\}] \end{aligned}$$

In the simulation, we compare the decay rate of ML/GLASS/MT and the decay rate of R^*/STAR/MDC in our setting for some specific t and small recombination rate ρ. In the simulation, we let $\tau_{AB} = s = 1.0$, and generate 100000 unit locus under the SMC model to compute the $\mathbb{P}[d_{AB}^{(l)} > \frac{1}{4}(1-e^{-\frac{4}{3}(s+t)})]$ and $\mathbb{P}[d_{AB}^{(l)} < \min\{d_{AC}^{(l)}, d_{BC}^{(l)}\}]$, then compute the decay rate of ML/GLASS/MT and the decay rate of R^*/STAR/MDC. We repeat this process 500 times for any specific t and ρ and compute the average. The program is written in Java and it takes slightly over 2 minutes for 500 runs in average for a specific t and ρ.

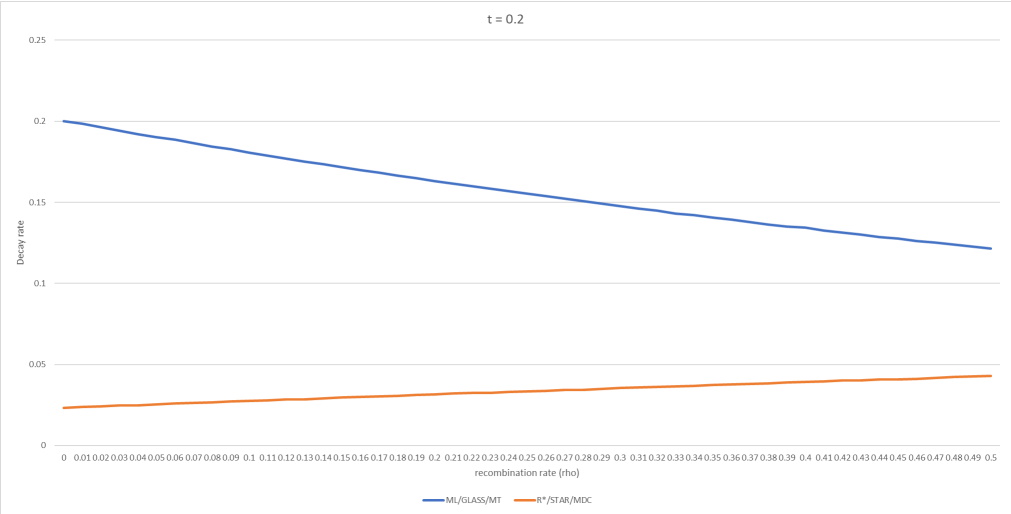


Figure 2: The decay rate of ML/GLASS/MT and the decay rate of R*/STAR/MDC versus ρ, when $t = 0.2$

Figure 3: The decay rate of ML/GLASS/MT and the decay rate of R*/STAR/MDC versus ρ, when $t = 0.2$

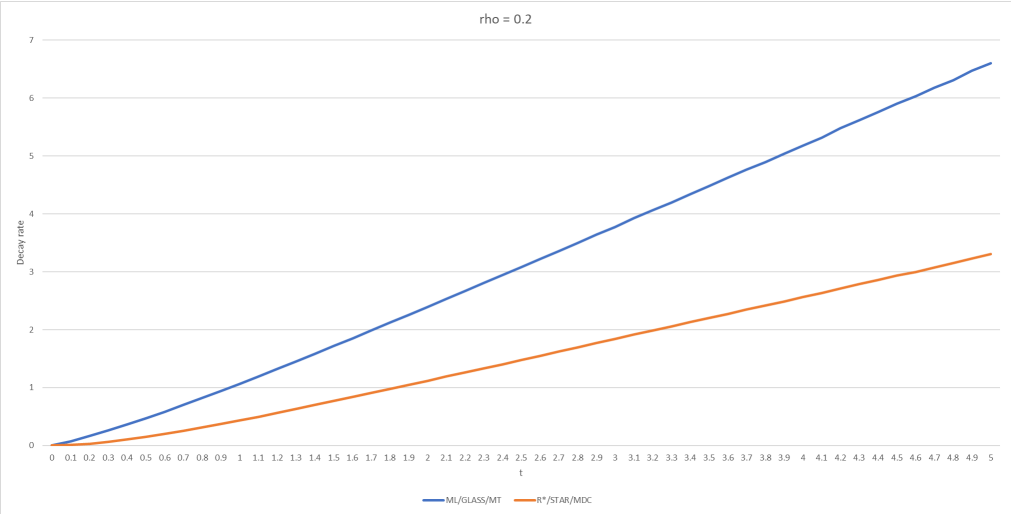


Figure 4: The decay rate of ML/GLASS/MT and the decay rate of R*/STAR/MDC versus t, when $\rho = 1.8$

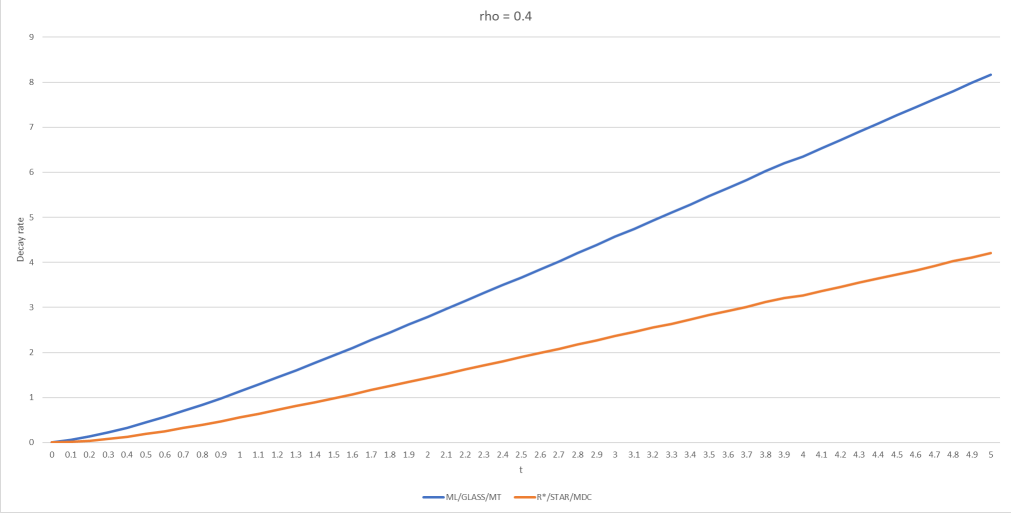


Figure 5: The decay rate of ML/GLASS/MT and the decay rate of R*/STAR/MDC versus t, when $\rho = 1.8$

CHAPTER 4

NETWORK RECONSTRUCTION

Contribution I lead all aspects of the work, including both theoretical part and simulation part, and I wrote the first draft of this section.

4.1 Introduction

In evolution biology, phylogenetic trees are widely used to uncover the evolutionary relationships between different species or taxa. A number of phylogenetic tree reconstruction methods have been invented and studied. Starting from the short quartet method (SQM)[20], the fast-converging methods, which guarantee to recover the tree given only polynomial length sequence, are gaining in popularity. The work of Daskalakis et al. in [12] improves SQM and introduces a phylogenetic tree reconstruction algorithm from the distorted metrics, which has a high probability to unearth the tree topology from any reasonable length sequence.

Nevertheless, reticulate events in evolution history, like hybridization, horizontal

gene transfer, recombination, are unable to be modeled by a single phylogenetic tree. **Phylogenetic networks** are more adequate for the data sets involving these events [31, 15, 16, 25, 52, 57, 59]. One way to describe an unrooted phylogenetic network is using **split networks**, which are first introduced by H. J. Bandelt et al. in [3]. An abundance of algorithms have been introduced to reconstruct a split network from information obtained from samples, and we can classify them into a few categories based on the information they need [29, 31]: Neighbor-Net [7] and the split decomposition method [3] take pairwise distances between taxa as input, while the consensus network method [27] and the super network method [30, 62] use a set of phylogenetic trees to reconstruct the network, and the median-joining algorithm [4] method and the quasi median-joining algorithm [2] require condense characters.

The distance-based reconstruction methods, Neighbor-Net and the split decomposition method, are the most popular algorithms. Neighbor-Net [7] uses the clustering idea which comes from Neighbor-Joining [54], a method of phylogenetic tree reconstruction. On the other hand, the split decomposition method [3] recursively enlarges the set of taxa and finds the splits on them. They have been accepted and widely used with the proof of their consistency [9, 3]. In addition, both Neighbor-Net and the split decomposition method have been proven to still be consistent if the error between real distances and their estimates from samples is bounded [43, 34, 3]. Nevertheless, both methods require that the estimation between every pair of taxa have a bounded error, and so if the diameter of the network is large, the methods will need impractical number of sample length to ensure consistency.

Our motivation in this project is to introduce a fast-converging split network algorithm, similar to [20] and [12], by relying on only short distances. Our algorithm is

analogous to the algorithm introduced in [12], with the fact that every edge in a phylogenetic tree is corresponding to a split in a split network. The algorithm eliminates the disadvantage of Neighbor-Net and the split decomposition method by shortening the radius of the trusted region from diameter of the network to a linear combination of the depth and the incompatibility of the network. In the remainder of this section, we will formalize the idea and provide all the details of the algorithm.

An important consideration is information theory on split networks. The identifiability of split networks with an Abelian group-based model, a Markov model on split networks, from character distribution have been proved by D. Bryant [6]. Moreover, the identifiability from distances of two most important type of split networks, circular networks and weakly compatible networks, have also been shown in [3] and [7]. Nevertheless, similar to the fact that short edges and deep parts of a phylogenetic tree are difficult to detect[12], light or deep splits in a split networks are hard to reconstruct, for any algorithms. In our algorithm, we reconstruct a subset of the original split set, and ensure that the subset contains all splits which are heavy enough. The difficulty of finding deep parts is reflected on the threshold of reliable distances. That is, to reconstruct a network with a deep part, we need to enlarge the threshold of reliable distances, which requires more samples to achieve.

4.2 Background

Before we introduce our algorithm, we want to provide some background on split networks.

Definition 4.1 (Split)**.** *A* **split** $S = \{S^1, S^2\}$ *on a set of taxa* $\mathcal{X}$ *is an unordered*

bipartition of $\mathcal{X}$ into two non-empty, disjoint sets: $S^1 \cap S^2 = \emptyset$, $S^1 \cup S^2 = \mathcal{X}$.

Definition 4.2 (Split Networks [3, 31]). *We say $\mathcal{N} = (\mathcal{X}, \mathcal{S}, w)$ is a* **weighted split network** *on a set of $\mathcal{X}$ if $\mathcal{S}$ is a set of splits on $\mathcal{X}$, and $w : \mathcal{S} \to (0, \infty)$ is the function which assigns a positive weight to every split in $\mathcal{S}$. We assume that any two splits $S_1 = \{S_1^1, S_1^2\}, S_2 = \{S_2^1, S_2^2\} \in \mathcal{S}$ are different, that is, $S_1^1 \neq S_2^1$ and $S_1^1 \neq S_2^2$*

Any edge e in a weighted phylogenetic tree $\mathcal{T} = (\mathcal{X}, V, E, w)$ on $\mathcal{X}$ define a split on $\mathcal{X}$ [31]: after deleting e, the leaves of $\mathcal{T}$ are separating into two non-empty, disjoint sets, say l^1 and l^2, and the split $S_e = \{S^1, S^2\}$ generated by e is defined as: S^i is the set contains all the taxon labels which occur in l^i, $i = 1, 2$. And so any phylogenetic tree can be represented as a split network. One quick question is its converse statement: Given a split network $\mathcal{N} = (\mathcal{X}, \mathcal{S}, w)$, is there a phylogenetic tree $\mathcal{T} = (\mathcal{X}, V, E, w)$ such that $\mathcal{S} \subset \mathcal{S}_E = \{S_e : e \in E\}$? The question can be answered using the concept of **compatibility**.

Definition 4.3 (Compatibility [10]). *Two splits $S_1 = \{S_1^1, S_1^2\}$ and $S_2 = \{S_2^1, S_2^2\}$ are called* **compatible**, *if at least one of the following interactions is empty:*

$$S_1^1 \cap S_2^1, S_1^1 \cap S_2^2, S_1^2 \cap S_2^1, S_1^2 \cap S_2^2$$

Otherwise, these two splits are called **incompatible**. *A set of splits $\mathcal{S}$ is called compatible if all pairs of splits in $\mathcal{S}$ are compatible.*

Lemma 4.4 (Compatibility Theorem [10]). *Given a split network $\mathcal{N} = (\mathcal{X}, \mathcal{S}, w)$, there is a phylogenetic tree $\mathcal{T} = (\mathcal{X}, V, E, w)$ such that $\mathcal{S} \subset \mathcal{S}_E = \{S_e : e \in E\}$ if and only if $\mathcal{S}$ is compatible.*

Nevertheless, there are many more split networks that cannot be realized as phylogenetic trees, and most of them are difficult to visualize in a comprehensive way [31]. Some restricted classes of split networks, therefore, have been introduced. The most important two classes are **circular networks** and **weakly compatible networks**.

Definition 4.5 (Circular Splits, Circular Networks [3, 31])**.** *A set of splits $\mathcal{S}$ on $\mathcal{X}$ is called* **circular***, if there exists a linear ordering $(x_1, \ldots, x_n)$ of the elements of $\mathcal{X}$ for $\mathcal{S}$ such that each splits $S \in \mathcal{S}$ has the form:*

$$S = \{\, \{x_p, \ldots, x_q\}\,,\ \mathcal{X} - \{x_p, \ldots, x_q\}\,\}$$

for appropriately chosen $1 < p \leq q \leq n$. We say a split network $\mathcal{N} = \{\mathcal{X}, \mathcal{S}, w\}$ is a **circular network** *if $\mathcal{S}$ is circular.*

Definition 4.6 (Weak Compatibility, Weakly Compatible Networks [3, 31])**.** *We say three splits $S_1 = \{S_1^1, S_1^2\}$, $S_2 = \{S_2^1, S_2^2\}$ and $S_3 = \{S_3^1, S_3^2\}$ are* **weakly compatible** *if*

1. *at least one of the following four intersections is empty:*

$$S_1^1 \cap S_2^1 \cap S_3^1, \quad S_1^1 \cap S_2^2 \cap S_3^2, \quad S_1^2 \cap S_2^1 \cap S_3^2 \quad \textit{and} \quad S_1^2 \cap S_2^2 \cap S_3^1$$

2. *at least one of the following four intersections is empty:*

$$S_1^2 \cap S_2^2 \cap S_3^2, \quad S_1^2 \cap S_2^1 \cap S_3^1, \quad S_1^1 \cap S_2^2 \cap S_3^1 \quad \textit{and} \quad S_1^1 \cap S_2^1 \cap S_3^2$$

A set of splits $\mathcal{S}$ on $\mathcal{X}$ is called weakly compatible if any three splits are weakly compatible, and we say a split network $\mathcal{N} = \{\mathcal{X}, \mathcal{S}, w\}$ is a **weakly compatible network** *if $\mathcal{S}$ is weakly compatible.*

The following two lemmas tell us the relation between the phylogenetic trees, circular networks, and weakly compatible networks [31].

Lemma 4.7. *The set of splits obtained from a phylogenetic tree is circular.*

Lemma 4.8. *Let $\mathcal{S}$ be a set of splits on $\mathcal{X}$. If $\mathcal{S}$ is circular, then $\mathcal{S}$ is weakly compatible.*

As we introduced above, Neighbor-Net and the split decomposition method are the two main algorithms of split network reconstruction. Two major differences between these two algorithms are:

- The the split decomposition method can deal with more general cases than Neighbor-Net: the output of the split decomposition method is a weakly compatible network, while Neighbor-Net outputs a circular network.

- Neighbor-Net is much faster than the split decomposition method in finding all possible splits (but without estimating the weight): Neighbor-Net takes $O(n^3)$ time, while the worst case of the split decomposition method needs $O(n^6)$ time, where n is the number of taxa.

  Remark 4.9. *In a general case, Neighbor-Net uses ordinary least square (OLS) or even non-negative-constraints least square (NNLS) to estimate the weight [7, 34], which has at least $O(n^6)$ time complexity. Nevertheless, if we assume that the error between real distance and the estimators has an upper bound for every pair of taxa, then the weight estimation step in Neighbor-Net only takes $O(n^2)$ time using equation (4) in [34], and the whole process of Neighbor-Net takes only $O(n^3)$ time.*

As we discussed above, the consistency of both algorithms is built on the assumption that the estimation of every pair of taxa has a bounded error from the true distance. However, if the diameter of the network is high, for example a tree with linear structure, then the requirement becomes unreasonable. The algorithm which we will introduce later improves these two methods by reducing the radius of accurate distances.

4.2.1 Previous work on phylogenetic tree reconstruction

In this section, we will explain the algorithm on phylogenetic tree reconstruction in [12] which inspires our algorithm on split network reconstruction. Suppose we have a weighted, unrooted phylogenetic tree $\mathcal{T} = (\mathcal{X}, V, E, w)$, where $\mathcal{X}$ is the set of taxa mapped to the leaves of the tree, V and E are, respectively, the vertices and edges of the tree, and $w : E \to (0, \infty)$ is the weight function. We equip $\mathcal{T}$ with the **additive metric**, $d : \mathcal{X} \times \mathcal{X} \to (0, \infty)$:

$$\forall x, y \in \mathcal{X},\ d(x, y) = \sum_{e \in path(x,y)} w(e)$$

where $path(x, y)$ is the set of edges on the path between x, y in $\mathcal{T}$.

It is well known that one can reconstruct the phylogenetic tree $\mathcal{T}$ given an additive metric d [12]. However, in practice, one can only obtain an estimate $\hat{d}$ of d, usually called the **distance matrix**, from samples. The accuracy of these estimates depends on the amount of data used [12]. Usually the estimates of long distances are unreliable, and the following definition describes this property.

Definition 4.10 (Distorted metric [45, 1])**.** *Suppose $\mathcal{T} = (\mathcal{X}, V, E, w, d)$ is a phylogenetic tree, and let $\tau, M > 0$. We say $\hat{d} : \mathcal{X} \times \mathcal{X} \to (0, \infty]$ is a (τ, M)-**distorted metric** of $\mathcal{T}$ or a (τ, M)-**distortion** of d if the following are true:*

1. *(Symmetry) For all $x, y \in \mathcal{X}$, $\hat{d}$ is symmetric, that is,*

$$\hat{d}(x, y) = \hat{d}(y, x)$$

2. *(Distortion) $\hat{d}$ is accurate on "short" distances, that is, for all $x, y \in \mathcal{X}$, if either $d(x, y) < M + \tau$ or $\hat{d}(x, y) < M + \tau$, then*

$$|d(x, y) - \hat{d}(x, y)| < \tau$$

As we mentioned above, the deep parts of a phylogenetic tree are hard to detect. The following definition of **depth** provides an idea about the deepness of an edge by measuring the distance between the edge and the closest leaves.

Definition 4.11 (Depth [12])**.** *In a phylogenetic tree $\mathcal{T} = (\mathcal{X}, V, E, w, d)$, the* **depth** *of an edge $e \in E$ is the length of the shortest path among all paths crossing e between two leaves:*

$$\Delta(e) = \min\{d(x, y) : x, y \in \mathcal{X}, e \in path(x, y)\}$$

and the **depth** *of $\mathcal{T}$ is the maximum depth over all the edges:*

$$\Delta(\mathcal{T}) = \max\{\Delta(e) : e \in E\}$$

The main theorem in [12] shows the following lemma.

Lemma 4.12. *For any phylogenetic tree $\mathcal{T} = (\mathcal{X}, V, E, w, d)$, if $\tau > 0$ and $M > 2\Delta(\mathcal{T}) + 5\tau$, there is a polynomial time algorithm which is guaranteed to reconstruct all the edges, whose weight is larger than 2τ, of $\mathcal{T}$ given a (τ, M)-distorted metric of $\mathcal{T}$.*

The algorithm in [12] consists of the following two steps:

1. Mini Reconstruction:

   The purpose of Mini Reconstruction is to uncover edges in a "small region". For any $x, y \in \mathcal{X}$ such that $\hat{d}(x,y) \leq \Delta(\mathcal{T}) + \tau$, define the small region around x, y by

   $$B(x,y) = \{z \in \mathcal{X} : \hat{d}(x,z) \vee \hat{d}(y,z) < 2\Delta(\mathcal{T}) + 5\tau\}$$

   Reconstruct all the edges between x and y by finding all bipartitions on $B(x,y)$ which separates x and y using $\hat{d}(x,y)$, $\hat{d}(x,z)$ and $\hat{d}(y,z)$ for all $z \in B(x,y)$. [12] show that because for all $z \in B(x,y)$, $\hat{d}(x,y)$, $\hat{d}(x,z)$ and $\hat{d}(y,z)$ are reliable, and so any bipartition or edge found on $B(x,y)$ is corresponding to a bipartition on $\mathcal{X}$, or an edge of the original phylogenetic tree $\mathcal{T}$.

2. Bipartition Extension:

   The purpose of Bipartition Extension is to extend a bipartition in a small region from Mini Reconstruction to a bipartition on the whole taxa $\mathcal{X}$, and then reconstruct $\mathcal{T}$ after collecting all the bipartitions or the edges of $\mathcal{T}$. The extension is executed in the following way. Starting from the bipartition $\{S^x, S^y\}$ on $B(x,y)$, apply the following process recursively until we have a bipartition on $\mathcal{X}$: If currently the bipartition is $\{\tilde{S}^x, \tilde{S}^y\}$, and taxa $z \in \mathcal{X} - (\tilde{S}^x, \tilde{S}^y)$, $z' \in \tilde{S}^x$ satisfy $\hat{d}(z,z') < \Delta(\mathcal{T}) + \tau$, then enlarge $\tilde{S}^x$ by adding z to $\tilde{S}^x$(Similar for $\tilde{S}^y$). [12] showed that for every bipartition S on $B(x,y)$ in a small region:

   - All taxa outside $B(x,y)$ have a way to connect to $B(x,y)$ right from the definition of the depth, and so the extension process always terminates.
   - For any bipartition $\{S^x, S^y\}$ on $B(x,y)$, there is an unique way to execute the extension. In other words, the extension, there is only one possible output for

any $\{S^x, S^y\}$ on $B(x, y)$. In addition, the output of the extension is always a bipartition, or an edge, of the original tree $\mathcal{T}$.

- Last, for any edge e of the original tree $\mathcal{T}$, there exists x, y such that the extension terminates in e starting from bipartition $\{S^x, S^y\}$ found on $B(x, y)$.

With all definitions, algorithms above, we can rebuild the whole tree once we collect all the bipartitions, or the edges, of the phylogenetic tree $\mathcal{T}$.

4.2.2 Main result

Our algorithm of split network reconstruction follows the work in [12] which we introduced in the previous section. The idea is based on the fact that a bipartition or an edge in a phylogenetic tree is corresponding to a split in a split network, and therefore uncovering all the bipartitions in a phylogenetic tree means uncovering all the splits in a split network. Nevertheless, there is a significant difference between phylogenetic trees and split networks: split networks allow the incompatibility between splits, and we need to modify some definitions and values to make the algorithm suitable to split networks. Suppose we have a weighted split network $\mathcal{N} = (\mathcal{X}, \mathcal{S}, w)$. We can extend the additive metric definition of a phylogenetic tree to the definition of **metric** of a split network.

Definition 4.13 (Metric)**.** *The* **metric** $d : \mathcal{X} \times \mathcal{X} \to (0, \infty)$ *of a split network* $\mathcal{N} = (\mathcal{X}, \mathcal{S}, w)$ *is defined as:*

$$\forall x, y \in \mathcal{X},\ d(x, y) = \sum_{S \in \mathcal{S}} w(S)\delta_S(x, y)$$

where $\delta_S(x, y)$ for any splits $S = \{S^1, S^2\}$ is the indicator of whether S separate x, y:

$$\delta_S(x, y) = \begin{cases} 0, & \text{if } x, y \in S^1 \text{ or } x, y \in S^2. \\ 1. & \text{otherwise.} \end{cases}$$

We use the word metric because d satisfies at least symmetry and triangle inequality [31]. In the remainder of the chapter, we assume that d also has identity property, that is, for any different $x, y \in \mathcal{X}$, there exists $S \in \mathcal{S}$ such that $\delta_S(x, y) = 1$, and so $d(x, y) > 0$.

Similar to phylogenetic trees, we cannot get the real metric d from samples but only an estimated distance matrix $\hat{d}$, and once again the estimates of long distance are not reliable. This property is shown with the following analogous definition of **distorted metric** on split networks.

Definition 4.14 (Distorted metric, [45, 1])**.** *Suppose $\mathcal{N} = (\mathcal{X}, \mathcal{S}, w, d)$ is a split network, and let $\tau, M > 0$. We say $\hat{d} : \mathcal{X} \times \mathcal{X} \to (0, \infty]$ is a (τ, M)-**distorted metric** of $\mathcal{N}$ or a (τ, M)-**distortion** of d if the following are true:*

1. *(Symmetry) For all $x, y \in \mathcal{X}$, $\hat{d}$ is symmetric, that is,*

$$\hat{d}(x, y) = \hat{d}(y, x)$$

2. *(Distortion) $\hat{d}$ is accurate on "short" distances, that is, for all $x, y \in \mathcal{X}$, if either $d(x, y) < M + \tau$ or $\hat{d}(x, y) < M + \tau$, then*

$$|d(x, y) - \hat{d}(x, y)| < \tau$$

The next definition we work on is **depth**, which describes how close an edge is to the leaves in a phylogenetic tree. We want to extend the definition of depth on split networks

so that it does not contradict the definition on phylogenetic trees, and, on the other hand, we need to consider the incompatibility between splits in a split network. Recall that, in a phylogenetic tree, every edge is a split, and any two edges are compatible. Hence, for every $e \in E$ in a phylogenetic tree,

$$\begin{aligned}
\Delta(e) &= \min\{d(x,y) : x,y \in \mathcal{X}, e \in path(x,y)\} \\
&= \min\{d(x,y) : x,y \in \mathcal{X}, \delta_{S_e}(x,y) = 1\} \\
&= \min\{\sum_{e' \in path(x,y)} w(e') : x,y \in \mathcal{X}, \delta_{S_e}(x,y) = 1\} \\
&= \min\{\sum_{S' : \delta_{S'}(x,y)=1} w(S') : x,y \in \mathcal{X}, \delta_{S_e}(x,y) = 1\} \\
&= \min\{\sum_{S' : \delta_{S'}(x,y)=1 \text{ and } S' \text{ is compatible with } S_e} w(S') : x,y \in \mathcal{X}, \delta_{S_e}(x,y) = 1\}
\end{aligned}$$

Consequently, a natural way to define the depth in a split network is:

Definition 4.15 (Depth)**.** *In a split network $\mathcal{N} = (\mathcal{X}, \mathcal{S}, w, d)$, and $S \in \mathcal{S}$. Suppose $x, y \in \mathcal{X}$ satisfy $\delta_S(x,y) = 1$. Define*

$$\Xi\mathcal{C}(x,y,S) = \{S' \in \mathcal{S} : \delta_{S'}(x,y) = 1 \text{ and } S' \text{ is compatible with } S\}$$

then we define the **depth** *of the split S as:*

$$\Delta(S) = \min\{\sum_{S' \in \Xi\mathcal{C}(x,y,S)} w(S') : x,y \in \mathcal{X}, \delta_S(x,y) = 1\}$$

and the **depth** *of $\mathcal{N}$ as the maximum depth among all $S \in \mathcal{S}$:*

$$\Delta(\mathcal{N}) = \max\{\Delta(S) : S \in \mathcal{S}\}$$

We need another definition when we take the incompatibility into consideration:

Definition 4.16 (Incompatible weight)**.** *In a split network* $\mathcal{N} = (\mathcal{X}, \mathcal{S}, w, d)$*, and* $S \in \mathcal{S}$*. Let*

$$\mathcal{I}(S) = \{S' \in \mathcal{S} : S' \textit{ is incompatible with } S\}$$

The **incompatible weight** *of the split* $S \in \mathcal{S}$ *is:*

$$\Omega(S) = \sum_{S' \in \mathcal{I}(S)} w(S')$$

and the **incompatible weight** *of* $\mathcal{N}$ *is the maximum incompatible weight among all* $S \in \mathcal{S}$*:*

$$\Omega(\mathcal{N}) = \max\{\Omega(S) : S \in \mathcal{S}\}$$

With all the definitions above, we can construct a similar algorithm for split network reconstruction:

1. Mini Reconstruction:

   The purpose of Mini Reconstruction is to uncover splits in a "small region". The definition of the small region depends on the following two factors:

   - The distances between taxa that we use in a small region to reconstruct splits needs to be reliable.
   - The size of a small region needs to be sufficiently large such that all taxa outside the small region have only one possible side to add in the process of Bipartition Extension.

   In the algorithm in [12], we can reconstruct edges between two close enough $x, y \in \mathcal{X}$ in small region with only the distance between x, y, and the distance from z

to x, y for every z in the small region. However, for split networks, in order to reconstruct splits between two close enough $x, y \in \mathcal{X}$ in a small region, either using the split decomposition method or the Neighbor-Net algorithm, we need all distances between any pair of taxa in the small region to be reliable. Hence, we need a different definition for the small region around x, y and show that with a proper choice of M, the distance between any two taxa in the small region is reliable.

Remark 4.17. *In this chapter, we use the split decomposition method to reconstruct the splits in a small region rather than Neighbor-Net because we can have a more general result: the split decomposition method can reconstruct weakly compatible networks, while Neighbor-Net can only reconstruct circular networks. However, if the original network is a circular network, Neighbor-Net, which is much faster than the split decomposition method, also returns the correct splits and estimates the weight from distorted metrics [43, 34]. We will discuss how we improve our algorithm by replacing the split decomposition with Neighbor-Net in section 4.6.*

2. Bipartition Extension:

   The purpose of Bipartition Extension is to extend a split in a small region from Mini Reconstruction to a split on the whole taxa $\mathcal{X}$, by recursively adding the taxa outside the small region to the bipartition or the split. The most important value is the connecting distance, as the value $\Delta(T) + \tau$ in the algorithm in [12], which needs to promise both

   - All taxa outside the small region have a way to connect to the small region.

- All taxa outside the small region have only one possible side to add in the process.

The second factor can be easily done by modifying the size of the small region, but the first one is much harder to show. In the algorithm in [12], the connectablity can be achieved from definition of the depth by choosing a connecting distance of about $\Delta(\mathcal{T})$. However, it is much harder in a split network to find the connecting distance and show it is proper. One main contribution of this chapter is showing that $\Delta(\mathcal{N}) + 2\Omega(\mathcal{N})$ is the proper value, with an example showing that this value is tight.

With this algorithm, our main result is:

Theorem 4.18. *Suppose $\mathcal{N} = (\mathcal{X}, \mathcal{S}, w, d)$ is a weakly compatible network. Given any $\tau > 0$ and $M > 3\Delta(\mathcal{N}) + 7\Omega(\mathcal{N}) + 10\tau$, there is a polynomial time algorithm to construct splits set $\hat{\mathcal{S}}$ with the weight estimator $\hat{w}$ from a (τ, M)-distortion $\hat{d}$ of d such that:*

- *$\mathcal{S}_{4\tau} \subset \hat{\mathcal{S}} \subset \mathcal{S}$, where $\mathcal{S}_{4\tau} = \{S \in \mathcal{S} : w(S) > 4\tau\}$.*
- *For any split $S \in \hat{\mathcal{S}}$, $|\hat{w}(S) - w(S)| < 2\tau$.*

4.2.3 Organization

The rest of the section is organized as follows. We first introduce the split decomposition method, which we use to reconstruct splits in small region in Mini Reconstruction, in section 4.3. The overall algorithm is detailed in section 4.4, and the proof of our main theorem is explained in 4.5. We show the lower bound of M in section 4.7, and discuss how we improve our algorithm in time complexity with Neighbor-Net in section 4.6.

4.3 Split decomposition method

The split decomposition method uses the idea of **d-splits**. In a split network $\mathcal{N} = (\mathcal{X}, \mathcal{S}, w, d)$, a split $S = \{S^1, S^2\} \in \mathcal{S}$ is a d-split if its **isolation index** $\alpha_d(S)$ is greater than 0 [31]:

$$\alpha_d(S) = \min\{\tilde{\alpha}_d(\{\{x_1, y_1\}, \{x_2, y_2\}\}) | x_1, y_1 \in S^1, x_2, y_2 \in S^2\} > 0$$

where

$$\begin{aligned}\tilde{\alpha}_d(\{\{x_1, y_1\}, \{x_2, y_2\}\}) &= \frac{1}{2}(\max\{d(x_1, y_1) + d(x_2, y_2), d(x_1, x_2) + d(y_1, y_2),\\ &\quad d(x_1, y_2) + d(y_1, x_2)\} - d(x_1, y_1) - d(x_2, y_2))\end{aligned}$$

The set of all d-splits on $\mathcal{X}$ is called the **split decomposition** of d. We can use the following algorithm to find the split decomposition of d.

Algorithm 4.19. *(split decomposition method [31])*
Given a distance matrix d on $\mathcal{X} = \{z_1, z_2, \ldots, z_n\}$, compute the set of all d-splits on $\mathcal{X}$ as follows:
Initially, set $\mathcal{X}_1 = \{z_1\}$ and $\mathcal{S}_1 = \emptyset$. Assume we have the set of all d-splits $\mathcal{S}_i$ on the first i taxa $\mathcal{X}_i = \{z_1, z_2, \ldots, z_i\}$. To obtain $\mathcal{S}_{i+1}$ on $\mathcal{X}_{i+1} = \{z_1, z_2, \ldots, z_{i+1}\}$, for each split $S = \{S^1, S^2\} \in \mathcal{S}_i$:

- *If $\alpha_d(\{S^1 \cup \{z_{i+1}\}, S^2\}) > 0$, then add $\{S^1 \cup \{z_{i+1}\}, S^2\}$ to $\mathcal{S}_{i+1}$.*
- *If $\alpha_d(\{S^1, S^2 \cup \{z_{i+1}\}\}) > 0$, then add $\{S^1, S^2 \cup \{z_{i+1}\}\}$ to $\mathcal{S}_{i+1}$.*
- *If $\alpha_d(\{\mathcal{X}_i, \{z_{i+1}\}\}) > 0$, then add $\{\mathcal{X}_i, \{z_{i+1}\}\}$ to $\mathcal{S}_{i+1}$.*

The result is given by $\mathcal{S}_n$.

In a weakly compatible network, all the splits are d-splits [3]. Therefore, the algorithm outputs all the splits in a weakly compatible network, with their isolation index. Since the isolation index only takes four values of the distortion $\hat{d}$, it is a good estimation of the weight. See section 4.33 for more details.

4.4 Algorithm

As we discussed in the introduction, the algorithm is similar to the phylogenetic tree reconstruction method which is introduced in [12]. It breaks into two parts: Mini Reconstruction and Bipartition Extension. The purpose of Mini Reconstruction is to uncover all the splits in "small regions" using the split decomposition method, and the purpose of Bipartition Extension is to extend each split we found in a small region to a split on all taxa by recursively adding taxa to the split. In detail, as shown in section 6, suppose we have a weakly compatible network $\mathcal{N} = (\mathcal{X}, \mathcal{S}, w, d)$ and $\tau > 0$: (See equation 4.1 and 4.2 for the definition of $B(x, y)$ and $\hat{\mathcal{S}}|_{C(x,y)}$)

Algorithm 4.20. *Network Reconstruction*

Main loop:

Input: $\mathcal{X}$, τ, $\Delta(\mathcal{N})$, $\Omega(\mathcal{N})$, *and a* (τ, M)*-distortion* $\hat{d}$ *of* d *with* $M > 3\Delta(\mathcal{N}) + 7\Omega(\mathcal{N}) + 10\tau$

Output: A set of splits $\hat{\mathcal{S}}$ *on* $\mathcal{X}$ *and a weight function* $\hat{w} : \hat{\mathcal{S}} \to (0, \infty)$

1. *Initially* $\hat{\mathcal{S}} = \emptyset$.

2. *Let* $EllipseRadius = 3\Delta + 7\Omega + 8\tau$, $ConnectDistance = \Delta + 2\Omega + \tau$.

3. *For each pair of taxa* $x, y \in \mathcal{X}$ *satisfying* $\hat{d}(x, y) \leq \Delta + \Omega + \tau$*:*

   $\hat{\mathcal{S}}|_{C(x,y)}$, $\hat{w}|_{C(x,y)}$ = *MiniReconstruction(*$\mathcal{X}$ *,*x*,* y*,* τ*,* $EllipseRadius$*,* $\hat{d}$*)*

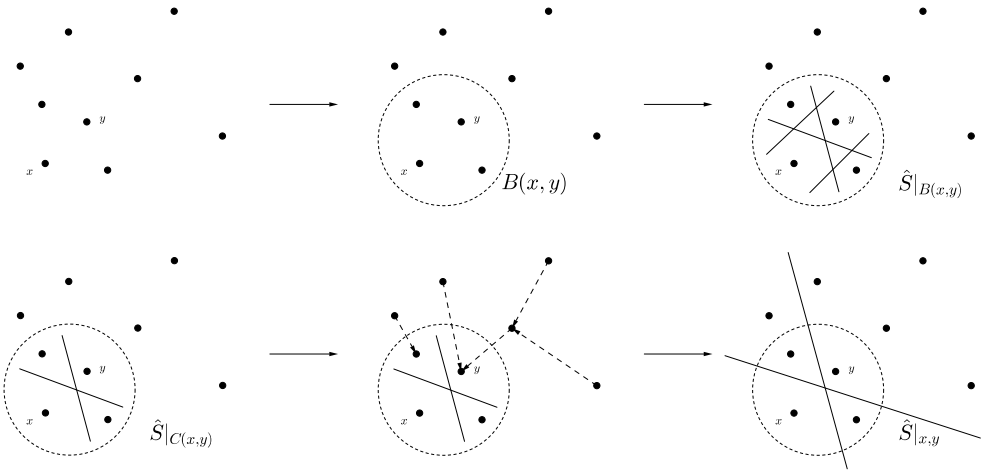


Figure 6: Steps of the algorithm

$\hat{\mathcal{S}}|_{x,y}$, $\hat{w}|_{x,y}$ = *BipartitionExtension*($\mathcal{X}$, $\hat{\mathcal{S}}|_{C(x,y)}$, $\hat{w}|_{C(x,y)}$, *ConnectDistance*, $\hat{d}$)

$\hat{\mathcal{S}} = \hat{\mathcal{S}} \cup \hat{\mathcal{S}}|_{x,y}$, *and* $w(\hat{S}|_{x,y}) = \hat{w}|_{x,y}(\hat{S}|_{x,y})$ *for any* $\hat{S}|_{x,y} \in \hat{\mathcal{S}}|_{x,y}$

4. *Return* $\hat{\mathcal{S}}$, $\hat{w}$

MiniReconstruction:

Input: $\mathcal{X}$, x, y, τ, *EllipseRadius*, $\hat{d}$

Output: A set of splits $\hat{\mathcal{S}}|_{C(x,y)}$ *on* $B(x, y)$, *and a weight function:* $\hat{w}|_{C(x,y)} : \hat{\mathcal{S}}|_{C(x,y)} \to (0, \infty)$

1. $B(x, y) = \{z \in \mathcal{X} : \hat{d}(z, x) + \hat{d}(z, y) \leq \mathit{EllipseRadius}\}$

2. *Apply the split decomposition method, algorithm 4.19, to find all the* $\hat{d}$*-splits and their isolation index on* $B(x, y)$. *Denote* $\hat{\mathcal{S}}|_{B(x,y)}$ *as the collection, and* $\hat{w}|_{B(x,y)} : \hat{\mathcal{S}}|_{B(x,y)} \to (0, \infty)$ *as the function recording the isolation index.*

3. $\hat{\mathcal{S}}|_{C(x,y)} = \{\hat{S}|_{C(x,y)} \in \hat{\mathcal{S}}|_{B(x,y)} : \delta_{\hat{S}|_{C(x,y)}}(x,y) = 1\}$. *And for any* $\hat{S}|_{C(x,y)} \in \hat{\mathcal{S}}|_{C(x,y)}$, $\hat{w}|_{C(x,y)}(\hat{S}|_{C(x,y)}) = \hat{w}|_{B(x,y)}(\hat{S}|_{C(x,y)})$.

4. *Return* $\hat{\mathcal{S}}|_{C(x,y)}$, $\hat{w}|_{C(x,y)}$

BipartitionExtension:

Input: $\mathcal{X}$, $\hat{\mathcal{S}}|_{C(x,y)}$, $\hat{w}|_{C(x,y)}$, *ConnectingDistance*, $\hat{d}$

Output: A set of splits $\hat{\mathcal{S}}|_{x,y}$ *on* $\mathcal{X}$, *and a weight function:* $\hat{w}|_{x,y} : \hat{\mathcal{S}}|_{x,y} \to (0,\infty)$

1. *Initially* $\hat{\mathcal{S}}|_{x,y} = \emptyset$

2. *For all* $S = \{S^1, S^2\} \in \hat{\mathcal{S}}|_{C(x,y)}$*:*

$w_0 = \hat{w}|_{C(x,y)}(S)$

While $\mathcal{X} - (S^1 \cup S^2) \neq \emptyset$*:*

Find $i \in 1,2$, $z \in S^i$, $z' \in \mathcal{X} - (S^1 \cup S^2)$ *such that* $\hat{d}(z,z') \leq$ *ConnectingDistance*

$S^i = S^i \cup \{z'\}$

$\hat{\mathcal{S}}|_{x,y} = \hat{\mathcal{S}}|_{x,y} \cup \{S\}$, $\hat{w}|_{x,y}(S) = w_0$

3. *Return* $\hat{\mathcal{S}}|_{x,y}$, $\hat{w}|_{x,y}$

Our main result is $\mathcal{S}_{4\tau} \subset \hat{\mathcal{S}} \subset \mathcal{S}$, where $\mathcal{S}_{4\tau} \subset \mathcal{S}$ contains all splits heavier than 4τ, and the estimator $\hat{w}$ satisfies $|w(S) - \hat{w}(S)| < 2\tau$ for any $S \in \hat{S}$. We will prove our main result in the next section.

There are at most $O(n^2)$ pairs of x, y which satisfy $\hat{d}(x,y) \leq \Delta + \Omega + \tau$. For each $B(x,y)$. the split decomposition method takes $O(n^6)$ time and generates at most $O(n^2)$ of $\hat{d}$-split [3], which implies that Mini Reconstruction takes $O(n^6)$ time, and the size of $\hat{\mathcal{S}}|_{C(x,y)}$ is at most $O(n^2)$. For every $\hat{S}|_{C(x,y)} \in \hat{\mathcal{S}}|_{C(x,y)}$, we use the following way to

implement the while loop in Bipartition Extension. Imagine an undirected graph with n vertices corresponding to the n taxa, and there is an edge between $x, y \in \mathcal{X}$ if and only if $\hat{d}(x, y) < \Delta + 2\Omega + \tau$. Then the while loop is like doing the traversal on the graph, which can be implemented using DFS and takes $O(n^2)$ time. This implies that Bipartition Extension takes $O(n^2 \cdot |\hat{\mathcal{S}}|_{C(x,y)}|)$ time. Thus, the running time of the whole algorithm is:

$$O(n^2) \cdot (O(n^6) + O(n^2) \cdot O(n^2)) = O(n^8)$$

4.5 Analysis

In this section, we will prove our main theorem:

Theorem 4.18. *Suppose $\mathcal{N} = (\mathcal{X}, \mathcal{S}, w, d)$ is a weakly compatible network. Given any $\tau > 0$ and $M > 3\Delta(\mathcal{N}) + 7\Omega(\mathcal{N}) + 10\tau$, there is a polynomial time algorithm to construct splits set $\hat{\mathcal{S}}$ with the weight estimator $\hat{w}$ from a (τ, M)-distortion $\hat{d}$ of d such that:*

- *$\mathcal{S}_{4\tau} \subset \hat{\mathcal{S}} \subset \mathcal{S}$, where $\mathcal{S}_{4\tau} = \{S \in \mathcal{S} : w(S) > 4\tau\}$.*
- *For any split $S \in \hat{\mathcal{S}}$, $|\hat{w}(S) - w(S)| < 2\tau$.*

In the remainder of this section, we will use the following notations: $\mathcal{N} = (\mathcal{X}, \mathcal{S}, w, d)$ is a weakly compatible network. Δ and Ω are the depth and the incompatible weight of $\mathcal{N}$, respectively. Also, let $\hat{d}$ be a (τ, M)-distorted metric of $\mathcal{N}$, with $M > 3\Delta + 7\Omega + 10\tau$. Moreover, for any $x, y \in \mathcal{X}$ with $\hat{d}(x, y) \leq \Delta + \Omega + \tau$, let $B(x, y)$ be the "small region":

$$B(x, y) = \{z \in \mathcal{X} : \hat{d}(z, x) + \hat{d}(z, y) \leq 3\Delta + 7\Omega + 8\tau\} \tag{4.1}$$

We denote $\hat{S}|_{B(x,y)}$ as any $\hat{d}$-split on $B(x,y)$ which is found via the split decomposition method with isolation index larger than 2τ in the Mini Reconstruction part, and $\hat{\mathcal{S}}|_{B(x,y)}$ as the collection of $\hat{S}|_{B(x,y)}$. Let $\hat{\mathcal{S}}|_{C(x,y)}$ be the subset of $\hat{\mathcal{S}}|_{B(x,y)}$ which contains all the splits separating x and y:

$$\hat{\mathcal{S}}|_{C(x,y)} = \{\hat{S}|_{C(x,y)} \in \hat{\mathcal{S}}|_{B(x,y)} : \delta_{\hat{S}|_{C(x,y)}}(x,y) = 1\} \tag{4.2}$$

The Bipartition Extension part will extend every split $\hat{S}|_{C(x,y)} \in \hat{\mathcal{S}}|_{C(x,y)}$ to a split $\hat{S}|_{x,y}$ on all the taxa $\mathcal{X}$, and suppose $\hat{\mathcal{S}}|_{x,y}$ are the collection of all $\hat{S}|_{x,y}$ (See section 6). The algorithm will output:

$$\hat{\mathcal{S}} = \cup_{x,y\in\mathcal{X}:\hat{d}(x,y)\leq\Delta+\Omega+\tau}\, \hat{\mathcal{S}}|_{x,y}$$

One of our main claim is $\mathcal{S}_{4\tau} \subset \hat{\mathcal{S}} \subset \mathcal{S}$. For convenient comparison purpose, for any $x, y \in \mathcal{X}$ satisfies $\hat{d}(x,y) \leq \Delta + \Omega + \tau$, we let:

$$\begin{aligned} \mathcal{S}|_{x,y} &= \{S|_{x,y} \in \mathcal{S} : \delta_{S|_{x,y}}(x,y) = 1\} \\ \mathcal{S}_{4\tau}|_{x,y} &= \{S_{4\tau}|_{x,y} \in \mathcal{S}_{4\tau} : \delta_{S_{4\tau}|_{x,y}}(x,y) = 1\} \end{aligned}$$

and

$$\begin{aligned} \mathcal{S}|_{C(x,y)} &= \{\{S^x \cap B(x,y), S^y \cap B(x,y)\} : S = \{S^x, S^y\} \in \mathcal{S}|_{x,y}\} \\ \mathcal{S}_{4\tau}|_{C(x,y)} &= \{\{S^x \cap B(x,y), S^y \cap B(x,y)\} : S = \{S^x, S^y\} \in \mathcal{S}_{4\tau}|_{x,y}\} \end{aligned}$$

We will show the following claims to prove our main theorem:

(A) For any $x, y \in \mathcal{X}$ satisfies $\hat{d}(x,y) \leq \Delta + \Omega + \tau$, $\mathcal{S}_{4\tau}|_{C(x,y)} \subset \hat{\mathcal{S}}|_{C(x,y)} \subset \mathcal{S}|_{C(x,y)}$.

(B) For any $x, y \in \mathcal{X}$ satisfies $\hat{d}(x,y) \leq \Delta + \Omega + \tau$, $\mathcal{S}_{4\tau}|_{x,y} \subset \hat{\mathcal{S}}|_{x,y} \subset \mathcal{S}|_{x,y}$.

(C) $\mathcal{S} = \cup_{x,y\in\mathcal{X}:\hat{d}(x,y)\leq\Delta+\Omega+\tau}\mathcal{S}|_{x,y}$, so $\mathcal{S}_{4\tau} = \cup_{x,y\in\mathcal{X}:\hat{d}(x,y)\leq\Delta+\Omega+\tau}\mathcal{S}_{4\tau}|_{x,y}$ and $\mathcal{S}_{4\tau} \subset \hat{\mathcal{S}} \subset \mathcal{S}$.

(D) The isolation index from the split decomposition method is a good estimation of the weight for all splits.

The proof consists of following parts:

1. Mini Reconstruction:

   - Executability: The split decomposition method is executable from [3].
   - Correctness: We will show that the distance between any pair of taxa in $B(x, y)$ in (τ, M)-distorted metric $\hat{d}$ is reliable (Lemma 4.31), and then prove claim (A) that $\mathcal{S}_{4\tau}|_{C(x,y)} \subset \hat{\mathcal{S}}|_{C(x,y)} \subset \mathcal{S}|_{C(x,y)}$ (Lemma 4.33).
   - Estimation on the weight: Lemma 4.34 indicates that the isolation index provides a good estimation on the weight function w, which concludes claim (D).

2. Bipartition Extension:

   - Executability: We will show that Bipartition Extension always returns one and only one split on $\mathcal{X}$ for every $\hat{S}|_{C(x,y)} \in \hat{\mathcal{S}}|_{C(x,y)}$ (Lemma 4.29).
   - Correctness: The same proposition (Lemma 4.29) also shows that Bipartition Extension extends any $S|_{C(x,y)}$ back to corresponding $S|_{x,y}$. This, with claim (A), proves claim (B). Last, claim (C) follows immediately after we show that for any $S \in \mathcal{S}$, there exists x, y such that $\hat{d}(x, y) \leq \Delta+\Omega+\tau$ and $\delta_S(x, y) = 1$ (Lemma 4.24).

We will use the following notations:

Notation 4.21. *For any $x, y \in \mathcal{X}$, we define:*

$$\Xi(x,y) = \{S \in \mathcal{S} : \delta_S(x,y) = 1\}, \xi(x,y) = \Xi^C(x,y) = \{S \in \mathcal{S} : \delta_S(x,y) = 0\}$$

and $\forall S \in \mathcal{S}$, we define:

$$\begin{aligned} \mathcal{C}(S) &= \{S' \in \mathcal{S} : S' \textit{ is compatible to } S\} \\ \mathcal{I}(S) = \mathcal{C}^C(S) &= \{S' \in \mathcal{S} : S' \textit{ is incompatible to } S\} \end{aligned}$$

For convenience, we also make the following notations:

$$\Xi\mathcal{C}(x,y,S) = \Xi(x,y) \cap \mathcal{C}(S) \quad \textit{and} \quad \xi\mathcal{C}(x,y,S) = \xi(x,y) \cap \mathcal{C}(S)$$

$$\Xi\mathcal{I}(x,y,S) = \Xi(x,y) \cap \mathcal{I}(S) \quad \textit{and} \quad \xi\mathcal{I}(x,y,S) = \xi(x,y) \cap \mathcal{I}(S)$$

4.5.1 Distance Lemmas

First we introduce some lemmas about the distance between taxa.

Lemma 4.22. *Suppose $x_1, x_2, y_1, y_2 \in \mathcal{X}$. Let w_{x_1} be the total weight that separates $\{x_1\}$ and three other points, with similar definitions for w_{y_1}, w_{x_2}, and w_{y_2}. Moreover, let w_1 be the total weight that separates $\{x_1, y_1\}$ and $\{x_2, y_2\}$, w_2 be the total weight that separates $\{x_1, x_2\}$ and $\{y_1, y_2\}$, and w_3 be the total weight that separate $\{x_1, y_2\}$ and $\{x_2, y_1\}$, as shown Lemma 7. then*

$$\begin{aligned} d(x_1,y_1) &= w_{x_1} + w_{y_1} + w_2 + w_3 & d(x_2,y_2) &= w_{x_2} + w_{y_2} + w_2 + w_3 \\ d(x_1,x_2) &= w_{x_1} + w_{x_2} + w_1 + w_3 & d(y_1,y_2) &= w_{y_1} + w_{y_2} + w_1 + w_3 \\ d(x_1,y_2) &= w_{x_1} + w_{y_2} + w_1 + w_2 & d(y_1,x_2) &= w_{y_1} + w_{x_2} + w_1 + w_2 \end{aligned}$$

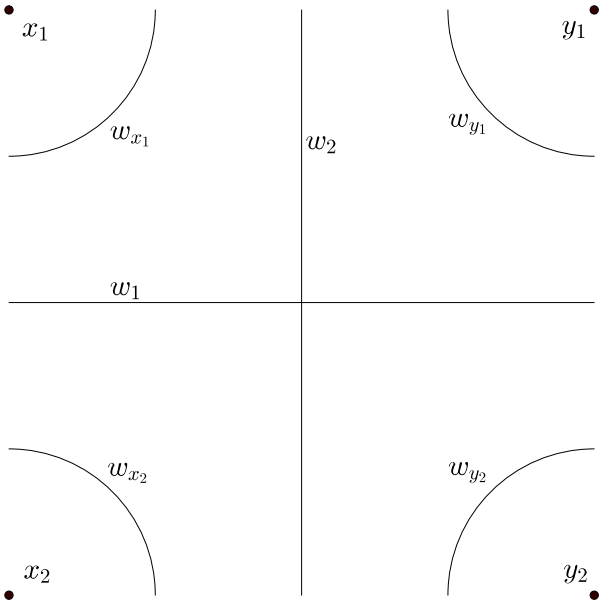


Figure 7: Lemma 4.22

Lemma 4.23. *If $x, y \in \mathcal{X}$ are the witness of the depth of a split $S_0 \in \mathcal{S}$, that is,*

$$\sum_{\Xi\mathcal{C}(x,y,S)} w(S') = \Delta(S)$$

then $d(x, y) \leq \Delta + \Omega$

Proof.

$$\begin{aligned} d(x,y) = \sum_{\Xi(x,y)} w(S') &= \sum_{\Xi\mathcal{C}(x,y,S)} w(S') + \sum_{\Xi\mathcal{I}(x,y,S)} w(S') \\ &\leq \sum_{\Xi\mathcal{C}(x,y,S)} w(S') + \sum_{\mathcal{I}(S)} w(S') = \Delta(S) + \Omega(S) \leq \Delta + \Omega \end{aligned}$$

□

Corollary 4.24. *For any split $S \in \mathcal{S}$, there exists $x, y \in \mathcal{X}$ such that $\delta_S(x, y) = 1$ and $d(x, y) \leq \Delta + \Omega$. So $\hat{d}(x, y) \leq \Delta + \Omega + \tau$, and $S|_{x,y}$ exists.*

Proof. Choose x, y to be the witness of the depth of S, then $\delta_S(x, y) = 1$ and by previous lemma, Lemma 4.23, $d(x, y) \leq \Delta + \Omega$. □

Lemma 4.25. *Suppose $x, y, z, z' \in \mathcal{X}$. If there exists a split $S \in \mathcal{S}$ that separates $\{z, x\}$ and $\{z', y\}$, then $d(z, x) + d(z, y) \leq 2d(z, z') + d(x, y) + 2\Omega$*

Proof. Let $x_1 = x$, $y_1 = y$, $x_2 = z$, $y_2 = z'$. Notice that any splits that separates $\{x = x_1, y = y_1\}$ and $\{z = x_2, z' = y_2\}$ will be incompatible with S, which implies that, as in Lemma 4.22, $w_1 \leq \Omega(S) \leq \Omega$. Therefore, by Lemma 4.22,

$$\begin{aligned}
d(z, x) + d(z, y) &= d(x_1, x_2) + d(x_2, y_1) \\
&= (w_{x_1} + w_{x_2} + w_1 + w_3) + (w_{x_2} + w_{y_1} + w_1 + w_2) \\
&= 2w_{x_2} + (w_{x_1} + w_{y_1} + w_2 + w_3) + 2w_1 \\
&\leq 2(w_{x_2} + w_{y_2} + w_2 + w_3) + (w_{x_1} + w_{y_1} + w_2 + w_3) + 2w_1 \\
&= 2d(x_2, y_2) + d(x_1, y_1) + 2w_1 \\
&= 2d(z, z') + d(x, y) + 2w_1 \leq 2d(z, z') + d(x, y) + 2\Omega
\end{aligned}$$

□

4.5.2 Bipartition Extension

We first prove the part that is related to the second step of the algorithm: Bipartition Extension. The following lemma shows that the while loop in Bipartition Extension always terminate.

Lemma 4.26. *For any $z, z' \in \mathcal{X}$, there exists a sequence of taxa $\{z_i\}_{i=0}^{k+1} \subset \mathcal{X}$ such that $z_0 = z$, $z_{k+1} = z'$, and for any $0 \leq i \leq k$, $d(z_i, z_{i+1}) \leq \Delta + 2\Omega$.*

Proof. For convenience, we say $z, z' \in \mathcal{X}$ are **connectable** if $\exists\ \{z_i\}_{i=0}^{k+1} \subset \mathcal{X}$, $z_0 = z$, $z_{k+1} = z'$, and for any $0 \leq i \leq k$, $d(z_i, z_{i+1}) \leq \Delta + 2\Omega$. Our goal is to prove that any two taxa in $\mathcal{X}$ are connectable. Our first observation is that the connectable property is an equivalence relation, which is an important fact in the proof.

The case $|\mathcal{S}| = 1$ is not interesting and easy to prove, so we assume that $|\mathcal{S}| \geq 2$. We will show later that, for any $S = \{S^1, S^2\} \in \mathcal{S}$, any two taxa in S^1 or in S^2 are connectable. If this claim holds, choose any two different splits in $\mathcal{S}$, $S_1 = \{S_1^1, S_1^2\}$ and $S_2 = \{S_2^1, S_2^2\}$. Notice that at least one of S_2^1, S_2^2 has a non-empty intersection with both S_1^1 and S_1^2, say both $S_2^1 \cap S_1^1$ and $S_2^1 \cap S_1^2$ are non-empty. Pick $v_1 \in S_2^1 \cap S_1^1$ and $v_2 \in S_2^1 \cap S_1^2$, then by the claim above, every taxon in S_1^1 is connectable to v_1 because they are both in S_1^1, every taxa in S_1^2 is connectable to v_2 because they are both in S_1^2, and v_1 and v_2 are connectable because they are both in S_2^1. Now since the connectable property is an equivalence relation, we know that any two taxa in $\mathcal{X}$ are connectable, and this finishes the proof (See graph 8). Therefore, what remains is showing the claim that, for any $S = \{S^1, S^2\} \in \mathcal{S}$, any two taxa in S^1 or in S^2 are connectable.

The way we prove the claim is using induction on the size of all possible S^1 and S^2. Let $\mathcal{S}^*$ be the set containing all possible S^1 and S^2:

$$\mathcal{S}^* = \{S^1, S^2 : S = \{S^1, S^2\} \in \mathcal{S}\}$$

The basic case is when $S_0 = \{S_0^1, S_0^2\} \in \mathcal{S}$ satisfies that $|S_0^1|$ is the smallest among all the elements in $\mathcal{S}^*$, and we want to show that any two taxa in S_0^1 are connectable. For any $z, z' \in S_0^1$, if $\Xi\mathcal{C}(z, z', S_0) \neq \emptyset$, then for any $S = \{S^1, S^2\} \in \Xi\mathcal{C}(z, z', S_0)$, both $S_0^1 \cap S^1$ and $S_0^1 \cap S^2$ are nonempty because one contains z and the other one contains z'. Since S, S_0 are incompatible, one of $S_0^2 \cap S^1$ and $S_0^2 \cap S^2$ need to be empty set. Without lost

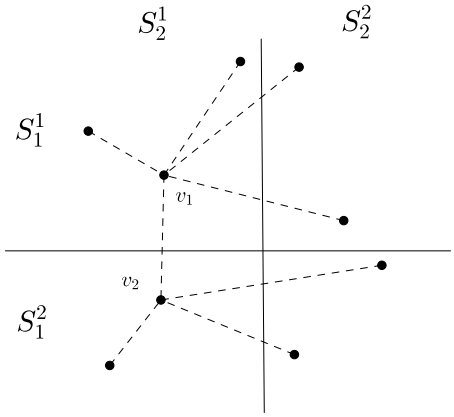


Figure 8: Lemma 4.26 Graph 1

of generality say $S_0^2 \cap S^1 = \emptyset$, then $S^1 \subset \mathcal{X} - S_0^2 = S_0^1$. Notice that one of z, z' is not in S^1, so $S^1 \subsetneq S_0^1$. This contradicts the assumption that $|S_0^1|$ is the smallest among $\mathcal{S}^*$. Hence, $\Xi\mathcal{C}(z, z', S_0) = \emptyset$. Therefore,

$$\begin{aligned} d(z,z') = \sum_{\Xi(z,z')} w(S') &= \sum_{\Xi\mathcal{C}(z,z',S_0)} w(S') + \sum_{\Xi\mathcal{I}(z,z',S_0)} w(S') \\ &= \sum_{\Xi\mathcal{I}(z,z',S_0)} w(S') \leq \sum_{\mathcal{I}(S_0)} w(S') = \Omega(S_0) \leq \Omega \end{aligned}$$

This shows that the claim holds for the basic case.

Now suppose we have a split $S_0 = \{S_0^1, S_0^2\} \in \mathcal{S}$ and, by induction hypothesis, assume that for any $S = \{S^1, S^2\} \in \mathcal{S}$, if $|S^1| < |S_0^1|$, then any two taxa in S^1 are connectable. By definition of the depth, there are two points $v_1 \in S_0^1$, $v_2 \in S_0^2$ such that:

$$\sum_{S \in \Xi\mathcal{C}(v_1,v_2,S_0)} w(S') = \Delta(S_0)$$

What we will show next is that every taxon in S_0^1 is connectable to v_1, and so any two

points in S_0^1 are connectable since connectability is an equivalence relation, and this finishes the proof of the claim and concludes the proposition.

Pick any $v \in S_0^1$. The set of all splits that separate v_1, v, $\Xi(v_1, v)$, can be broken into three categories, as in figure 9:

$$\begin{aligned}\Xi(v_1, v) &= \Xi\mathcal{I}(v_1, v, S_0) \cup \Xi\mathcal{C}(v_1, v, S_0) \\ &= \Xi\mathcal{I}(v_1, v, S_0) \cup [\Xi\mathcal{C}(v_1, v, S_0) \cap \Xi(v_1, v_2)] \cup [\Xi\mathcal{C}(v_1, v, S_0) \cap \xi(v_1, v_2)]\end{aligned}$$

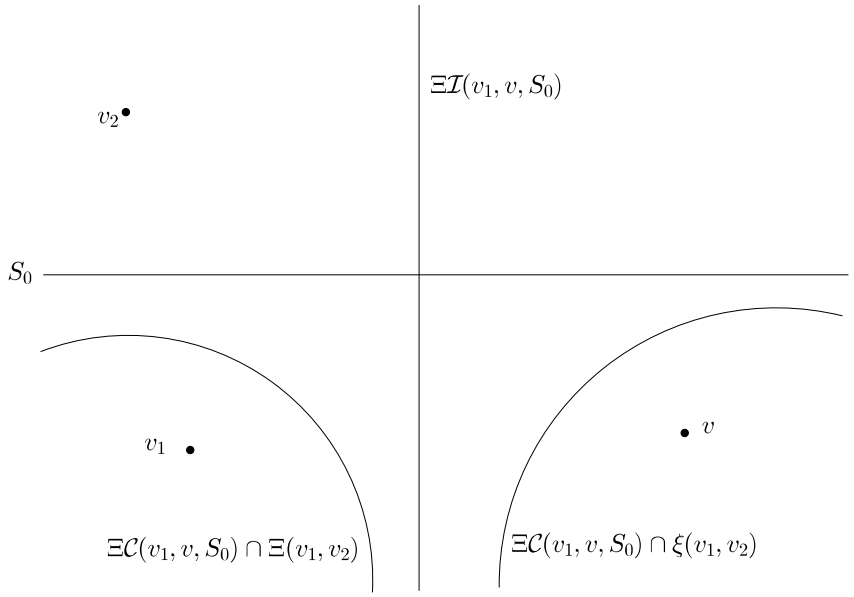


Figure 9: Lemma 4.26 Graph 2

If $\Xi\mathcal{C}(v_1, v, S_0) \cap \xi(v_1, v_2) = \emptyset$, then

$$\begin{aligned} d(v_1, v) = \sum_{\Xi(v_1,v)} w(S') &= \sum_{\Xi\mathcal{I}(v_1,v,S_0)} w(S') + \sum_{\Xi\mathcal{C}(v_1,v,S_0)\cap\Xi(v_1,v_2)} w(S') \\ &\leq \sum_{\mathcal{I}(S_0)} w(S') + \sum_{\Xi\mathcal{C}(v_1,v_2,S_0)} w(S') = \Omega(S_0) + \Delta(S_0) + \leq \Omega + \Delta \end{aligned}$$

which shows that v_1, v are connectable.

If $\Xi\mathcal{C}(v_1, v, S_0) \cap \xi(v_1, v_2)$ is nonempty, pick $S_M = \{S_M^1, S_M^2\} \in \Xi\mathcal{C}(v_1, v, S_0) \cap \xi(v_1, v_2)$, with the assumption that $v \in S_M^1$ and $v_1, v_2 \in S_M^2$, such that S_M is "maximal". That is, for any $S = \{S^1, S^2\} \in \Xi\mathcal{C}(v_1, v, S_0) \cap \xi(v_1, v_2)$ with $v \in S^1$, $S_M^1 \subset S^1$ if and only if $S = S_M$. Again by definition of the depth, there exists $u_1 \in S_M^1$, $u_2 \in S_M^2$ such that:

$$\sum_{\Xi\mathcal{C}(u_1,u_2,S_M)} w(S') = \Delta(S_M)$$

Notice that because $S_M \in \mathcal{C}(S_0)$, and $v_1 \in S_0^1 \cap S_M^2$, $v_2 \in S_0^2 \cap S_M^2$, $v \in S_0^1 \cap S_M^1$, one must have $S_0^2 \cap S_M^1 = \emptyset$. As a result, $S_M^1 \subset \mathcal{X} - S_0^2 = S_0^1$. Since $v_1 \in S_0^1 - S_M^1$, $|S_M^1| < |S_0^1|$. By induction hypothesis, any two taxa in S_M^1 are connectable. In particular, v, u_1 are connectable. We will now show that v_1, u_1 are connectable, then v_1, v are connectable and the proof ends.

We first show that the set:

$$\mathscr{S} = \{S \in \mathcal{S} : S \in \mathcal{C}(S_0) \cap \mathcal{C}(S_M), S \text{ separates } \{v_1, v_2\} \text{ and } \{u_1, u_2\}\}$$

is empty. Suppose not and assume $S = \{S^1, S^2\} \in \mathscr{S}$. Without loss of generality, we assume that $\{v_1, v_2\} \subset S^1$ and $\{u_1, u_2\} \subset S^2$. Recall that $\{v, u_1\} \subset S_M^1$ and $\{v_1, v_2, u_2\} \subset S_M^2$. If $v \in S^1$, then $\{v_1, v, u_1, u_2\}$ are the witness that S_M and S are incompatible, as the left graph in Figure 10, contradiction. Thus, $v \in S^2$ and so S satisfies

1. $S \in \mathcal{C}(S_0)$ because $S \in \mathscr{S}$, and

2. $\delta_S(v_1, v) = 1$, and $\delta_S(v_1, v_2) = 0$.

Consequently, $S \in \Xi\mathcal{C}(v_1, v, S_0) \cap \xi(v_1, v_2)$. Because we choose S_M to be "maximal", and $S \neq S_M$ ($\delta_S(u_1, u_2) = 0$ but $\delta_{S_M}(u_1, u_2) = 1$), there exists $w \in S^1_M - S^2$. Therefore,

$$\{v, u_1, w\} \subset S^1_M, \{v_1, v_2, u_2\} \subset S^2_M \text{ and } \{v_1, v_2, w\} \subset S^1, \{v, u_1, u_2\} \subset S^2$$

And so $\{w, v, v_1, u_2\}$ are the witness that S, S_M are incompatible, as the right graph in Figure 10, contradiction. This shows that $\mathscr{S}$ is empty.

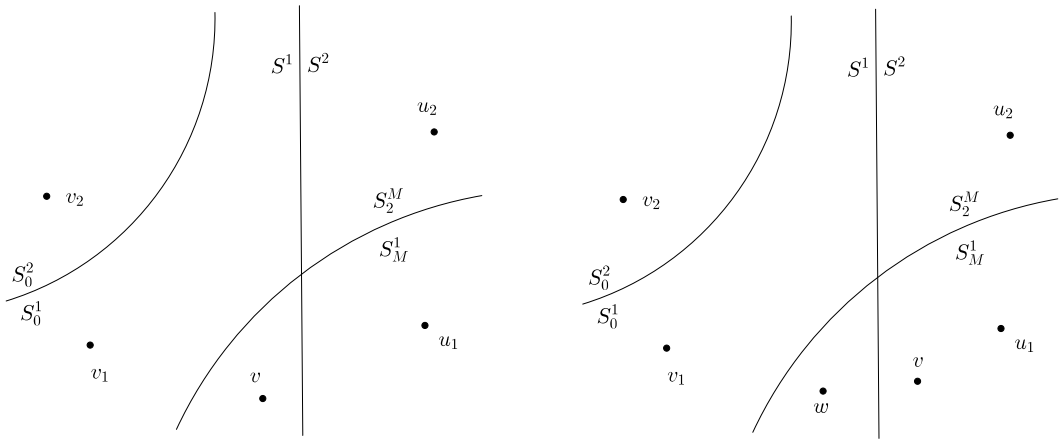


Figure 10: Lemma 4.26 Graph 3

Now we start the proof that v_1, u_1 are connectable. By Lemma 4.22. with $x_1 = v_1$,

$x_2 = v_2$, $y_1 = u_1$, $y_2 = u_2$, we have

$$\begin{aligned}
d(v_1, u_1) + d(v_2, u_2) &= d(x_1, y_1) + d(x_2, y_2) \\
&= (w_{x_1} + w_{y_1} + w_2 + w_3) + (w_{x_2} + w_{y_2} + w_2 + w_3) \\
&= w_{x_1} + w_{x_2} + w_{y_1} + w_{y_2} + 2w_2 + 2w_3 \\
&\leq w_{x_1} + w_{x_2} + w_{y_1} + w_{y_2} + 2w_1 + 2w_2 + 2w_3 \\
&= (w_{x_1} + w_{x_2} + w_1 + w_3) + (w_{y_1} + w_{y_2} + w_1 + w_3) + 2w_2 \\
&= d(x_1, x_2) + d(y_1, y_2) + 2w_2 \\
&= d(v_1, v_2) + d(u_1, u_2) + 2w_2
\end{aligned}$$

where w_2 is the total weight of the splits that separate $\{x_1, x_2\} = \{v_1, v_2\}$ and $\{y_1, y_2\} = \{u_1, u_2\}$. From the previous paragraph, we know that any split S that separates $\{v_1, v_2\}$ and $\{u_1, u_2\}$ is either incompatible with S_0 or incompatible with S_M, and it satisfies both $\delta_S(v_1, v_2) = 0$ and $\delta_S(u_1, u_2) = 0$. Thus,

$$w_2 \leq \sum_{\xi \mathcal{I}(v_1, v_2, S_0)} w(S') + \sum_{\xi \mathcal{I}(u_1, u_2, S_M)} w(S')$$

Therefore,

$$\begin{aligned}
& d(v_1,u_1)+d(v_2,u_2) \\
\leq\ & d(v_1,v_2)+d(u_1,u_2)+2w_2 \\
\leq\ & \sum_{\Xi(v_1,v_2)} w(S') + \sum_{\Xi(u_1,u_2)} w(S') + 2\cdot[\sum_{\xi\mathcal{I}(v_1,v_2,S_0)} w(S') + \sum_{\xi\mathcal{I}(u_1,u_2,S_M)} w(S')] \\
=\ & [\sum_{\Xi\mathcal{C}(v_1,v_2,S_0)} w(S') + \sum_{\Xi\mathcal{I}(v_1,v_2,S_0)} w(S')] + [\sum_{\Xi\mathcal{C}(u_1,u_2,S_M)} w(S') + \sum_{\Xi\mathcal{I}(u_1,u_2,S_M)} w(S')] \\
& +2\cdot[\sum_{\xi\mathcal{I}(v_1,v_2,S_0)} w(S') + \sum_{\xi\mathcal{I}(u_1,u_2,S_M)} w(S')] \\
\leq\ & \sum_{\Xi\mathcal{C}(v_1,v_2,S_0)} w(S') + \sum_{\Xi\mathcal{C}(u_1,u_2,S_M)} w(S') + 2\cdot[\sum_{\Xi\mathcal{I}(v_1,v_2,S_0)} w(S') + \sum_{\xi\mathcal{I}(v_1,v_2,S_0)} w(S')] \\
& +2\cdot[\sum_{\Xi\mathcal{I}(u_1,u_2,S_M)} w(S') + \sum_{\xi\mathcal{I}(u_1,u_2,S_M)} w(S')] \\
=\ & \Delta(S_0)+\Delta(S_M)+2\sum_{\mathcal{I}(S_0)} w(S') + 2\sum_{\mathcal{I}(S_M)} w(S') \\
=\ & \Delta(S_0)+\Delta(S_M)+2\Omega(S_0)+2\Omega(S_M) \leq 2\Delta+4\Omega
\end{aligned}$$

If $d(v_1,u_1) \leq \Delta+2\Omega$, then v_1,u_1 are connectable. If $d(v_1,u_1) > \Delta+2\Omega$, then by the inequality above, $d(v_2,u_2) \leq \Delta+2\Omega$. Recall that v_1,v_2,u_1,u_2 satisfy

$$\sum_{S\in\Xi\mathcal{C}(v_1,v_2,S_0)} w(S') = \Delta(S_0) \ \text{ and } \sum_{\Xi\mathcal{C}(u_1,u_2,S_M)} w(S') = \Delta(S_M)$$

so by Lemma 4.23

$$d(v_1,v_2) \leq \Delta+\Omega \ \text{ and } \ d(u_1,u_2) \leq \Delta+\Omega$$

Hence, $\{v_1,v_2,u_2,u_1\}$ is the sequence which shows that v_1,u_1 are connectable. Either way we have v_1,u_1 are connectable, and this finishes the proof. □

Remark 4.27. *It might be questioned about the tightness of $\Delta+2\Omega$, especially the factor 2. The graph in Figure 11 shows why the factor 2 is tight. In the graph, every vertex*

denotes a taxa, and every line denotes a split with weight 1. It is not hard to check that in this split network, $\Delta = 1$, $\Omega = 3$, *and the smallest distance between taxa in the left part and taxa in the right part is 6. We can generalize this network by extending the number of splits in 4 different set of splits from* 3 *to* n. *Then in the general network,* Δ *is still 1, but* Ω *is* n, *and the smallest distance between taxa on the left and taxa on the right becomes* $2n$. *This shows the tightness of factor 2.*

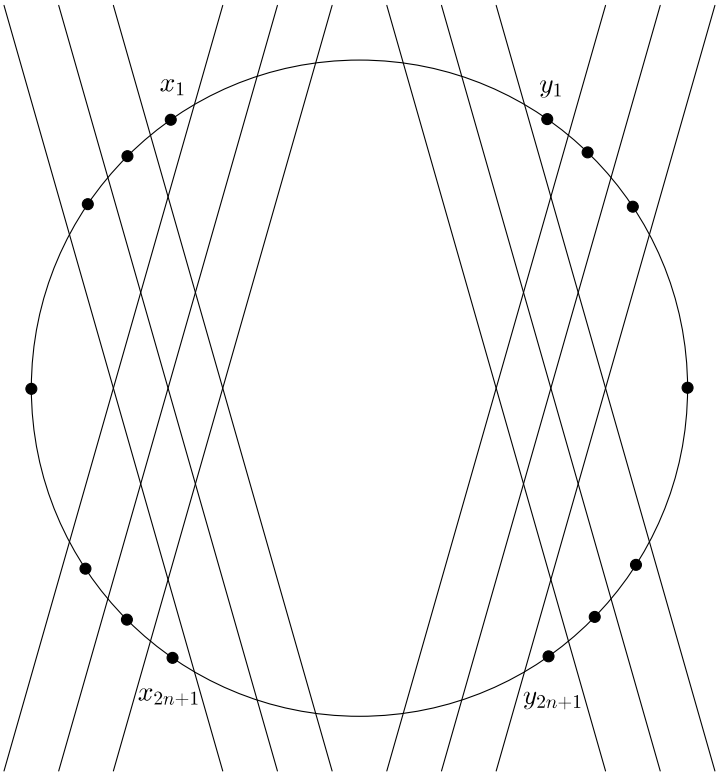


Figure 11: Example of the factor 2 is tight

Furthermore, if we add four more splits to the network:

$$\{x_1, \mathcal{X} - \{x_1\}\}, \{x_{2n+1}, \mathcal{X} - \{x_{2n+1}\}\}, \{y_1, \mathcal{X} - \{y_1\}\}, \{y_{2n+1}, \mathcal{X} - \{y_{2n+1}\}\}$$

then $\Delta = 2$, $\Omega = n$, and the smallest distance between taxa on the left and that on the right is $2n + 2$. This shows the tightness of $\Delta + 2\Omega$.

The next lemma gives an idea why Bipartition Extension always chooses the right side when adding taxa outside the small region.

Lemma 4.28. *Suppose $x, y \in \mathcal{X}$ satisfies $\hat{d}(x, y) \leq \Delta + \Omega + \tau$ and $S = \{S^x, S^y\} \in \mathcal{S}|_{x,y}$ is a split that separates x and y. If $z \in S^x - B(x, y)$, then $\forall z' \in S^y$, $\hat{d}(z, z') > \Delta + 2\Omega + \tau$.*

Proof. Since $z \in \mathcal{X} - B(x, y)$, we know that $\hat{d}(z, x) + \hat{d}(z, y) > 3\Delta + 7\Omega + 8\tau$. If $d(z, x) + d(z, y) \leq 3\Delta + 7\Omega + 6\tau < M$, then both $d(z, x)$ and $d(z, y)$ are less than M. By definition of distorted metrics,

$$\hat{d}(z, x) + \hat{d}(z, y) \leq (d(z, x) + \tau) + (d(z, y) + \tau) \leq 3\Delta + 7\Omega + 8\tau$$

which causes a contradiction. Hence, $d(z, x) + d(z, y) > 3\Delta + 7\Omega + 6\tau$. Moreover, because $z \in S^x$ and $z' \in S^y$, we know that S separate $\{z, x\}$ and $\{z', y\}$, and hence by Lemma 4.25, $d(z, z') \geq \frac{1}{2}(d(z, x) + d(z, y) - d(x, y) - 2\Omega)$. In addition, since we choose x, y such that $\hat{d}(x, y) \leq \Delta + \Omega + \tau < M$, we know that $d(x, y) \leq \Delta + \Omega + 2\tau$. Combining all the above information,

$$\begin{aligned} d(z, z') &\geq \frac{1}{2}(d(z, x) + d(z, y) - d(x, y) - 2\Omega) \\ &> \frac{1}{2}[(3\Delta + 7\Omega + 6\tau) - (\Delta + \Omega + 2\tau) - 2\Omega] \\ &= \frac{1}{2}(2\Delta + 4\Omega + 4\tau) = \Delta + 2\Omega + 2\tau \end{aligned}$$

which implies that $\hat{d}(z, z') > \Delta + 2\Omega + 2\tau - \tau = \Delta + 2\Omega + \tau$

□

The previous two lemmas lead us to an important proposition:

Proposition 4.29. *For any $S|_{x,y} = \{S^x, S^y\} \in \mathcal{S}|_{x,y}$, let*

$$S|_{C(x,y)} = \{S^x \cap B(x,y), S^y \cap B(x,y)\} \in \mathcal{S}|_{C(x,y)}$$

be the corresponding split on $B(x,y)$. Then there is one and only one possible output when applying Bipartition Extension on $S|_{C(x,y)}$, and the output is $S|_{x,y}$.

Proof. By Lemma 4.26, we know for any $z \in \mathcal{X} - B(x,y)$, z is connectable to $B(x,y)$. Hence, there is always an output when we applying Bipartition Extension on $S|_{C(x,y)}$. We now prove the output is $S|_{x,y}$ by contradiction. Suppose the output is not $S|_{x,y}$, and z is the first taxon that goes to the wrong side. That is, at the moment when adding z, the split $\{\tilde{S}^x, \tilde{S}^y\}$ we are enlarging still satisfies $\tilde{S}^x \subset S^x$ and $\tilde{S}^y \subset S^y$, but, without loss of generality, $z \in S^x$ is being added to $\tilde{S}^y$. The reason that we add z to $\tilde{S}^y$ needs to be that there exists $z' \in \tilde{S}^y$ such that $\hat{d}(z,z') \leq \Delta + 2\Omega + \tau$. Notice that because $z \notin \tilde{S}^x \cup \tilde{S}^y$, z is not in $B(x,y)$. Hence, $z \in S^x - B(x,y)$. This shows that $z \in S^x - B(x,y)$, $z' \in S^y$ satisfy $\hat{d}(z,z') \leq \Delta + 2\Omega + \tau$, and contradicts the result in Lemma 4.28. Hence, the output is always $S|_{x,y}$, and this concludes the proof. □

Corollary 4.30. *For any $S|_{x,y}, S'|_{x,y} \in \mathcal{S}|_{x,y}$, suppose $S|_{C(x,y)}, S'|_{C(x,y)} \in \mathcal{S}|_{C(x,y)}$ are the corresponding splits on $B(x,y)$, then $S|_{C(x,y)} = S'|_{C(x,y)}$ if and only if $S|_{x,y} = S'|_{x,y}$.*

Proof. The if part is right from the definition. For the other direction, by Lemma 4.29, both $S|_{x,y}$ and $S'|_{x,y}$ are the output when we applying Bipartition Extension to $S|_{C(x,y)}$, and the property that Bipartition Extension has unique output ends the proof. □

4.5.3 Mini Reconstruction

The Lemma 4.19 of the split decomposition method tells us that we can find out all the $\hat{d}$-splits $\hat{\mathcal{S}}|_{B(x,y)}$ in $B(x,y)$ with $\hat{d}(x,y) \leq \Delta + \Omega + \tau$. What we need to show is the correctness of the Mini Reconstruction. First, we want to show that for any $z, z' \in B(x,y)$, $d(z,z') < M$, which implies that the distance matrix is reliable in $B(x,y)$.

Proposition 4.31. $\forall z, z' \in B(x,y)$, $d(z,z') \leq 3\Delta + 7\Omega + 10\tau < M$

Proof. Because $z \in B(x,y)$, $\hat{d}(z,x) + \hat{d}(z,y) \leq 3\Delta + 7\Omega + 8\tau$, and so both $\hat{d}(z,x)$ and $\hat{d}(z,x)$ need to be smaller or equal to $3\Delta + 7\Omega + 8\tau < M$. Hence, because $\hat{d}$ is a (τ, M)-distortion of d, $d(z,x) \leq \hat{d}(z,x) + \tau$ and $d(z,y) \leq \hat{d}(z,y) + \tau$, and so

$$d(z,x) + d(z,y) \leq \hat{d}(z,x) + \hat{d}(z,y) + 2\tau \leq 3\Delta + 7\Omega + 10\tau$$

Similarly, $d(z',x) + d(z',y) \leq 3\Delta + 7\Omega + 10\tau$. Therefore, because d satisfies triangle inequality [31],

$$\begin{aligned} d(z,z') &\leq \frac{1}{2}[d(z,x) + d(z,y) + d(z',x) + d(z',y)] \\ &= \frac{1}{2} \cdot 2 \cdot (3\Delta + 7\Omega + 10\tau) = 3\Delta + 7\Omega + 10\tau \end{aligned}$$

□

Next, we want to show $\mathcal{S}_{4\tau}|_{C(x,y)} \subset \hat{\mathcal{S}}|_{C(x,y)} \subset \mathcal{S}|_{C(x,y)}$ and that the isolation index that we obtained from the split decomposition method provides a very good estimation on the weight function. The next lemma, which is shown as Theorem 3 in [3], is essential for our upcoming proposition.

Lemma 4.32. *Let $\mathcal{S}'$ be any collection of weakly compatible splits of $\mathcal{X}'$. For each $S' \in \mathcal{S}'$, choose some $w(S') > 0$ and consider*

$$d := \sum_{S' \in \mathcal{S}'} w(S')\delta_{S'}$$

Then $\mathcal{S}'$ is the set of all d-splits. Moreover, the isolation index $\alpha_d(S')$ equals $w(S')$ for each $S' \in \mathcal{S}'$.

We will use this lemma to show the following proposition.

Proposition 4.33. *For any $x, y \in \mathcal{X}$ satisfies $\hat{d}(x,y) \leq \Delta+\Omega+\tau$, $\mathcal{S}_{4\tau}|_{C(x,y)} \subset \hat{\mathcal{S}}|_{C(x,y)} \subset \mathcal{S}|_{C(x,y)}$.*

Proof. Because $\mathcal{S}$ is weakly compatible,

$$\begin{aligned} \mathcal{S}|_{B(x,y)} = \{S|_{B(x,y)} &= \{S^1 \cap B(x,y), S^2 \cap B(x,y)\} \\ &: S = \{S^1, S^2\} \in \mathcal{S}, S^1 \cap B(x,y) \neq \emptyset, S^2 \cap B(x,y) \neq \emptyset\} \end{aligned}$$

is also weakly compatible since the intersection will be smaller when we restrict taxa on the set on $B(x,y)$. Moreover, if we assign

$$w(S|_{B(x,y)}) = \sum_{S=\{S^1,S^2\} \in \mathcal{S}: \{S^1 \cap B(x,y), S^2 \cap B(x,y)\} = S|_{B(x,y)}} w(S)$$

then for any $z, z' \in B(x,y)$,

$$d(z,z') = \sum_{S|_{B(x,y)} \in \mathcal{S}|_{B(x,y)}} w(S|_{B(x,y)})\delta_{S|_{B(x,y)}}(z,z')$$

By previous lemma, with $\mathcal{X}' = B(x,y)$ and $\mathcal{S}' = \mathcal{S}|_{B(x,y)}$, we know that $\mathcal{S}|_{B(x,y)}$ is the set of all d-splits in $B(x,y)$, with the property that the isolation index $\alpha_d(S|_{B(x,y)}) = w(S|_{B(x,y)})$ for any $S|_{B(x,y)} \in \mathcal{S}|_{B(x,y)}$. In particular, for any $S|_{C(x,y)} \in \mathcal{S}|_{C(x,y)} \subset \mathcal{S}|_{B(x,y)}$,

by Lemma 4.30, there is only one split $S|_{x,y}$ corresponding to $S|_{C(x,y)}$, and therefore $\alpha_d(S|_{C(x,y)}) = w(S|_{x,y})$.

Because $\hat{d}$ is a (τ, M)-distortion of d and by Lemma 4.31, for every pair of taxa in $B(x,y)$, the difference of $\hat{d}$ and d is smaller than τ. Therefore, for any $x_1, x_2, y_1, y_2 \in B(x,y)$, it is routine to show

$$|(d(x_1,y_1) + d(x_2,y_2)) - (\hat{d}(x_1,y_1) + \hat{d}(x_2,y_2))| < 2\tau$$

and

$$\begin{aligned}&|\max\{d(x_1,y_1)+d(x_2,y_2), d(x_1,x_2)+d(y_1,y_2), d(x_1,y_2)+d(y_1,x_2)\}\\ &\quad - \max\{\hat{d}(x_1,y_1)+\hat{d}(x_2,y_2), \hat{d}(x_1,x_2)+\hat{d}(y_1,y_2), \hat{d}(x_1,y_2)+\hat{d}(y_1,x_2)\}| < 2\tau\end{aligned}$$

So by the definition of $\tilde{\alpha}_d$,

$$|\tilde{\alpha}_d(\{\{x_1,y_1\},\{x_2,y_2\}\}) - \tilde{\alpha}_{\hat{d}}(\{\{x_1,y_1\},\{x_2,y_2\}\})| < 2\tau$$

As the result, for any bipartition S' of $B(x,y)$,

$$|\alpha_d(S') - \alpha_{\hat{d}}(S')| < 2\tau$$

Hence, for any bipartition S' on $B(x,y)$ which separates x, y, $\alpha_{\hat{d}}(S') > 2\tau$ only if $\alpha_d(S') > 0$, or S' is a d-split on $B(x,y)$. This shows that $\hat{\mathcal{S}}|_{C(x,y)} \subset \mathcal{S}|_{C(x,y)}$. On the other hand, for any split $S_{4\tau}|_{C(x,y)} \in \mathcal{S}_{4\tau}|_{C(x,y)}$ on $B(x,y)$ with corresponding split $S_{4\tau}|_{x,y} \in \mathcal{S}_{4\tau}|_{x,y}$ on $\mathcal{X}$, because $\alpha_d(S_{4\tau}|_{C(x,y)}) = w(S_{4\tau}|_{x,y}) > 4\tau$, we have $\alpha_{\hat{d}}(S_{4\tau}|_{C(x,y)}) > 2\tau$. This shows that $\mathcal{S}_{4\tau}|_{C(x,y)} \subset \hat{\mathcal{S}}|_{C(x,y)}$. Combining both statements concludes the claim. □

Combining both Lemma 4.33 and Lemma 4.29, we know that $\mathcal{S}_{4\tau}|_{x,y} \subset \hat{\mathcal{S}}|_{x,y} \subset \mathcal{S}|_{x,y}$. The last proposition we need to show is the isolation index provides a good estimation.

Proposition 4.34. *For any $\hat{S}|_{x,y} \in \hat{\mathcal{S}}|_{x,y} \subset \mathcal{S}|_{x,y}$, if $\hat{S}|_{C(x,y)}$ is the corresponding split on $B(x,y)$, then we have $|w(\hat{S}|_{x,y}) - \alpha_{\hat{d}}(\hat{S}|_{C(x,y)})| < 2\tau$.*

Proof. As we shown in the previous proof, for any $\hat{S}|_{C(x,y)} \in \hat{\mathcal{S}}|_{C(x,y)} \subset \mathcal{S}|_{C(x,y)}$, we have $\alpha_d(\hat{S}|_{C(x,y)}) = w(\hat{S}|_{x,y})$, and $|\alpha_d(\hat{S}|_{C(x,y)}) - \alpha_{\hat{d}}(\hat{S}|_{C(x,y)})| < 2\tau$, and the statement follows. □

4.6 Improvement in Time Complexity with Circular Network

As said in the introduction, we use the split decomposition method in the Mini Reconstruction step so that the algorithm can deal with any weakly compatible network. However, if the original network is a circular network, we can use Neighbor-Net, which is much faster in finding circular ordering, in our Mini Reconstruction step. In detail, we modify our MiniReconstruction method in Algorithm 4.20 in the following way:

MiniReconstruction:

Input: $\mathcal{X}$, x, y, τ, EllipseRadius, $\hat{d}$

Output: A set of splits $\hat{\mathcal{S}}|_{C(x,y)}$ on $B(x,y)$, and a weight function: $\hat{w}|_{C(x,y)} : \hat{\mathcal{S}}|_{C(x,y)} \to (0,\infty)$

1. $B(x,y) = \{z \in \mathcal{X} : \hat{d}(z,x) + \hat{d}(z,y) \leq EllipseRadius\}$
2. Apply Neighbor-Net to find the circular ordering in $B(x,y)$. Denote $\hat{\mathcal{S}}|_{C(x,y)}$ to be all possible splits that separate x, y in this ordering. Estimate the weight for every $\hat{S}|_{C(x,y)} \in \hat{\mathcal{S}}|_{C(x,y)}$ and assign the value to $\hat{w}|_{C(x,y)}(\hat{S}|_{C(x,y)})$.

3. Return $\hat{\mathcal{S}}|_{C(x,y)}$, $\hat{w}|_{C(x,y)}$

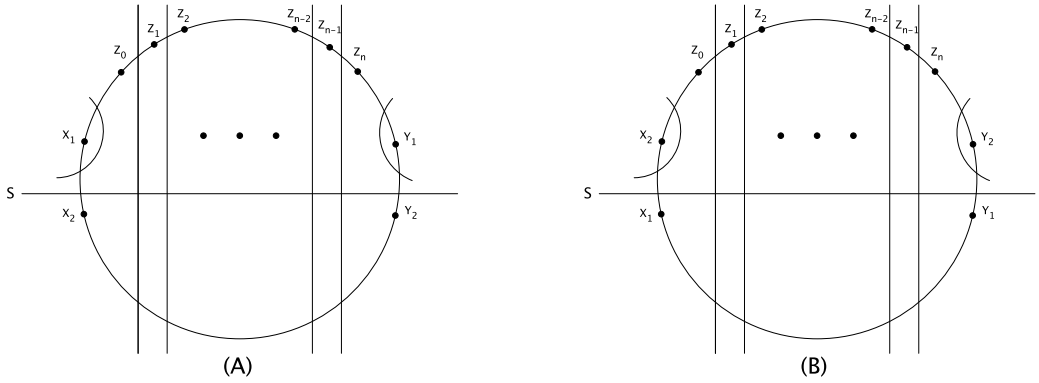
Neighbor-Net takes only $O(n^3)$ to recover the circular ordering, but it is slow when estimating the weight in general: as indicated in [7], it applies the ordinary least squares (OSL) or even the non-negative-constraints least squares (NNLS) to estimate the weights, which takes at least $O(n^6)$ running time in worst case. Nevertheless, as we have shown in Lemma 4.31, the distance between any pairs of taxa in $B(x, y)$ is reliable, and so we can estimate the weight using equation (4) in [34]. This takes only at most $O(n^2)$ time. So the overall time complexity is:

$$O(n^2) \cdot (O(n^3) + O(n^2) \cdot O(n^2)) = O(n^6)$$

This is faster than Lemma 4.20. Moreover, the final splits set $\hat{\mathcal{S}}$ and the estimator $\hat{w}$ still satisfy the statements in our main theorem(Theorem 4.18): The consistency of Neighbor-Net with distortion has been shown in [9, 34], and the executability and correctness of Bipartition Extension is shown in Section 4.5, with the fact that circular networks are one type of weakly compatible networks.

4.7 Lower Bound

In this section, we want to show if we want to reconstruct $\mathcal{S}$ from a (τ, M)-distorted metric for any circular network $\mathcal{N} = (\mathcal{X}, \mathcal{S}, w, d)$, M needs to be larger than a certain linear function of $\Delta(\mathcal{N})$ and $\Omega(\mathcal{N})$. [12] provides an example showing that if we want to reconstruct $\mathcal{S}$ from a (τ, M)-distorted metric for any phylogenetic tree, M needs to be at least about $2\Delta(\mathcal{N})$. We claim here is that M needs to larger than about $\Omega(\mathcal{N})$ to ensure that the reconstruction can succeed for any circular network $\mathcal{N}$.



(a) Figure A

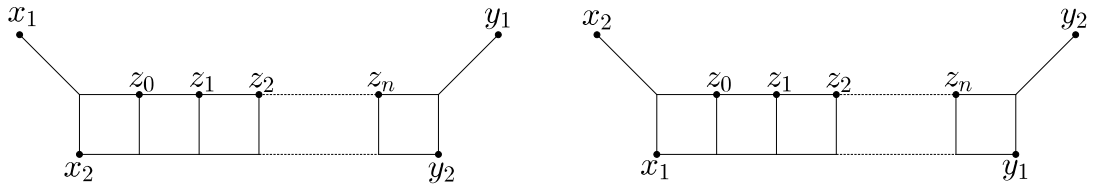
(b) Figure B

Figure 12: Examples

Consider these two circular networks in Figure 12 (which will shown as Figure 13 if using *SplitsTree* [29]). In these two circular networks, $\mathcal{X} = \{x_1, x_2, y_1, y_2\} \cup \{\cup_i z_i\}$, and the n vertical lines, the horizontal line, and the the little arcs around x_i and y_i are the splits. Suppose all the splits have weight 1, then the depth of both graphs is equal to 1, and the incompatible weight of both graphs is equal to n. The split sets of these two circular networks are different: the split $S = \{\{x_2, y_2\}, \mathcal{X} - \{x_2, y_2\}\}$ is in graph (A) but can not be found in graph (B). Nevertheless, in both circular networks:

- $d(z_i, x_j) = i + 1$, $0 \leq i \leq n$, $1 \leq j \leq 2$
- $d(z_i, y_j) = n - i + 1$, $0 \leq i \leq n$, $1 \leq j \leq 2$
- $d(x_1, x_2) = d(y_1, y_2) = 2$
- $d(x_1, y_2) = d(x_2, y_1) = n + 2$

The only difference is in graph (A), $d(x_1, y_1) = n + 2$ and $d(x_2, y_2) = n$, while in graph



(a) Figure A

(b) Figure B

Figure 13: Examples

(B), $d(x_2, y_2) = n+2$ and $d(x_1, y_1) = n$. If we choose the distance matrix $\hat{d}$ by assigning

- $\hat{d}(x_1, y_1) = \hat{d}(x_2, y_2) = n + 1$
- $\hat{d} = d$ for all other pairs

then $\hat{d}$ is a $(\tau, n-1)$-distorted metric for both graphs, for any $1 > \tau > 0$. Hence, these two circular networks are indistinguishable from $\hat{d}$. Because the depth of the circular network is fixed to 1, but n could be arbitrary large, we know M needs to consist of a linear factor of $\Omega(\mathcal{N})$ to make any possible algorithm works.

4.8 Simulation

In the following simulation, we consider one type of circular network. We start from a circular network that demonstrates a "linear" tree, then randomly add a few extra splits. In detail, suppose $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ is the set of taxa, then we initially assign $\mathcal{S}$ as:

$$\mathcal{S} = \{\{\{x_i\}, \mathcal{X} - \{x_i\}\} : 1 \leq i \leq n\} \cup \{\{\{x_1, \ldots, x_k\}, \{x_{k+1}, \ldots, x_n\}\} : 2 \leq k \leq n-1\}$$

After that, we add a few more splits, $\{\{x_i, \ldots, x_j\}, \mathcal{X} - \{x_i, \ldots, x_j\}\}$. $1 \leq i < j \leq n-1$ into $\mathcal{S}$ (the number of additional splits is randomly chosen from 0 to $\mathcal{X}/4$). Last, for every splits $S \in \mathcal{S}$, we let$w(S) = 0.01$. This type of network usually has relatively small depth and small incompatible weight compared to its diameter. The following figure, Figure 14 (which will shown as Figure 15 if using *SplitsTree* [29]), is one example of the circular network with 6 taxa.

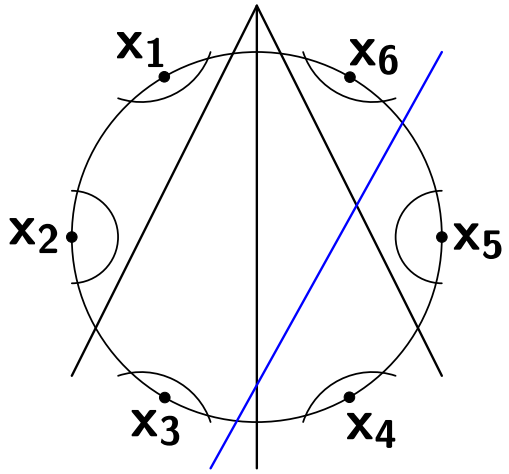


Figure 14: One circular network example

To simulate the states of the taxa, we use Abelian group-based model introduced in [8] with 4 states, namely $\{0, 1, 2, 3\}$. Given a split network $\mathcal{N} = (\mathcal{X}, \mathcal{S}, w)$, to simulate the states of all taxa on a specific site, we first independently generate a random variable ξ_S associates to each split $S \in \mathcal{S}$, with the following distribution:

$$\mathbb{P}[\xi_S = 0] = \frac{1}{4}(1 + 3e^{-\frac{4}{3}w(e)}), \mathbb{P}[\xi_S = 1] = \mathbb{P}[\xi_S = 2] = \mathbb{P}[\xi_S = 3] = \frac{1}{4}(1 - e^{-\frac{4}{3}w(e)})$$

Next, fixed any taxa x, and assign any possible state to x with equal probability $\frac{1}{4}$. Then

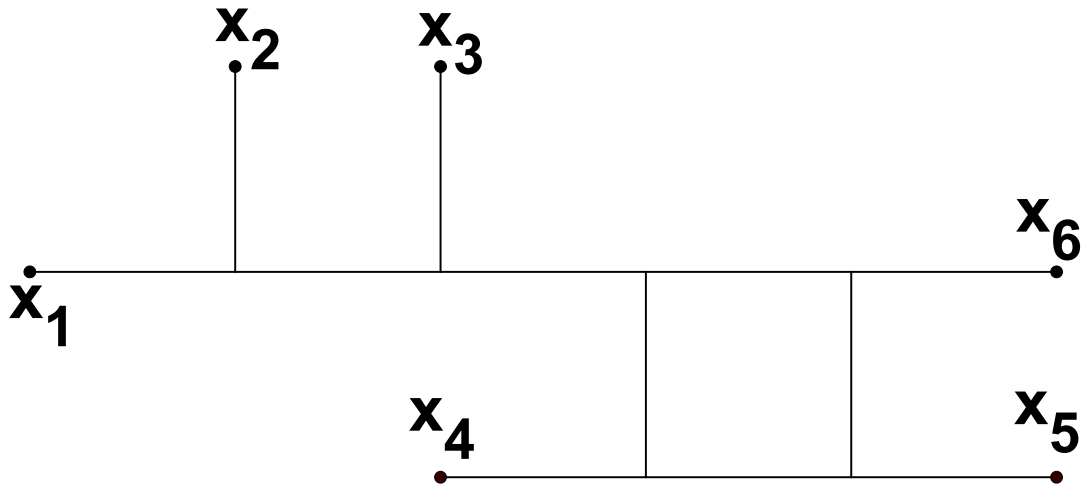


Figure 15: One circular network example

for any other taxa y, the state of y is:

$$state(y) = (state(x) + \sum_{\delta_S(x,y)=1} \xi_S) \mod 4$$

In the simulation, we first generate a split network, then we produce 10000 sites using the model above, with $w(S) = 0.01$ for all splits. Then we compute the Jukes-Cantor distance for every pair of taxa as the estimator $\hat{d}$. Next, we apply all three algorithms on the estimators $\hat{d}$, and respectively compute the Robinson-Foulds Distance between the original split set and the output split set. We run the process 100 times and the result is shown in the following Figure 16. As the figure shows, our algorithm has better result in this type of circular network. The project is implemented in Java, and the running time for 100 runs is, for example, few seconds for 52 taxa.

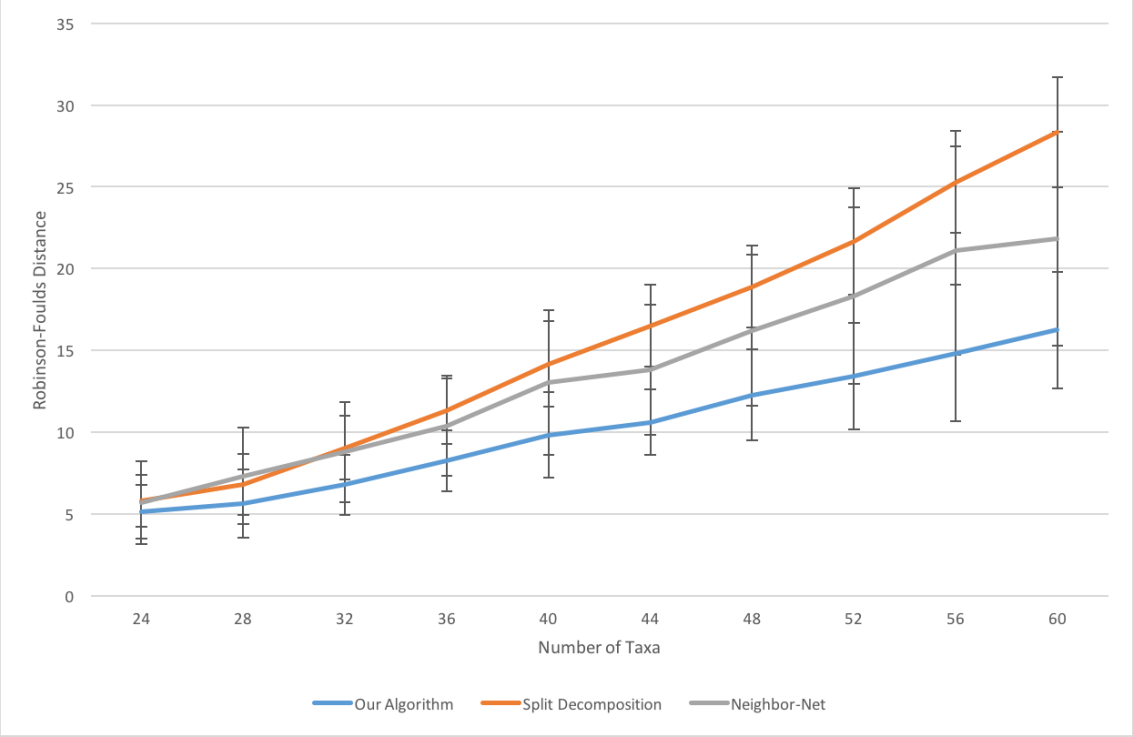


Figure 16: The accuracy of the algorithms

CHAPTER 5

DISCUSSION AND FUTURE WORK

My thesis consists of three projects that discuss quite different fields. Each one provides some concrete results, and in the mean time, each one shows part of the blueprint of future work in its main topic.

In Chapter 2, we bring the idea of branching number into the discussion over the accuracy of the Fitch algorithm, and give a stronger result than what we previously knew. Nevertheless, the result is restricted to the symmetric two-state Jukes-Cantor model, which might not be that attractive for those biologists who focus on DNA sequences. One possible future project is to extend the results to a four-state evolution model, to see if the branching number still helps in estimating the accuracy of the Fitch-Hartigan algorithm.

Next, Chapter 3 illustrates how we compute the accuracy of each category of multi-locus methods once we have a full understanding in one gene, under both ILS and recombination. As we say in the chapter, the consistency of R^*/STAR/MDC is being

left as a conjecture. Another possible future work is to uncover the explicit formula of the decay rate to support the simulation. These incomplete parts are good future project topics.

Last, in Chapter 4, we introduced a new algorithm to reconstruct a circular network, or more generally a weakly compatible network. The algorithm does work well, compared to Neighbor-Net and the split decomposition method, in some specific types of phylogenetic networks. However, the improvement is not that obvious when it comes to a general networks with few taxa. One possible future project is to compare these three algorithms in a large, general random tree, and determine under what conditions our algorithm is better than the other two.

BIBLIOGRAPHY

[1] *Distance-based phylogeny reconstruction (fast-converging), 2003; king, zhang, zhou*, SpringerReference.

[2] S. C. Ayling and T. A. Brown, *Novel methodology for construction and pruning of quasi-median networks*, BMC Bioinformatics, 9 (2008), p. 115.

[3] H.-J. Bandelt and A. W. Dress, *A canonical decomposition theory for metrics on a finite set*, Advances in Mathematics, 92 (1992), pp. 47–105.

[4] H. J. Bandelt, P. Forster, and A. Rohl, *Median-joining networks for inferring intraspecific phylogenies*, Molecular Biology and Evolution, 16 (1999), pp. 37–48.

[5] D. Bryant, *A classification of consensus methods for phylogenetics*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, (2003).

[6] ——, *Extending tree models to splits networks*, Algebraic Statistics for Computational Biology, (2004), pp. 322–334.

[7] D. Bryant, *Neighbor-net: An agglomerative method for the construction of phylogenetic networks*, Molecular Biology and Evolution, 21 (2004), pp. 255–265.

[8] D. Bryant, *Extending tree models to split networks*, Algebraic Statistics for Computational Biology (L Pachter and B Sturmfels, editors), Cambridge University Press, (2005), pp. 297–310.

[9] D. Bryant, V. Moulton, and A. Spillner, *Consistency of the neighbor-net algorithm*, Algorithms for Molecular Biology, 2 (2007), p. 8.

[10] P. Buneman, *The recovery of trees from measures of dissimilarity*, Mathematics the the Archeological and Historical Sciences, (1971), pp. 387–395.

[11] C. W. Cunningham, K. E. Omland, and T. H. Oakley, *Reconstructing ancestral character states: a critical reappraisal*, Trends in Ecology & Evolution, 13 (1998), pp. 361–366.

[12] C. Daskalakis, E. Mossel, and S. Roch, *Phylogenies without branch bounds: Contracting the short, pruning the deep*, Lecture Notes in Computer Science Research in Computational Molecular Biology, (2009), pp. 451–465.

[13] J. H. Degnan, M. Degiorgio, D. Bryant, and N. A. Rosenberg, *Properties of consensus methods for inferring species trees from gene trees*, Systematic Biology, 58 (2009), pp. 35–54.

[14] J. H. Degnan and N. A. Rosenberg, *Gene tree discordance, phylogenetic inference and the multispecies coalescent*, Trends in Ecology and Evolution, 24 (2009), pp. 332–340.

[15] C. F. Delwiche and J. D. Palmer, *Rampant horizontal transfer and duplication of rubisco genes in eubacteria and plastids*, Molecular Biology and Evolution, 13 (1996), pp. 873–882.

[16] W. F. Doolittle, *Phylogenetic classification and the universal tree*, Science, 284 (1999), pp. 2124–2128.

[17] R. Durrett, *Probability models for DNA sequence evolution*, Springer, second ed., 2008.

[18] ——, *Probability: theory and examples*, Cambridge University Press, fourth ed., 2010.

[19] S. V. Edwards, L. Liu, and D. K. Pearl, *High-resolution species trees without concatenation*, Proceedings of the National Academy of Sciences, 104 (2007), pp. 5936–5941.

[20] P. L. Erds, M. A. Steel, L. A. Szkely, and T. J. Warnow, *A few logs suffice to build (almost) all trees (i)*, Random Structures and Algorithms, 14 (1999), pp. 153–184.

[21] J. Felsenstein, *Cases in which parsimony or compatibility methods will be positively misleading*, Systematic Zoology, 27 (1978), p. 401.

[22] W. M. Fitch, *Distinguishing homologous from analogous proteins*, Systematic Zoology, 19 (1970), p. 99.

[23] ——, *Toward defining the course of evolution: Minimum change for a specific tree topology*, Systematic Zoology, 20 (1971), p. 406.

[24] O. Gascuel and M. Steel, *Inferring ancestral sequences in taxon-rich phylogenies*, Mathematical Biosciences, 227 (2010), pp. 125–135.

[25] R. Griffiths and P. Marjoram, *Ancestral inference from samples of dna sequences with recombination*, Journal of Computational Biology, 3 (1996), pp. 479–502.

[26] R. C. Griffiths and P. Marjoram, *An ancestral recombination graph*, Progress in Population Genetics and Human Evolution The IMA Volumes in Mathematics and its Applications, (1997), pp. 257–270.

[27] B. Holland and V. Moulton, *Consensus networks: A method for visualising incompatibilities in collections of trees*, Lecture Notes in Computer Science Algorithms in Bioinformatics, (2003), pp. 165–176.

[28] R. R. Hudson, *Properties of a neutral allele model with intragenic recombination*, Theoretical Population Biology, 23 (1983), pp. 183–201.

[29] D. H. Huson, *Application of phylogenetic networks in evolutionary studies*, Molecular Biology and Evolution, 23 (2005), pp. 254–267.

[30] D. H. Huson, T. Dezulian, T. Klpper, and M. A. Steel, *Phylogenetic super-networks from partial trees*, Lecture Notes in Computer Science Algorithms in Bioinformatics, (2004), pp. 388–399.

[31] D. H. Huson, R. Rupp, and C. Scornavacca, *Phylogenetic networks: concepts, algorithms and applications*, Cambridge University Press, 2010.

[32] M. Kimura, *A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences*, Journal of Molecular Evolution, 16 (1980), pp. 111–120.

[33] H. C. Lanier and L. L. Knowles, *Is recombination a problem for species-tree analyses?*, Systematic Biology, 61 (2012), pp. 691–701.

[34] D. Levy and L. Pachter, *The neighbor-net algorithm*, Advances in Applied Mathematics, 47 (2011), pp. 240–258.

[35] G. Li, M. Steel, and L. Zhang, *More taxa are not necessarily better for the reconstruction of ancestral character states*, Systematic Biology, 57 (2008), pp. 647–653.

[36] L. Liu and D. K. Pearl, *Species trees from gene trees: Reconstructing bayesian posterior distributions of a species phylogeny using estimated gene tree distributions*, Systematic Biology, 56 (2007), pp. 504–514.

[37] L. Liu, L. Yu, and D. K. Pearl, *Maximum tree: a consistent estimator of the species tree*, Journal of Mathematical Biology, 60 (2009), pp. 95–106.

[38] L. Liu, L. Yu, D. K. Pearl, and S. V. Edwards, *Estimating species phylogenies using coalescence times among sequences*, Systematic Biology, 58 (2009), pp. 468–477.

[39] W. P. Maddison, *Calculating the probability distributions of ancestral states reconstructed by parsimony on phylogenetic trees*, Systematic Biology, 44 (1995), p. 474.

[40] ——, *Gene trees in species trees*, Systematic Biology, 46 (1997), p. 523.

[41] P. Marjoram and J. D. Wall, *Fast "coalescent" simulation*, BMC Genetics, 7 (2006).

[42] G. A. T. Mcvean and N. J. Cardin, *Approximating the coalescent with recombination*, Philosophical Transactions of the Royal Society B: Biological Sciences, 360 (2005), pp. 1387–1393.

[43] R. Mihaescu, D. Levy, and L. Pachter, *Why neighbor-joining works*, Algorithmica, 54 (2007), pp. 1–24.

[44] S. Mirarab, M. S. Bayzid, and T. Warnow, *Evaluating summary methods for multilocus species tree estimation in the presence of incomplete lineage sorting*, Systematic Biology, 65 (2014), pp. 366–380.

[45] E. Mossel, *Distorted metrics on trees and phylogenetic forests*, IEEE/ACM Transactions on Computational Biology and Bioinformatics, 4 (2007), p. 108116.

[46] E. Mossel and S. Roch, *Incomplete lineage sorting: Consistent phylogeny estimation from multiple loci*, IEEE/ACM Transactions on Computational Biology and Bioinformatics, 7 (2010), pp. 166–171.

[47] M. Page, *The maximum likelihood approach to reconstructing ancestral character states of discrete characters on phylogenies*, Systematic Biology, 48 (1999), pp. 612–622.

[48] M. Pagel, A. Meade, and D. Barker, *Bayesian estimation of ancestral character states on phylogenies*, Systematic Biology, 53 (2004), pp. 673–684.

[49] P. Pamilo and M. Nei, *Relationships between gene trees and species trees.*, Molecular Biology and Evolution, (1988).

[50] L. Pauling, E. Zuckerkandl, T. Henriksen, and R. Lvstad, *Chemical paleogenetics. molecular "restoration studies" of extinct forms of life.*, Acta Chemica Scandinavica, 17 supl. (1963), pp. 9–16.

[51] Y. Peres, *Probability on trees: An introductory climb*, Lecture Notes in Mathematics Lectures on Probability Theory and Statistics, (1999), pp. 193–280.

[52] L. H. Rieseberg, *Hybrid origins of plant species*, Annual Review of Ecology and Systematics, 28 (1997), pp. 359–389.

[53] S. Roch, *An analytical comparison of multilocus methods under the multispecies coalescent: The three-taxon case*, Biocomputing 2013, (2012).

[54] N. Saitou and M. Nei, *The neighbor-joining method: a new method for reconstructing phylogenetic trees.*, Molecular Biology and Evolution, (1987), pp. 406–425.

[55] B. A. Salisbury and J. Kim, *Ancestral state estimation and taxon sampling density*, Systematic Biology, 50 (2001), pp. 557–564.

[56] D. Schluter, T. Price, A. O. Mooers, and D. Ludwig, *Likelihood of ancestor states in adaptive radiation*, Evolution, 51 (1997), p. 1699.

[57] P. H. A. Sneath, *Cladistic representation of reticulate evolution*, Systematic Zoology, 24 (1975), p. 360.

[58] M. A. Steel, *Distributions on bicoloured evolutionary trees*, Bulletin of the Australian Mathematical Society, 41 (1990), p. 159.

[59] M. Syvanen, *Cross-species gene transfer; implications for a new theory of evolution*, Journal of Theoretical Biology, 112 (1985), pp. 333–343.

[60] S. Tavar, *Some probabilistic and statistical problems in the analysis of dna sequences*, Lectures on Mathematics in the Life Sciences, 17 (1986), pp. 57–86.

[61] C. Than and L. Nakhleh, *Species tree inference by minimizing deep coalescences*, PLoS Computational Biology, 5 (2009).

[62] J. Whitfield, S. Cameron, D. Huson, and M. Steel, *Filtered z-closure supernetworks for extracting and visualizing recurrent signal from incongruent gene trees*, Systematic Biology, 57 (2008), pp. 939–947.

[63] G. U. Yule, *A mathematical theory of evolution, based on the conclusions of dr. j. c. willis, f.r.s.*, Philosophical Transactions of the Royal Society B: Biological Sciences, 213 (1925), pp. 21–87.

[64] J. Zhang and M. Nei, *Accuracies of ancestral amino acid sequences inferred by the parsimony, likelihood, and distance methods*, Journal of Molecular Evolution, 44 (1997).

[65] L. Zhang, J. Shen, J. Yang, and G. Li, *Analyzing the fitch method for reconstructing ancestral states on ultrametric phylogenetic trees*, Bulletin of Mathematical Biology, 72 (2010), pp. 1760–1782.