Inferring phylogenetic trees under the general Markov model via a minimum spanning tree backbone

A dissertation

submitted towards the degree

Doctor of Natural Sciences (Dr. rer. nat.)

of the Faculty of Mathematics and Computer Science

of Saarland University

by

Prabhav Kalaghatgi

Saarbrücken, 2020

Day of the colloquium: July 2, 2020

Dean of the faculty: Prof. Dr. Thomas Schuster

Chair of the committee: Prof. Dr. Kurt Mehlhorn

Reporters:

First reviewer: Prof. Dr. Dr. Thomas Lengauer

Second reviewer: Prof. Dr. Tobias Marschall

Third reviewer: Prof. Dr. Arndt von Haeseler

Academic assistant: Dr. Christina Backes

Acknowledgments

No work is done in vacuum. I count myself lucky to have started my research career in bioinformatics at the MPI for informatics. In retrospect, I could not have asked for a better start than working with Glenn Lawyer. There was much that I learnt from Glenn's work on network-based modeling. I enjoyed sharing an office with Mathieu Flinders. The best advice that I received on how to conduct research was when Thomas Lengauer told me not to straight-jacket my thoughts. I am grateful for the advice given by the more experienced researchers that I worked with, specifically, Nico Pfeifer and Olga Kalinina. I have always gravitated towards Thomas, and consequently I defined myself as an independent researcher. That said, it was exciting to work in collaboration with Rolf Kaiser's group. It was enjoyable to work with Elena Knops. Our scientific retreats to the Austrian alps were challenging at first, but eventually I learnt my way on the slopes. It was gratifying to be able to make quick updates to the geno2pheno system which would not have been possible without prior work and helpful suggestions from Bastian Beggel, Alejandro Pironti, Matthias Döring, Joachim Büch and Georg Friedrich.

Thank you Alaknanda for some of the best moments of my life. Your indomitable spirit has been an inspiration through difficult times. In many ways I have grown older in Saarbrücken. Thank you Mittul and Juhi for your friendship that I could always count on. Thank you Pranav for being supportive through hard times. On a similar note, I thank my parents Ramesh Kalaghatgi and Suhasini Kalaghatgi. On a more special note, I thank my mother for all the times when she called to check up on me, although I wish she didn't worry as much. I am one of those children who wanted to prove their worth to their father. That said, I was inspired by evolution from a fairly young age, and an early exposure to nature, thanks to a father who worked in the forest department didn't hurt one bit.

Thank you Sreyo and Sneha for all the fun times that we've had. Thank you Fatemeh, Valentina, Mikko and Dilip for being challenging badminton partners. Thank you Esha for being a source of comfort. Thank you Shaleen for inspiring me through your work. Thank you Fatemeh for being a good listener. Thank you Anna for your counsel and for being a good friend. Thank you Peter for our conversations about science. Thank you Tomas for worrying about society at large and for doing something about it. Thank you Siva for your helpful support.

I hope to cherish our shared memories until my memories are overwritten or the birth-death process consumes me.

”I have deeply regretted that I did not proceed far enough at least to understand something of the great leading principles of mathematics, for men [minds] thus endowed seem to have an extra sense.” — Charles Darwin (1887)

Abstract

Phylogenetic trees are models of the evolutionary relationships among species, with species typically placed at the leaves of trees. We address the following problems regarding the calculation of phylogenetic trees. (1) Leaf-labeled phylogenetic trees may not be appropriate models of evolutionary relationships among rapidly evolving pathogens which may contain ancestor-descendant pairs. (2) The models of gene evolution that are widely used unrealistically assume that the base composition of DNA sequences does not evolve. Regarding problem (1) we present a method for inferring generally labeled phylogenetic trees that allow sampled species to be placed at non-leaf nodes of the tree. Regarding problem (2), we present a structural expectation maximization method (SEM-GM) for inferring leaf-labeled phylogenetic trees under the general Markov model (GM) which is the most complex model of DNA substitution that allows the evolution of base composition. In order to improve the scalability of SEM-GM we present a minimum spanning tree (MST) framework called MST-backbone. MST-backbone scales linearly with the number of leaves. However, the unrealistic location of the root as inferred on empirical data suggests that the GM model may be overtrained. MST-backbone was inspired by the topological relationship between MSTs and phylogenetic trees that was introduced by Choi et al. (2011). We discovered that the topological relationship does not necessarily hold if there is no unique MST. We propose so-called vertex-order based MSTs (VMSTs) that guarantee a topological relationship with phylogenetic trees.

Kurzfassung

Phylogenetische Bäume modellieren evolutionäre Beziehungen zwischen Spezies, wobei die Spezies typischerweise an den Blättern der Bäume sitzen. Wir befassen uns mit den folgenden Problemen bei der Berechnung von phylogenetischen Bäumen. (1) Blattmarkierte phylogenetische Bäume sind möglicherweise keine geeigneten Modelle der evolutionären Beziehungen zwischen sich schnell entwickelnden Krankheitserregern, die Vorfahren-Nachfahren-Paare enthalten können. (2) Die weit verbreiteten Modelle der Genevolution gehen unrealistischerweise davon aus, dass sich die Basenzusammensetzung von DNA-Sequenzen nicht ändert. Bezüglich Problem (1) stellen wir eine Methode zur Ableitung von allgemein markierten phylogenetischen Bäumen vor, die es erlaubt, Spezies, für die Proben vorliegen, an inneren des Baumes zu platzieren. Bezüglich Problem (2) stellen wir eine strukturelle Expectation-Maximization-Methode (SEM-GM) zur Ableitung von blattmarkierten phylogenetischen Bäumen unter dem allgemeinen Markov-Modell (GM) vor, das das komplexeste Modell von DNA-Substitution ist und das die Evolution von Basenzusammensetzung erlaubt. Um die Skalierbarkeit von SEM-GM zu verbessern, stellen wir ein Minimale Spannbaum (MST)-Methode vor, die als MST-Backbone bezeichnet wird. MST-Backbone skaliert linear mit der Anzahl der Blätter. Die Tatsache, dass die Lage der Wurzel aus empirischen Daten nicht immer realistisch abgeleitet warden kann, legt jedoch nahe, dass das GM-Modell möglicherweise übertrainiert ist. MST-backbone wurde von einer topologischen Beziehung zwischen minimalen Spannbäumen und phylogenetischen Bäumen inspiriert, die von Choi et al. 2011 eingeführt wurde. Wir entdeckten, dass die topologische Beziehung nicht unbedingt Bestand hat, wenn es keinen eindeutigen minimalen Spannbaum gibt. Wir schlagen so genannte vertex-order-based MSTs (VMSTs) vor, die eine topologische Beziehung zu phylogenetischen Bäumen garantieren.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter provides an introduction to phylogenetic trees, explains the limitations of current methods that are used to infer phylogenetic trees, and highlights the steps taken in this thesis towards inferring phylogenetic trees under more realistic models of evolution than those that are commonly used. The phylogenetic terminology that is introduced in this chapter is explained in detail in Chapter 2.

1.1 What are phylogenetic trees?

The word species is Latin for kind or type. A *species* is canonically defined as a group of *organisms* (individual life forms) that are capable of mating with each other, and giving birth to fertile offspring (de Queiroz, 2005). Darwin (1859) hypothesized that living species have descended from a common origin. The only illustration in "The origin of species" depicts a birth-death process that started from ancestral species that have gone extinct, and proceeded to give rise to living species (see Figure 1.1). How do new species come to exist?

The information that is necessary for the reproduction of organisms is present in the form of deoxyribonucleic acid (DNA) molecules known as genomes. The Dobzhansky-Muller model of speciation states that if the members of a species split and form mutually exclusive reproducing populations, then, over generations of isolated reproduction, each population will independently accumulate changes in their genomes, and members from separated populations will not be able to successfully reproduce, thus forming distinct species (Johnson, 2008). Phylogenetic trees are models of how species are related to each other. The process of speciation enables a hierarchical classification of species. Each level of the hierarchical classification is a taxonomic rank, with species being the lowest taxonomic rank. The term taxa is used instead of species if the phylogenetic tree under consideration has a higher taxonomic rank at the leaves instead of species.

All organisms are cellular, and are capable of reproducing on their own via the use of molecules that are synthesized within their cells (Alberts et al., 2002). Viruses are parasites that cannot replicate on their own; instead, viruses replicate using the molecules that exist within the cells of the organisms that they infect. Viruses evolve rapidly and don't easily fit the species definition. The term taxa is used in this thesis to describe organisms and viruses that are related via common descent.

The functions of organisms are carried out by ribonucleic acid (RNA) molecules and amino-acid molecules known as proteins. A characteristic feature of RNA molecules and proteins is that they are synthesized as linear polymers, and are subsequently modified in order to form functional molecules. *Genes* are DNA sequences that contain the information that specifies the order of RNA monomers and amino-acid monomers in RNA molecules and proteins, respectively, Epp (1997). Genes that are transcribed into RNA, and subsequently translated in protein(s) are called protein-coding genes.

The nucleic acids DNA and RNA are polymers of nucleotides. Each nucleotide is comprised of a sugar (deoxyribose for DNA, and ribose for RNA) that is attached to a phosphate group, and a nucleobase/base

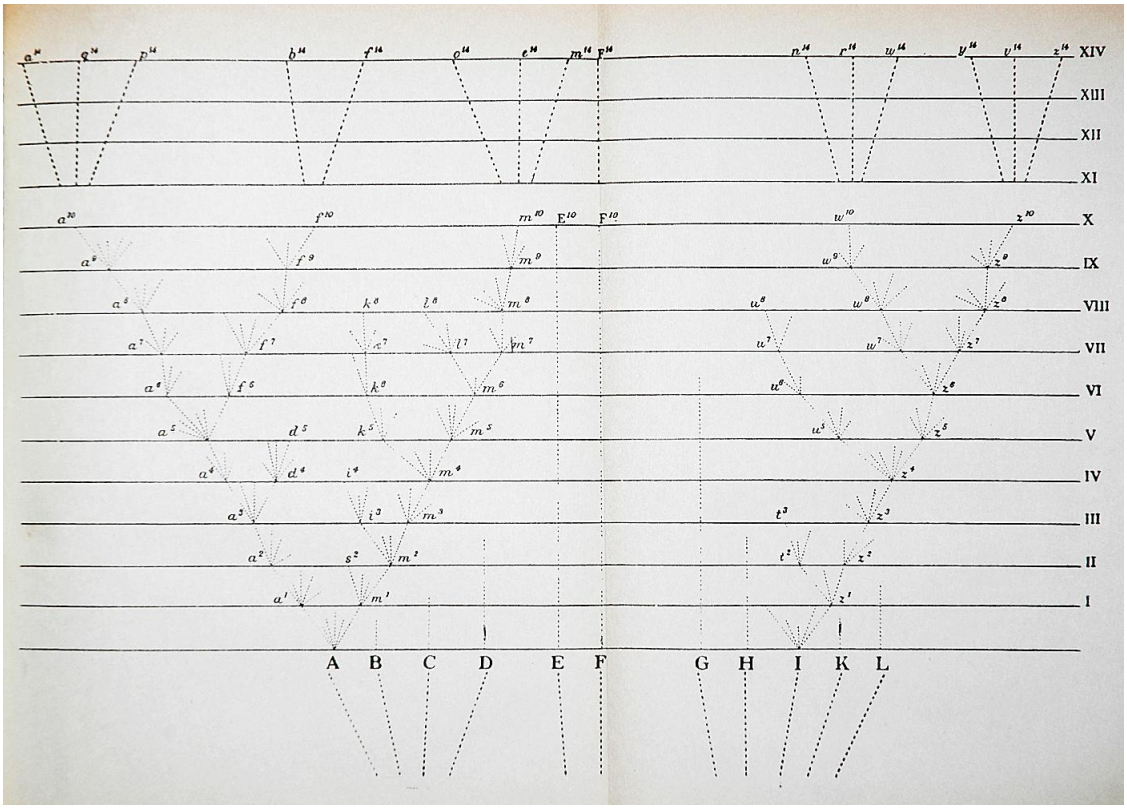


Figure 1.1: The only illustration in "The origin of species" by Darwin (1859). Time's arrow is directed from bottom to top. Species A through L are ancestral species that are hypothesized to have given birth to living species. The dashed lines indicate genetic lineages.

(see Figure 1.2). The bases that constitute DNA are Adenine (A), Thymine (T), Guanine (G) and Cytosine (C). RNA is comprised of the bases Uracil (U), Adenine (A), Guanine (G) and Cytosine (C). Adenine and Guanine are purines, and Thymine, Uracil and Cytosine are pyrimidines. DNA sequences in genomes form double-stranded helical structures such that a pyrimidine on one strand pairs with a purine on the complementary strand (see Figure 1.2) . The base pairs A:T and G:C are referred to as Watson-Crick pairs.

New genes are created by (*i*) gene duplication, (*ii*) mutations that transform non-genic genomic regions to genes, (*iii*) gene fusion, and (*iv*) horizontal gene transfer (Andersson et al., 2015).

1.1.1 Gene trees and species trees

A set of genes is said to be homologous if the genes have evolved from a common ancestral gene (see Figure 1.3 for a set of homologous HIV-1 *pol* gene sequences). A *gene tree* is a tree-structured representation of the evolutionary history of homologous genes. Two evolutionarily related genes are said to have *diverged* from a common ancestral gene if the two genes have accumulated distinct mutations when compared to the common ancestral gene. The branches of a gene tree are scaled in units of DNA substitutions per site.

Homologous genes are either orthologs or paralogs (Koonin, 2005). Orthologous genes are genes from different species that have diverged from a common ancestral gene. Paralogous genes are genes that have evolved from a common ancestral gene via gene duplication that is subsequently followed by divergence.

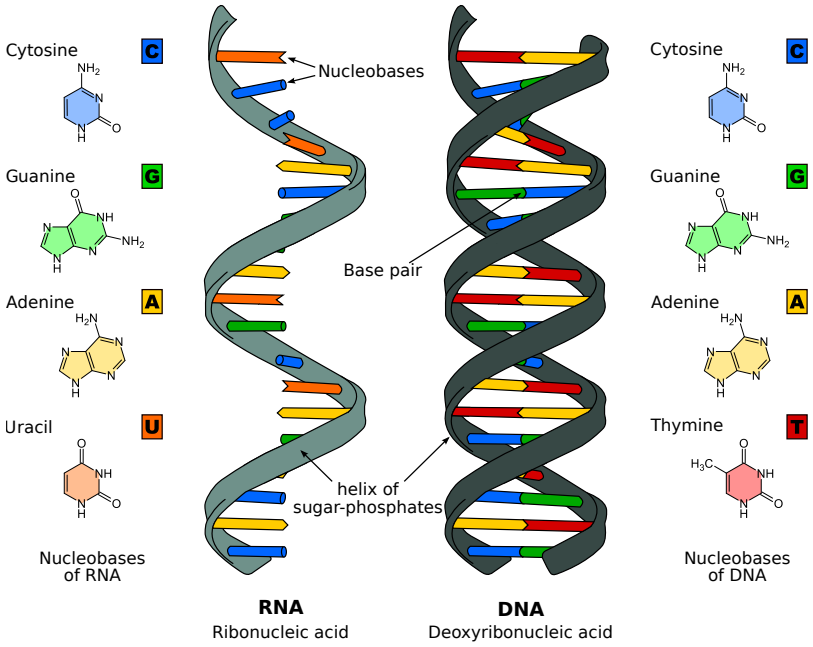


Figure 1.2: The structure and composition of ribonucleic acids (RNA) and deoxyribonucleic acids (DNA) (Figure adapted from Wikimedia (2017)).

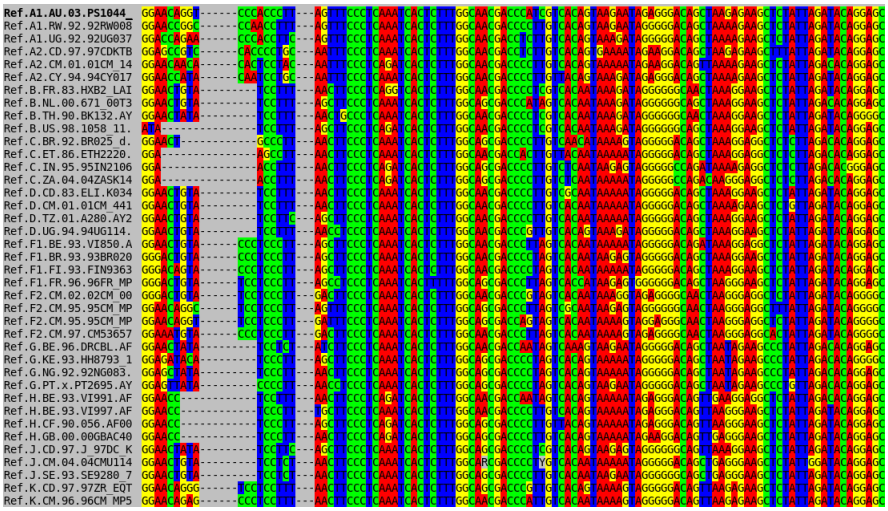


Figure 1.3: A set of orthologous HIV-1 *pol* gene sequences. The sequences shown above were downloaded from the HIV sequence database that is hosted at the Los Alamos National Laboratory (HIVLANL). SEAVIEW (Galtier et al., 1996) was used to visualize the sequence alignment.

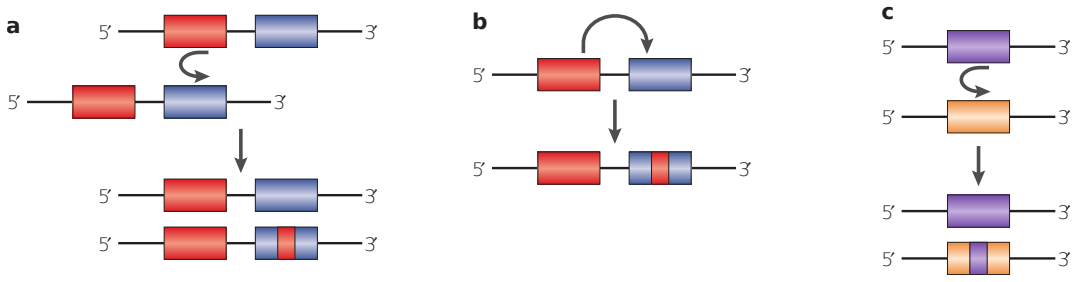


Figure 1.4: Types of gene conversion: (a) conversion between paralogs located on different chromosomes (b) conversion between paralogs located on the same chromosome, (c) conversion between alleles. The curved arrows are directed from donor sequence to the acceptor sequence. The acceptor sequence contains a double-strand break that is repaired using the donor sequence. The straight arrows are oriented in the direction of the gene conversion process. The illustration shown above has been adapted with permission from Chen et al. (2007).

A *species tree* is a tree-structured representation of the evolutionary history of a set of related species. Gene trees that are constructed using orthologous genes can help infer the evolutionary relationships of the species from whose genomes the orthologous genes were sampled.

There are limitations to tree-like models of species relationships. Hybridization and horizontal gene transfer cannot be represented using species trees. Additionally, gene evolution via recombination cannot be modeled using gene trees.

1.2 Evolution of GC content

DNA molecules are replicated during cellular reproduction. DNA replication is not an error-free process. *Mutations* are changes in the DNA sequence of daughter DNA molecules when compared with the corresponding DNA sequence in the parental DNA molecule. Mutations that result in the change of a single nucleotide are known as point mutations. DNA substitutions are point mutations that change one nucleotide with another nucleotide.

Methylated Cytosine that spontaneously deaminates to Thymine results in a base pair mismatch (G:C converts to G:T). Subsequently DNA replication at the site containing G:T in the parental strand would create one of two distinct base pairs G:C and A:T, respectively, in the daughter strands, consequently reducing the GC content in one daughter strand. Methylated Cytosine is found in the CpG dinucleotides of genomes.

Transitions are DNA substitutions where a purine is replaced by a purine (*e.g.*, A is replaced by G), or a pyrimidine is replaced by a pyrimidine (*e.g.*, C is replaced by T). Each transition event results in a change in GC content. Transversions are DNA substitutions where a purine is replaced by a pyrimidine (*e.g.*, A is replaced by T or C), or vice-versa. Transversions do not necessarily change GC content. There are four possible transitions and eight possible transversions. If each nucleotide substitution was equally likely then the ratio ti/tv of transitions (ti) to transversions (tv) would be around 0.5. Empirical findings suggest that ti/tv is around two and four for *Drosophila melanogaster* (Begun et al., 2007), and *Homo sapiens* (Hodgkinson and Eyre-Walker, 2010), respectively. GC content may change because transitions are more frequent than transversions.

Double-strand breaks (DSB) that occur either as part of *meiosis* (a type of cell division that is used to produce the gametes: sperm cells and egg cells) or due to replication errors such as stalled DNA replication, can result in cell death if left unrepaired. Gene conversion is one of the end products of repairing DSB using homologous recombination. Gene conversion usually occurs by replacing the sequence in the gene

that contains the DSB (acceptor sequence) with the sequence from an intact gene (donor sequence) that is homologous to the acceptor sequence (Chen et al., 2007). Gene conversion can occur between paralogs or between alleles, *i.e.*, variants of a gene that are found on the same genetic locus (see Figure 1.4). The repair of the acceptor sequence involves the DNA mismatch repair machinery. Gene conversion is said to be biased if DNA mismatches are repaired in a manner that is biased towards one purine-pyrimidine pair over the other. GC-biased gene conversion would increase GC content, whereas AT-biased gene conversion reduces GC content. GC-biased gene conversion that occurs during meiosis is thought to have contributed to the non-uniform distribution of GC content along chromosomes (Duret and Galtier, 2009).

1.3 Current approaches for inferring of phylogenetic trees

The commonly used model of evolutionary relationships is a tree with observed species placed at the leaves and unobserved ancestors placed at branching points. The widely adopted approach to inferring gene trees involves modeling gene evolution using probabilistic models (Felsenstein, 2003). The probabilistic modeling approach can be formulated as a combinatorial optimization problem that involves selecting a combination of phylogenetic tree and model parameters that maximizes the likelihood score. Phylogeny inference via maximum-likelihood is *NP-hard* (Chickering, 1996; Roch, 2006; Chor and Tuller, 2006), and the corresponding decision problem is *NP-complete*.

Leaf-labeled trees may not be appropriate for modeling the relationships among rapidly evolving pathogens such as viruses that are sampled over similar time-scales as their evolution. Choi et al. (2011) model evolutionary relationships using trees that allow internal nodes to be labeled, and describe a minimum spanning tree method for constructing generally labeled trees using a clustering algorithm known as Chow-Liu grouping. Minimum spanning trees (MSTs) can be computed quickly using fast algorithms (Kruskal, 1956). As Kalaghatgi et al. (2016b) implemented Chow-Liu grouping, they discovered that Choi et al. (2011)'s proof of correctness that was based on additive distances was incorrect. Kalaghatgi et al. (2016b) modified a distance-based clustering method known as neighbor-joining that is popular in the field of phylogeny inference in order to construct generally labeled trees in a manner that is guaranteed to be correct that distances are additive. The method introduced in Kalaghatgi et al. (2016b) is called family-joining and is described in Chapter 3. Kalaghatgi and Lengauer (2017) corrected the proof by Choi et al. (2011) and performed a detailed analysis of the amount of phylogenetic information that is contained in minimum spanning trees (see Chapter 4)

Current approaches for inferring phylogenetic trees search through the set of possible phylogenetic trees in order to find a tree that maximizes the likelihood score. The large computational cost of optimizing the likelihood score has led to the wide-spread adoption of time-reversible models of gene evolution (Kozlov et al., 2019; Nguyen et al., 2015; Hohna et al., 2016). It is not possible to identify the location of the root of a phylogenetic tree under a time-reversible model of evolution (Felsenstein, 1981).Jermiin et al. (2004) used sequences simulated under a non-stationary model in order to claim that phylogenetic trees inferred under time-reversible models are systematically biased. The evolutionary history of genes is not known. Consequently, systematic error in phylogenies inferred using empirical data is determined by measuring the similarity of distinct gene trees from the same set of species (Naser-Khdour et al., 2019). Systematic error is inferred if gene sequences tend to be closer to each other on the basis of base composition and not species relationships. Sheffield et al. (2009) claim to have found evidence for systematic bias in the phylogenetic trees of beetle mitochondria that were inferred using time-reversible models of sequence evolution. Additionally, Sheffield et al. (2009) claim to have overcome systematic bias using methods that perform phylogeny inference under non-stationary models of sequence evolution. Current methods that perform phylogeny inference under non-stationary models of gene evolution are not scalable, and have not been widely applied (Betancur-R et al., 2013).

All of the phylogeny inference software that is commonly used makes use of phylogenetic trees with branch lengths that are scaled in units of substitutions per site. The parameter known as branch length is utilized to construct time-calibrated phylogenetic trees which are phylogenetic trees with branch lengths scaled in units of calendar time.

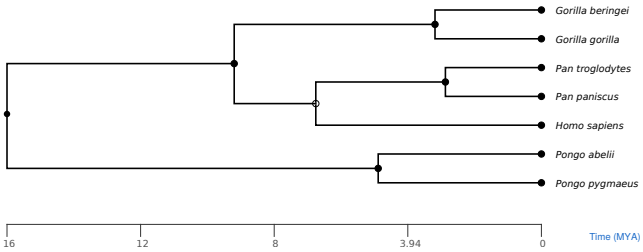


Figure 1.5: Humans (*Homo sapiens*) and chimpanzees (*Pan trogdolytes*) diverged from a common ancestor (represented as an unfilled circle) around 7 million years ago (MYA). The time-calibrated phylogenetic tree that is shown above was created using TimeTree (Hedges et al., 2006).

1.4 Time-calibrated phylogenetic trees

The branches of a phylogenetic tree are usually scaled in units of molecular substitutions per site. The rate at which molecular substitutions take place can be used to scale the branches of a phylogenetic tree in units of calendar time resulting in the construction of time-calibrated phylogenetic trees. The time-calibrated phylogenetic tree shown in Figure 1.5 dates the divergence of humans (Homo sapiens) and chimpanzees (Pan troglodytes) from their most recent common ancestor to a time point around 7 million years ago (Hedges et al., 2006). Molecular clocks are widely used to construct time-calibrated phylogenetic trees, and are described below.

The molecular clock hypothesis assumes that the molecular substitution rate (nucleotide substitution rate or amino-acid substitution rate) is constant (Zukerkandl and Pauling, 1965). Observations of constant substitution rates of the amino-acid sequences of *hemoglobin* and *cytochrome c* among closely related species provided empirical evidence for the molecular clock assumption (Zukerkandl and Pauling, 1962; Margoliash, 1963).

Each branching point in a phylogenetic tree corresponds to a time point when an ancestral species diverged into multiple descendant species. Estimates of one or more divergence times are used to calibrate a molecular clock. Subsequently the calibrated molecular clock is used to scale all branches of the phylogenetic tree in units of time. Sampling times of rapidly evolving pathogens such as HIV provide an alternate source of data for calibrating molecular clocks. Models that are used for calibrating molecular clocks are discussed in Subsection 2.8.3.

1.5 Overview of contributions made in this thesis

The general Markov model (GM; Barry and Hartigan (1987)) is the most complex non-stationary model of DNA substitutions. In contrast to commonly used models that are parameterized in terms of branch lengths, the GM model is parameterized in terms of transition matrices (also known as conditional probability distributions). Currently, there is no scalable method for inferring phylogenetic trees under the GM model. The main contribution of this thesis is to show that MSTs can be used to constrain the search for phylogenetic trees, thereby allowing the use of more complex models of gene evolution than the models that are widely used. We used the minimum spanning tree framework to perform phylogeny inference under the GM model.

The total contributions made in this thesis are: (*i*) a method that models ancestor-descendant relationship among serially sampled pathogens by placing species at ancestor nodes of phylogenetic trees (Kalaghatgi et al., 2016a), (*ii*) a rigorous analysis of the relationship between phylogenetic trees and minimum spanning

trees (Kalaghatgi and Lengauer, 2017), and (*iii*) a computationally efficient framework for inferring phylogenetic trees under the general Markov model (unpublished).

The following parts of the thesis are structured as described below. Chapter 2 gives a technical overview of current approaches for inferring phylogenetic trees. Chapters 3, 4, and 5 provide detailed description of each contribution made in this thesis.

Chapter 2

Background

A brief introduction to some of the graph-theoretic terminology that is used in this thesis is provided in Section 2.1. Probabilistic models that are used for inferring phylogenetic trees are introduced in Section 2.4. Approaches to optimize model parameters are discussed in Section 2.5 and Section 2.6. Methods for placing the root on unrooted phylogenetic trees are discussed in Section 2.8. The chapter concludes with a summary of the contributions that have been made in this thesis.

2.1 Graph-theoretic terminology

The graph-theoretic notions that are presented here have been adapted from "Data Structures and Network Algorithms" by Tarjan (1992), and "Phylogeny: discrete and random processes in evolution" by Steel (2016).

Graphs are models of pairwise relationships among objects. Objects are represented by *nodes* or *vertices*. Pairwise relationships between vertices are referred to as *edges*. Given a set of edges E between vertices in the set V, a graph G is an ordered pair (V, E).

A graph G is either *undirected* in which case each edge is an unordered pair of distinct vertices, or G is *directed* in which case each edge is an ordered pair of distinct vertices. In order to avoid repeating definitions for directed graphs and undirected graphs the notation $[u, v]$ is used to either represent an undirected edge $\{u, v\}$ or a directed edge (u, v), using the context to resolve the ambiguity. If $[u, v]$ is any edge then u and v are its *ends*; $[u, v]$ is said to be incident to u and v, and u and v are said to be incident to $[u, v]$. If $\{u, v\}$ is an undirected edge then u and v are *adjacent*. A directed edge (u, v) *exits* u and *enters* v. An edge $[u, u]$ is a *self-loop*.

If v is a vertex in an undirected graph then the *degree* of v is the number of vertices that are adjacent to v. If v is a vertex in a directed graph then the *in-degree* of v is the number of directed edges that enter v, and the *out-degree* of v is the number of directed edges that exit v. A vertex v in an undirected graph is a *leaf* if the degree of v is one. A vertex v in a directed graph is a leaf if the in-degree of v is one, and the out-degree of v is zero. Any vertex that is not a leaf is an *internal* vertex. A *terminal* edge is an edge that is incident to a leaf. An *internal* edge is any edge that is not a terminal edge.

The *undirected version* of a directed graph can be obtained by replacing each edge (u, v) with the edge $\{u, v\}$. Conversely, the *directed version* of an undirected graph can be obtained by replacing each edge $\{u, v\}$ with the edges (u, v) and (v, u).

An *edge-weighted* graph is a graph $G = (V, E)$ such that each edge in E is assigned a real number called the *weight* of the edge. The edge weights of an edge-weighted graph $G = (V, E)$ are denoted by $\mathbf{w} = \{w_e : e \in E\}$. The terms edge length and edge weight are used interchangeably in this thesis. A spanning tree of a graph G is a connected subgraph of G with no cycles. A *minimum spanning tree* (MST) of an edge-weighted undirected graph G is a spanning tree of G with the minimum sum of edge weights.

Contraction of an edge $\{u, v\}$ in an undirected graph $G = (V, E)$ comprises the following operations: (i) adding a new vertex w to V, (ii) adding edges $\{w, n\}$ to E for each n that is adjacent either to u or to v,

(*iii*) removing u and v from V, and (*iv*) removing each edge from E that is incident either to u or to v. Contraction of an edge (u, v) in an directed graph $G = (V, E)$ comprises the following operations: (*i*) adding a new vertex w to V, (*ii*) adding edges (w, n) to E for each n such that there is an edge in E that enters n and exits either u or v, (*iii*) adding edges (n, w) to E for each n such that there is an edge in E that is exits n and enters either u or v, (*iv*) removing u and v from V, and (v) removing each edge from E that is incident either to u or to v.

Vertices a and b are said to be *neighbors* if there is an edge that is incident to a and b. Given two non-leaf vertices v and w such that a and b are neighbors of v and w, respectively. v and w are said to *swap* their neighbors if a neighbor of w is a neighbor of v, and vice-versa, subsequent to the swap operation.

A *path* in a graph *from* vertex v_1 *to* vertex v_k is an ordered set of vertices $(v_1, v_2, \ldots, v_k)$ such that $[v_i, v_{i+1}]$ is an edge for $i \in [1, \ldots, k-1]$. The path *contains* vertex v_i for $i \in [1, \ldots, k]$ and edge $[v_i, v_{i+1}]$ for $i \in [1, \ldots, k-1]$. Vertices $v_1$and v_k are the ends of the path. An edge $[u, v]$ is contained in a path if there is an index $i \in [1, \ldots, k-1]$ such that $[v_i, v_{i+1}]$ equals $[u, v]$. A path is *simple* if the vertices contained in the path are distinct. A path in a directed graph is a *cycle* if k is greater than one, and v_k equals v_1, and the edges in the path are distinct. A path in an undirected graph is a cycle if k is greater than one, v_k equals v_1, and the edges in the path are distinct. A graph with no cycles is *acyclic*. If there is a path from vertex v to vertex w then w is *reachable* from v. An undirected path in a directed graph is a path in the undirected version of the graph. The weighted path length $p_T^w(v_1, v_k)$ is the sum of edge weights of the edges that are contained in the path. The unweighted path length $p_T^u(v_1, v_k)$ is the number of edges that are contained in the path.

An undirected graph is *connected* if every vertex is reachable from every other vertex, and *disconnected* otherwise. An undirected graph is said to be *complete* if each vertex is adjacent to every other vertex. A *tree* is a connected undirected graph with no cycles. A disconnected graph is a *forest* if each component of the graph is a tree. A directed graph is said to be *weakly connected* if the undirected version of the graph is connected. A directed graph is said to be *strongly connected* if every vertex is reachable from every other vertex. The *diameter* of a tree is the largest unweighted path length of all paths in the tree. A rooted tree $T = (V, E)$ is a directed graph such that the undirected version of T is a tree, and all the edges in E are directed away from a single vertex known as the root. If v and w are distinct vertices in a rooted tree such that v is contained in the path from the root to w then v is an *ancestor* of w, and w is a *descendant* of v. If (v, w) is an edge in a rooted tree then v is the *parent* of w, and w is a *child* of v. The *least common ancestor (lca)* of any pair of distinct vertices u and v is the vertex $\text{lca}_T(u, v)$ that is a common ancestor of u and v such that no descendant of $\text{lca}_T(u, v)$ is a common ancestor of u and v.

A *tree traversal* is the process of visiting each of the vertices in a rooted tree exactly once. A *preorder* tree traversal visits parents before children. The *postorder* tree traversal visits children before parents.

A graph $G_s = (V_s, E_s)$ is said to be a *subgraph* of a graph $G = (V, E)$ if $V_s \subseteq V$, and $E_s \subseteq E$. A *subtree* $\tau_v = (V_{\tau_v}, E_{\tau_v})$ of a rooted tree $T = (V, E)$ is any weakly connected subgraph of T such that the descendants in T of each non-leaf vertex in V_{τ_v} are contained in V_{τ_v}. A subtree $\tau_v = (V_{\tau_v}, E_{\tau_v})$ is said to be rooted at vertex v in V_{τ_v} if each other vertex in V_{τ_v} is a descendant of v. A subtree $\tau_v = (V_{\tau_v}, E_{\tau_v})$ of an undirected tree $T_u = (V, E)$ is a connected subgraph of T such that there is exactly one edge $\{u, v\}$ in E_T such that v is in V_{τ_v} and u is in $V_T \backslash V_{\tau_v}$. The subtree τ_v of the undirected tree T is said to be rooted at v. The edges of any subtree are directed away from the root of the subtree.

2.1.1 Phylogenetic trees

A *rooted phylogenetic tree* $T = (V, E)$ is a rooted tree with two types of vertices in $V = \{\mathcal{H}, \mathcal{L}\}$: hidden vertices $\mathcal{H}$ representing unknown ancestral gene sequences, and labeled vertices $\mathcal{L}$ representing observed gene sequences. An *unrooted phylogenetic tree* is tree with hidden vertices and labeled vertices. Phylogenetic trees are assumed to be rooted unless specified otherwise.

A *leaf-labeled phylogenetic tree* is a phylogenetic tree such that each labeled vertex is a leaf (see Figure 2.1A). A *generally labeled phylogenetic tree* is a phylogenetic tree such that all leaves are labeled but not all labeled vertices are leaves (see Figure 2.1B). A generally labeled phylogenetic tree with no hidden vertices is

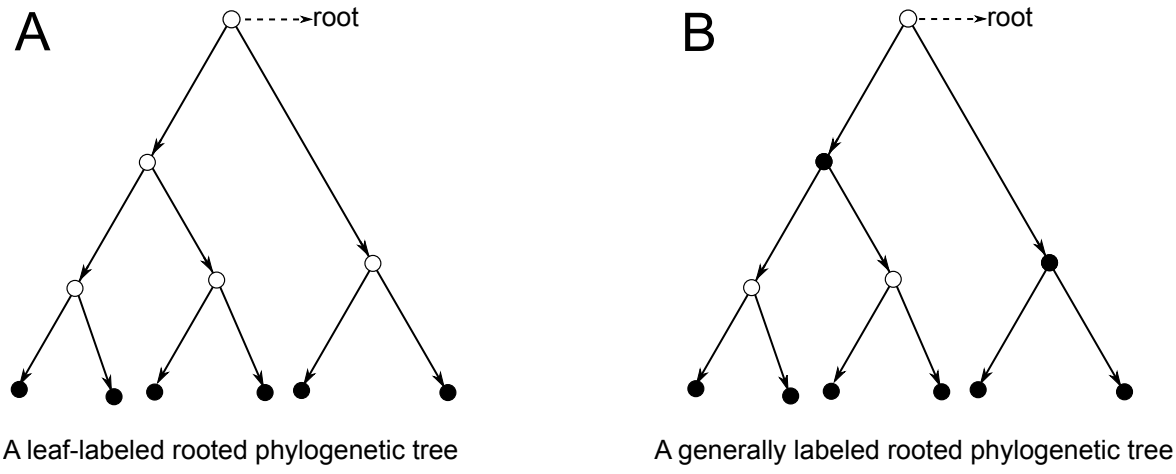


Figure 2.1: Types of phylogenetic trees. A leaf-labeled phylogenetic tree is shown in panel A. A generally labeled phylogenetic tree is shown in panel B. Labeled vertices and unlabeled vertices are represented by filled circles and unfilled circles, respectively.

a *fully labeled phylogenetic tree*. The phylogenetic trees that are inferred by majority of current software are unrooted leaf-labeled phylogenetic trees. Phylogenetic trees are assumed to be leaf-labeled unless specified otherwise. *Ultrametric trees* are rooted phylogenetic trees such that each leaf is equidistant from the root.

Given an unrooted phylogenetic tree $T = (V_T, E_T)$ and any edge $e = \{v, w\}$ in E_T, consider the subtrees τ_v and τ_w of T that are rooted at v and w, respectively. Let $\mathcal{L}_{\tau_v}$ be the set of labeled vertices in τ_v, and let $\mathcal{L}_{\tau_w}$ be the set of labeled vertices in τ_w. $\mathcal{L}_{\tau_v}|\mathcal{L}_{\tau_w}$ denotes a *split* in T_u that is induced by the edge $e = \{v, w\}$. $\mathcal{L}_{\tau_v}$ and $\mathcal{L}_{\tau_w}$ are the *sides* of the split $\mathcal{L}_{\tau_v}|\mathcal{L}_{\tau_w}$. A split is said to be a *trivial split* if the cardinality of one side of the split equals one. Given a rooted tree T_ρ, a group of species is said to be *monophyletic* if the species are the leaves of a subtree in T_ρ.

Given an unrooted phylogenetic tree $T = (V_T, E_T)$. The distance between a vertex set $V_s \subset V_T$ and a vertex $v \in V_T \backslash V_s$ is defined as the unweighted path length of the shortest path in T from v to a vertex in V_s. Given a split $\mathcal{L}_{\tau_v}|\mathcal{L}_{\tau_w}$ that is induced by an edge $\{v, w\}$ the side $\mathcal{L}_{\tau_v}$ is said to closer to v in comparison to w. Conversely $\mathcal{L}_{\tau_w}$ is said to be closer to w in comparison to v.

Given any non-leaf vertex u of a rooted tree, let $u.l$ and $u.r$ be the children of u. Consider the subtrees $\tau_{u.l}$ and $\tau_{u.r}$ that are rooted at $u.l$ and $u.r$, respectively. Without loss of generality (wlog), the subtrees $\tau_{u.l}$ and $\tau_{u.r}$ are said to be the left subtree and the right subtree that *subtend* from vertex u. The imbalance of a rooted tree, as quantified using Colless's index (I_C see equation 2.1; Colless (1982)), is a measure of how differently sized the left subtree and the right subtree that subtend from each non-leaf vertex are, where the the size of a subtree is the number of leaves that are contained in the subtree.

$$I_C = \sum_{u \in \mathcal{H}} (|\mathcal{L}_{\tau_{u.l}}| - |\mathcal{L}_{\tau_{u.r}}|) \tag{2.1}$$

Two special cases of phylogenetic trees are described below. A rooted caterpillar is a rooted phylogenetic tree such that all hidden vertices are contained in a single path (see Figure 2.2 A). A balanced tree is a rooted phylogenetic tree for which the path from each leaf to the root contains the same number of edges (see Figure 2.2 B).

The following restrictions are placed on the degrees of vertices in phylogenetic trees. Non-leaf vertices in rooted phylogenetic trees are restricted to have an out-degree that is at least one. Non-leaf vertices in unrooted phylogenetic trees are restricted to have a degree that is at least three. A rooted phylogenetic

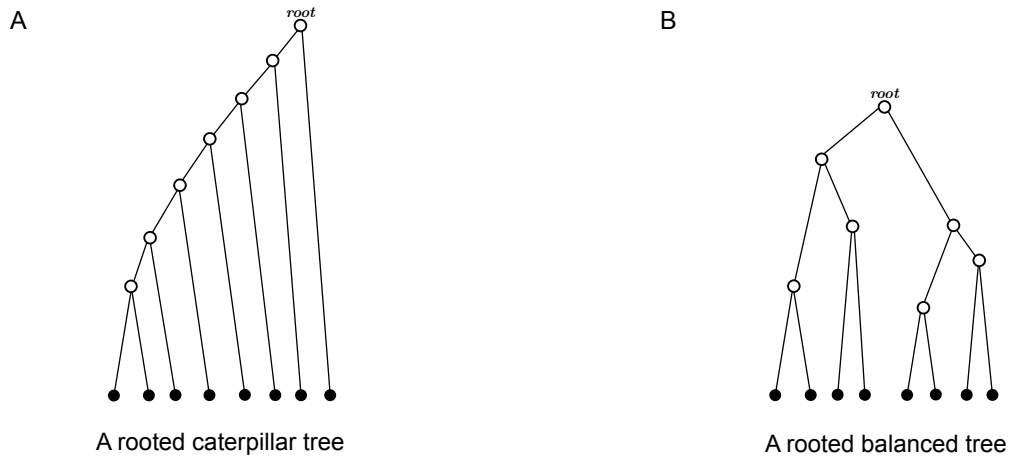


Figure 2.2: Imbalance of rooted phylogenetic trees. The tree shown on panel A is least balanced, and is known as a rooted caterpillar phylogenetic tree. The tree shown on panel B is maximally balanced, and is called a rooted balanced phylogenetic tree.

tree $T_\rho = (V_{T_\rho}, E_{T_\rho})$ is said to be *fully resolved* if each non-leaf vertex in V_{T_ρ} has out-degree two. A fully resolved rooted phylogenetic tree is also referred to as a *bifurcating* tree. An unrooted phylogenetic tree $T = (V_T, E_T)$ is said to be fully resolved if each non-leaf vertex in V_T has degree three. Hidden vertices in rooted phylogenetic trees with out-degree greater than two are *polytomies*. A hidden vertex in an unrooted phylogenetic tree is a polytomy if the degree of the hidden vertex is greater than three. Phylogenetic trees are assumed to be fully resolved unless specified otherwise.

Edge lengths are numerical values that are assigned to the edges of a phylogenetic tree. Edge lengths are usually scaled in units of substitutions per site. The edge lengths of a *time-calibrated phylogenetic tree* are scaled in units of time. The terms edge and branch are used interchangeably in this thesis. Additionally, the terms edge length and branch length are also used interchangeably. The term *phylogeny* as used in this thesis is short for phylogenetic tree.

We define below two graph operations involving the removal of vertices and the insertion of vertices in phylogenetic trees with undirected edges. Given a phylogenetic tree $T = (V_T, E_T)$ with undirected edges let $\mathbf{t}$ denote the edge lengths of edges in E_T. Let vertices u and v be adjacent to a vertex w with degree two. *Suppressing* the vertex w involves (i) removing the edges $\{u, w\}$ and $\{v, w\}$ from E_T, (ii) removing the vertex w from V_T, (iii) adding the edge $\{u, v\}$ to E_T, (iv) removing the edge lengths $t_{\{u,w\}}$ and $t_{\{v,w\}}$ from $\mathbf{t}$, and adding the edge length $t_{\{u,v\}} = t_{\{u,w\}} + t_{\{v,w\}}$ to $\mathbf{t}$. Conversely, *inserting* a vertex w along an edge $\{u, v\}$ in E_T at a non-negative distance δ away from u such that δ is smaller than $t_{\{u,v\}}$ involves (i) adding w to V_T, (ii) removing $\{u, v\}$ from E_T, (iii) adding $\{u, w\}$ and $\{v, w\}$ to E_T, (iv) removing $t_{\{u,v\}}$ from $\mathbf{t}$, and (v) adding $t_{\{u,w\}} = \delta$ and $t_{\{v,w\}} = t_{\{u,v\}} - \delta$ to $\mathbf{t}$.

Given an unrooted phylogenetic tree $T = (V_T, E_T)$ with edge lengths $\mathbf{t} = \{t_e : e \in E_T\}$ let $e = \{a, b\}$ be an edge in E_T and let t_e in $\mathbf{t}$ be the edge length of e. *Rooting* T along the edge $e = \{a, b\}$ at a distance d (such that (s.t.) $0 \leq \delta \leq t_{\{a,b\}}$) from a involves (i) inserting a vertex ρ at distance δ along e away from a, (ii) directing all edges in E_T away from ρ, and (iii) replacing the length of each undirected edge $t_{\{u,v\}}$ in $\mathbf{t}$ with $t_{(u,v)} = t_{\{u,v\}}$ such that E_T contains (u, v). Conversely, given a rooted phylogenetic tree $T = (V_T, E_T)$ with edge lengths $\mathbf{t} = \{t_e : e \in E_T\}$ let ρ in V_T be the root of T. Constructing the unrooted version of T involves (i) replacing each directed edge in E_T with the undirected version of the edge, (ii) replacing the length of each directed edge $t_{(u,v)}$ in $\mathbf{t}$ with $t_{\{u,v\}} = t_{(u,v)}$, and (iii) suppressing the root ρ.

Given a phylogenetic tree $T = (V_T, E_T)$ the edge lengths $\mathbf{t}$ of edges in E_T are denoted by $\mathbf{t} = \{t_e : e \in E_T\}$. An unrooted phylogenetic tree $T = (V_T, E_T)$ with edge lengths is equipped with a distance function $d_T : V_T \times V_T \to \mathbb{R}^+$ over pairs of vertices in V_T. The *tree-distance* $d_T(u, v)$ between vertices u and v in V_T

is the weighted path length $p^w_T(u,v)$. Tree-distances of T are *additive* in T, and are referred to as *additive distances* of T. Tree-distances of a rooted tree $T = (V, E)$ is computed on the basis of the unrooted version of T. The location of the root can not be recovered using tree-distances.

The *topology* of a phylogenetic tree $T = (V_T, E_T)$ is the graph structure comprising the vertex set V_T and the edge set E_T. Edge lengths are not included in the topology of a phylogenetic tree.

2.2 Three ways to score trees: parsimony, likelihood, and tree length

Phylogeny inference is a combinatorial optimization problem. The three scores that are commonly used are parsimony, likelihood and tree length. Parsimony and likelihood are character-based scores and are defined with respect to (wrt) a leaf-labeled phylogenetic tree $T = (V = \{\mathcal{H}, \mathcal{L}\}, E)$ and a multiple sequence alignment $\mathcal{X}_\mathcal{L} = \{\mathcal{X}^i_l : l \in \mathcal{L} \wedge 1 \leq i \leq k\}$, where k is the number of columns in the alignment, and the states represented by V are characters from alphabet of size a. The number of leaves in T is denoted by n. Tree length is a distance-based score, where distances are estimates of tree-distances.

The *maximum parsimony score* is the minimum number of state changes required to generate the states that are observed at the leaves of a phylogenetic tree. Given an assignment to the states in $\mathcal{X}_\mathcal{H}$, the total number of state changes $c_T(\mathcal{X}_\mathcal{H}|\mathcal{X}_\mathcal{L})$ over edges E is computed as

$$c_T(\mathcal{X}_\mathcal{H}|\mathcal{X}_\mathcal{L}) = \sum_{i=1}^{k} \sum_{(u,v)\in E} \delta(\mathcal{X}^i_u, \mathcal{X}^i_v)$$

where $\delta(x, y)$ is the Kroenecker delta function that is defined as

$$\delta(x,y) = \begin{cases} 1 & x = y \\ 0 & x \neq y \end{cases}$$

The maximum parsimony score $c^*_T(\mathcal{X}_\mathcal{L})$ is computed by selecting a character assignment $\mathcal{X}_\mathcal{H}$ that minimizes the total number of character changes, *i.e.*,

$$c^*_T(\mathcal{X}_\mathcal{L}) = \operatorname*{argmin}_{\mathcal{X}_\mathcal{H}} c_T(\mathcal{X}_\mathcal{H}|\mathcal{X}_\mathcal{L}) \tag{2.2}$$

The maximum parsimony score can be computed in time $O(na^2k)$ using Fitch's algorithm (Fitch, 1971). A maximum parsimony estimate is a phylogenetic tree T and character assignment for hidden states $\mathcal{X}_\mathcal{H}$ that minimizes the maximum parsimony score. Given a character assignment that minimizes the number of changes over the edges of a rooted tree, it is possible to change the location of the root without modifying the maximum parsimony score. Consequently, it is not possible to infer the location of the root using the maximum parsimony score (Felsenstein, 2003).

The *likelihood score* is defined on the basis of a probabilistic generative model M. The likelihood score is computed by (i) conditioning the joint probability distribution over $\mathcal{X}_\mathcal{L}$ wrt observed states $\mathcal{X}_\mathcal{L}$, and (ii) marginalizing over the $a^{|\mathcal{H}|}$ possible assignments to the hidden states $\mathcal{X}_\mathcal{H}$. It is often necessary to assume that each column of $\mathcal{X}_\mathcal{L}$ has been generated independently by a common model in order to have a sufficiently large sample for estimating model parameters. The likelihood score $\ell_T(M|\mathcal{X}_\mathcal{L})$ is defined as

$$\ell_T(M|\mathcal{X}_\mathcal{L}) = \prod_{i=1}^{k} \sum_{\mathcal{X}^{i}_{h}:h\in\mathcal{H}} P(\{\mathcal{X}^i_v : v \in V\}|M) \tag{2.3}$$

where $P(\{\mathcal{X}^i_v : v \in V\}|M)$ is the conditional probability distribution over the states in column i of the sequence alignment. The *maximum likelihood score* $\ell^*_T(\mathcal{X}_\mathcal{L})$ of a tree T is defined as

$$\ell^*_T(\mathcal{X}_\mathcal{L}) = \operatorname*{argmax}_M \ell_T(M|\mathcal{X}_\mathcal{L})$$

The likelihood score can be computed in time $O(na^2k)$ using Felsenstein's tree pruning algorithm (Felsenstein, 1981). The maximum likelihood estimate (MLE) of phylogenetic trees is the combination of model M and tree T that maximizes the likelihood score. Likelihood scores can be used to infer the location of the root if the underlying Markov model is not time-reversible, as is explained in detail in Section 2.4.

The *tree length* score is a distance-based score that is defined as the sum of edge lengths, where edge lengths are usually estimated by regressing weighted path lengths on estimates of tree-distances (Desper and Gascuel, 2002). Tree length can be computed in time $O(n)$, and the estimation of edge lengths via ordinary-least-squares regression (OLS) can be performed in time $O(n^2)$ (Bryant, 1997). A minimum tree length estimate is the combination of edge lengths and tree that minimizes tree length. Trees that are inferred based on tree length are unrooted by definition because distances do not generally contain information about the location of the root.

2.3 Statistical consistency

An estimator is said to be statistically consistent if, given k samples of data that are generated under a model θ, the estimated model $\hat{\theta}$ converges to the generative model θ as k tends to infinity. Felsenstein (1978) used a two-state model of evolution to show that the maximum parsimony estimator is not statistically consistent. On the other hand, the maximum likelihood estimator is statistically consistent (RoyChoudhury, 2014). The minimum tree length estimator is statistically consistent if the distance estimator converges to tree-distances as sample size k tends to infinity. Developers of distance-based methods use model-based estimates of tree-distances in order to ensure statistical consistency.

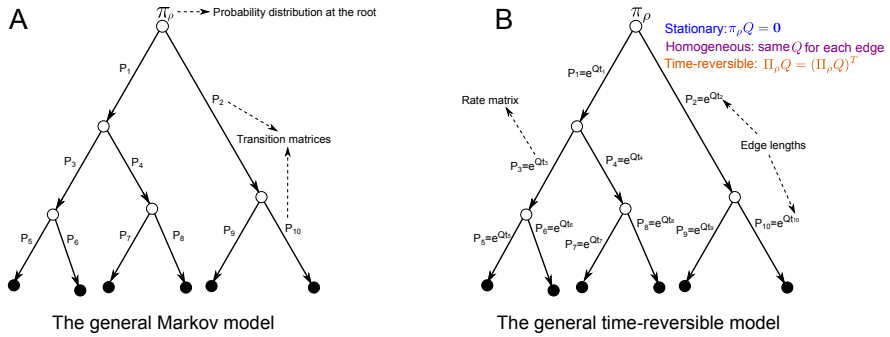


Figure 2.3: Two hidden Markov models on leaf-labeled rooted phylogenetic trees are shown here. The general Markov model (GM) is shown in panel A. The GM model is parameterized in terms of transition matrices. The general time-reversible model (GTR) is shown in panel B. The GTR model is parameterized in terms of rate matrices. The restrictions that are placed on the rate matrices of the GTR model are shown in panel B. The observed states of the hidden Markov models are represented by the labeled vertices of the phylogenetic tree.

2.4 Hidden Markov models on trees

Model-based approaches to phylogenetic tree inference assume that the observed gene sequences have evolved from a common ancestral sequence according to a tree-structured graphical model. The appeal of using models for inferring phylogenetic trees is that the parameters of the fitted models enable us to make statements about the nature of evolutionary processes that have brought about the observed genomic changes.

The probabilistic models that are used for modeling sequence evolution are hidden Markov models (HMM) on rooted phylogenetic trees. A HMM M on a phylogenetic tree $(T = V, E)$ specifies the joint probability distribution over vertices in V. Each nucleobase is assumed to have evolved independently according to a common model (independent and identically distributed (iid) assumption). Let seq(v) be the sequence represented by a vertex v, and let $\mathcal{X}_v^i$ be the variable representing the character at site i of seq(v). Let $\mathcal{X}_v$ denote the ordered set $(\mathcal{X}_v^1, \mathcal{X}_v^2, \ldots, \mathcal{X}_v^k)$ of characters in the sequence seq(v) comprising k characters. Let $\mathcal{X}_\mathcal{L}$ and $\mathcal{X}_\mathcal{H}$ denote the set of sequences for labeled vertices $\mathcal{L}$ and hidden vertices $\mathcal{H}$.

Two types of hidden Markov models (HMM) on phylogenetic trees will be discussed in this subsection: discrete-time HMM (DT-HMM) and continuous-time HMM (CT-HMM). DT-HMM are parameterized in terms of transition matrices. A square matrix P is a transition matrix if (i) each element of P is non-negative, and (ii) the sum of elements of each row of P equals one. CT-HMM are parameterized in terms of rate matrices. A square matrix Q is a rate matrix if (i) each off-diagonal element of Q is non-negative, and (ii) the sum of elements of each row of Q equals zero.

Barry and Hartigan (1987) introduced a DT-HMM on rooted phylogenetic trees that is referred to as the general Markov model (GM). The parameters of a GM model $M_{\text{GM}} = (\pi_\rho, \mathbf{P})$ on a phylogenetic tree $T = (V, E)$ comprise (i) a root probability distribution π, and (ii) the set of transition matrices $\mathbf{P} = \{P_e : e \in E\}$ (see Figure 2.3A). Each entry $P(a, b)$ of a transition matrix P specifies the conditional probability of observing state b given state a. The sum of elements of each row of P equals one.

The Markov models that are commonly used for inferring phylogenetic trees are CT-HMM. A continuous-time hidden Markov model $M_{\text{CT}} = (\pi_\rho, \mathbf{Q}, \mathbf{t})$ on a phylogenetic tree $T = (V, E)$ is parameterized in terms of (i) a root probability distribution π_ρ, (ii) the set of rate matrices $\mathbf{Q} = \{Q_e : e \in E\}$, and ($iii$) the set of edge lengths $\mathbf{t} = \{t_e : e \in E\}$. The transition matrix P_e for edge e is computed as $P_e = e^{Q_e t_e}$.

If a probability distribution π_s satisfies the condition that $\pi_s Q = \mathbf{0}$ then it follows that $\pi_s P = \pi_s$, where $P = e^{Qt}$ for any non-zero positive t (Steel, 2016). π_s is said to be the stationary distribution of

Q. The summation $-\Sigma_i \pi_s(i)Q(i,i)$ is the expected number of substitutions per unit time for a stationary homogeneous continuous-time Markov process that is defined by the rate matrix Q (Steel, 2016). It is common to scale Q such that $-\Sigma_i \pi_s(i)Q(i,i)$ is equal to one, where π_s is the stationary distribution of Q, because edge lengths are scaled in units of substitutions per site. A rate matrix Q is said to be a normalized rate matrix if $-\sum_i \pi_s(i)Q(i,i)$ equals one. All the rate matrices that are referred to in the following text are normalized rate matrices unless specified otherwise.

There are two classes of CT-HMM that will be discussed below. The first class of models are time-reversible models that are characterized by a property that makes it impossible to identify the location of the root (Felsenstein, 1981). The second class of models are Lie Markov models; Lie Markov models are a hierarchical family of Markov models that are closed under matrix multiplication (Sumner et al., 2012; Woodhams et al., 2015).

2.4.1 Time-reversible models

In the following paragraph we define the three constraints that are commonly placed on CT-HMM on phylogenetic trees: (i) stationarity, (ii) homogeneity, and (iii) time-reversibility.

Given a CT-HMM $M_{\mathrm{CT}} = (\pi_\rho, \mathbf{Q}, \mathbf{t})$ on a phylogenetic tree T. M_{CT} is stationary if $\pi_\rho Q$ equals $\mathbf{0}$ for each rate matrix Q in $\mathbf{Q}$. M_{CT} is homogeneous if the rate matrices in $\mathbf{Q}$ are identical. A stationary and homogeneous CT-HMM on a phylogenetic tree $M_{\mathrm{CT}} = (\pi_\rho, \mathbf{Q}, \mathbf{t})$ is said to be time-reversible if $\pi(a)P_{(u,v)}(a,b)$ equals $\pi(b)P_{(v,u)}(b,a)$ for each pair of adjacent vertices u, v. Time-reversibility is enforced by constraining ΠQ to be symmetric for each Q in $\mathbf{Q}$, where Π is a diagonal matrix such that $\Pi(i,i) = \pi(i)$. The widely used general time-reversible (GTR; Tavare (1986)) model is a stationary, homogeneous, and time-reversible CT-HMM on rooted phylogenetic trees (see Figure 2.3B). The unrestricted model (UNREST; Yang (1994b)) is the stationary and homogeneous CT-HMM on a phylogenetic tree which does not impose any constraints on the parameters of the rate matrix.

The model parameters of probabilistic models are estimated by maximizing the likelihood score (see equation 2.3). Time-reversible CT-HMM on rooted phylogenetic trees share the following property that makes it impossible to infer the location of the root using the likelihood score. Given a time-reversible CT-HMM $M_{\mathrm{TR}} = (\pi_\rho, \mathbf{Q}, \mathbf{t})$ on a phylogenetic tree T_ρ, let $T = (V_T, E_T)$ be the unrooted version of T_ρ. Let $e = \{u, v\}$ be any edge in E_T, and let δ be a non-negative number that is smaller than t_e. Let $T_\rho^{e,\delta}$ be the phylogenetic tree that is constructed by rooting T at distance δ away from vertex u along edge $e = \{u, v\}$. Let $M_{\mathrm{TR}}^{e,\delta} = (\pi_\rho^{e,\delta}, \mathbf{Q}^{e,\delta}, \mathbf{t}^{e,\delta})$ be a CT-HMM on $T_\rho^{e,\delta}$ such that $\pi_\rho^{e,\delta}$ equals π_ρ, $\mathbf{Q}^{e,\delta}$ equals $\mathbf{Q}$, and $\mathbf{t}^{e,\delta}$ is the set of edge lengths of $T_\rho^{e,\delta}$. The likelihood scores $\ell_{T_\rho}(M_{\mathrm{TR}}|\mathcal{X}_\mathcal{H})$ and $\ell_{T_\rho^{e,\delta}}(M_{\mathrm{TR}}|\mathcal{X}_\mathcal{H})$ are identical for any edge e in E_{T_u} and any non-negative δ that is smaller than t_e (Felsenstein, 1981). The property of time-reversible CT-HMM on rooted phylogenetic trees which is that the likelihood score does not depend on the location of the root is known as Felsenstein's pulley principle (Felsenstein, 1981).

2.4.2 Lie Markov models

Lie Markov models (Sumner et al., 2012; Fernández-Sánchez et al., 2015; Woodhams et al., 2015) are a set of nested Markov models that were designed to ensure statistical consistency in case of incomplete species sampling (see Figure 2.4). It turns out that the GTR model is not statistically consistent if there is incomplete sampling (Sumner et al., 2012), as explained in detail below.

Consider the following scenario. The gene sequences of four species l_1, l_2, l_3 and l_4 have evolved according a non-homogeneous Markov model (model A in Figure 2.4). However, sequences are only available for species l_1, l_2 and l_4 because of incomplete sampling (see Figure 2.4 B). The Markov model that is used for inference is shown in Figure 2.4 C. Assume that each transition matrix shown in Figure 2.4 belongs to a set $\mathbf{P}$ of transition matrices. In order to ensure that the set $\mathbf{P}$ of transition matrices is statistically consistent wrt incomplete sampling, it is necessary that there exists a transition matrix P_6 in $\mathbf{P}$ such that $P_6 = P_3 P_5$. Statistical consistency wrt incomplete sampling is guaranteed if $\mathbf{P}$ is closed under matrix multiplication.

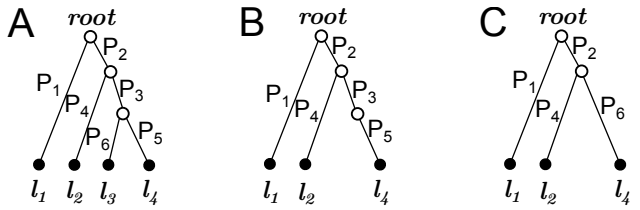


Figure 2.4: Panel A shows a non-homogeneous Markov model on a phylogenetic tree with leaves $l_1$through l_4. Sequences are only available for species l_1,l_2 and l_4 because of incomplete sampling (panel B). Let P_6 be a transition matrix such that $P_6 = P_3P_5$ (panel C).

Continuous-time HMM on trees are more widely used for phylogeny inference than discrete-time HMM on trees because the parameter known as branch length is used to construct phylogeny-based models such as the molecular clock (see subsection 2.8.3). Consequently, the development of Lie Markov models has been restricted to CT-HMM (Sumner et al., 2012; Fernández-Sánchez et al., 2015; Woodhams et al., 2015).

Let $e^{Q_3t_3}$ and $e^{Q_5t_5}$ be continuous-time realizations of the transition matrices P_3 and P_5 that are shown in Figure 2.4. Let the rate matrices Q_3 and Q_5 belong to a set $\mathbf{Q}$ of rate matrices. The set $\mathbf{Q}$ of rate matrices is said to form a Lie algebra (Steel, 2016) if

1. Each matrix in $\mathbf{Q}$ is closed under addition and scalar multiplication, and
2. The matrix commutator $[Q_1, Q_2] := Q_1Q_2 - Q_2Q_1$ for each matrix pair in $\mathbf{Q}$ is closed.

An operation on a set $\mathbf{S}$ is said to be closed if the operation on elements in $\mathbf{S}$ always maps to elements in $\mathbf{S}$.

If a set $\mathbf{Q}$ of rate matrices form a Lie algebra then, according to the Baker-Campbell-Hausdoff formula (Campbell, 1987) it follows that for each pair of rate matrices Q_1, Q_2in $\mathbf{Q}$ there exists a rate matrix Q_3 in $\mathbf{Q}$ such that

$$e^{Q_1t_1+Q_2t_2} = e^{Q_3t_3} \tag{2.4}$$

Thus, in order to ensure statistical consistency for the case of incomplete sampling, it suffices that the rate matrices that are used to parameterize a CT-HMM on a phylogenetic tree belong to a set of rate matrices that form a Lie algebra.

Fernández-Sánchez et al. (2015) constructed a hierarchy of 37 Lie Markov models such that the rate matrix of each Lie Markov model is a linear combination of a common set of basis matrices (see Figure 2.5). A rate matrix Q_{Lie}that forms Lie algebra can be expressed as $Q = \Sigma\alpha_i\mathcal{B}_i$ where $\mathcal{B}_i$ is a basis matrix and α_i is a non-negative weight. Consequently, each element of a rate matrix that forms Lie algebra can be expressed as a linear combination of weights. The nomenclature of Lie Markov models is explained below using the example RY5.6B that is listed in Woodhams et al. (2015). The columns of rate matrix $Q_{RY5.6b}$ are indexed A, G, C, T. The first two columns are indexed with purines, and the latter two columns are indexed with pyrimidines. The Lie Markov models have been developed with nucleotide-pair symmetry in mind. The model RY5.6B has purine/pyrimidine (RY) symmetry which will be explained later in this subsection. Note that in contrast to the convention of having the rows of rate matrices sum to 0, as has been adopted in this thesis, the convention used in the development of Lie Markov models is to have the columns of rate matrices sum to 0. The rate matrix $Q_{RY5.6b}$ is constructed as the linear combination $aA + a_2A_2 + dD + e_1E_1 + e_2E_2$

$$Q_{RY5.6b} = \begin{pmatrix} -3a + d + e_1 & a + 2a_2 + d + e_1 & a\text{-}a_2 + d + e_1 & a - a_2 + d + e_1 \\ a + 2a_2 + d - e_1 & -3a + d\text{-}e_1 & a - a_2 + d\text{-}e_1 & a\text{-}a_2 + d\text{-}e_1 \\ a - a_2 - d + e_2 & a - a_2 - d + e_2 & -3a - d + e_2 & a + 2a_2 - d + e_2 \\ a - a_2 - d - e_2 & a - a_2 - d - e_2 & a + 2a_2 - d - e_2 & -3a - d - e_2 \end{pmatrix} \tag{2.5}$$

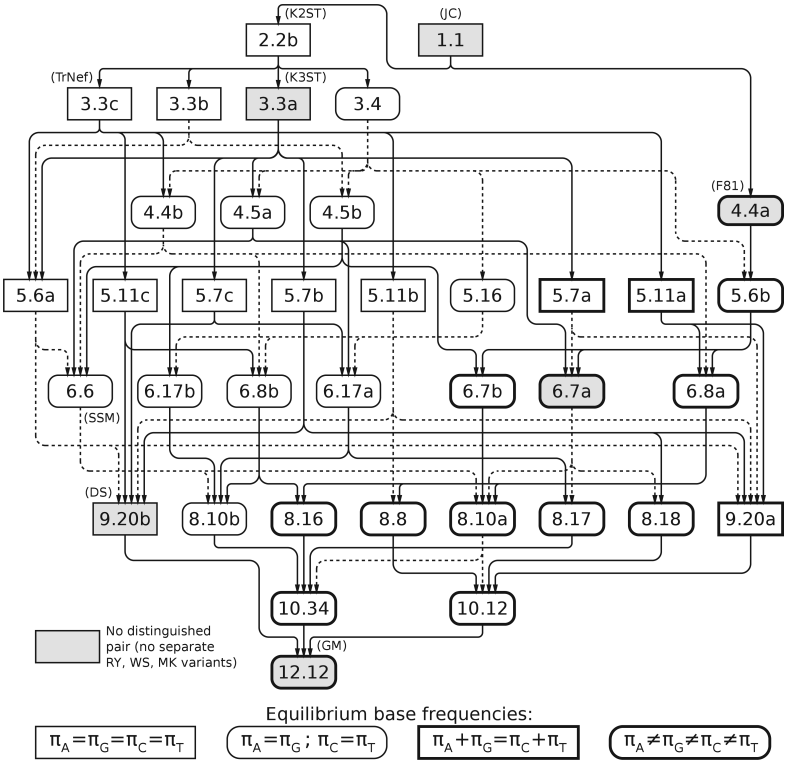


Figure 2.5: The set of Lie Markov models that have been developed by Sumner et al. (2012) and Fernández-Sánchez et al. (2015) (Figure adapted from Woodhams et al. (2015)). Arrows are directed from special models to more general models. Alternate names of models are shown in parentheses. Model 12.12 is the GM model. The use of dotted lines and solid lines is for visual clarity. The nucleotide pairing symmetry that is inherent in each Lie Markov model is not shown in the model name. Boxes that are shaded do not have distinct RY, WS or MK variants. The shape of the box that outlines model names represents constraints on the stationary distribution of the Lie Markov model.

where $A, A_2, D, E_1$and E_2 are basis matrices, and a, a_2, d, e_1 and e_2 are non-negative weights. The number 5 in the model name 5.6b indicates the number of free parameters of the model. Rate matrices are constrained to have non-negative entries in their off-diagonal element. Fernández-Sánchez et al. (2015) add a non-negativity constraint, *e.g.*, $a + 2a_2 + d + e_1 \geq 0$, for each off-diagonal element. The inequality constraints define a convex polyhedral cone, enabling a reparameterization of the off-diagonal entries of the rate matrix as a convex combination of the rays (linearly independent vectors) of the convex polyhedral cone. A set S of vectors is said to be linearly independent if the no vector v in the set can be derived as a linear combination of $S \backslash v$.The number 6 in the model name is the number of linearly independent vectors of the convex polyhedral cone. The reparameterized version of $Q_{RY5.6b}$ is given below

$$Q_{RY5.6b} = \begin{pmatrix} * & \alpha + \rho_A & \beta + \rho_A & \beta + \rho_A \\ \alpha + \rho_G & * & \beta + \rho_G & \beta + \rho_G \\ \beta + \rho_C & \beta + \rho_C & * & \alpha + \rho \\ \beta + \rho_T & \beta + \rho_T & \alpha + \rho_T & * \end{pmatrix} \tag{2.6}$$

where $*$ is set such that each column sums to zero. Note that any modification of the index of the rate matrix that preserves purine/pyrimidine grouping will result in a row and column permutation operation such that there is no change in the resulting rate matrix (Woodhams et al., 2015). The suffix "b" in the model name is used to distinguish between multiple rate matrices that share the same number of free parameters, and the same number of parameters in the reparameterized versions.

In addition to purine/pyrimidine grouping {{A,G},{C,T}}, two additional groupings have been developed: {{A,T},{G,C}} which is denoted by WS, and {{A,C},{G,T}} which is denoted by MK (Woodhams et al., 2015). R, Y, W, S, M, and K are the IUPAC ambiguity codes for the pairs: {A,G}, {C,T}, {A,T}, {G,C}, {A,C}, {G,T}. Six out of 37 Lie Markov models are identical wrt RY, WS, and MK grouping. In total there are 99 Lie Markov models that have been implemented in IQ-TREE v1.6.1 (Nguyen et al., 2015) and BEAST v2.0 (Bouckaert et al., 2014).

2.4.3 Mixture models that account for heterogeneous rate of substitutions across sites

Base-pair substitutions have been observed to occur at different rates across sites. For example, nucleotides in the third codon position are substituted more frequently than the nucleotides in the first two codon positions because of redundancy in the genetic code at the third codon position. There is a family of models that is commonly modeled to account for site-heterogeneity in substitution rates comprising: (*i*) the invariable sites model (I) restricts a proportion of sites to be invariable by setting edge lengths to zero (Steel et al., 2000), (*ii*) a mixture model that draws rates from a discrete Gamma distribution Γ_k with k classes (Yang, 1994a), and (*iii*) a mixture model R_k that allows k rates to vary independently instead of constraining the rates to be drawn from the same probability distribution (Yang, 1995). The free-rate model is known as the discrete-rate CAT model as implemented in FastTree.

2.5 Tree-search under continuous-time HMM on trees

Let $\ell^*_{T_\rho}(D) = \underset{M}{\operatorname{argmax}}\, \ell_{T_\rho}(M|D)$ denote the maximum likelihood score of a phylogenetic tree T_ρ. The maximum likelihood (ML) problem is the combinatorial optimization problem of finding a phylogenetic tree T_ρ such that $\ell^*_{T_\rho}(D)$ is maximum. The total number of distinct rooted phylogenetic trees with n leaves is $\prod_{i=0}^{n-2}(2i+1)$ for $n \geq 2$ (Felsenstein, 2003). The total number of distinct unrooted phylogenetic trees with n leaves is $\prod_{i=0}^{n-3}(2i+1)$ for $n \geq 3$ (Felsenstein, 2003).

The general approach to approximate the ML problem is to compute initial trees using fast approaches such as neighbor-joining (Saitou and Nei, 1987) or stepwise addition (Wagner, 1961). Subsequently, the tree space is explored via tree modification operations such that incremental improvements to $\ell^*_{T_\rho}(D)$ are smaller than a threshold that is specified *a priori*. Tree modification operations that are used in practice are nearest

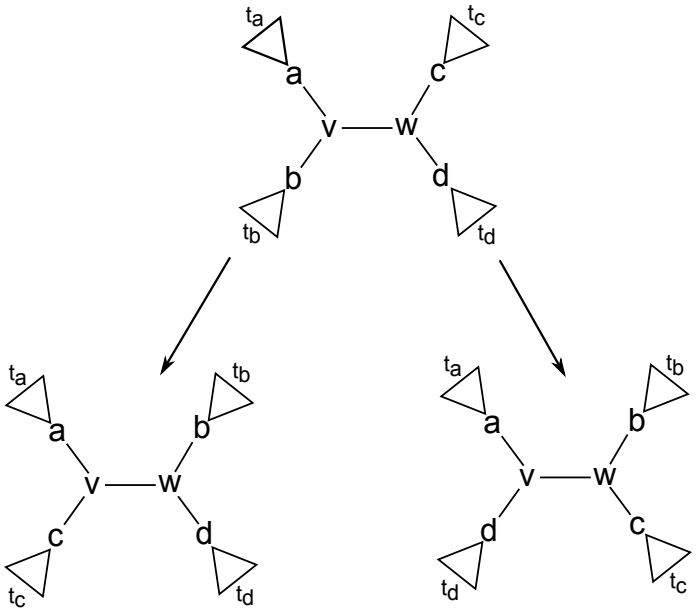


Figure 2.6: Nearest neighbor interchange (NNI) moves for unrooted phylogenetic trees operating on the edge $\{v, w\}$.

neighbor interchange (NNI) , subtree prune and regraft (SPR), and tree bisection and reconnection (TBR) Steel (2016).

Tree modification operations on unrooted phylogenetic trees are described below because the majority of software implement time-reversible CT-HMM on phylogenetic trees. Tree modification operations on rooted phylogenetic trees are not described in this thesis.

2.5.1 Searching through tree space

A nearest neighbor interchange (NNI) move involves swapping the neighbors of adjacent non-leaf vertices. An NNI move on an unrooted phylogenetic tree $T = (V_T, E_T)$ is defined with respect to any edge $\{v, w\}$ in E_T that is not incident to a leaf, and neighbors n_v of v and n_w of w, respectively. The two possible NNI moves involving an edge $\{v, w\}$ are shown in Figure 2.6.

A subtree prune and regraft (SPR) move involves removing (pruning) a subtree and inserting (grafting) the subtree at a new location. An SPR move on an unrooted phylogenetic tree $T = (V_T, E_T)$ is defined with respect to a subtree $\tau_v = (V_{t_v}, E_{\tau_v})$ of T, an edge $\{y, z\}$ in $E_T \backslash E_{\tau_v}$, and a non-negative distance d smaller than $t_{\{y,z\}}$. Given a subtree τ_v, an edge $e = \{y, z\}$, and a distance d, an SPR move involves the following steps (i) removing the edge $\{v, w\}$ in E_T such that w is not in V_{τ_v}, (ii) suppressing the vertex w, (iii) selecting an edge $e = \{y, z\}$ in $E_T \backslash E_{\tau_v}$, (iv) adding a vertex x at a feasible distance d from y along the edge e, and (v) adding the edge $\{v, x\}$ to E_T resulting in a connected graph.

A tree bisection and reconnection (TBR) move involves removing an edge and connecting the subsequently disconnected components by a newly added edge. A TBR move on an unrooted phylogenetic tree $T = (V, E)$ that removes edge $\{v, w\}$ in E is performed as follows. The edge $\{v, w\}$ in E is removed resulting in the construction of connected components $C_v = (V_{C_v}, E_{C_v})$ containing v, and $C_w = (V_{C_w}, E_{C_w})$ containing w, respectively. Subsequently vertices v and w are suppressed. Finally two edges $e_v = (y_v, z_v)$ in E_{C_v}, and $e_w = (y_w, z_w)$ in E_{C_w} are selected. A new vertex v' is inserted along e_v at distance d_v away y_v, and a new

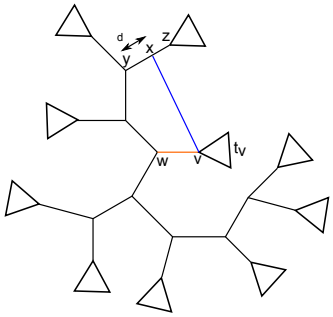


Figure 2.7: A subtree prune and regraft (SPR) move on an unrooted phylogenetic tree. The SPR move shown above involves removing the edge colored in orange and adding the edge colored in blue.

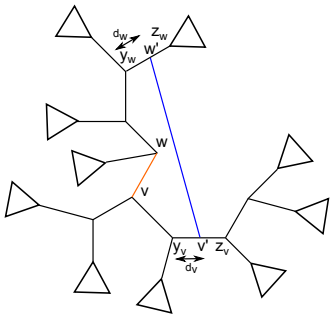


Figure 2.8: A tree bisection and reconnection (TBR) move on an unrooted phylogenetic tree. The TBR move shown above involves removing the edge colored in orange and adding the edge colored in blue.

vertex w' is inserted along e_w at distance d_w away y_w. Finally the edge $\{v', w'\}$ is added to E_u (see Figure 2.8).

The tree space of rooted phylogenetic trees can be explored using modified versions of the NNI, SPR and TBR moves described above, and are not explained in detail in this thesis.

Parameter optimization is performed alongside tree modification operations. Approaches for optimizing the parameters of CT-HMM on phylogenetic trees are described in the following Subsection.

2.5.2 Optimizing the parameters of CT-HMM on phylogenetic trees

The parameters of CT-HMM on phylogenetic trees include the free parameters of rate matrices and edge lengths. Optimization of the parameters mentioned above involves increasing the likelihood score such that incremental changes to the likelihood score are below a threshold that is specified *a priori*. A commonly adopted strategy is to iteratively optimize rate matrix parameters for fixed edge lengths, and optimize edge lengths for fixed rate matrix parameters (Yang, 2000). Edge lengths are usually optimized sequentially using Brent's method or Newton-Raphson's method (Bryant et al., 2005; Nocedal and Wright, 2006).

The task of computing the likelihood score (equation 2.3) is computationally demanding. The likelihood score can be computed in $O(nkA^2)$ using a dynamic programming algorithm (Felsenstein's tree pruning algorithm) where n is the number of leaves in the phylogenetic tree, k is the number of columns in the sequence alignment, and A is the number of states in the HMM which is four for DNA substitution models (Felsenstein, 1981).

The tree pruning algorithm computes the likelihood score in equation 2.3 as follows. The *conditional likelihood* $L_u^i(x)$ of observing character x at base pair (site) i at an unlabeled vertex u is computed recursively as

$$L_u^i(x) = \left(\sum_y P_{(u,v)}(y|x) L_v^i(y)\right)\left(\sum_z P_{(u,w)}(z|x) L_w^i(z)\right) \tag{2.7}$$

where v and w are the children of u. The *conditional likelihood vector* L_u^i is the marginal probability

$$L_u^i = \sum_{\mathcal{X}_h^i : h \in \mathcal{H}_{\tau_u} \setminus \{u\}} P(\{\mathcal{X}_v^i : v \in V_{\tau_u}\}, M_{\tau_u})$$

where $\mathcal{H}_{\tau_u}$ is the set of hidden vertices in the subtree $\tau_u = (V_{\tau_u}, E_{\tau_u})$ that is rooted at vertex u. M_{τ_u} is the set of transition matrices $M_{\tau_u} = \{P_e : e \in E_{\tau_u}\}$ The conditional likelihood vector L_u^i at a leaf u is defined as follows. Let $\mathcal{X}_l^i$ be the character that is observed at site i of the sequence that is represented by leaf l.

$$L_l^i(x) = \begin{cases} 0 & x \neq \mathcal{X}_l^i \\ 1 & x = \mathcal{X}_l^i \end{cases}$$

The likelihood score L^i for site i is computed as

$$L^i = \sum_x \pi_\rho(x) L_\rho^i(x),$$

where ρ is the root. Under the iid assumption the total likelihood score L is given by $L = \Pi_i L^i$. The log likelihood score ℓ is computed instead of the likelihood score in order to avoid numerical underflow.

$$\ell = \sum_{i=1}^{k} \log L^i$$

The computational burden of computing the log likelihood score is high. One technique that is commonly used to reduce computational burden is to compute site likelihood scores L^i for sites with distinct site patterns.

The site pattern for site (base pair) i is the ordered set of characters $(\mathcal{X}_l^i : l \in \mathcal{L}_T)$ that are present in a column of a multiple sequence alignment. Identical site patterns have identical likelihood scores (Felsenstein, 1981). It is standard practice to compute conditional likelihood vectors for each unique site pattern, and reuse conditional likelihood vectors for each repeated site pattern (Felsenstein, 1981).

$$\ell = \sum_i w^i \log L^i$$

where w^i is the number of times that site pattern i repeats.

Additionally, conditional likelihood vectors are rescaled in order to avoid numerical underflow, and log transformed values of scaling factors are added to the log likelihood score (Yang, 2000).

The task of optimizing the lengths of newly added edges, and the lengths of edges that are modified subsequent to a tree modification operation is made less computationally demanding by reusing the conditional likelihood vectors (see equation 2.7) that correspond to the root vertices of subtrees that are unchanged subsequent to the tree modification operation.

A commonly used compute-time saving technique is to use a fast-to-compute score that can be used to avoid tree-rearrangements that are likely to reduce the likelihood score (Hordijk and Gascuel, 2005). The alternative criterion that is used instead of maximum likelihood is minimum evolution. The score that is optimized by minimum evolution is tree length.

The parameters of rate matrices are optimized using quasi-Newton methods such as Broyden-Fletcher-Goldfarb-Shano (Fletcher, 1987; Yang, 2000) or gradient-free methods such as Powell's method (Powell, 1964; Holder et al., 2008). The computational burden of optimizing the parameters of the rate matrix of a homogeneous CT-HMM on a phylogenetic tree $T_\rho = (V_{T_\rho}, E_{T_\rho})$ cannot be reduced by reusing conditional likelihood vectors subsequent to each tree-modification operation because a change to any entry of a rate matrix parameter results in a change in the transition matrix for each edge in E_{T_ρ}. In practice the free parameters of the rate matrix are optimized infrequently after a considerable number of tree modification operations have been performed (Sullivan et al., 2005). The use of non-homogeneous CT-HMM further increases the number of rate matrix parameters that need to be estimated. A brief overview of methods that perform phylogeny inference under non-stationary non-homogeneous Markov models are presented below.

Galtier and Gouy (1998) introduced a non-stationary non-homogenous CT-HMM that allows base composition to vary over each the edges of a phylogenetic tree. Boussau and Gouy (2006) implemented a tree search algorithm called nhPhyML that searches for maximum likelihood phylogenetic trees under the Galtier and Gouy (1998) model. Yang and Roberts (1995) implemented a non-homogenous CT-HMM such that the rate matrix for each edge is a rate matrix of the HKY model. Additionally, Yang and Roberts (1995) allowed base composition to vary across edges. p4 (Foster, 2004), PHASE (Gowri-Shankar and Rattray, 2007), and PhyloBayes (Blanquart and Lartillot, 2006) perform Bayesian inference via Markov chain Monte Carlo (MCMC) sampling under non-stationary non-homogeneous CT-HMM. Jayaswal et al. (2005) provide a method for fitting the general Markov model which takes as input an unrooted phylogenetic tree. In recent work Williams et al. (2015) developed a MCMC sampling scheme for inferrinxg phylogenetic trees under non-reversible, and non-homogenous CT-HMM. None of the methods mentioned above are applicable to large data sets comprising more than 1000 species.

The CT-HMM that are currently implemented by most of the widely used software are stationary, homogeneous, and time-reversible (RAxML-NG (Kozlov et al., 2019), PhyML (Guindon et al., 2010), FastTree (Price et al., 2010), RevBayes (Hohna et al., 2016) and BEAST v1.10 (Suchard et al., 2018)). IQ-TREE v1.6.1 (Nguyen et al., 2015) and BEAST v2.0 (Bouckaert et al., 2014) implement Lie Markov models and time-reversible models that are not Lie Markov models, such as the GTR model. Bettisworth and Stamatakis (2020) described a method called RootDigger for placing the root on an unrooted tree using the UNREST model.

Popular programs that perform model selection for DNA substitution models such as ModelTest-NG (Darriba et al., 2020), ModelFinder (Kalyaanamoorthy et al., 2017) and Smart Model Selection (Lefort et al., 2017) evaluate the GTR model, and special cases of the GTR model such as the Tamura-Nei 93 model (TN93; Tamura and Nei (1993)) and the Hasegawa-Kishino-Yano 85 model (HKY85; Hasegawa et al. (1985)).

IQ-TREEv1.6.1 (Nguyen et al., 2015) performs model selection using Lie Markov models and time-reversible models that are not Lie Markov models, such as the GTR model.

It is common practice to select the most appropriate CT-HMM on phylogenetic trees using criteria such as Akaike information criterion (AIC; Akaike (1974)), and Bayesian information criterion (BIC; Schwarz (1978)), which are defined as

$$\text{AIC} = -2\log - \text{likelihood} + 2m \tag{2.8}$$

$$\text{BIC} = -2\log - \text{likelihood} + m\log k, \tag{2.9}$$

where m is the number of free parameters and k is the number of observations. The number of observations equals the number of alignment columns.

2.5.3 Matrix exponentiation of rate matrices

It is necessary to exponentiate rate matrices in order compute the likelihood score using CT-HMM. The wide use of time-reversible CT-HMMs models is commonly justified on the basis of mathematical convenience because it is always possible to exponentiate time-reversible rate matrices using eigenvalue decomposition (Felsenstein, 2003). This is because (i) given a time-reversible rate matrix Q, the matrix ΠQ is symmetric where the diagonal matrix Π has the stationary distribution of Q as its diagonal elements, (ii) symmetric matrices with real entries are guaranteed to be diagonalizable such that the diagonal matrix comprises real numbers (Golub and Van Loan, 1996), (iii) it is mathematically easy to exponentiate diagonalizable matrices that contain real numbers as explained below. We explain how time-reversible rate matrices are exponentiated using the derivation given by Bryant et al. (2005). Given a time-reversible rate matrix Q construct the matrix $\Pi^{1/2}Q\Pi^{-1/2}$, which is symmetric because it can be constructed by multiplying the symmetric matrix $\Pi^{-1/2}$ to the left of ΠQ, and multiplying $\Pi^{-1/2}$ to the right of $\Pi^{1/2}Q$. Diagonalize $\Pi^{1/2}Q\Pi^{-1/2}$ as BDB^{-1}. Note that Q can be factorized as ADA^{-1} where A is $\Pi^{-1/2}B$. Compute the matrix exponential e^Q using the Taylor series expansion as follows.

$$\begin{aligned}
e^Q &= \sum_{k=0}^{\infty} \frac{(Q)^k}{k!} \\
&= \sum_{k=0}^{\infty} \frac{(Q)^k}{k!} \\
&= \sum_{k=0}^{\infty} \frac{\left(ADA^{-1}\right)^k}{k!} \\
&= \sum_{k=0}^{\infty} \frac{\left(ADA^{-1}ADA^{-1}\dots ADA^{-1}\right)}{k!} (\text{there are } k \ ADA^{-1} \text{ terms}) \\
&= \sum_{k=0}^{\infty} \frac{\left(AD^kA^{-1}\right)}{k!} \\
&= A\left(\sum_{k=0}^{\infty} \frac{D^k}{k!}\right)A^{-1} \\
&= Ae^DA^{-1}
\end{aligned}$$

where the matrix exponential of a diagonal matrix D is the diagonal matrix with the scalar exponential e^{d_i} as the i^{th} diagonal entry where d_i is the i^{th} diagonal entry of D. ΠQ is not necessarily symmetric if the rate matrix is the unrestricted rate matrix (UNREST). Consequently, it is not always possible to diagonalize

the UNREST rate matrix such that the diagonal elements are guaranteed to be real. The alternate way of exponentiating rate matrices that does not involve eigenvalue decomposition is to numerically approximate the Taylor series expansion. We used the numerical approximation techniques that are implemented in the scientific computing package for python, Scipy (Virtanen et al., 2020), and the C++ library Eigen v3 (Guennebaud and Benoit, 2010) in order to exponentiate unrestricted rate matrices.

2.6 Related work on the general Markov model

All popular phylogeny inference software exclusively implements CT-HMM on phylogenetic trees. The general Markov model (GM) is a DT-HMM on phylogenetic trees. The following section discusses related work on inferring phylogenetic trees under the assumption that sequences were generated according to a general Markov model.

2.6.1 Barry and Hartigan's paper

The GM model on phylogenetic trees was introduced by Barry and Hartigan (1987), who stated that their model was not identifiable, *i.e.*, it is not possible to identify the GM model using the likelihood score because there are several models that yield identical likelihood scores. Barry and Hartigan reparameterized the GM model in terms of edge-wise joint probability matrices, and provided an EM algorithm for optimizing model parameters. The EM algorithm for the reparameterized version of the GM model has been implemented by Jayaswal et al. (2005). Additionally, Barry and Hartigan introduced a distance measure which is more widely known than the general Markov model. The distance measure has come to be known as the logDet which is defined as follows. Given species u and v, let $F_{(u,v)}$ be the estimated joint probability matrix such that $F_{(u,v)}(x, y)$ is the fraction of sites at which character $\mathcal{X}_u$ equals x, and $\mathcal{X}_v$ equals y.

$$\text{logDet}(u, v) = -\ln|\det(F_{(u,v)})| \tag{2.10}$$

logDet distances are tree-distances for all u, v s.t. $u \neq v$. The notion of tree-distances is defined wrt phylogenetic trees with edge lengths. Given a GM model on a rooted phylogenetic tree T_ρ, Steel (1994) showed that there exists an edge length function λ that is defined on the edges of the unrooted version T of T_ρ such that logDet distances (see equation 2.10) are additive in T with respect to λ. Given a GM model M on a rooted tree T_ρ, logDet distances are additive in the unrooted version of T under the assumption that $F_{(u,v)}$ equals the joint probability distribution over $\mathcal{X}_u$ and $\mathcal{X}_v$ that is defined by M on T_ρ (Steel, 1994). The widely known distance-based clustering method neighbor-joining (NJ; Saitou and Nei (1987)) is statistically consistent if distances are tree-distances in the model tree. NJ using logDet distances is one the most common methods to construct trees under the GM model (Sheffield et al., 2009).

2.6.2 Phylogenetic invariants

The typical approach to tree search involves computing the odds that the observed site patterns were generated by the combination of tree and parameters of interest, and selecting the combination of tree and model parameters that has the greatest odds of generating the observed data. Phylogenetic invariants are a radically different way of finding trees. Invariants of a HMM on a tree are polynomials in site pattern frequencies that vanish (evaluate to zero) at observed site pattern frequencies if the observed data was generated by the HMM on trees under consideration (Allman and Rhodes, 2007). Note that invariants provide a way of selecting topologies without having to concern oneself with parameter estimation. Consequently, invariant-based methods can be used to infer the topology of a GM model on trees without having to learn model parameters. A simple example of a phylogenetic invariant will be given below. Consider a GM model on a two-species tree $T = (V = \{\rho, a, b\}, E = \{(\rho, a), (\rho, b)\})$. The joint probability over $\{a, b\}$ is given by

$$P(\mathcal{X}_a = j, \mathcal{X}_b = k) = p_{jk} = \sum_{i=1}^{4} \pi_\rho(i) P_{(\rho,a)}(i, j) P_{(\rho,b)}(i, k)$$

The joint distribution $P(\mathcal{X}_a, \mathcal{X}_b)$ can be expressed as a 4×4 matrix, with p_{jk} as the entry in row j and column k, such that each entry is a degree-3 polynomial comprising four terms. The following polynomial, known as *the stochastic invariant*, is an example of a phylogenetic invariant.

$$\sum_{j,k} p_{jk} - 1$$

The stochastic invariant is a trivial invariant because any joint probability distribution must sum to one. Invariants can be used to infer the topology of the generative model can be identified if the invariants are constructed based on topological information about the underlying model. Consider a further simplification of the example shown above where the state at the root is constrained to be A, *i.e.*, $\pi_\rho = (1, 0, 0, 0)$. p_{jk} can be expressed as

$$p_{jk} = \pi_\rho(1) P_{(\rho,a)}(1, j) P_{(\rho,b)}(1, k) = P_{(\rho,a)}(1, j) P_{(\rho,b)}(1, k)$$

It follows that the polynomial $p_{jk}p_{mn} - p_{jm}p_{kn}$ is an invariant because

$$p_{jk}p_{mn} = p_{jm}p_{kn} = P_{(\rho,a)}(1, j) P_{(\rho,b)}(1, k) P_{(\rho,a)}(1, m) P_{(\rho,b)}(1, k)$$

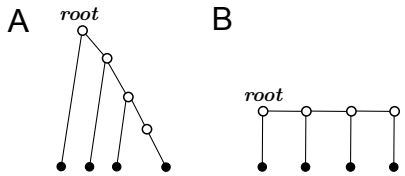
In general, invariants can be used to discriminate between competing topologies based on how closely the corresponding polynomials approach zero if evaluated at observed site pattern frequencies. The construction of invariants is a daunting task. There are an exponential number of polynomials in an exponential number of variables that need to be evaluated for an exponential number of tree topologies. Phylogenetic invariants for the general Markov model do not depend on the location of the root (Allman and Rhodes, 2008). Consequently, all the invariant-based methods that have been developed for the GM model have either been used to construct unrooted trees, or to score splits. There are two programs that make use of invariants. The first program is by Nicholas Eriksson. Eriksson computes unrooted trees in a neighbor-joining like fashion by identifying splits using singular-value decomposition (Eriksson, 2005). The second program SpitSup (Allman et al., 2017) takes as input a multiple sequence alignment and a set of splits, and scores the splits by constructing split-specific invariants and evaluating the invariants using observed site pattern frequencies. Additionally, SplitSup can perform a scanning window analysis to assign window-specific scores to input splits.

Pachter and Sturmfels (2005) note that the parameters of the GM model on phylogenetic trees can be optimized using an expectation-maximization algorithm (EM). An EM algorithm for the GM model on phylogenetic trees is described in the following Section.

2.6.3 Expectation-maximization

Consider a phylogenetic tree T'_{CAT} that is constructed by removing one edge and one labeled vertex from the edge set and the vertex set, respectively, of a rooted caterpillar tree such that each hidden vertex has exactly one child vertex that is labeled. T'_{CAT} resembles the hidden Markov model (Cappé et al., 2005) that is commonly used for finding DNA sequence patterns (see Figure 2.9 A that was adapted from Pachter and Sturmfels (2005)). The parameters of the hidden Markov model can be optimized using an expectation-maximization algorithm (EM; Dempster et al. (1977)), known as the Baum-Welch algorithm (Cappé et al., 2005).

EM algorithms are a class of algorithms that are used to infer the parameters of models with hidden variables. If all the variables of interest were observed then the desired maximum likelihood estimates of model parameters could be inferred in closed form. If there are hidden variables then one can make the problem of parameter estimation feasible by (i) filling in values for hidden variables using a suboptimal estimate of



Two views of the rooted caterpillar-like tree

Figure 2.9: A phylogenetic tree with each hidden vertex having one labeled vertex as a child is shown in panel A. The tree shown in panel A is redrawn in panel B to resemble the hidden Markov model. The Figure shown above has been adapted from Pachter and Sturmfels (2005).

model parameters, and subsequently (ii) using the complete data set to estimate model parameters. Steps (i) and (ii) are performed iteratively such that the likelihood score increases with each iteration.

Koller and Friedman (2009) describe an EM algorithm for optimizing the parameters of Bayesian networks. A Bayesian network generalizes the general Markov model on phylogenetic trees by allowing multiple parents. The EM algorithm described by Koller and Friedman (2009) makes use of Pearl's belief propagation algorithm (Pearl, 1982) for performing the expectation step, and closed-form solutions for the maximization step. The belief propagation algorithm was developed for the special case of Bayesian networks where the number of parents is limited to one, in which case the Bayesian network is the general Markov model on phylogenetic trees.

First, we described the maximization step for the case where there are no hidden variables. Subsequently, we show how to compute expectation statistics that are sufficient to optimize the parameters of a suboptimal general Markov model on a leaf-labeled phylogenetic tree.

2.6.3.1 Maximization step:

If there are no hidden variables then the maximum likelihood estimate can be computed in closed form (Koller and Friedman, 2009). Consider a GM model M_{GM} on a fully labeled phylogenetic tree $T_{\mathrm{full}} = (V_{\mathrm{full}}, E_{\mathrm{full}})$.

Let $\overline{C_u}$ be the normalized observed count matrix for any vertex u in V_{full}, which can be computed as

$$\overline{C_u}(x) = \frac{1}{k} \sum_{i=1}^{k} \delta(\mathcal{X}_u^i, x), \tag{2.11}$$

where x denotes nucleotides and $\delta(x, y)$ is the Kroenecker delta function.

Let $\overline{C_{(u,v)}}$ be the normalized observed count matrix for any edge (u, v) in E_{full}, which can be computed as

$$\overline{C_{(u,v)}}(x, y) = \frac{1}{k} \sum_{i=1}^{k} \delta(\mathcal{X}_u^i, x) \times \delta(\mathcal{X}_v^i, y) \tag{2.12}$$

The maximum likelihood estimate (MLE) of parameters of M_{GM} can be computed in closed form as follows (Koller and Friedman, 2009):

$$\pi_\rho^{\mathrm{MLE}}(x) = \overline{C}_\rho(x), \text{ and} \tag{2.13}$$

$$P_{(u,v)}^{\mathrm{MLE}}(x, y) = \frac{\overline{C}_{(u,v)}(x, y)}{\overline{C}_u(x)} \tag{2.14}$$

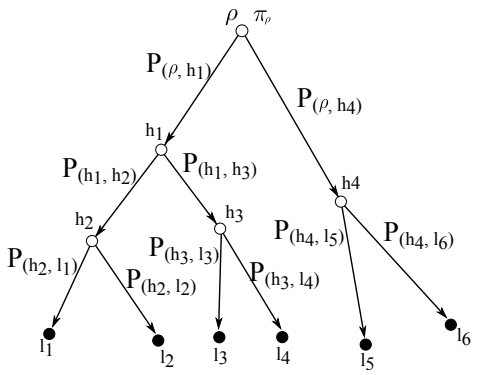


Figure 2.10: A general Markov model (GM) on a phylogenetic tree $T = (V_T, E_T)$. Each edge e in E_T is labeled with transition matrix P_e. The root ρ is labeled with the root probability distributionπ_ρ.

2.6.3.2 Expectation step:

If we have suboptimal estimates of a GM model M on a leaf-labeled phylogenetic tree then the parameter estimates of M that are guaranteed to improve the likelihood score can be computed using the expected values of the count matrices listed in equation 2.11 and equation 2.12.

The expected counts $E_M[\overline{C}_u(x)]$ of variable $\mathcal{X}_u$ can be computed as follows (Koller and Friedman, 2009):

$$E_M\left[\overline{C}_{(u)}(x)\right] = \sum_{i=1}^{k} P(\mathcal{X}_u^i = x), \tag{2.15}$$

where $P(\mathcal{X}_u^i)$ is the marginal probability

$$P(\mathcal{X}_u^i) = \sum_{\mathcal{X}_h^i : h \in \mathcal{H}_{T_\rho} \setminus \{u\}} P(\{\mathcal{X}_v^i : v \in V_{T_\rho}\}|M)$$

Similarly, the expected counts $E_M\left[\overline{C}_{(u,v)}(x,y)\right]$ of variable pair $\mathcal{X}_u, \mathcal{X}_v$ can be computed as follows

$$E_M\left[\overline{C}_{(u,v)}(x,y)\right] = \sum_{i=1}^{k} P(\mathcal{X}_u^i = x, \mathcal{X}_v^i = y), \tag{2.16}$$

where $P(\mathcal{X}_u^i, \mathcal{X}_v^i)$ is the marginal probability

$$P(\mathcal{X}_u^i, \mathcal{X}_v^i) = \sum_{\mathcal{X}_h^i : h \in \mathcal{H}_{T_\rho} \setminus \{u,v\}} P(\{\mathcal{X}_v^i : v \in V_{T_\rho}\}|M)$$

The marginal probabilities listed in equation 2.15 and equation 2.16 can be computed efficiently using the belief propagation algorithm by Pearl (1982), as described below.

Belief propagation makes use of a graphical structure known as clique tree. Here we define clique trees for the special case of phylogenetic trees. Given a phylogenetic tree T, a clique tree $T^{\text{CT}} = (V_{T^{\text{CT}}}, E_{T^{\text{CT}}})$ of $T = (V_T, E_T)$ is a undirected tree such that each edge in E_T is represented by a distinct vertex in $V_{T^{\text{CT}}}$. Figure 2.11 depicts the clique tree for a GM model on the phylogenetic tree shown in Figure 2.10. Each vertex of a clique tree is referred to as a clique. The *scope* of a clique is the variable pair that is represented by the clique. For instance the scope of clique $C_{(h_3,l_3)}$ is $(\mathcal{X}_{h_3}, \mathcal{X}_{l_3})$. Operations on a clique tree are defined in terms of *factors* which are clique-specific functions that are defined on the variables included in the scope

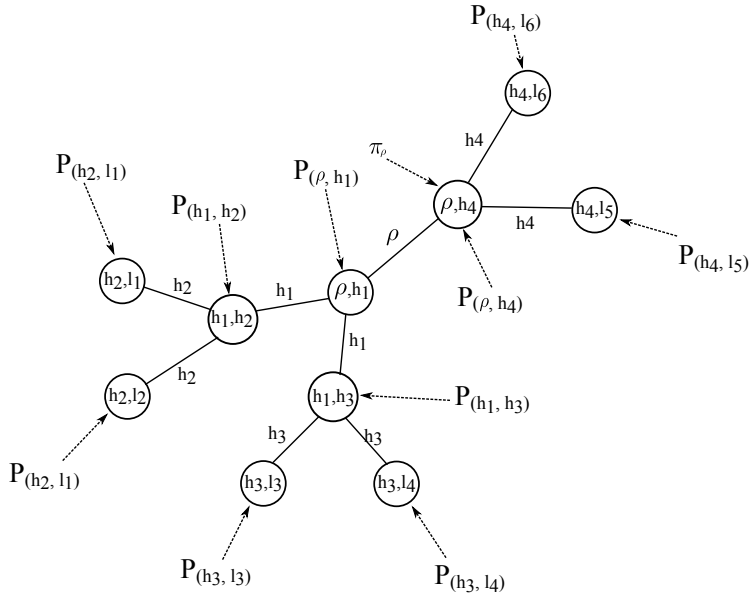


Figure 2.11: A clique tree of the phylogenetic tree T that is shown in Figure 2.10. Each vertex of a clique tree represents an edge in T. Each model parameter of the HMM on T is assigned to exactly one vertex of the clique tree (shown with dashed edges). The parameters π_ρ and $P_{(\rho,h_4)}$ are assigned to the clique $C_{(\rho,h_4)}$. Each edge of the clique tree is labeled with the scope of messages that are sent across the edge.

of one of more cliques. Factors provide a means of reparameterizing joint probability distributions in terms of clique-specific parameters. Three types of factors will be introduced in this section: potentials, messages, and beliefs.

The *potential* $\psi_{(u,v)}$ of a clique $C_{(u,v)}$ is a measure of co-occurrence of variables $\mathcal{X}_u$ and $\mathcal{X}_v$. The potential of a clique is initialized using one or more model parameters such that each model parameter is assigned to one clique, as defined in equation 2.17. An example of parameter assignment is shown in Figure 2.11.

$$\psi_{(u,v)}(x,y) = \begin{cases} P_{(u,v)}(x,y) & \text{if the factor } P_{(u,v)} \text{ is assigned to the clique } C_{(u,v)} \\ \pi_u(y)P_{(u,v)}(x,y) & \text{if factors } P_{(u,v)} \text{ and } \pi_u \text{ are assigned to the clique } C_{(u,v)} \end{cases} \tag{2.17}$$

Conditioning on observed data is performed by restricting the potential of cliques that contain an observed variable in their scope. Consider a clique $C_{(h,l)}$ that contains the observed variable $\mathcal{X}_l$ in its scope. Let the initial potential $\psi^{\text{init}}_{(h,l)}$ of $C_{(h,l)}$ be

$$\psi^{\text{init}}_{(h,l)} = \begin{array}{c|cccc} & \text{A} & \text{T} & \text{G} & \text{C} \\ \text{A} & 0.89 & 0.01 & 0.05 & 0.05 \\ \text{T} & 0.05 & 0.85 & 0.04 & 0.06 \\ \text{G} & 0.02 & 0.02 & 0.95 & 0.01 \\ \text{C} & 0.03 & 0.01 & 0.01 & 0.95 \end{array} \tag{2.18}$$

where rows and columns are indexed by nucleotides. Let the observed state $\mathcal{X}_l^i$ for $\mathcal{X}_l$ be G for site i. The initial potential $\psi^{\text{init}}_{(h,l)}$ is restricted to state G by setting to zero all entries in each column of $\psi^{\text{init}}_{(h,l)}$ that is not indexed by G, resulting in the following matrix.

$$\psi_{(h,l)} = \begin{array}{c|cccc} & \text{A} & \text{T} & \text{G} & \text{C} \\ \text{A} & 0 & 0 & 0.05 & 0 \\ \text{T} & 0 & 0 & 0.04 & 0 \\ \text{G} & 0 & 0 & 0.95 & 0 \\ \text{C} & 0 & 0 & 0.01 & 0 \end{array} \tag{2.19}$$

A *message* is a factor that is computed by marginalizing over a variable in the domain of the potential of a clique. The message μ_v is computed by marginalizing $\psi_{(u,v)}$ over the variable $\mathcal{X}_u$. The message μ_h for the factor $\psi_{(h,l)}$ shown in equation 2.19 is the column vector

$$\mu_h = \begin{array}{c} 0.05 \\ 0.04 \\ 0.95 \\ 0.01 \end{array}$$

Messages can be multiplied into potentials via matrix multiplication. The belief $\beta_{(u,v)}$ of a clique $C_{(u,v)}$ for site i is the marginal probability $P(\mathcal{X}_u^i, \mathcal{X}_v^i) = \sum_{\mathcal{X}_h^i : h \in \mathcal{H}_{T_\rho} \setminus \{u,v\}} P(\{\mathcal{X}_v^i : v \in V_{T_\rho}\}|M)$. Belief propagation is an operation on a clique tree that computes the belief of each clique by passing messages along the edges of a clique tree as defined in Algorithm 1.

Algorithm 1: Belief propagation

Input: A GM model M_{GM} on $T = (V = \{\mathcal{H}, \mathcal{L}\}, E)$, and the observed state for each labeled vertex.
Initialize:
Compute clique tree $T_{\text{CT}} = (V_{\text{CT}}, E_{\text{CT}})$
Assign parameters of M_{GM} to cliques in V_{CT}
Set initial potential $\psi_{(u,v)}^{\text{init}}$ of each clique $C_{(u,v)}$ in V_{CT}
Pick any non-leaf vertex in T_{CT} as the root clique, and direct all edges away from the root
Let $V_{\text{CT}}^{\text{pre}}$ and $V_{\text{CT}}^{\text{post}}$ be the ordered sets comprising vertices in V_{CT} that are visited in preorder traversal on T_{CT}, and postorder traversal on T_{CT}, respectively
Set potential of each clique to the initial potential of the clique
For clique $C_{(v,w)}$ in $V_{\text{CT}}^{\text{post}}$
  If w is a labeled vertex
    Condition on $\mathcal{X}_w$ by restricting $\psi_{(v,w)}$ based on observed state $\mathcal{X}_w$
  Else
    Multiply messages from each child clique into $\psi_{(v,w)}$
  If $C_{(v,w)}$ is not the root clique
    Let vertex v represent the variable that is common to $C_{(v,w)}$ and the parent clique of $C_{(v,w)}$
    Compute message μ_v by marginalizing $\psi_{(v,w)}$ over variable χ_w
    Send message μ_v to parent clique
For clique $C_{(a,b)}$ in $V_{\text{CT}}^{\text{pre}}$
  Set belief $\beta_{(a,b)}$ as $\psi_{(a,b)}^{\text{init}}$ multiplied with messages received from each neighbor of $C_{(a,b)}$
  For each child clique $C_{(x,y)}$
    Let vertex v represent the variable that is common to $C_{(a,b)}$ and $C_{(x,y)}$ (there is exactly one such variable)
    Set $\psi_{(a,b)}$ as $\psi_{(a,b)}^{\text{init}}$ multiplied with messages received from each each neighbor of $C_{(a,b)}$ except $C_{(x,y)}$
    Compute message μ_v by marginalizing $\psi_{(a,b)}$ over variable χ_v
    Send message μ_v to $C_{(x,y)}$
Output: Beliefs of each clique

Executing belief propagation for site i results in computing the marginal probabilities $P(\mathcal{X}_u^i, \mathcal{X}_v^i)$ for each pair of adjacent vertices. The marginal probability $P(\mathcal{X}_u^i)$ for each variable $\mathcal{X}_u$ can be computed by selecting a probability distribution $P(\mathcal{X}_u^i, \mathcal{X}_v^i)$ that contains $\mathcal{X}_u$ in its domain, and marginalizing over the variable $\mathcal{X}_v$. Expected count matrices can be computed using equation 2.15 and equation 2.16. The MLE of model parameters can be computed using equation 2.13 and equation 2.14 where the observed count matrices are substituted with expected count matrices.

2.7 Divide-and-conquer approaches

The large computational cost incurred in searching through tree space for maximum likelihood phylogenetic trees motivated the development of divide-and-conquer methods that compute local phylogenetic trees for small sets of species, and combine local phylogenetic trees into a global phylogenetic tree. The trees that are inferred by the divide-and-conquer approaches described below combine local unrooted phylogenetic trees into a global unrooted phylogenetic tree. The method that is used for constructing local phylogenetic trees is referred to as the base method. Quartet puzzling (QP; Strimmer and von Haeseler (1996)) is a divide-and-conquer method that works as follows. First all possible sets of four species are constructed. For each such quartet of species a quartet with the maximum likelihood is selected. Let $\{a, b|c, d\}$ denote a quartet tree that contains the split {a,b}|{c,d}. The selected quartets are combined (the puzzling step) to construct a global phylogenetic tree. St. John et al. (2003) compare the reconstruction accuracy of QP with neighbor-joining and find that NJ performs better. The relatively poor performance of QP is because quartet trees comprising distantly related species are not reliably estimated.

Erdös et al. (Erdös et al., 1999a,b) designed an efficient quartet-based method called the dyadic closure method that outperforms QP because the dyadic closure method only considers quartets comprising closely related species. The dyadic closure method was subsequently implemented as a family of methods referred to as disc-covering methods (DCM). The DCM (Huson et al. (1999)) partition species on the basis of a threshold graph. Given pairwise distances d over species V a threshold graph $G = (V, E)$ is constructed by adding edges $\{u, v\}$ such that $d(u, v)$ is smaller than a threshold that is selected *a priori*. The most sophisticated DCM is Rec-I-DCM3 (Roshan et al., 2004) which stands for recursive-iterative-DCM3. Roshan et al. (2004) applied Rec-I-DCM3 to search for maximum parsimony (MP) phylogenetic trees. TNT by Goloboff (1999) is a popular method for finding MP phylogenetic trees. Roshan et al. (2004) used TNT as the base-method of Rec-I-DCM3 for inferring local MP phylogenetic trees and report a substantial reduction in compute-time combined with an improvement in reconstruction accuracy when compared with the use of TNT to infer global MP phylogenetic trees.

Adkins designed a Kruskal-like agglomerative clustering algorithm that starts with a forest of singletons, iteratively joins a tree pair, and terminates if the forest is a tree (Adkins, 2010). The tree-pair to join at each iteration is selected as follows. A tree-pair $T_1 = (V_1, E_1)$ and $T_2 = (V_2, E_2)$, and edges $\{u_1, v_1\}$ and $\{u_2, v_2\}$ in E_1 and E_2, respectively, are selected that minimize the length of the interior edge of the quartet $\{u_1, v_1|u_2, v_2\}$, where length of the interior edge is computed as follows. First the trees T_1 and T_2 are disconnected by removing edges $\{u_1, v_1\}$ and $\{u_2, v_2\}$. Subsequently, ancestral sequences are inferred at u_1, v_1, u_2 and v_2. The length of the interior edge is estimated using pairwise distances computed using the ancestral sequences. Given a tree pair $T_1 = (V_1, E_1)$ and $T_2 = (V_2, E_2)$, and edges $\{u_1, v_1\}$ and $\{u_2, v_2\}$ in E_1 and E_2, tree-joining is performed by removing the edges $\{u_1, v_1\}$ and $\{u_2, v_2\}$, adding vertices w_1, w_2 and edges $\{u_1, w_1\}$,$\{v, w_1\}$,$\{u_2, w_2\}$, $\{v_2, w_2\}$, and $\{w_1, w_2\}$. The algorithm terminates if the the forest is a tree. Adkins noted that some of the trees and edges selected are incorrect because the algorithm does not have a global view of how the initial set of observed sequences are related to each other.

Chow-Liu grouping by Choi et al. (2011) improves upon Adkins's algorithm by using a minimum spanning tree to maintain a global view. However, Chow-Liu grouping is a distance-based divide-and-conquer method that uses minimum spanning trees (MSTs) as follows. An MST is computed using pairwise distances. Subsequently, each non- leaf vertex v of the MST is visited, and a local phylogenetic tree is computed over the set $v \cup N(v)$ where $N(v)$ comprises the neighbors of v. The MST is used as a guide tree to join the local phylogenetic trees into a global phylogenetic tree. Huang et al. (2014) demonstrated that Chow-Liu

grouping enables distributed computation of local trees followed by merger of local trees with respect to minimum spanning tree. Chow-Liu grouping is discussed in detail in Chapter 4. In contrast to Adkins's algorithm, Chow-Liu grouping does not perform ancestral state reconstruction and thus avoids parameter estimation that is necessarily required when inferring ancestral states using a Markov model.

In recent work Zhang et al. (2019) presented a minimum spanning tree-based method called incremental tree construction (INC) for building distance-based phylogenetic trees. INC involves the following steps: (i) computing an MST M using pairwise distance estimates, (ii) selecting the first three vertices of the MST that are visited by performing a BFS/DFS starting at any leaf of the MST, (iii) constructing a three-leaf tree T using the vertices selected in step (ii), incrementally growing T by adding vertices in M that are adjacent to vertices in T on the basis of constraints derived from quartet trees constructed using Buneman's four point condition (Buneman, 1971). Buneman's four-point condition is defined wrt to an unrooted tree $T = (V, E)$ as follows. Any four vertices x, y, u, v in V that satisfy the following inequality

$$d_T(x, y) + d_T(u, v) \leq \max \{\{d_T(x, v) + d_T(u, y)\}, \{d_T(x, v) + d_T(u, y)\}\} \tag{2.20}$$

are the leaves of a quartet tree which contains the split $\{x, y\}|\{u, v\}$. INC is similar to Chow-Liu grouping in that a global constraint tree (the MST) is used to guide tree construction. In contrast to Chow-Liu grouping, INC does not allow distributed computation of local trees.

Le et al. (2019) implemented INC, and a related method called INC-ML which uses constraints based on trees provided by ML inference methods such as RAxML in order to incrementally construct a phylogenetic tree. Le et al. (2019) performed a comparative analysis on simulated data and found that the reconstruction accuracy of INC-ML was worse than RAxML but much better than INC.

All the divide-and-conquer methods mentioned above compute unrooted phylogenetic trees. A majority of phylogeny inference software infer unrooted trees. We discuss how unrooted trees are rooted in practice.

2.8 Placing the root on unrooted phylogenetic trees

The phylogenetic trees that are inferred using time-reversible Markov models are unrooted phylogenetic trees. Unrooted phylogenetic trees are less meaningful than rooted phylogenetic trees. Three methods that are commonly used for inferring rooted phylogenetic trees are discussed below. The methods are (i) outgroup-based rooting, (ii) midpoint rooting, and (iii) molecular clock based rooting. The current section ends with a discussion of how molecular clocks can be used to construct time-calibrated phylogenetic tree, *i.e.*, rooted phylogenetic trees with edge lengths that are scaled in units of time.

2.8.1 Outgroup-based rooting

If one has prior knowledge that the species in a subtree $\tau_v = (V_{\tau_v}, E_{\tau_v})$ of an unrooted phylogenetic tree $T = (V_T, E_T)$ are distantly related to species that are not in τ_v then it is reasonable to root the tree along the edge $\{u, v\}$ in $E_T \backslash E_{\tau_v}$. The species in τ_v are referred to as *outgroup*, and the species that are not in the outgroup are referred to as *ingroup*. For instance if the unrooted phylogenetic tree comprised sequences from two species of cats and twenty species of dogs then it seems reasonable to consider the cats as an outgroup.

2.8.2 Midpoint rooting

Given an unrooted phylogenetic tree with edge lengths, let p be a path with the longest sum of edge lengths. Let the end points of p be u and v. Midpoint rooting is performed by selecting an edge and placing the root ρ along the edge such that the distance $d_{\rho u}$ from the root to u equals the distance $d_{\rho v}$ from the root to v.

2.8.3 Molecular clock based rooting

The existence of a molecular clock was proposed based on the empirical observation that protein sequences seemed to evolve at a constant substitution rate (Zukerkandl and Pauling, 1965). The rate at which sequences

evolve is known as *substitution rate*. Strict molecular clocks assume that the substitution rate is identical for each edge of a rooted phylogenetic tree. Relaxed molecular clocks allow substitution rate to vary across the edges of a rooted phylogenetic tree. The discussion in this thesis focuses on strict molecular clocks. Unless specified otherwise the term molecular clock refers to a strict molecular clock.

A molecular clock imposes constraints on the edge lengths of phylogenetic trees. Two types of constraints are considered below that differ with respect to rapidity of molecular evolution relative to the time scale of sampling. Slowly-evolving species, *e.g.,* all living species of the animal kingdom, are modeled as being contemporaneously sampled. The distance from the root to each species in a time-calibrated phylogenetic tree of slowly-evolving species is constrained to be identical. Fast-evolving species such as HIV are modeled as being sampled at distinct time points. The distance from the root to any species in a time-calibrated phylogenetic tree of fast-evolving species is proportional to the sampling time of the species. The edge lengths of a rooted phylogenetic tree are said to be clock-like if root-to-leaf distances satisfy the constraints of a molecular clock. Distance-based methods and model-based methods for rooting phylogenetic trees are described below.

Distance-based methods root an unrooted phylogenetic tree such that the edge lengths of the rooted phylogenetic tree best satisfy the clock-based constraints. Mai et al. (2017) provide a method for rooting phylogenetic trees by minimizing the variance of root-to-leaf distances. Phylogenetic trees of fast-evolving species can be rooted by placing the root such that the sum-of-squared-errors of regressing path length from root-to-leaf on sampling time via ordinary-least-squares (OLS) regression is minimized (Rambaut et al., 2016).

Model-based methods can be used to root an unrooted phylogenetic tree as follows. Given the topology of a rooted phylogenetic tree, edge lengths are optimized under a CT-HMM according to the constraints of the molecular clock. The optimal rooted phylogenetic tree is inferred by selecting the combination of rooted phylogenetic tree topology and constrained edge lengths that maximize the likelihood score (Huelsenbeck et al., 2002; Rambaut, 2000).

The edge lengths of non-calibrated rooted phylogenetic trees are in units of substitution per site. Calibrating a phylogenetic tree involves scaling edge lengths in units of calendar time using substitution rates which are estimated as follows. Substitution rate can be computed by dividing the tree-distance from the ancestor to its living descendants by the divergence time as follows (Kumar and Hedges, 1998). Let $\text{divTime}(u, v)$ be an estimate of the time since species u and v diverged from their most recent common ancestor a. The substitution rate λ is computed as follows:

$$\lambda = \frac{d_{uv}}{2 \times \text{divTime}(u, v)} \tag{2.21}$$

where d_{uv} is the sum of edge lengths on the shortest path from u to v. The factor of two in the denominator of equation 2.21 is present because (i) $d_{uv} = d_{au} + d_{av}$ for tree-distances, and (ii) $d_{au} = d_{av} = \lambda \times \text{divTime}(u, v)$ for clock-like distances. A time-calibrated phylogenetic tree can be constructed by scaling edge lengths using the estimated substitution rate. Probabilistic approaches can also be used for estimating substitution rates (Rambaut and Bromham, 1998). Molecular clocks for fast-evolving species are calibrated using sampling times that correspond to leaf ages. For instance root-to-leaf distances can be regressed on sampling times (Rambaut et al., 2016) via OLS regression. Substitution rate can be computed using the slope of the linear model that is fitted via OLS.

2.8.4 Drawbacks of current methods for rooting phylogenetic trees

Methods that use molecular clocks for rooting fail if substitution rates vary across species (Li, 1993). Midpoint rooting usually gives realistic estimates of the location of the root only if the species under consideration can be split into two sets of pairwise distantly related species. Outgroup-based rooting requires the use of distantly related species. Methods that rely on the selection of distantly related species have the following shortcoming. Sequences that corresponding to distantly related species may share common characters due to independent evolution of the common character during evolutionary history resulting in distantly related species being erroneously close to each other in the inferred phylogenetic tree. This phenomenon is known as

long-branch-attraction (Felsenstein, 1978) and is thought to be a source of error for outgroup-based rooting (Graham et al., 2002).

Molecular clock based rooting gives a realistic estimate of the location of the root if the assumption of a molecular clock is appropriate. The use of molecular clocks has been criticized because of observation of variation in the substitution rates among closely related lineages (Li, 1993).

2.9 Summary of contributions made in thesis

2.9.1 Modeling ancestor-descendant relationships using generally labeled trees

Fast-evolving species that have been sampled at distinct time points may contain ancestor-descendant pairs. Leaf-labeled phylogenetic trees that are commonly used to represent evolutionary relationships do not allow sampled species to have ancestor-descendant relationships. In Kalaghatgi et al. (2016a) we developed a clustering method called family-joining for inferring generally labeled phylogenetic trees that better represent the evolutionary relationships among fast-evolving species using generally labeled phylogenetic trees that place species at non-leaf vertices. Family-joining compared favorably with related methods: sampled ancestors by Gavryushkina et al. (2014) and recursive-grouping and Chow-Liu recursive grouping by Choi et al. (2011). Family-joining was validated on empirical data using HIV-1 sequences that were sampled from individuals from a known transmission chain. The inferred phylogenetic tree was compatible with 9 out of 10 transmission events. Further details about family-joining are provided in Chapter 3.

2.9.2 Conditions under which MSTs share a topological correspondence with phylogenetic trees

Choi et al. (2011) introduced a minimum spanning tree (MST)-based method called CLGrouping, for constructing tree-structured probabilistic graphical models, a statistical framework that is commonly used for inferring phylogenetic trees. While CLGrouping works correctly if there is a unique MST, we observed an indeterminacy in the method in the case where there are multiple MSTs. We demonstrated the indeterminacy of CLGrouping using a synthetic quartet tree and a tree over primate genera. The indeterminacy of CLGrouping can be removed if the input MST shares a topological relationship with the corresponding phylogenetic tree. In Kalaghatgi and Lengauer (2017) we introduced so-called vertex order based MSTs (VMSTs) that are guaranteed to have the desired topological relationship. We related the number of leaves in the VMST to the degree of parallelism that is offered by CLGrouping. We provided polynomial-time algorithms for constructing VMSTs and for selecting a VMST with the optimal number of leaves. Details regarding the indeterminacy of Chow-Liu group, and a rigorous analysis of algorithms for constructing VMSTs are provided in Chapter 4.

2.9.3 Structural expectation-maximization under the general Markov model via a minimum spanning tree backbone

The Markov models that are commonly used in phylogeny inference such as the GTR model are stationary, homogeneous and time-reversible. Stationary models assume that base frequencies do not change over the course of evolutionary history. The GC content of bacterial genomes ranges from 13% to 75% across bacterial species (Agashe and Shankar, 2014) indicating that the assumption of stationarity is unreasonable. The current strategy of searching through tree space for maximum likelihood phylogenetic trees is computationally demanding. Simpler models such as the GTR model are used because they have a small number of free parameters that need to be estimated. The general Markov model (GM; Barry and Hartigan (1987)) is a non-stationary, non-homogeneous, non-reversible Markov that allows for variation in GC content. A method for performing tree-search under the GM model is missing. In Chapter 5 we adapt the structural expectation-maximization framework (Friedman et al., 2002) to perform tree search under the GM model (SEM-GM). SEM-GM is a computationally expensive method. Inspired by the topological correspondence between phylogenetic trees and minimum spanning trees due to Choi et al. (2011) we developed a framework

called MST-backbone for constraining the search through tree space. We applied MST-backbone to improve the scalability of SEM-GM without loss in performance. On simulated data with substantial variation in GC content we demonstrated that the use of stationary models leads to a worse performance when compared to the GM model. We validated our framework on multiple empirical datasets. Our method inferred rooted trees under the GM model for two experimental phylogeny data sets with recall of 0.8. The unrooted topology of the inferred phylogenetic trees appeared to be realistic for a majority of empirical datasets. However the location of the root was not robustly supported. The location of the root was robustly recovered using stationary homogeneous non-reversible Markov models. Details regarding the MST-backbone framework and the validation studies are provided in Chapter 5.

Chapter 3

Modeling ancestor-descendant relationships using generally labeled trees

The work that is presented in this chapter has been published in Kalaghatgi et al. (2016a).

Fast-evolving species that have been sampled at multiple time points may contain ancestor-descendant pairs. The current approach to modeling evolutionary relationships makes use of leaf-labeled phylogenetic trees. Leaf-labeled phylogenetic trees place all the sampled species at the leaves, and do not model direct ancestor-descendant relationships. In this chapter we model evolutionary relationships using so-called generally labeled phylogenetic trees. Generally labeled trees allow sampled species to be placed at non-leaf vertices. We present a clustering method called family joining (FJ) for constructing unrooted generally labeled phylogenetic trees. FJ compares favorably with respect to related methods on simulated data. FJ was validated using HIV-1 *env* gene sequences that were sampled from individuals that were part of a partially known HIV transmission network.

3.1 Current methods for modeling ancestor-descendant relationships

Leaf-labeled phylogenetic trees are widely used to model evolutionary relationships, and are appropriate models of evolutionary relationships among distantly related species such as the group of extant marine mammals that includes manatees, walruses, whales, and dolphins (Foote et al., 2015). Pathogens such as HIV replicate within individuals that are infected with the pathogen. A set of pathogens that are sampled from individuals that are part of a common transmission network may contain ancestor-descendant pathogen pairs. The evolutionary relationships of fast-evolving species such as HIV are better represented using generally labeled phylogenetic trees that allow species to be placed at non-leaf vertices.

To account for ancestor-descendant relationships Jombart et al. (2011) model evolutionary relationships using a directed acyclic graph (DAG) with no hidden vertices. Fully labeled DAGs do not account for unsampled ancestral species. Additionally the DAGs that are used by Jombart et al. (2011) are not necessarily connected. A disconnected graph with no hidden vertex does not fully represent the evolutionary relationships.

Three types of methods are compared in this chapter. The first type is a likelihood-based method called sampled ancestors (Gavryushkina et al., 2014) that performs Bayesian inference over phylogenetic trees via Markov chain Monte Carlo sampling. The second type of method performs agglomerative clustering. The agglomerative clustering methods discussed in this chapter include recursive grouping (RG; Choi et al.

(2011)), neighbor-joining with edge contraction (NJc; Saitou and Nei (1987), Choi et al. (2011)), and family-joining (FJ) which is the method that was developed by Kalaghatgi et al. (2016a). The agglomerative clustering methods construct unrooted generally labeled phylogenetic trees. The final type of method is a supertree method called Chow-Liu grouping that uses RG as the base method for constructing unrooted generally labeled phylogenetic tree.

3.1.1 Sampled ancestor trees

Gavryushkina et al. (2014) provide a method for constructing so-called sampled ancestor (SA) trees that are rooted generally labeled phylogenetic trees with labeled ancestors restricted to having a single child. The restriction on the the number of children of a non-leaf labeled vertex seems unnecessary. The authors infer sampled ancestor trees via Bayesian inference that is performed using Markov chain Monte Carlo (MCMC) sampling. The procedure of sampling phylogenetic trees via MCMC sampling is not applicable to reasonably sized data sets comprising more than a few hundred species.

3.1.2 Agglomerative clustering methods

A common feature of all the agglomerative clustering methods discussed in this chapter is the use of so-called active vertex set V_a that is initialized as the set of observed species. V_a is partitioned iteratively into one of more generally labeled phylogenetic trees which are combined in order to construct a connected generally labeled phylogenetic tree.

Recursive grouping (RG) iteratively partitions closely related vertices in V_a into clusters called families (MacQueen, 1967). The relationships of vertices in a family are modeled as an unrooted generally labeled phylogenetic tree (Choi et al., 2011) on the basis of distances for each vertex pair, and a distance threshold ϵ. Vertices in V_a that are present in a family are removed from V_a. Newly introduced hidden vertices in each phylogenetic tree are added to V_a. This procedure is iterated until a connected unrooted phylogenetic tree is constructed.

3.1.3 Neighbor joining with edge contraction

Neighbor-joining with edge contraction (NJc) constructs a neighbor-joining tree (Saitou and Nei, 1987). Subsequently all edges that are shorter than a small threshold ϵ are contracted in order to construct an unrooted generally labeled phylogenetic tree.

3.1.4 Chow-Liu Recursive grouping

Chow-Liu grouping (CLGrouping) is a minimum spanning tree (MST)-based supertree method that was introduced by Choi et al. (2011). Choi et al. (2011) provided an application of CLGrouping that uses RG as the base method for constructing generally labeled phylogenetic trees. The application is referred to as CLRG. CLRG starts by constructing a minimum spanning tree M over all the labeled vertices. Subsequently for each non-leaf vertex v_i, the vertex set V_i consisting of v_i and its neighbors is constructed and a generally labeled phylogenetic tree T_i over V_i is constructed using RG. The subgraph in M that is induced by V_i is replaced by T_i.

Choi et al. (2011) compared the performance of RG, CLRG, and NJc on simulated data where only the tree topology was varied. In that study, no method clearly outperformed the others.

The work in the current chapter presents a novel agglomerative clustering method called family-joining (FJ) that constructs generally labeled phylogenetic trees. Additionally, we perform a comparative analysis on the basis of a large variety of simulation scenarios. Finally we validate FJ using HIV sequences sampled from individuals that are part of a known HIV transmission network.

3.2 Family joining: a clustering approach for constructing generally labeled phylogenetic trees

First we provide a brief description of the main steps of FJ. Subsequently each step is explained in detail.

3.2.1 Overview of family-joining (FJ)

The family-joining (FJ) method takes as input distances d between each species pair, and a distance threshold ϵ. FJ consists of the two following algorithms. (i) A distance-based algorithm for constructing the topology of an unrooted generally labeled phylogenetic tree $T = (V_T, E_T)$, and (ii) an algorithm for computing edge lengths by regressing weighted path lengths in E_T of each pair of labeled vertices on distances between labeled vertex pairs, via ordinary least squares (OLS) regression.

Tree topologies are inferred using the following agglomeration clustering procedure. A vertex set V_a is initialized with the set of species. At each iteration we select from V_a, the vertex pair i, j that optimizes the neighbor-joining objective, as defined by Saitou and Nei (1987), see equation 3.1. Subsequently, we classify the selected vertex pair i, j as being either parent-child or siblings on the basis of a threshold ϵ, see equation 3.2. If they are found to be siblings we check if there is another vertex that is the parent of both the siblings. If no such vertex is found, a hidden vertex is introduced as the parent of both the siblings. The distance matrix is augmented by adding distances from the newly introduced hidden vertex to each of the other vertices in V_a, obtained using the formula by Studier and Keppler (1988), see equation 3.5. Rows and columns of the distance matrix corresponding to the children are removed, and the procedure is iterated until a connected graph is obtained.

Subsequently, we estimate edge lengths using OLS regression. For efficient calculation of OLS edge lengths we extended the algorithm by Bryant (1997), which was designed for leaf-labeled trees, to generally labeled trees. OLS edge lengths may be negative, which has no biological interpretation. To account for this all edges that are shorter than ϵ, and are incident to a hidden vertex are contracted.

The trees that are constructed by family-joining are unrooted generally labeled phylogenetic trees. In this chapter two vertices are said to be siblings if they are adjacent to a common vertex. What we refer to as siblings is referred to as neighbors by Saitou and Nei (1987) in the context of the neighbor-joining algorithm. Labeled vertices that are adjacent to each other are said to be in a parent-child relationship.

The inference of tree topology is described in Subsection 3.2.2. Edge length estimation is discussed in Subsection 3.2.3. A time complexity analysis of family-joining is performed in Subsection 3.2.4. The statistical consistency of family-joining is discussed in Subsection 3.2.6. Methods for selecting the distance threshold ϵ via model selection is discussed in Subsection 3.3.3.

3.2.2 Inferring tree topology

Given distances d between each pair of labeled vertices $\mathcal{L}_T$, and a distance threshold ϵ, the topology of an unrooted generally labeled phylogenetic tree $T = (V_T = \{\mathcal{L}_T, \mathcal{H}_T\}, E_T)$ is inferred using the algorithm GetTreeTopology. Criteria for selecting an appropriate ϵ are discussed later in Subsection 3.3.3. An overview of GetTreeTopology is provided in Algorithm 2. GetTreeTopology initializes a so-called active vertex set V_{a} with the set of all labeled vertices. $\mathcal{H}_T$ and E_T are initialized as empty sets. GetTreeTopology performs agglomerative clustering where the following actions are performed at each iteration.

A pair of vertices i, j in V_a is selected such that i, j minimize the neighbour-joining criterion (Saitou and Nei, 1987) given below.

$$(n-2)d(i,j) - \sum_{k \neq i} d(i,k) - \sum_{k \neq j} d(j,k) \tag{3.1}$$

where n is the number of vertices in V_a.

Neighbors i and j are classified as parent-child or siblings based on the following quantity.

$$\Delta(i,j) = \sum_{k \neq i,j} \frac{d(j,i) + d(i,k) - d(j,k)}{2(n-2)}$$

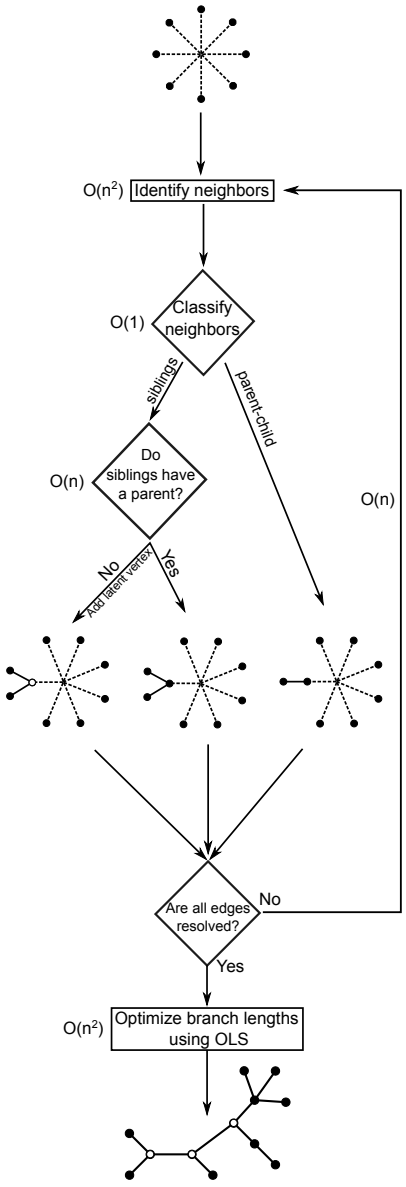


Figure 3.1: An illustration of the family-joining algorithm. The main steps have been labeled with their worst-case time complexity.

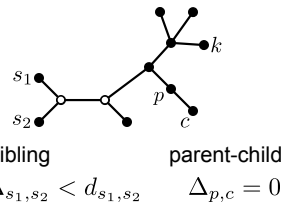


Figure 3.2: Relation types. Siblings s_1 and s_2 are leaves that are adjacent to a common vertex. The leaf c is a child of the vertex p. Filled circles represent labeled vertices, and unfilled circles represent hidden vertices. The measure Δ_{ij} is used to classify neighbors i, j as siblings or parent-child.

The motivation for using $\Delta(i,j)$ is as follows. If distances d are additive in T then it can be easily shown that:

$$\begin{aligned} &\Delta(i,j) = 0 && \text{if } i \text{ is the parent of } j \\ &\Delta(i,j) = d(i,j) && \text{if } j \text{ is the parent of } i \text{ ,} \\ &0 < \Delta(i,j) < d(i,j) && \text{if } i \text{ and } j \text{ are siblings} \end{aligned}$$

The criteria mentioned above are proved as follows. If i is the parent of j then the path from j to any vertex $k \neq i, j$ will visit i (see Figure 3.2). Thus $d(j,k) = d(j,i) + d(i,k)$, which gives $\Delta(i,j) = 0$ and $\Delta(j,i) = d(i,j)$. If i and j are siblings then $d(j,k) = d(j,u) + d(u,k)$ where u is the vertex adjacent to both i and j. Similarly $d(i,k) = d(i,u) + d(u,k)$, which gives $\Delta(i,j) = d(i,u)$. It follows that $0 < \Delta(i,j) < d(i,j)$.

When using distances that are estimated from sequences we use a threshold ϵ for classifying neighbors as parent-child or sibling. Specifically i is the parent of j if $|\Delta(i,j)| < \epsilon$. Neighbors i, j are said to be in a parent-child relationship if

$$\min\{|\Delta(i,j)|, |\Delta(j,i)|\} < \epsilon \tag{3.2}$$

If i and j are in a parent-child relationship, then wlog let i be the parent of j. An edge $\{i,j\}$ is added to E_T. All distances $d(j,m)$ where $m \in V_a \backslash \{j\}$ are removed from d. Subsequently, j is removed from V_a.

If i and j are found to be siblings then we search for another vertex k in V_a that minimizes the following quantity.

$$|d(i,k) + d(k,j) - d(i,j)| \tag{3.3}$$

If $|d(i,k) + d(k,j) - d(i,j)| < 2\epsilon$ then k is the parent of i and j. Edges $\{k,i\}$ and $\{k,j\}$ are added to E_T and distances $d(l,m)$ for each $l \in \{i,j\} \wedge m \in V_a \backslash \{l\}$ are removed from d. Subsequently, i and j are removed from V_a. We tried one additional criterion for checking if there is a vertex k that is the parent of i and j. We computed

$$\min\{|\Delta(k,i)|, |\Delta(k,j)|\}, \tag{3.4}$$

and considered k to be the parent of i and j if $\min\{|\Delta(k,i)|, |\Delta(k,j)|\} < 2\epsilon$. We found that reconstruction accuracy on simulated data was higher when we used the quantity in equation 3.3 as opposed to equation 3.4 (see Supplementary Figure A.4). This is probably because the quantity in equation 3.3 is more robust to noise in the estimates of large distances. We have used the criterion derived from equation 3.3 in the rest of this chapter.

If k is not the parent of i and j, a hidden vertex h is introduced as the parent of i and j. Edges $\{h,i\}$ and $\{h,j\}$ are added to E_T. Vertices i and j are removed from V_a. Vertex h is added to V_a and $\mathcal{H}_T$. Subsequently, distances $d(h,m)$ from vertex h to any other vertex m in $V_\mathrm{a} \backslash \{i,j\}$ are calculated using the following estimate by Studier and Keppler (1988), and added to d.

$$d(h,m) = (d(i,m) + d(j,m) - d(i,j))/2 \qquad \text{for } m \neq i,j \tag{3.5}$$

Subsequently i and j are removed from V_a, and each distance $d(l,m)$ where $l \in \{i,j\} \wedge m \in V_a \backslash \{l\}$ is removed from d.

The agglomeration step described above is repeated until the number of vertices in V_{a} is less than four. After each iteration the number of vertices in V_{a} decreases either by one or two. If V_{a} has reached the size three, we check using equation 3.3 if there are vertices i, j, and k in V_{a} such that k is the parent of both i and j. If we find such vertices, corresponding edges are added. Otherwise a hidden vertex h is introduced and edges $\{h, i\}$, $\{h, j\}$, and $\{h, k\}$ are added to E_T. If V_{a} has reached size two then an edge $\{i, j\}$ is added to E_T, where i, j are the vertices in V_a.

Algorithm 2: GetTreeTopology.

Input: Labeled vertices $\mathcal{L}$, pairwise distances d for each vertex pair in $\mathcal{L}$, and a distance threshold ϵ
Initialize: $\mathcal{H}_T \leftarrow \varnothing$, $E_T = \varnothing$, $V_a \leftarrow \mathcal{L}$
While$|V_a| > 3$ **do**
  Pick vertices i, j from V_a that minimize equation 3.1
  Identify relationship between i, j using equation 3.2
  If i, j are in parent-child relationship **then**
    Let j be the child
    Add edge $\{i, j\}$ to E_T
    Remove j from V_a
    Remove distances $d(j, m)$ from d for each $m \in V_a \backslash \{j\}$
  Else
    Remove i and j from V_a
    Pick a k from V_a that minimizes equation 3.3
    If i and j are children of k **then**
      Add edges $\{i, k\}$ and $\{j, k\}$ to E_T
    Else
      Introduce vertex h, add h to $\mathcal{H}_T$, and add h to V_a
      Add edges $\{i, h\}$and $\{j, h\}$to E_T
      Get distances $d(h, m)$ for each $m \in V_a \backslash \{i, j, h\}$ using equation 3.5 , and add the distances to d
    Remove distances $d(l, m)$ from d for each $l \in \{i, j\} \wedge m \in V_a \backslash \{l\}$
If $|V_a| = 2$ **then**
    $i, j \leftarrow V_a$
    Add edge $\{i, j\}$ to E_T
Else
    Pick i, j, k from V_a that minimize equation 3.3
    If i and j are children of k **then**
      Add edges $\{i, k\}$ and $\{j, k\}$ to E_T
    Else
      Introduce vertex h, and add h to $\mathcal{H}_T$
      Add edges $\{i, h\}$,$\{j, h\}$, and $\{k, h\}$ to E_T
Output: $T = (V_T = \{\mathcal{L}_T, \mathcal{H}_T\}, E_T)$

3.2.3 Estimation of edge lengths

Edge lengths $\mathbf{t} = \{t_e : e \in E_T\}$ of $T = (V_T, E_T)$ are estimated by OLS. This is done by solving $\mathbf{A}\mathbf{t}^c = \mathbf{d}^c$ where $\mathbf{d}^c$ is the column vector

$$\mathbf{d}^c = \begin{bmatrix} d(1,2) \\ d(1,3) \\ \vdots \\ d(n-2,n) \\ d(n-1,n) \end{bmatrix}$$

containing entries $d(i,j)$ such that $i < j$. $\mathbf{A}$ is the edge incidence matrix of T and is constructed as follows. If the m^{th} entry of $\mathbf{d}^c$ is d_{ij}, then

$$a_{me} = \begin{cases} 1 & \text{if the path from } i \text{ to } j \text{ contains } e \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

$\mathbf{A}$ has the dimension $n(n-1)/2 \times |E_T|$ where $|E_T|$ is the number of edges in T, n is the number of labeled vertices, and $\mathbf{t}^c$ is the column vector of edge lengths that we wish to estimate.

The OLS estimate of edge lengths is given by

$$\mathbf{t}^c = (\mathbf{A}^t\mathbf{A})^{-1}\mathbf{A}^t\mathbf{d}^c. \tag{3.7}$$

We do not make the assumption that distances are additive for the estimation of OLS edge lengths. There is a $O(n^2)$ algorithm for computing the OLS edge lengths (Bryant, 1997) for leaf-labeled trees. We show that this algorithm extends to generally labeled trees. The main steps involved in this computation are computing first $\mathbf{A}^t\mathbf{d}^c$ and then $(\mathbf{A}^t\mathbf{A})^{-1}\mathbf{A}^t\mathbf{d}^c$. We describe both of these steps below.

Computing $\mathbf{A}^t\mathbf{d}^c$

The i^{th} entry of $\mathbf{A}^t\mathbf{d}^c$, $\delta_i^t\mathbf{d}^c$, is the sum of all distances $d(a,b)$ such that $a \in A_i$ and $b \in B_i$ where $A_i|B_i$ is the split that is induced by edge e_i. δ_i is the i^{th} column of $\mathbf{A}$. Edges are visited in order of increasing distance from leaves for efficient computation of $\mathbf{A}^t\mathbf{d}^c$. The distance $d^{\mathcal{L}_T}(e_i)$ of an edge e_i from leaves $\mathcal{L}_T$ is defined below.

$$d^{\mathcal{L}_T}(e_i) = \underset{v \in e_i, l \in \mathcal{L}_T}{\operatorname{argmin}} \; p_T^u(v,l)$$

where $p_T^u(v,l)$ is the unweighted path length of the path in T from v to l.

We first compute $\delta_i^t\mathbf{d}^c$ for every terminal edge e_i as follows.

$$\delta_i^t\mathbf{d}^c = \sum_{j, j \neq i} d(i,j) \tag{3.8}$$

Next we compute $\delta_i^t\mathbf{d}^c$ for every internal edge e_i which is visited in the order of increasing distance from leaves. Consider the non-leaf vertex α such that there is only edge e_i that is incident to α such that $\delta_i^t\mathbf{d}^c$ has not been calculated. Consider the list E_j of edges $e_{j_1}, \ldots, e_{j_m}$ such that e_i and each edge in E_j are incident to a common vertex.

Let $C_i|\bar{C}_i$ be the split that is induced by edge e_i such that α is closer to C_i in comparison to $\bar{C}_i$. Similarly C_{j_k} is the side of the split induced by e_{j_k} that is closer to α.

$\delta_i^t\mathbf{d}^c$ is computed as follows depending on whether or not α is labeled:

Case 1: Vertex α is not labeled

$$\begin{aligned} \delta_i^t\mathbf{d}^c &= \sum_k \sum_{a \in C_{j_k}, b \in C_i} d(a,b) \\ &= \sum_k \delta_{jk}^t\mathbf{d}^c - 2\sum_{k<l} \sum_{a \in C_{j_k}, b \in C_{jl}} d(a,b) \end{aligned} \tag{3.9}$$

Case 2: Vertex α is labeled.

$$\begin{aligned} \delta_i^t\mathbf{d}^c &= \sum_k \sum_{a \in C_{j_k}, b \in C_i} d(a,b) + \sum_{b \in C_i} d(\alpha,b) \\ &= \sum_k \delta_{jk}^t\mathbf{d}^c - 2\sum_{k<l} \sum_{a \in C_{j_k}, b \in C_{jl}} d(a,b) - \sum_k \sum_{b \in C_{jk}} d(\alpha,b) + \sum_{b \in C_i} d(\alpha,b) \end{aligned} \tag{3.10}$$

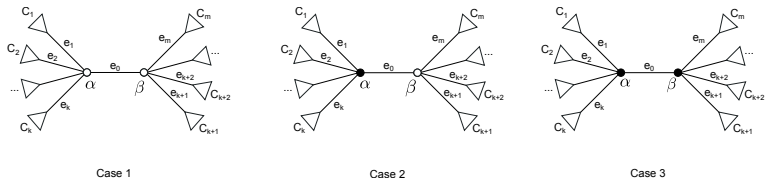


Figure 3.3: The three cases for the internal edge e_0. Case 1: Both α and β are not labeled. Case 2: Only α is labeled. Case 3: Both α and β are labeled. The triangles represent subtrees.

Computing $(\mathbf{A}^t\mathbf{A})^{-1}(\mathbf{A}^t\mathbf{d}^c)$

The current section lists the formulae by Bryant (1997) which allows the computation of the OLS estimate of edge lengths in closed form. Edges can be visited in any order to facilitate the computation of edge lengths. First we compute the edge length for an internal edge.

Consider the internal edge $e_0 = \{\alpha, \beta\}$ shown in Figure 3.3 Case 1 such that edges $e_1, e_2, \ldots, e_k$ are incident to α but are not incident to β, and edges $e_{k+1}, e_{k+2}, \ldots, e_m$ are incident to β but are not incident to α. Let $\mathcal{L}_\alpha|\mathcal{L}_\beta$ be the split in T that is induced by $\{\alpha, \beta\}$ such that the side $\mathcal{L}_\alpha$ is closer to α in comparison to β. Let n_α and n_β be the sizes of $\mathcal{L}_\alpha$ and $\mathcal{L}_\beta$, respectively.

For each edge e_i define $W_i = \sum_{x\in A_i, y\in B_i} p_{xy}$ where A_i and B_i are the sides of the split induced by edge e_i. The notation p_{xy} is used instead of $p_T^w(x, y)$ to denote the weighted path length of the path from x to y where edge lengths are determined by OLS. It turns out that $W_i = \delta_i^t \mathbf{d}^c$.

For each edge e_i such that $1 \le i \le k$, let C_i be the side of the split induced by e_i that is closer to α in comparison to β. For each edge e_i such that $k+1 \le i \le m$, let C_i be the side of the split induced by e_i that is closer to β in comparison to α. Let n_i be the cardinality of C_i. Define

$$Y_i = \begin{cases} \sum_{x\in C_i} p_{\alpha x}, & \text{if } 1 \le i \le k \\ \sum_{x\in C_i} p_{\beta x}, & \text{if } k+1 \le i \le m \end{cases}$$

For the case where neither α nor β are labeled Bryant (1997) showed that

$$\underline{W} = (nI - 2N)\underline{Y} + NU\underline{Y} + t_{e_0} N\underline{v}$$

where N is the $m \times m$ diagonal matrix with $(n_1, n_2, \ldots, n_m)$ on the diagonal, I is the identity matrix, $\underline{Y} = (Y_1, Y_2, \ldots, Y_m)^T$, U is the $m \times m$ matrix of ones, $\underline{v}$ is the vector with n_β in positions 1 to k followed by n_α in positions $k+1$ to m, $\underline{W} = (W_1, W_2, \ldots, W_m)^T$, n is the total number of labeled vertices, and t_{e_0} is the edge length of the edge e_0 .

Similarly for the internal edge e_0

$$W_0 = \underline{v}^T\underline{Y} + n_\alpha n_\beta t_{e_0}$$

Letting $X = (nN^{-1} - 2I + U)$ and substituting $\underline{Y}$ gives the following estimate of edge length t_{e_0}.

$$t_{e_0} = \frac{W_0 - \underline{v}^T X^{-1} N^{-1} \underline{W}}{n_\alpha n_\beta - \underline{v}^T X^{-1} \underline{v}} \tag{3.11}$$

For cases where only α is labeled, and both α and β are labeled, respectively, the derivation of the above mentioned equations is similar to that described in Bryant (1997) and is provided in Appendix A.1.1.

The formula, equation 3.11, for edge length is valid only if X^{-1} exists. Bryant (1997) showed that X is invertible as long as there is at most one zero on the diagonal of the matrix $(nN^{-1} - 2I)$. The i^{th} diagonal element is zero if $n_i/n = 2$ which occurs if there is an edge where both parts of the split have equal size. Even in generally labeled trees there can be at most one such edge.

There are two cases to consider for terminal edges depending on whether or not α is labeled (see Figure 3.4). In both cases the derivation of the edge length formula is similar to what has been described for internal edges and is presented in Appendix A.1.2.

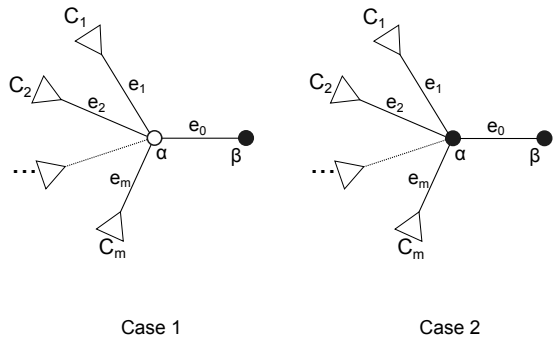


Figure 3.4: The two cases for the terminal edge e_0. α is not labeled in case 1, and is labeled in case 2. The triangles represent subtrees.

OLS edge lengths may be negative which has no biological interpretation. After estimating the edge lengths all edges that are shorter than ϵ and are incident to a hidden vertex are contracted. The length of every edge that has a negative length is set to 10^{-7}. 10^{-7} is smaller than the smallest non-zero distance estimate computed in any of the simulation scenarios.

3.2.4 Time-complexity analysis of family-joining

At first glance it appears that the neighbor identification step requires $\Omega(n^3)$ time. This can be reduced to $O(n^2)$ with the observation that the neighbor-joining objective can be reformulated as follows (Studier and Keppler, 1988):

$$\begin{aligned} &(n-2)d(i,j) - R_i - R_j \\ &\text{where } R_i = \sum_{k \neq i} d(i,k) \end{aligned} \tag{3.12}$$

From equation 3.12 it is evident that initializing each row sum R_i with the original distances takes $O(n)$ time. Updating each R_i after each agglomeration step is done by subtracting distances from children and, if applicable, adding distances to the newly introduced hidden vertices. Thus the process of updating each R_i takes $O(1)$ time. Additionally, storing all the R_i in memory requires $O(n)$ space which incurs very little memory overhead compared to the $O(n^2)$ space required to store all the pairwise distances. If all distances and row sums are stored in memory then identifying the neighbors takes $O(n^2)$ time. Note that Δ_{ij} can also be reformulated for faster computation as follows.

$$\begin{aligned} \Delta(i,j) &= \sum_{k \neq i,j} \frac{d(j,i) + d(i,k) - d(j,k)}{2(n-2)} \\ &= \frac{d(j,i)}{2} + \frac{(\sum_{k \neq i,j} d(i,k)) - (\sum_{k \neq i,j} d(j,k))}{2(n-2)} \\ &= \frac{d(j,i)}{2} + \frac{(d(i,j) + \sum_{k \neq i,j} d(i,k)) - (d(j,i) + \sum_{k \neq i,j} d(j,k))}{2(n-2)} \\ &= \frac{d(j,i)}{2} + \frac{(\sum_{k \neq i} d(i,k)) - (\sum_{k \neq j} d(j,k))}{2(n-2)} \\ &= \frac{d(j,i)}{2} + \frac{R_i - R_j}{2(n-2)}. \end{aligned}$$

Thus, once the neighbors $\{i,j\}$ have been identified, it takes $O(1)$ time to compute both $\Delta(i,j)$ and $\Delta(j,i)$. It takes $O(n)$ time to find the vertex k which minimizes $|d(k,i) + d(k,j) - d(i,j)|$.

The worst-case time-complexity of GetTreeTopology is $O(n^3)$. The time-complexities associated with the main steps of GetTreeTopology are shown in Figure 3.1.

The worst-case time-complexity of the procedure for estimating edge lengths via OLS regression is given below. The procedure involved two main steps: (i) computation of $\mathbf{A}^t\mathbf{d}^c$, and (ii) computation of edge lengths in closed form.

Computing $\mathbf{A}^t\mathbf{d}^c$ involved summation of entries of the distance vector (see equation 3.8). Since each element of the distance vector is summed over just once, $\mathbf{A}^t\mathbf{d}^c$ is computed in $O(n^2)$ time. Given $\mathbf{A}^t\mathbf{d}^c$, each edge length can be calculated in $O(n)$ time (Bryant, 1997). Since there are $O(n)$ edges the worst-case time-complexity of computing OLS edge lengths is $O(n^2)$.

Thus, the worst-case time-complexity of FJ is $O(n^3) + O(n^2) = O(n^3)$.

3.2.5 Vertex augmentation procedure to construct leaf-labeled phylogenetic trees

We made use of software for simulating sequences under the GTR + Γ model (Seq-Gen; Rambaut and Grassly (1997)), and computing likelihood scores under the GTR + Γ model (RAxML; Stamatakis (2014)). The software mentioned above is not designed to handle generally labeled phylogenetic trees. In order to use Seq-Gen and RAxML we use the following vertex augmentation procedure that converts generally labeled phylogenetic trees to leaf-labeled phylogenetic trees. Leaf-to-leaf distances are only slightly perturbed subsequent to the application of the vertex augmentation procedure.

Let a generally labeled tree $T_g = (V_{T_g} = \{\mathcal{L}_{T_g}, \mathcal{H}_{T_g}\}, E_{T_g})$ with hidden vertices that have degree greater than three be given. Let $\mathbf{t}_g$ be the set of edge lengths of T_g. The desired leaf-labeled tree $T_l = (V_{T_l} = \{\mathcal{L}_{T_l}, \mathcal{H}_{T_l}\}, E_{T_l})$ with edge length $\mathbf{t}_l$ is constructed as follows. $\mathcal{L}_{T_l}, \mathcal{H}_{T_l}, E_{T_l}$, and $\mathbf{t}_l$ are initialized as $\mathcal{L}_{T_g}, \mathcal{H}_{T_g}, E_{T_g}$ and $\mathbf{t}_g$, respectively.

If there is a labeled vertex l in T_l with degree greater than one then (i) a new hidden vertex h is added to $\mathcal{H}_{T_l}$, (ii) edge $\{h, m\}$ is added to E_{T_l} for each $\{h, l\}$ in E_{T_l}, and $\{h, l\}$ is removed from E_{T_l}, (iii) edge length $t_{\{h,m\}} = t_{\{l,m\}}$ is added to $\mathbf{t}_l$ for each $t_{\{l,m\}}$ in $\mathbf{t}_l$, and $t_{\{l,m\}}$ is removed from $\mathbf{t}_l$, (iv) edge $\{h, l\}$ is added to E_T, and (v) edge length $t_{\{h,l\}} = \epsilon_{\text{small}}$ is added to $\mathbf{t}_l$.

If there is a hidden vertex h_m in T_l with degree greater than three then (i) a new hidden vertex h_n is added to $\mathcal{H}_{T_l}$, (ii) vertices i, j that are adjacent to h_m are selected at random, (iii) edges $\{i, h_m\}$ and $\{j, h_m\}$ are removed from E_{T_l} and edges $\{i, h_n\}$ and $\{j, h_n\}$ are added to E_{T_l}, (iii) edge lengths $t_{\{i,h_n\}} = t_{\{i,h_m\}}$ and $t_{\{j,h_n\}} = t_{\{j,h_m\}}$ are added to $\mathbf{t}_l$, and edge lengths $t_{\{i,h_m\}}$ and $t_{\{j,h_m\}}$ are removed from $\mathbf{t}_l$, (iv) edge $\{h_m, h_n\}$ is added to E_{T_l}, and (v) edge length $t_{\{h_n,h_m\}} = \epsilon_{\text{small}}$ is added to $\mathbf{t}_l$.

The vertex augmentation operations mentioned above are performed iteratively until there is no labeled vertex in T_l that is not a leaf, and there is no hidden vertex in T_l with degree greater than three.

If ϵ_{small} is set to zero then leaf-to-leaf distances are unchanged subsequent to the vertex augmentation operations. However RAxML requires does not allow the use of edges with length zero. We set ϵ_{small} to a small value of 10^{-7}.

3.2.6 Statistical consistency

We establish the statistical consistency of FJ by applying it to distances d that are additive in a generally labeled phylogenetic tree T such that T may contain hidden vertices with degree greater than three. We assume that all edge lengths of T are strictly greater than zero.

Theorem 1. *Given distances d that are additive in an unrooted generally labeled phylogenetic tree T, and any ϵ that is smaller than the smallest entry in d. Let T_{FJ} be the generally labeled tree that is constructed by applying FJ to distances d, and threshold ϵ. T_{FJ} equals T.*

Proof. We are given $T = (V_T = \{\mathcal{L}_T, \mathcal{H}_T\}, E_T)$, distances d that are additive in T, and a threshold ϵ.

Let $\mathcal{T}_{\max}$ be the set of all trees such that for each tree $T_0 = (V_{T_0} = \{\mathcal{L}_{T_0}, \mathcal{H}_{T_0}\}, E_T)$ in $\mathcal{T}_{\max}$. (i) The leaf-set $\mathcal{L}_{T_0}$equals $\mathcal{L}_T$, (ii) The degree of each hidden vertex in T_0 equals three, and (iii) d is the additive in T_0. Some of the edge lengths of T_0 may be zero.

Table 3.1: Simulated data sets were constructed by varying either the tree type, fraction of hidden vertices, type of contracted edge, number of labeled vertices, sequence length or edge length. All settings that were considered for each parameter are shown below. Default parameter settings are indicated using (d).

Tree type		balanced	random (d)	caterpillar	
Fraction of hidden vertices	0.5	0.37	0.25 (d)	0.12	0
Type of contracted edge	leaf/hidden	labeled/hidden	any/hidden (d)	hidden/hidden	
Average edge length	0.001	0.004	0.016	0.064	0.256
Number of labeled vertices	20	40	80	160 (d)	320
Sequence length	250	500	1000 (d)	2000	4000

Given any T_0 in $\mathcal{T}_{\max}$, each split in T is a split in T_0. If this were not true then there would be distinct splits $A|B$ in T, and $A_0|B_0$ in T_0, and four labeled vertices i, j, k and l such that (i) i, j would be in A, (ii) k, l would be in B, (iii) i, k would be in A_0, and (iv) j, l would be in B_0.

Applying Buneman's 4-point condition (see equation 2.20) would result in the following contradictory inequalities:

$$d(i,j) + d(k,l) < d(i,k) + d(j,l) \text{ for } T$$
$$d(i,j) + d(k,l) \geq d(i,k) + d(j,l) \text{ for } T_0$$

The inequality is strict for T as all edge lengths in T are greater than zero.

Thus any tree in $\mathcal{T}_{\max}$ can be constructed using the vertex augmentation operation described in Subsection 3.2.5 with ϵ_{small} set to zero.

Applying the neighbor-joining algorithm using distances in d yields a leaf-labeled tree T_{NJ} such that each hidden vertex in T_{NJ} has degree three. T_{NJ} belongs to $\mathcal{T}_{\max}$ because d is additive in T_{NJ}. It follows that neighbors in T_{NJ} are either parent-child or siblings in T. Since d is additive any ϵ that is smaller than the smallest entry in d can be used for correctly classifying neighbors as parent-child or siblings.

It follows that each iteration of the topology construction algorithm of FJ correctly adds parent-child, and sibling edges. Thus the topology of the tree T_{FJ} is identical to the topology of T. □

3.3 Comparative analysis on simulated data

3.3.1 Simulation scenarios

Simulated sequences were generated by evolving sequences along the edges of generally labeled trees. The simulation scenarios that were considered in this study are described below.

Simulated data sets were constructed by varying either the tree type, fraction of vertices that are hidden, type of contracted edge, number of labeled vertices, sequence length or edge length. Each of these parameters is described in detail below. An overview of the parameter settings is provided in Table 3.1.

Three types of trees were generated: balanced, caterpillar and random. We chose caterpillar trees because it has been shown that the accuracy of the neighbor identification step (see equation 3.1), which forms a part of FJ, is inversely related to tree diameter (St. John et al., 2003). Balanced trees are leaf-labeled phylogenetics trees with minimum diameter. A random tree $T = (V_T = \{\mathcal{L}_T, \mathcal{H}_T\}, E_T)$ was generated as follows. (i) An active vertex set V_a was initialized with the labeled vertices $\mathcal{L}_T$. Subsequently, the following steps were performed iteratively until a connected tree was constructed. (i) A vertex pair $\{i, j\}$ was selected at random and removed from V_a, (ii) a new hidden vertex h was introduced and added to V_a and V_T, and (iii) edges $\{h, i\}$, and $\{h, j\}$ were added to E_T. The trees that are generated are leaf-labeled phylogenetic trees such that each the degree of each hidden vertex was three.

The fraction of hidden vertices ranges from zero to $(n-2)/(2n-2)$ where n is the number of labeled vertices. We simulated generally labeled trees by varying the fraction of hidden vertices over this range in four equal steps.

Generally labeled phylogenetic trees with the desired proportion of labeled vertices were constructed by contracting the edges of leaf-labeled phylogenetic trees with degree-3 hidden vertices. Depending on the type of simulation experiment, the following edges were contracted: leaf/hidden, labeled/hidden, hidden/hidden, and any/hidden.

Given the topology of generally labeled trees, edge lengths were drawn from the uniform distribution $U(1, 100)$, and scaled such that the expected edge length was equal to a pre-selected edge length average. The following edge length averages were generated: 0.001, 0.004, 0.016, 0.064, and 0.256 subs/site. Sequence evolution was performed as follows. A vertex was randomly selected as the root and sequences were evolved along the edge according to a GTR+Γ model of substitution (Lanave et al., 1984). The parameters of the GTR model were set using estimates from a real data set (Waddell and Steel, 1997). The parameters shape and scale of the Γ model were set to 1 which resulted in a moderate variation of substitution rate across sites. Seq-Gen was used for simulating sequence evolution (Rambaut and Grassly, 1997). Sequence lengths took values of 250, 500, 1000, 2000, and 4000 bp. The number of labeled vertices (species) took values of 20, 40, 80, 160, and 320. As Seq-Gen only takes leaf-labeled trees as input, the simulated generally labeled trees were converted to leaf-labeled trees using the vertex augmentation procedure described in Subsection 3.2.5 with ϵ_{small} set to 10^{-7}.

Simulation scenarios were defined by varying each parameter over its range while keeping the remaining parameters fixed at their default setting. This procedure results in 22 simulation scenarios. The default settings for each parameter are described below.

For the categorical parameters tree type and contracted edge type, the respective default settings were random and any/hidden. These settings were selected as the defaults as they do not restrict the generation of generally labeled trees. The continuous parameter, fraction of vertices that are hidden, which has a bounded range, the midpoint was considered as the default value. For the following continuous parameters with no upper bound: number of labeled vertices, sequence length, and average edge length, we selected the range and default settings such that the trend in performance over each parameter range was apparent. The default setting for the number of labeled vertices was 160, for the sequence length it was 1000 bp, for the average branch length was 0.016 subs/site.

100 trees and corresponding sequences were simulated for each setting of parameter values.

We provided sampling times for SA which constructs rooted trees under a molecular clock. In order to provide sampling times we rooted simulated trees along edges that were selected at random. We defined the sampling time of a labeled vertex as the weighted path length from the root to the labeled vertex. Note that this method of defining sampling times is equivalent to assuming a strict molecular clock with a clock rate of 1.0. When substitution rates (subs./site/time) follow a strict molecular clock, the distance from the root to each labeled vertex is proportional to the time elapsed since divergence from the root. SA recovers the correct clock rate of 1.0 under the strict molecular clock model in all scenarios except two where the average branch length is very small (0.001 and 0.004; see Supplementary Figure A.3)

3.3.2 Maximum likelihood distances

The estimated distances that were used in this study are maximum likelihood (ML) distances that were estimated under the GTR +Γ model using RAxMLv8.2.8. The procedure for computing ML distances is described below. First a maximum parsimony tree was constructed using stepwise addition and the parameters of the substitution model GTR+Γ were optimized. The optimized substitution model was used to compute maximum likelihood distances for all species pairs as follows.

Given parameters of a GTR + Γ model, and sequences for species l_1 and l_2. Let T_{12} be the two-leaf phylogenetic tree where l_1 and l_2 are the leaves of the tree. The maximum likelihood distance is the sum of edge lengths where edge lengths are optimized via maximum likelihood.

3.3.3 Model selection

Values of ϵ are inversely related to the number of hidden vertices and thus inversely related to model complexity. We performed model selection using three estimates of test error, Akaike information criterion

(AIC), Bayesian information criterion (BIC), and cross-validation error.

Likelihood was computed using RAxML as follows. RAxML was provided with a tree topology and edge lengths, and a GTR + Γ model was optimized such that tree topology and edge lengths were fixed. Because RAxML is not designed for generally labeled trees we constructed leaf-labeled trees using the vertex augmentation operations described in Subsection 3.2.6 with ϵ_{small} set to 10^{-7}.

For computing cross-validation error the original sequence alignment with k columns was partitioned into B validation alignments by randomly sampling k/B columns without replacement. For each validation alignment, the corresponding training alignment was constructed using the complimentary set of $k - k/B$ alignment columns. This procedure was repeated R times, giving RB training and validation alignments in total. ML distances were computed for all training and validation alignments. For a fixed value of ϵ, FJ trees were constructed for each training distance matrix. We set R to 10 and tried two values for B, *i.e.*, 3 and 5. Test error was computed as the residual sum of squares between the fitted distances (weighted path length on the tree) and the corresponding distances computed from the validation alignment. We then found the ϵ that minimized expected test error as this would yield the most generalizable model.

$$\arg\min_{\epsilon} \sum_{b=1}^{B} \sum_{i,j} \underbrace{(d_{T(\epsilon,b)}(i,j)}_{\text{distance in fitted tree}} - \underbrace{d_{V(b)}(i,j))^2}_{\text{distance in validation set}}$$

where $T(\epsilon, b)$ is the tree constructed at threshold ϵ using distances from the b^{th} training alignment and $V(b)$ is the b^{th} validation alignment. Model selection was performed by identifying the value of ϵ that minimizes the estimate of test error.

3.3.4 Performance metrics

Reconstruction accuracy was quantified using precision and recall as defined below.

$$\begin{aligned} \text{Pr}_S(T, \hat{T}) &= \frac{|\mathcal{S}_{\text{all}}(T) \cap \mathcal{S}_{\text{all}}(\hat{T})|}{|\mathcal{S}_{\text{all}}(\hat{T})|}, \text{ and} \\ \text{Re}_S(T, \hat{T}) &= \frac{|\mathcal{S}_{\text{all}}(T) \cap \mathcal{S}_{\text{all}}(\hat{T})|}{|\mathcal{S}_{\text{all}}(T)|}, \end{aligned}$$

where $\mathcal{S}_{\text{all}}(T)$ and $\mathcal{S}_{\text{all}}(\hat{T})$ are the set of all splits in the simulated tree T and the reconstructed tree $\hat{T}$, respectively. Note that $\mathcal{S}_{\text{all}}(T)$ contains the split of every edge in T, including the terminal edges. Precision and recall range from zero to one. Precision is equal to one only if all the splits in the reconstructed tree are present in the simulated tree. Similarly, recall is equal to one only if all the splits in the simulated tree are present in the reconstructed tree. Note that we do not report Robinson-Foulds (RF) distance in this chapter since the RF distance would be biased against methods that do not allow polytomies (hidden vertex with degree greater than three). The *Robinson-Foulds distance* ($\text{RF}_S(T, \hat{T})$) is computed as the fraction of unique splits that are present in one tree and not the other.

$$\text{RF}_S(T, \hat{T}) = 1 - \frac{|\mathcal{S}_{\text{all}}(T) \cap \mathcal{S}_{\text{all}}(\hat{T})|}{|\mathcal{S}_{\text{all}}(T) \cup \mathcal{S}_{\text{all}}(\hat{T})|}$$

Each of the reconstruction methods that we tested can achieve the highest and the lowest possible value of recall. Among the reconstruction methods that were compared, only SA can not achieve a precision of one if the simulated tree contains polytomies. We feel that both precision and recall are important measures of reconstruction accuracy.

3.3.5 Implementation details

We used the sampled ancestors package (Gavryushkina et al., 2014) of BEASTv2.3.0 (Drummond et al., 2012). The following models were considered: the GTR model for substitution, the four-category Γ model

Table 3.2: Methods with the highest precision. F, N, R, C, and S stand for FJ-BIC, NJc-BIC, RG-BIC, CLRG-BIC, and SA, respectively.The default setting for each simulation parameter is indicated with (d).

Tree type		Balanced	Random (d)	caterpillar	
		F	F	C	
Type of contracted edge	leaf/hidden	labeled/hidden	any/hidden (d)	hidden/hidden	
	F,N	F	F	R	
Fraction of hidden vertices	0.5	0.37	0.25 (d)	0.12	0
	N	N,C	F	F	C
Average branch length	0.001	0.004	0.016	0.064	0.256
	C	F	F	F	C
Number of labeled vertices	20	40	80	160(d)	320
	F	F	F	F	F
Sequence length	250	500	1000(d)	2000	4000
	F,C	F	F	F,N,C	F,N,C

for rate heterogeneity across sites, the strict molecular clock model and the fossilized birth death model for generating trees. Uniform priors were set for all model parameters. For all datasets, 10^8 states were visited using Markov chain Monte Carlo (MCMC) and every 10^5 state was sampled. The first 5% of the sampled states were discarded as burn-in and the effective sample size (ESS) was computed for all model parameters using the R package CODA (Plummer et al., 2006). ESS were found to be greater than 200 for all parameters across all the MCMC chains indicating that the chains were sufficiently long. The trees that are produced by BEAST are rooted and contain the maximum number of hidden vertices. The sampled trees were post-processed by unrooting them and contracting all terminal edges of length zero. We reported the average precision and recall of the post-processed sampled trees from the true tree.

RG and CLRG require the setting of two thresholds, ϵ_s and ϵ_l. ϵ_s is used for performing the relationship test. RG and CLRG additionally contract branches that are smaller than this threshold. We optimized ϵ_s using BIC. The second threshold, ϵ_l is used to filter out large distances and only distances below this threshold are used when performing the relationship test. We set ϵ_l to a large value of 0.5.

The distance threshold ϵ for NJc was selected using BIC.

3.3.6 Results

We present the results of applying FJ-BIC, NJc-BIC, RG-BIC, CLRG-BIC and SA to all simulated data sets. For methods which have the suffix BIC, we performed threshold selection by minimizing Bayesian information criterion (BIC). For FJ, we also tested FJ-AIC and FJ-CV which optimized Akaike information criterion (AIC), and cross-validation error (CV), respectively. As FJ-AIC and FJ-CV never performed better than FJ-BIC in any simulation scenario we do not show the results in the current chapter. The results for FJ-AIC and FJ-CV can be found in Supplementary Figure A.4. A change in precision or recall is considered to be statistically significant if the corresponding Welch's t-test has a p-value that is smaller than 0.01. A method is said to have the highest precision or recall if no other method has significantly higher precision or recall, respectively.

Tree type

FJ-BIC and NJc-BIC had significantly higher precision and recall on balanced trees than on caterpillar trees. This behavior is expected, since the accuracy of the step of FJ, in which neighbors are identified, is inversely related to tree diameter (St. John et al., 2003). Even on caterpillar trees, which have large diameters, FJ-BIC and NJc-BIC have moderately large (median) precision/recall values of 0.79/0.81 and 0.76/0.87 respectively (see Figure 3.5A). RG-BIC performs poorly on caterpillar trees in comparison to balanced trees, which is in agreement with previous work (Choi et al., 2011). In contrast, CLRG-BIC

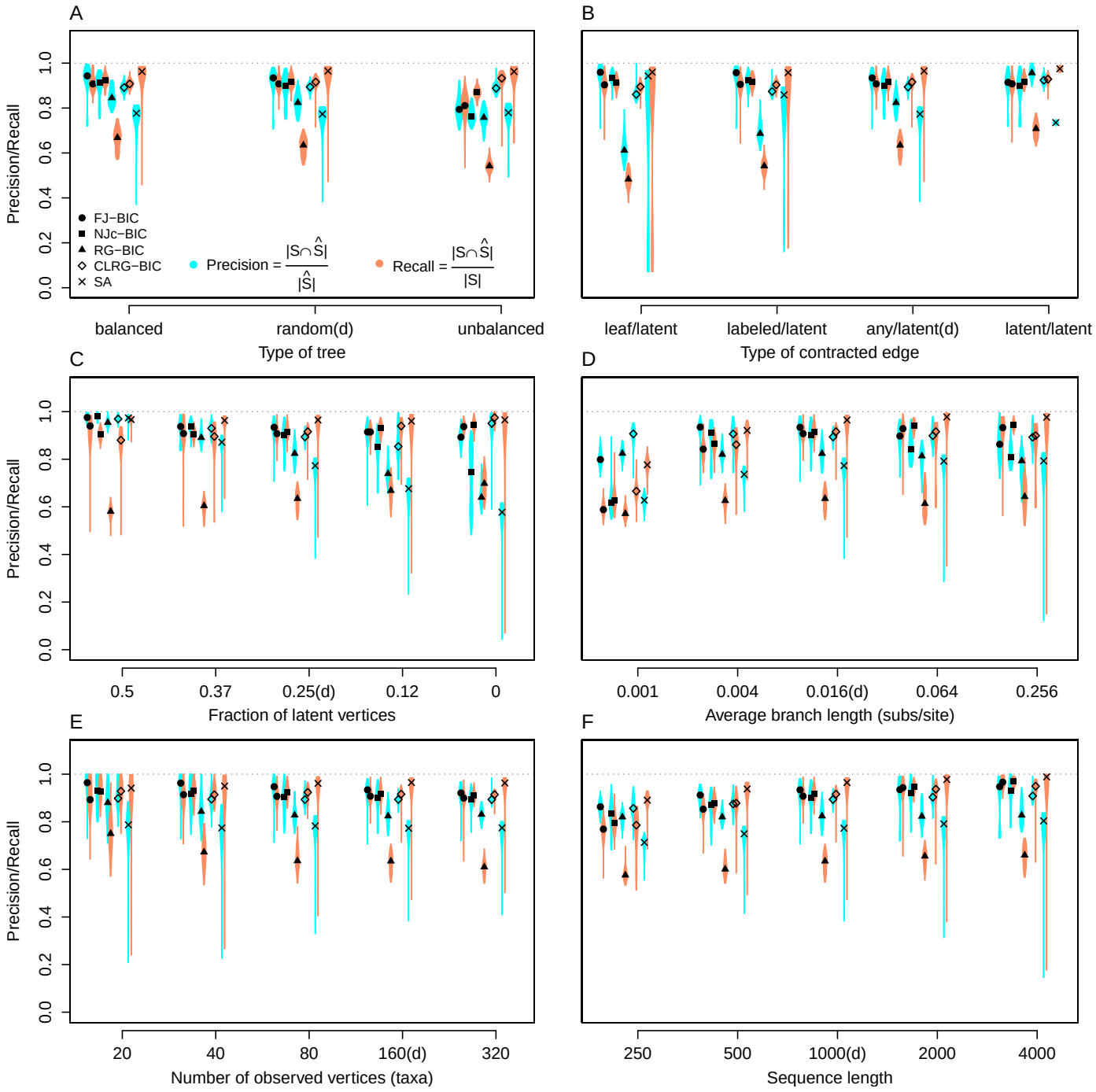


Figure 3.5: A comparison of the reconstruction accuracy of all methods in six simulation categories. One parameter (x-axes) was varied in each category. The default parameter settings are denoted as (d) on each x-axis. For each parameter setting, 100 data sets were created. Precision is shown in blue and recall is shown in orange.

Table 3.3: Methods with the highest recall. F, N, R, C, and S stand for FJ-BIC, NJc-BIC, RG-BIC, CLRG-BIC, and SA, respectively. The default setting for each simulation parameter is indicated with (d).

Tree type		Balanced	Random (d)	caterpillar	
		N,S	F,N,C,S	C,S	
Type of contracted edge	leaf/hidden	labeled/hidden	any/hidden (d)	hidden/hidden	
	F,N,C	N	F,N,C,S	S	
Fraction of hidden vertices	0.5	0.37	0.25 (d)	0.12	0
	S	S	F,N,C,S	N,C,S	C
Average branch length	0.001	0.004	0.016	0.064	0.256
	S	S	F,N,C,S	C,N,S	N,S
Number of labeled vertices	20	40	80	160	320
	N,C	N,C	N,C,S	F,N,C,S	N,C,S
Sequence length	250	500	1000	2000	4000
	C,S	S	F,N,C,S	F,N,C,S	F,N,C,S

performs significantly better on caterpillar trees than on balanced trees with median precision/recall values of 0.89/0.93 and 0.89/0.91, respectively. CLRG constructs an MST and then iteratively applies RG to the neighborhood of each non-leaf vertex. The better performance of CLRG-BIC on caterpillar trees is most likely due to the MST being topologically similiar to the caterpillar tree. SA has a median precision and recall of 0.77 and 0.96, respectively, across all tree types. SA has low precision because SA restricts the maximum out degree of labeled vertices to two, and the maximum out degree of hidden vertices to three.

Type of contracted edge

FJ-BIC has significantly higher precision than other methods for all types of contracted edges, except hidden/hidden. SA has a high median recall of 0.96 for all types of contracted edges. However the recall values of SA are not significantly higher than those of FJ-BIC if the contracted edge is leaf/hidden. This is due to a large variance in the performance of SA, quantified with an inter-quantile range of 0.26 (see Figure 3.5B). SA has high median precision of 0.94 if the contracted edge is leaf/hidden. Contracting leaf/hidden edges results in trees in which a labeled vertex can have up to one child and all other non-leaf vertices have degree three. The high performance of SA in this category is because these are the same type of trees which SA samples when optimizing tree topology. SA has lower performance when any other edge type is contracted. RG-BIC and CLRG-BIC have significantly higher precision and recall if hidden/hidden edges are contracted, when compared to precision and recall for other edge types.

Fraction of vertices that are hidden

All methods have a median precision higher than 0.95 (see Figure 3.5C) for leaf-labeled trees which have the maximal fraction (0.5) of hidden vertices. In this simulation scenario, with a median recall of 0.97, SA has significantly higher recall than other methods, even though FJ-BIC also has a high median recall of 0.94. A common trend for each method is that precision reduces and recall rises with a decrease in the fraction of hidden vertices. FJ-BIC has a median precision and recall greater than 0.89 across all settings of fraction of hidden vertices. CLRG-BIC has significantly higher precision and recall than other methods when all vertices are labeled. This is because the CLRG algorithm involves the construction of a MST which should be topologically similar to the completely labeled tree.

Average edge length

All methods perform poorly on trees with short average branch lengths of 0.001 subs/site with median recall smaller than 0.8 each (see Figure 3.5D). This is because a significant portion of the simulated sequences are

identical. Thus, in FJ-BIC, NJc-BIC, RG-BIC, and CLRG-BIC there is a preference for choosing parent-child relationship over siblings. CLRG-BIC has significantly higher precision than other methods if branch lengths are either very small or very large. FJ-BIC has high precision if the average branch length is between 0.004 and 0.064. In trees with larger branch lengths there is a high chance that sequences undergo multiple substitutions at the same site. This effect has been termed site saturation and results in an underestimation of the evolutionary distance. Additionally, estimates of large distances are associated with large variance (Hoyle and Higgs, 2003) which results in the selection of wrong neighbors in the neighbor-joining step. CLRG-BIC has higher performance for trees with large branch lengths because the input to CLRG-BIC is the MST and the construction of the MST is probably robust to noise in distance estimates. The performance of SA is not greatly affected by long branches.

Number of labeled vertices (species)

RG shows significant reduction in precision/recall as tree size (number of species) is increased with corresponding median values changing from 0.88/0.75 (5 labeled vertices) to 0.83/0.61 (80 labeled vertices) (see Figure 3.5E). The change in precision and recall shown by SA is not significant. FJ-BIC and CLRG-BIC show a significant drop in precision with increasing tree size, but recall does not change significantly. Even for trees with 320 species, FJ-BIC has high median precision and recall of 0.92 and 0.9 respectively. NJc-BIC shows significant reduction in both precision and recall with increasing tree size , with median precision/recall changing from 0.93/0.93 to 0.89/0.91.

Sequence length

The performance of all methods improves with increasing sequence length. For all settings of sequence length, FJ-BIC is one of the best performing methods (see Figure 3.5F). FJ-BIC is among the methods with significantly high recall for sequences of length 1000 bp to 4000 bp. SA is one of the methods with significantly high recall for all settings of sequence length.

3.3.6.1 Summary of results

For the simulations performed using the default parameter settings, the methods listed in order of decreasing median precision are FJ-BIC (0.93), NJc-BIC (0.9), CLRG-BIC (0.89), RG-BIC (0.82), and SA (0.77), and the methods listed in order of decreasing median recall are SA (0.96), NJc-BIC (0.92), CLRG-BIC (0.92), FJ-BIC (0.91) and RG-BIC (0.63). In 15 out of the 22 simulated scenarios FJ-BIC is among the methods with significantly high precision (see Table 3.2). In 17/22 simulated scenarios SA is among the methods with significantly high recall (see Table 3.3). In 13/22 simulated scenarios NJc-BIC is among the methods with significantly high recall. FJ-BIC has a median recall that is greater than 0.9 in 16/22 simulated scenarios. The remaining scenarios are (*i*) trees with 20 species (recall of 0.89), (*ii*) trees in which branches are very short (0.001 and 0.004 subs/site; recall of 0.6 and 0.84 respectively), (*iii*) caterpillar trees (0.81), and (*iv*) trees constructed using short sequences (250 and 500 bp; recall of 0.77 and 0.85 respectively).

3.3.7 Comparison of time-complexities and run times

We report the worst-case time-complexity for the clustering procedures. FJ and NJ run in time $O(n^3)$. RG runs in time $O(n^4)$ which makes it infeasible to run on large datasets. CLRG runs in $O(n^2 \log n + n_i \delta_{\max}^3(\text{MST}))$ where n_i is the number of non-leaf vertices of the MST and $\delta_{\max}(\text{MST})$ is the largest vertex degree in the MST. Model selection with BIC or AIC requires the repeated optimization of the likelihood function with respect to parameters of the substitution model. Computing the likelihood with Felsenstein's tree pruning algorithm (Felsenstein, 1981) takes $O(nA^2L)$ time where L is the sequence length and A is the size of the alphabet. A is four for genetic sequences and 20 for protein sequences. We used RAxML for computing likelihoods, and optimizing parameters of substitution model. SA performs Bayesian inference by MCMC sampling, a stochastic procedure whose runtime depends on how easily the MCMC chain moves through the space of trees and model parameters. The observed run times (see Figure 3.6) suggest that

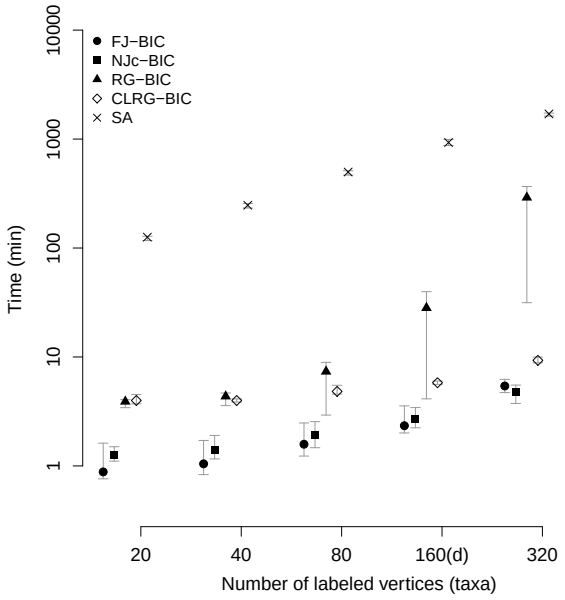


Figure 3.6: A comparison of run times of all methods in the scenario where the number of labeled vertices was varied. Run times are shown on a log-scale.

FJ-BIC and NJc-BIC are the fastest methods for trees containing up to 320 species, with both the methods having a median run time of 5.4 and 4.8 minutes respectively. CLRG-BIC took around 9.3 minutes to reconstruct trees containing 320 species and showed the slowest growth in run time. RG showed the largest growth in run time taking 4.8 hours for reconstructing trees with 320 species. SA was run with MCMC chain-length set to 10^8 states. SA took around two hours to construct trees containing 20 species and 30 hours for constructing trees containing 320 species.

3.4 Validation of family joining using HIV transmission network data

We applied FJ-BIC to a dataset of HIV-1 subtype C *env* gene sequences that were sampled from 11 hosts who are part of a partially known transmission chain (Lemey et al., 2005; Vrancken et al., 2014). We discarded 31 sequences which had gaps and analyzed the remaining 181 sequences of length 1376 bp. The hosts are labeled $A, B, C, D, E, F, G, H, I, K,$ and L. Sequences from multiple time points are available for $A, B, C, D, E,$ and H. The sampling times for all sequences are known. All the host pairs who were involved in a transmission event are known, and were inferred by interviewing the hosts. The direction of transmission is known for all transmission events except for the transmission between A and B.

Additionally we compared the bootstrap support of branches in the FJ-BIC tree with the branches in the maximum likelihood tree constructed using RAxMLv8.2. (Stamatakis, 2014). We first identified the most appropriate model of substitution using JModelTest2 (Darriba et al., 2012). The models that we considered were limited to the set of time-reversible Markov models that were made available by JModelTest2. Variants of all available substitution models which included a parameter for invariant sites (I) and/or a Gamma model (Γ) for across-site rate variation were also tested. GTR+Γ+I was the best model, *i.e.*, the one with the smallest AIC score. We constructed a tree with RAxML using the original sequence alignment and the GTRCATI model of substitution, which we refer to as the RAxML tree. The CAT model is an alternative

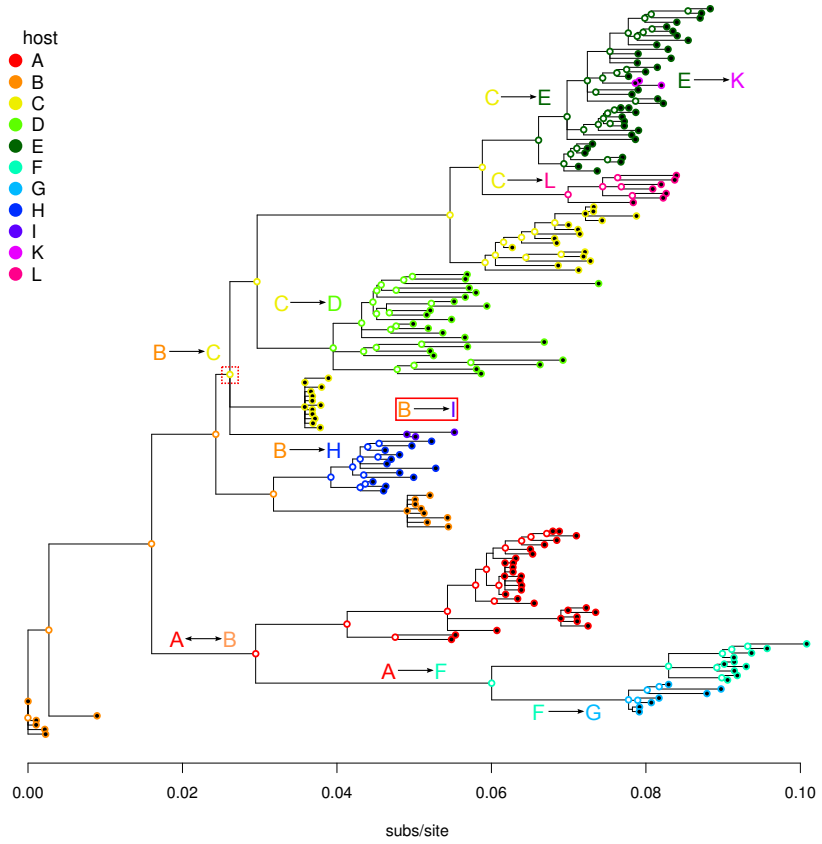


Figure 3.7: The FJ-BIC tree of 181 HIV-1 *env* gene sequences sampled from hosts involved in a known transmission chain. Each vertex is represented by a circle whose inner color is black if the vertex is labeled and white if the vertex is hidden. The outer color of each circle indicates the host of the corresponding vertex. Edges indicating transmission events have been labeled. 9/10 transmission events are compatible with the FJ-BIC tree. The red box highlights the transmission event $B \to I$ which is not compatible with the FJ tree.

to the Gamma model that enables fast computation (Stamatakis, 2006). We inferred a generally labeled tree using FJ-BIC.

The FJ-BIC tree was rooted assuming a strict molecular clock model. We define the optimal position of the root as that position which minimizes the sum of squared residuals (RSS) of regressing distances from the root to each labeled vertex against sampling times. We searched for the optimal position of the root as follows. First we placed the root at the midpoint of each edge, and selected the edge that minimized the RSS. Subsequently, we searched along the edge for the position of the root which minimized the RSS.

Compatibility of the FJ-BIC tree with known transmission events

In order to check if the known transmission events are compatible with a rooted tree we needed to label all hidden vertices with a host. Hidden vertices were visually labeled with hosts via maximum parsimony. The labeling that we applied resulted in the minimum possible total cost of 10 (see Figure 3.4).

Given a rooted tree with all vertices labeled with a host, we define a transmission event $(X \to Y)$ to be compatible with the tree if there is an edge that exits a vertex labeled X and enters a vertex labeled Y. 9 out of 10 transmission events are compatible with the FJ-BIC tree. The direction of transmission between A and B is not known. The FJ-BIC tree suggests that A was infected by B. The branch of the FJ-BIC tree

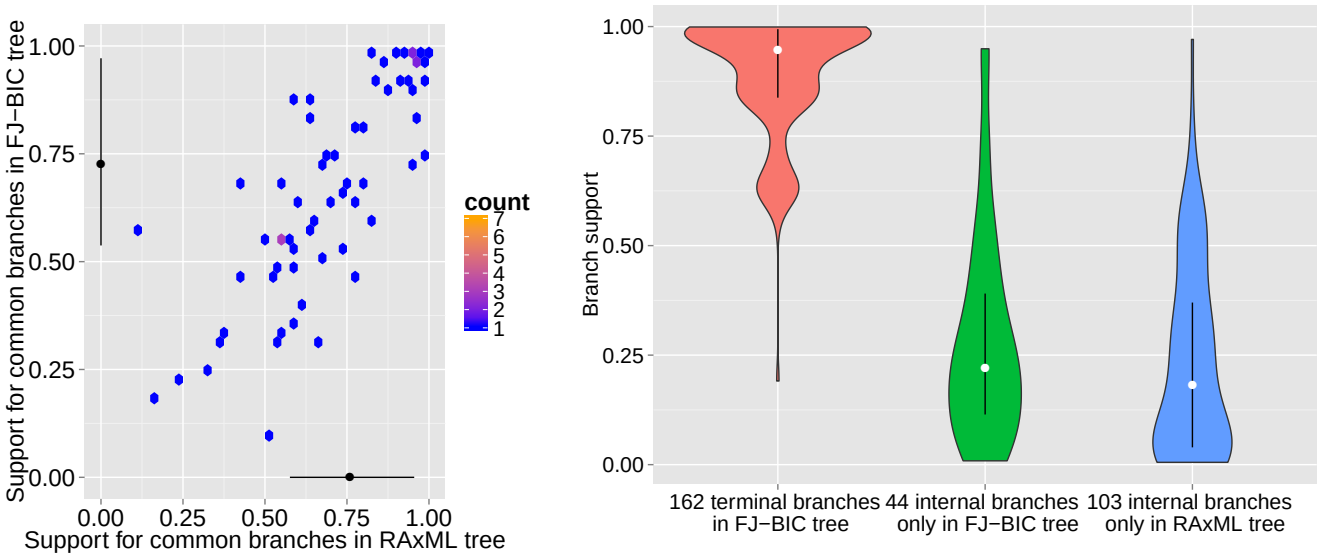


Figure 3.8: Left: Comparing the support of common branches in the FJ-BIC tree and the RAxML tree. Right: Supports for branches that are only present in either the FJ-BIC tree or the RAxML tree.

that suggests this transmission event has been labeled with the known transmission event $A \leftrightarrow B$. 8 out the remaining 9 transmission events are compatible with the FJ-BIC tree and branches indicative of these transmission events are labeled in Figure 3.4. The transmission event $B \rightarrow I$ is not compatible with the FJ-BIC tree (red solid box in Figure 3.4) which may be due to insufficient sampling. Only three sequences were available from host I. It is possible that the polytomy present inside the red dotted box in Figure 3.4 may be resolved if more sequences from I were available, in such a way that the resulting tree would be compatible with the transmission $B \rightarrow I$.

Branch support in the FJ-BIC tree and the RAxML tree

The bootstrap support of an edge is defined as the number of bootstrap replicate trees that contain the split that is induced by the edge. Given trees T_1 and T_2 that have the same set of labeled vertices. An edge e_1 in T_1 is said to be contained in T_2 if the split in T_1 that is induced by e_1 is contained in T_2. The bootstrap support of edges in the FJ-BIC tree and the RAxML tree were computed using 1000 replicates. Since each labeled vertex is a leaf in all bootstrap RAxML trees, all terminal branches of the RAxML tree trivially have a support of one. The support of a terminal edge in the generally labeled tree that are constructed by FJ-BIC is not necessarily one.

75 internal edges were common to the FJ-BIC tree and the RAxML tree. The median (IQR) supports for the common edges were 0.73 (0.43) and 0.76 (0.38) in the FJ-BIC and the RAxML tree respectively. Supports for the common edges in both trees were strongly correlated (Pearson's $\rho = 0.84$, see Figure 3.8). There are 44 and 103 internal edges that are present only in the FJ-BIC tree and the RAxML tree respectively with lower median (IQR) edge supports of 0.22 (0.28) and 0.18 (0.33) respectively (see Figure 3.8). The 124 terminal edges in the FJ-BIC tree have a median (IQR) branch support of 0.95 (0.16).

On average an internal edge in the FJ-BIC tree has a higher support than an internal edge in the RAxML tree. 36% of FJ-BIC edges and 25% of RAxML edges have bootstrap supports greater than 0.7.

3.5 Summary and Outlook

In this chapter a distance-based clustering method called FJ for constructing generally labeled trees was presented. Given pairwise distances between 320 species, FJ-BIC took around 5.4 min (± 0.76) to estimate a tree. The FJ algorithm treats short edges as unreliable and identifies an optimal threshold for contracting short edges by minimizing test error. We tested three methods: FJ-AIC, FJ-BIC, and FJ-CV, which minimize AIC, BIC and CV error, respectively. BIC was the best model selection criterion. When compared with related methods FJ-BIC was the best at reconstructing generally labeled phylogenetic trees across a wide range of simulation settings. FJ-BIC was applied to HIV sequences sampled from individuals that were involved in a known transmission chain. The FJ-BIC tree was compatible with ten out of eleven transmission events. On average, internal edges in the FJ-BIC tree were found to have higher support than internal edges in the tree constructed using RAxML. A method for reconstructing phylogenetic trees with high precision circumvents the need for time-consuming bootstrap analyses.

As part of this study we tried implementing the distance-based supertree method by Choi et al. (2011) called Chow-Liu grouping because we were interested in better understanding how minimum spanning trees can be used in the inference of phylogenetic trees. During our attempt at implementing CLGrouping with NJ as the base method we discovered that given the input distances that are additive in a phylogenetic tree T there are instances where the phylogenetic tree that is reconstructed using CLGrouping(NJ) differs from T. Since NJ is guaranteed to recover T if NJ is applied to distances that are additive in T, the indeterminacy of CLGrouping appeared to stem from an issue with the input MST that is used by the supertree method. Additionally, we noticed that if we used the MST that was constructed by the authors' Matlab implementation of CLGrouping then there was no indeterminacy in CLGrouping(NJ). The cause of indeterminacy of CLGrouping is clarified in the following chapter.

Chapter 4

Topological relationship between MSTs and phylogenetic trees

The work that is presented in this chapter has been published in Kalaghatgi and Lengauer (2017).

Choi et al. (2011) claimed that minimum spanning trees (MSTs) constructed using tree-distances share a topological relationship with corresponding phylogenetic trees. We discovered that the topological relationship does not necessarily hold if the MST is not unique. We proposed so-called vertex-order based MSTs (VMSTs) that are guaranteed to share a topological relationship with phylogenetic trees. We show that the number of leaves in a VMST is an indicator of the amount of phylogenetic information that is contained in the VMST. Additionally, we provide a polynomial-time algorithm for selecting a VMST with the maximum amount of phylogenetic information.

4.1 Motivation

Choi et al. (2011) introduced a distance-based divide-and-conquer method called Chow-Liu grouping (CLGrouping). The distances that are used in this chapter are tree-distances that are defined on phylogenetic trees. CLGrouping makes use of the minimum spanning trees (MSTs) of a graph structure that is referred to as the distance graph. Distance graphs are constructed as follows. Given distances d for each vertex pair in V the distance graph $G = (V, E)$ is an edge-weighted undirected complete graph over V such that for any edge $\{u, v\}$ the edge-weight $w_{\{u,v\}}$ equals the distance $d(u, v)$.

CLGrouping consists of two stages. The first stage involves the construction of a minimum spanning tree (MST) M of the distance graph G. The second stage iterates over the non-leaf vertices of M and, for each non-leaf vertex i that is visited, a vertex set V_i comprising i and the neighbors of i is constructed. Subsequently a phylogenetic tree T_i is constructed using distances between vertices in V_i. In the final step of the iteration, the graph in M, which is induced by V_i is replaced by T_i (see Figure 4.1E for an illustration). If i is not the first vertex to be visited then V_i may contain newly introduced hidden vertices. Let h_j be a hidden vertex that was introduced when processing the labeled vertex j. The distance from h_j to a labeled vertex l in V_i is computed as $d(h_j, l) = d(j, l) - d(j, h_j)$. The distance between two hidden vertices h_j and h_k is computed as $d(h_j, h_k) = d(j, k) - d(j, h_j) - d(k, h_k)$.

The order in which the non-leaf vertices are visited is not specified by the Choi et al. and does not seem to be important. CLGrouping terminates once all the non-leaf vertices of M have been visited once.

This procedure is called Chow-Liu grouping because the MSTs that are constructed using tree-distances are topologically equivalent to Chow-Liu trees (Chow and Liu, 1968), for certain probability distributions. Please refer to Choi et al. (2011) for further details.

Choi et al. (2011) compared the reconstruction accuracy of neighbor joining (NJ; (Saitou and Nei, 1987)), a popular distance-based clustering method, with CLGrouping(NJ) which is an application of CLGrouping

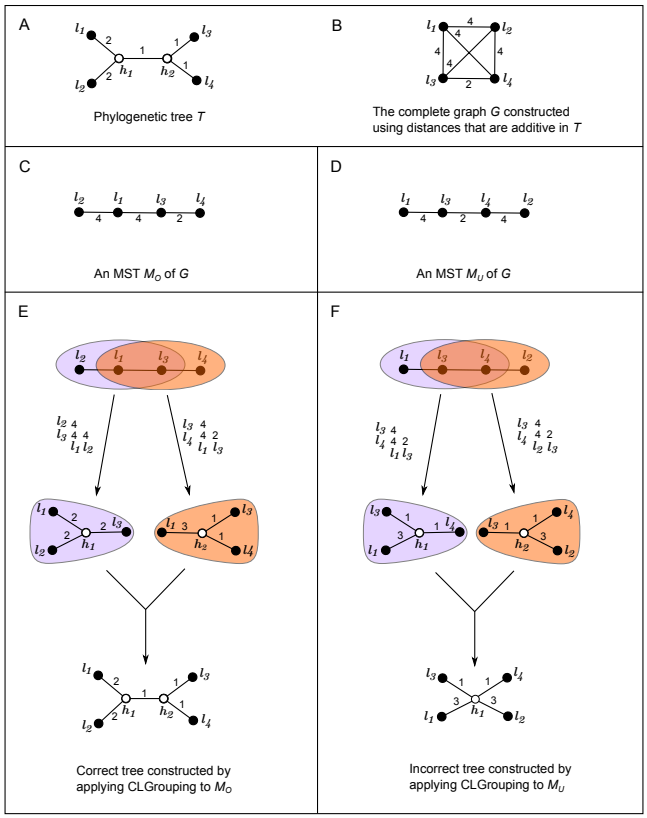


Figure 4.1: The example used to demonstrate that CLGrouping may not reconstruct the correct tree if there are multiple MSTs. The phylogenetic tree T that is used in this example is shown in panel A. The distance graph G of T is shown in panel B. Two MSTs of G, M_O and M_U are shown in panels C and D, respectively. Panels E and F show the intermediate steps, and the final result of implementing CLGrouping using M_O and M_U respectively. CLGrouping reconstructs the original phylogenetic tree if it uses M_O but not if it is uses M_U.

that uses NJ as the base method. Choi et al. (2011) showed that CLGrouping(NJ) is more accurate than NJ at reconstructing phylogenetic trees with large diameter. Huang et al. (2014) showed that CLGrouping affords a high degree of parallelism because phylogenetic tree reconstruction can be performed independently for each vertex group.

Unless specified otherwise each mention of CLGrouping in the following text in this chapter refers to the application of CLGrouping with NJ as the base method. Additionally, distances that are used in this chapter are assumed to be tree distances. Unless specified otherwise the proofs that are included in this chapter are applicable to generally labeled phylogenetic trees.

4.2 Indeterminacy of Chow-Liu grouping

4.2.1 A quartet tree

We demonstrate the indeterminacy of CLGrouping for the quartet tree T (Figure 4.1). Two MSTs M_U and M_O of the distance graph G of T were constructed by hand. The intermediate steps, and the final result of applying CLGrouping to M_U and M_O are shown in Figure 4.1E and Figure 4.1F, respectively. CLGrouping reconstructs the original phylogenetic tree if it is applied to the VMST M_O but not if it is applied to M_U.

4.2.2 A primate phylogenetic tree

In this subsection we demonstrate the indeterminacy of CLGrouping using the phylogeny over the primate genera (Pozzi et al., 2014). CLGrouping will infer the correct topology if the input MST shares the topological relationship with phylogenetic trees that was introduced by Choi et al. (2011).

Methodological details

The primate phylogeny was downloaded from the TimeTree database which is a comprehensive collection of published phylogenies (Hedges et al., 2006; Kumar and Hedges, 2011; Hedges et al., 2015). The branches of the primate phylogeny are scaled in units of calendar time. The primate phylogeny contains three branches of length zero that cannot be inferred from the corresponding tree metric. A modified primate phylogenetic tree T was constructed by contracting all branches of length zero. One hundred MSTs were constructed using the following procedure. Kruskal's algorithm was applied to the edges of the distance graph of T that were arranged in a randomly shuffled order. We applied CLGrouping to each MST, and computed the topological distance between each output phylogeny and the primate phylogeny using the Robinson-Foulds distance (Robinson and Foulds, 1981). The *Robinson-Foulds distance* ($\mathrm{RF}_S(T, \hat{T})$) is computed as the fraction of unique splits that are present in one tree and not the other.

$$\mathrm{RF}_S(T, \hat{T}) = 1 - \frac{|\mathcal{S}_{\mathrm{all}}(T) \cap \mathcal{S}_{\mathrm{all}}(\hat{T})|}{|\mathcal{S}_{\mathrm{all}}(T) \cup \mathcal{S}_{\mathrm{all}}(\hat{T})|}$$

where $\mathcal{S}_{\mathrm{all}}(T)$ is the set of all splits that are contained in the tree T.

We selected a CLGrouping tree that maximizes the RF distance from the primate phylogeny. The selected CLGrouping tree is 0.4 RF distance away from the primate phylogeny and is shown in Figure 4.2. In order to enable a visual comparison we rooted the CLGrouping tree at the midpoint of the least imbalanced edge. The primate phylogeny is an ultrametric tree and has been rooted such that the root is equidistant from the leaves. As can be seen, both the trees in Figure 4.2 are substantially different.

Topological relationship between MSTs and phylogenetic trees

The correctness of CLGrouping depends on a topological relationship between MSTs and phylogenetic trees that was introduced by Choi et al. (2011).

In order to establish the topological relationship between minimum spanning trees and phylogenetic trees Choi et al. (2011) introduced the notion of a surrogate vertex.

The *surrogate vertex* of a hidden vertex is the closest labeled vertex w.r.t. tree distance. Choi et al. (2011) claim that minimum spanning trees can be constructed by contracting all edges along the path from each hidden vertex h to the surrogate vertex of h. In the example shown in Figure 4.1, the MST M_O can be constructed by contracting the edges $\{h_1, l_1\}$, and $\{h_2, l_3\}$. Clearly there is no selection of surrogate vertices such that M_U can be constructed by contracting the path between each hidden vertex and the corresponding surrogate vertex.

Choi et al. (2011) assume that for any MST there exists a selection of surrogate vertices such that the MST can be constructed by contracting paths between each hidden vertex and the corresponding surrogate vertices. The indeterminacy of CLGrouping only occurs if there are multiple MSTs. The problem of selecting surrogate vertices for the case where multiple labeled vertices are closest to hidden vertices is discussed below.

Let the surrogate vertex set $S(h)$ of a vertex h be the set of all labeled vertices that are closest to h. Consider two hidden vertices h_1 and h_2, such that there are multiple labeled vertices, l_1 and l_2, that are common to the corresponding surrogate vertex sets $S(h_1)$ and $S(h_2)$. Choi et al. (2011) assume that it is always possible to apply the following tie-breaking rule for implicitly selecting the corresponding surrogate vertices. A labeled vertex that is common to $S(h_1)$ and $S(h_2)$ (either l_1 or l_2) is selected as the surrogate vertex of both h_1 and h_2.

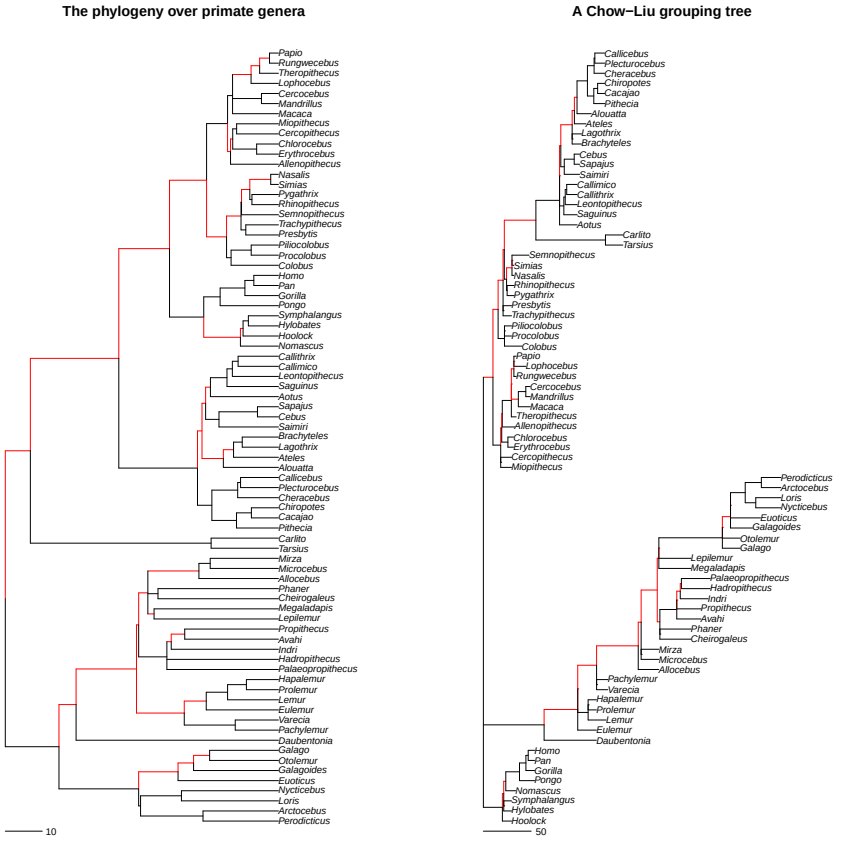


Figure 4.2: Left: The empirically established phylogeny T over primate genera (Hedges et al., 2006; Pozzi et al., 2014). Right: A phylogeny that was constructed by applying CLGrouping to an MST of the distance graph of T. The edges that are highlighted in red correspond to splits that are contained in one tree but not the other tree.

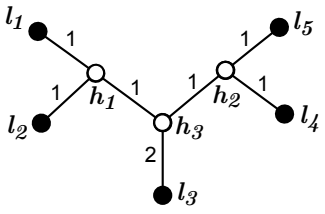


Figure 4.3: The phylogenetic tree that is used to demonstrate that the tie-breaking rule as defined by Choi et al. (2011) cannot be applied in general.

This rule for selecting surrogate vertices cannot be applied in general. We demonstrate this with an example. For the tree shown in Figure 4.3 we have $S(h_1) = \{l_1, l_2\}$, $S(h_2) = \{l_4, l_5\}$, and $S(h_3) = \{l_1, l_2, l_3, l_4, l_5\}$. There is no selection of surrogate vertices that satisfies the tie-breaking rule.

4.3 Vertex order based MSTs

In order to construct an MST that is guaranteed to have the desired topological correspondence with the phylogenetic tree, we propose the following definition of a surrogate vertex.

Definition 1. Given a phylogenetic tree $T = (V_T = \{\mathcal{L}_T, \mathcal{H}_T\}, E_T)$ and distances d_T that are additive in T, let there be a total order $<_V$ over the set of all labeled vertices of T. The vertex order based surrogate vertex of a vertex v in V_T is the labeled vertex in $\mathcal{L}_T$ that is closest w.r.t. d_T, and smallest w.r.t to the vertex order $<_V$. That is,

$$s(v) = \underset{l \in \mathcal{L}_T}{\operatorname{argmin}} \, (d_T(l, v), l_{<_V}),$$

where $l_{<_V}$ is the rank of l in the order $<_V$, and the lexicographic order is applied to the ordered pair following "argmin" in the formula.

The inverse surrogate set $S^{-1}(l)$ of a labeled vertex l is the set of all vertices whose surrogate vertex is l. Note that each labeled vertex is contained in its inverse surrogate set.

In order to ensure that the surrogate vertices are selected on the basis of tree distances and vertex order, it is necessary that information pertaining to vertex order is used when selecting the edges of the MST. We use Kruskal's algorithm (Kruskal, 1956) for constructing the desired MST. Since Kruskal's algorithm takes as input a set of edges sorted w.r.t. edge weight, we modify the input by sorting edges with respect to edge weight and vertex order as follows. It is easy to modify other algorithms for constructing MSTs in such a way that vertex order is taken into account.

Definition 2. Given an edge-weighted graph $G = (V, E)$, and a total order $<_V$ over the vertices in V. Let $w_{\{u,v\}}$ be the weight of the edge $\{u, v\}$. Edges in E are sorted w.r.t. edge weight and vertex order using the lexicographic order that is defined below. Let the sorting be defined using the total order $<_E$. For each pair of edges $\{a, b\}$ and $\{c, d\}$ in E,

$$\{a, b\} <_E \{c, d\}, \text{ if and only if}$$

$$(w_{\{a,b\}}, \min(a_{<_V}, b_{<_V}), \max(a_{<_V}, b_{<_V})) < (w_{\{c,d\}}, \min(c_{<_V}, d_{<_V}), \max(c_{<_V}, d_{<_V}))$$

where the tuples are compared lexicographically

The modified algorithm for constructing a vertex order based MST (VMST) is described in Algorithm 3.

Algorithm 3: Constructing a vertex order based MST (VMST)
Input: $(G = (V, E), <_V)$
$E_{<_V} \leftarrow$ edges in E ordered w.r.t. edge weight and vertex order
$M_{<_V} \leftarrow$ MST constructed by applying Kruskal's algorithm to $E_{<_V}$
Output: $M_{<_V}$

Using the notion of VMSTs we will prove Lemma 1, and consequently show that the indeterminacy of CLGrouping can be removed if CLGrouping is applied to a VMST.

Lemma 1. *Adapted from parts (i) and (ii) of Lemma 8 in Choi et al. (2011). Given a phylogenetic tree $T = (V_T, E_T)$ and a total order $<_{\mathcal{L}_T}$ over the labeled vertices in T, let $G = (V_G, E_G)$ be the distance graph of T. Let $M = (V_M, E_M)$ be the VMST constructed by applying Algorithm 3 to $(G,<_{\mathcal{L}_T})$. The surrogate vertex of each hidden vertex is defined with respect to the tree metric d_T and a vertex order as given in Definition 1. M is related to T as follows.*

1. If $l \in V_M$ and $h \in S^{-1}(l)$ s.t. $h \neq l$, then every vertex in the path in T that connects l and h belongs to the inverse surrogate set $S^{-1}(l)$.

2. For any two vertices that are adjacent in T, their surrogate vertices, if distinct, are adjacent in M, *i.e.*, for all $i, j \in V_T$ with $s(i) \neq s(j)$,

$$\{i, j\} \in E_T \Rightarrow \{s(i), s(j)\} \in E_M.$$

Proof. (i). Assume that there is a vertex u on the path between h and l, such that $s(u) = k \neq l$. Since $s(u) = k$ implies that $(d_T(u, k), k_{<_V}) < (d_T(u, l), l_{<_V})$, we have $d_T(u, k) \leq d_T(u, l)$, with equality holding only if $k_{<_V} < l_{<_V}$.

There are seven ways to position k w.r.t. h, u, and l (see Figure 4.4). We only consider the general positions.

For case 1 we have
$$\begin{aligned} & d_T(h, l) \leq d_T(h, k) \\ \Leftrightarrow & d_T(h, j) + d_T(j, u) + d_T(u, l) \leq d_T(h, j) + d_T(j, k) \\ \Leftrightarrow & d_T(j, u) + d_T(u, l) \leq d_T(j, k) \\ \Rightarrow & d_T(u, l) < d_T(u, j) + d_T(j, k) \\ \Leftrightarrow & d_T(u, l) < d_T(u, k) \text{ (contradiction since } s(u) = k) \end{aligned}$$

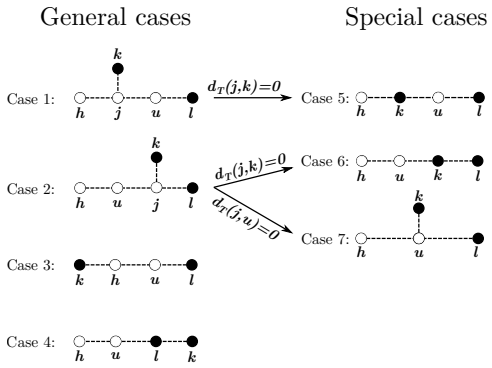


Figure 4.4: The cases that were considered in the proof of Lemma 1 part (i). Each case specifies one of the seven possible positions of a labeled vertex k w.r.t. hidden vertices h and u, and a labeled vertex l. Hidden vertices are represented with white circles and labeled vertices are represented with black circles. Each dashed line represents a path between the two vertices at its end points. The condition on top of each solid arrow describes how the special cases can be constructed from the corresponding general cases.

For case 2 we have
$$\begin{aligned} & d_T(h,l) \leq d_T(h,k) \\ \Leftrightarrow & d_T(h,u) + d_T(u,j) + d_T(j,l) \leq d_T(h,u) + d_T(u,j) + d_T(j,k) \\ \Leftrightarrow & d_T(u,j) + d_T(j,l) \leq d_T(u,j) + d_T(j,k) \\ \Leftrightarrow & d_T(u,l) \leq d_T(u,k) \text{ (contradiction since } s(u) = k) \end{aligned}$$

For case 3 we have
$$\begin{aligned} & d_T(h,l) \leq d_T(h,k) \\ \Leftrightarrow & d_T(h,u) + d_T(u,l) \leq d_T(h,k) \\ \Rightarrow & d_T(u,l) < d_T(h,k) + d_T(h,u) \\ \Leftrightarrow & d_T(u,l) < d_T(u,k) \text{ (contradiction since } s(u) = k) \end{aligned}$$

For case 4 we have
$$\begin{aligned} & d_T(u,k) = d_T(u,l) + d_T(l,k) \\ \Rightarrow & d_T(u,k) > d_T(u,l) \text{ (contradiction since } s(u) = k) \end{aligned}$$

(ii). Consider the edge $\{i,j\}$ in E_T such that $s(i) \neq s(j)$. Let V_i and V_j be the vertex sets of the connected components that are constructed by removing the edge $\{i,j\}$, such that V_i and V_j contain i and j, respectively. Let $\mathcal{L}_i$ and $\mathcal{L}_j$ be sets of labeled vertices that are defined as $V_i \cap V_M$ and $V_j \cap V_M$ respectively. From part (i) of Lemma 1 we know that $s(i) \in \mathcal{L}_i$ and $s(j) \in \mathcal{L}_j$. Consider the labeled vertices $l_i \in \mathcal{L}_i \backslash \{s(i)\}$ and $l_j \in \mathcal{L}_j \backslash \{s(j)\}$.

We have

$$\begin{aligned} d_T(l_i,l_j) &= d_T(l_i,i) + d_T(i,j) + d_T(l,j) \\ &\geq d_T(s(i),i) + d_T(i,j) + d_T(s(j),j) \\ &= d_T(s(i),s(j)) \end{aligned}$$

It follows that

$$d_T(s(i),s(j)) \leq d_T(l_i,l_j), \tag{4.1}$$

with equality holding only if

$$s(i)_{<_V} < l_{i<_V} \text{ and } s(j)_{<_V} < l_{j<_V}. \tag{4.2}$$

The cut property of MSTs states that given a graph $G = (V, E)$, for each pair V_1, V_2 of disjoint sets such that $V_1 \cup V_2 = V$, each MST of G contains one of the smallest edges (w.r.t. edge weight) which have one endpoint in V_1 and the other endpoint in V_2. Thus M contains at most one of the following edges $\{l_i, l_j\}, \{s(i), l_j\}, \{l_i, s(j)\}$ and $\{s(i), s(j)\}$. Note that the vertex order based MST M is constructed using edges that are sorted w.r.t. edge weight and the vertex order $<_V$. Let the ordered set of edges be defined using the total order $<_E$ over E.

From equations (4.1) and (4.2) we have

$$(d_T(s(i), s(j)), \min(s(i)_{<_V}, s(j)_{<_V}), \max(s(i)_{<_V}, s(j)_{<_V})) < (d_T(l_i, l_j), \min(l_{i<_V}, l_{j<_V}), \max(l_{i<_V}, l_{j<_V}))$$

Thus, according to Definition 2, it follows that $\{s(i), s(j)\} <_E \{l_i, l_j\}$. Through a similar construction it can be shown that $\{s(i), s(j)\} <_E \{s(i), l_j\}$ and $\{s(i), s(j)\} <_E \{l_i, s(j)\}$. It follows that $\{s(i), s(j)\} \in E_M$. □

CLGrouping can be shown to be correct using Lemma 1 and the rest of the proof that was provided by Choi et al. (2011).

The authors of CLGrouping provide a Matlab implementation of their algorithm. The implementation takes as input a distance matrix which has the following property: the row index, and the column index of each labeled vertex is equal. The MST that is constructed in the authors implementation is a vertex order based MST. The vertex order is equal to the order over the column/row indices of the labeled vertices. The implementation provided by Choi et al. (2011) correctly reconstructs the model tree even if there are multiple MSTs in the underlying distance graph.

4.4 An optimality criterion for selecting vertex order

4.4.1 Split information in a VMST

Consider a minimum spanning tree to be a generally labeled unrooted phylogenetic tree with no hidden vertices. The notion of split that is usually only used for unrooted phylogenetic trees can be easily extended to minimum spanning trees. In the following lemma we will show that each split that occurs in a VMST is also contained the corresponding phylogenetic tree.

Lemma 2. *Given a phylogenetic tree $T = (V_T = \{\mathcal{L}_T, \mathcal{H}_T\}, E_T)$. Let $M = (V_M, E_M)$ be a VMST of T. Each split $V_a|V_b$ in M is a split in T.*

Proof. Without loss of generality let $\{a, b\}$ in E_M be the edge that induces the split $V_a|V_b$ such that V_a contains a, and V_b contains b.

From Lemma 1 part (ii) we know that a and b are the surrogate vertices of hidden vertices that are adjacent in T. Let h_a and h_b be hidden vertices in $\mathcal{H}_T$ that are adjacent in T such that a is the surrogate vertex of h_a, and b is the surrogate vertex of h_b.

Consider the split $\mathcal{L}_{h_a}|\mathcal{L}_{h_b}$ in T that is induced by edge $\{h_a, h_b\}$ in E_T. Without loss of generality let $\mathcal{L}_{h_a}$contain a, and let $\mathcal{L}_{h_b}$contain b.

From Lemma 1 it follows that M is constructed by contracting paths in T between each hidden vertex and the corresponding surrogate vertex. By construction of M from path operations it follows each vertex in V_a is contained in $\mathcal{L}_a$ but not $\mathcal{L}_b$. Conversely each vertex in V_b is contained in $\mathcal{L}_b$ but not $\mathcal{L}_a$. Since $V_a \cup V_b = \mathcal{L}_T = \mathcal{L}_a \cup \mathcal{L}_b$, and $V_a \cap V_b = \varnothing = \mathcal{L}_a \cap \mathcal{L}_b$, it follows that each split in M is a split in T □

Consider an unrooted phylogenetic tree T and a corresponding VMST M. Each terminal edge in M induces a trivial split in T. Each internal edge in M induces a nontrivial split in T. With respect to maximizing the number of non-trivial splits, an optimal VMST would have the minimum number of leaves.

In the context of parallel programming, Huang et al. (2014) showed that it is possible to parallelize CLGrouping by independently constructing phylogenetic trees over the vertex group that is associated with

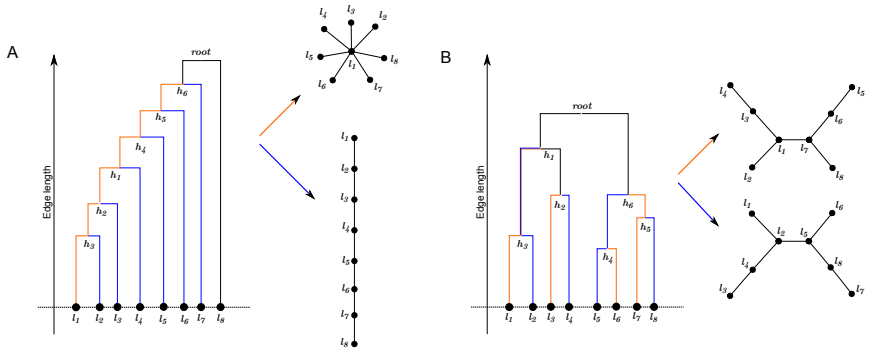


Figure 4.5: Each panels above shows (*i*) an ultrametric tree (left), (*ii*) a VMST with the maximum number of leaves (VMST_{max}, top right), and (*iii*) a VMST with the minimum number of leaves (VMST_{min}, bottom right). The edge contraction operations in orange and blue were used to construct VMSTs with the maximum number of leaves, and minimum number of leaves, respectively. The difference in the number leaves between VMST_{max} and VMST_{min} is largest for the caterpillar tree shown in panel A, and smallest for the balanced tree shown in panel B.

each non-leaf vertex, and merging them in order to construct the full phylogenetic tree. The step involving tree mergers requires a shared memory architecture.

Thus, with respect to parallelism, an optimal VMST would have the maximum number of vertex groups, and equivalently, the minimum number of leaves.

4.4.2 Tree shape

In order to relate the shape of a phylogenetic tree to the number of leaves in a corresponding VMST, we consider ultrametric caterpillar trees and ultrametric balanced trees (Semple and Steel, 2003).

A VMST of a rooted phylogenetic tree is constructed by suppressing the root of the phylogenetic tree, followed by contracting paths in the unrooted phylogenetic tree between each hidden vertex and the corresponding surrogate vertex.

Consider an ultrametric caterpillar tree. There exists a vertex-order based MST VMST_{max} which has a star topology that can be constructed by contracting edges between each hidden vertex and one labeled vertex that is in the surrogate vertex set of each hidden vertex (see Figure 4.5 A). VMST_{max} has the maximum number of leaves and does not contain any information regarding the splits of the phylogenetic tree.

Instead, if a vertex-order based MST VMST_{min} was to be constructed by contracting edges between each hidden vertex h and a labeled vertex that is adjacent to h, then the number of the vertex groups would be $n-2$, where n is the number of vertices in the phylogenetic tree. VMST_{min} has the minimum number of leaves (two), and the maximum amount of split information about the phylogenetic tree.

Consider a phylogenetic tree $T = (V_T = \{\mathcal{L}_T, \mathcal{H}_T\}, E_T)$ which is an ultrametric balanced tree. For each leaf $l_1$in $\mathcal{L}_T$ there is another leaf l_2 in $\mathcal{L}_T$ such that l_1 and l_2 are adjacent to the same hidden vertex h in $\mathcal{H}_T$. Since l_1 and l_2 are closest to h, the surrogate vertex of h is either l_1 or l_2. In each VMST of T, either l_1 or l_2 will be a leaf in the VMST. Since this is true for all leaves in $\mathcal{L}_T$, each VMSTs of T will have $|\mathcal{L}_T|/2$ leaves (see Figure 4.5 B).

Whether or not the phylogenetic trees that are estimated from real data are ultrametric depends on the set of organisms that are being studied. Genetic sequences that are sampled from closely related organisms have been estimated to undergo substitutions at a similar rate, resulting in ultrametric phylogenetic trees (dos Reis et al., 2016). With respect to the phenomenon of adaptation by natural selection, phylogenetic trees are caterpillar-like if there is strong selection; the longest path from the root represents the best-adapted lineage.

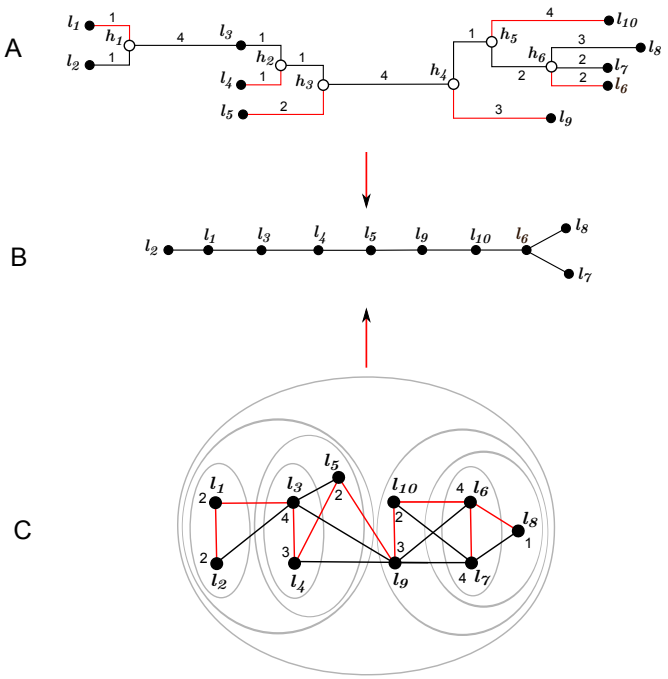


Figure 4.6: Panel A shows a generally labeled phylogenetic tree T with surrogate vertices selected such that the edge contraction would construct the VMST with the minimum number of leaves shown in Panel B. Panel C shows the VMST (in red) superimposed with the common laminar family and the MST union graph. Additionally, each vertex has been labeled with the corresponding $\delta_{\max}$.

4.4.3 Overview of our approach

Our approach to selecting optimal VMSTs makes use of three notions,(i), the maximum degree $\delta_{\max}$ of each vertex across all MSTs, (ii), the so-called *MST union graph* which is a graph containing all the edges that are present in at least one MST, and, (iii), a common structure over the MSTs that can be defined as a laminar family.

The intuition behind our approach is as follows. From Lemma 1 it follows that each non-leaf vertex of a VMST is a surrogate vertex. Thus we want to choose a vertex order such that we maximize the number of distinct surrogate vertices. In Section 4.5, we show that such a vertex order can be constructed by arranging vertices in order of non-decreasing $\delta_{\max}$. In Section 4.4.4 we show how the common laminar family and the MST union graph can be used to compute $\delta_{\max}$. The construction is exemplified graphically in Figure 4.6.

On a related note, the general problem of selecting an MST with the minimum number of leaves (MLMST) is in **NP-complete** by reduction from the Hamiltonian path problem. MLMST specializes the problem of finding spanning trees with the minimum number of leaves which is also in **NP-complete** by a similar reduction (Salamon and Wiener, 2008).

4.4.4 A structure that is common to all MSTs of a graph

In this subsection we will prove the existence of a so-called common laminar family over the vertex set of an edge-weighted graph G. A collection $\mathcal{F}$ of subsets of a set S is a laminar family over S if, for any two intersecting sets in $\mathcal{F}$, one set contains the other. That is to say, for each pair S_1, S_2 in $\mathcal{F}$ such that $|S_1| \leq |S_2|$, either $S_1 \cap S_2 = \varnothing$, or $S_1 \subset S_2$.

The common laminar family defines a representation of a tree structure that is common to each MST of G. The notion of a laminar family has been utilized previously by Ravi and Singh (2006) for designing an approximation algorithm for computing a minimum-degree MST.

Semple and Steel (2003) note that each rooted phylogenetic tree can be uniquely described as a laminar family over the set of labeled vertices. Laminar family representations of rooted phylogenies are used for comparing and combining information from multiple rooted phylogenetic trees. Later in this section we show that the laminar family representation of an ultrametric tree is equivalent to the common laminar family.

Lemma 3. *Given an edge-weighted graph $G = (V, E)$ with k distinct weight classes $W = \{w_1, w_2, \ldots, w_k\}$, and an MST M of G, let F_i be the forest that is formed by removing all edges in G that are heavier than w_i. Let C_i be the collection comprising the vertex set of each component of F_i. Consider the collection $\mathcal{F}$ which is constructed as follows: $\mathcal{F}_C = \left\{\cup_{i=1}^{k} C_i\right\} \cup V$. The following is true:*

1. $\mathcal{F}_C$ is a laminar family over V
2. Each vertex set in $\mathcal{F}_C$ induces a connected graph in each MST of G

Proof. (i). Consider any two vertex sets V_1 and V_2 in $\mathcal{F}$. Let w_1 and w_2 be the weights of the heaviest edges in the subgraphs of M that are induced by V_1 and V_2, respectively. Let F_1 and F_2 be the forests that are formed by removing all edges in M that are heavier than w_1 and w_2, respectively. Let C_1 and C_2 be the collections comprising the vertex set of each component in F_1 and F_2, respectively.

By construction, we have $V_1 \in C_1$ and $V_2 \in C_2$. Consider the case where $w_1 = w_2$. Since $C_1 = C_2$, it follows that $V_1 \cap V_2 = \varnothing$. If $w_1 \neq w_2$, then without loss of generality, let $w_1 < w_2$. F_2 can be constructed by adding to F_1 all edges in M that are no heavier than w_2. The vertex set of each component in F_1 that is not in F_2 induces a connected subgraph in exactly one component of F_2. If $V_1 \in C_1 \cap C_2$ then $V_1 \cap V_2 = \varnothing$. Otherwise, if $V_1 \in C_1 \backslash C_2$, then V_1 is a subset of exactly one set in C_2. This implies that either $V_1 \subset V_2$, or $V_1 \cap V_2 = \varnothing$. Thus $\mathcal{F}_C$ is a laminar family over V.

(ii). Let V_i be the vertex set of a component in the graph G_i of G that is created by removing all edges in G_i that are heavier than w_i. It follows that V_i induces a connected graph in each minimum spanning forest of G_i. Consider an MST M of G. Removing all edges in M that are heavier than w_i constructs a minimum spanning forest F of G. Thus V_i induces a connected graph in M. It follows that V_i induces a connected graph in each MST of G. By construction $V_i \in \mathcal{F}_C$. □

4.4.5 Ultrametric trees

Semple and Steel (2003) note that the hierarchical structure of a rooted tree can be represented using a laminar family. We show that the laminar family $\mathcal{F}_T$ that represents an ultrametric tree T is equivalent to the laminar family $\mathcal{F}_C$ that is common to all the MSTs of the distance graph associated with T.

Lemma 4. *We are given an ultrametric tree $T = (V_T, E_T)$ and the corresponding distance graph G. Let $\mathcal{F}_C$ be the laminar family that is common to each MST of G. Let $\mathcal{F}_T$ be the laminar family representation of T. The following is true.*

$$\mathcal{F}_T = \mathcal{F}_C.$$

Proof. Consider a vertex set $V_w \subset \mathcal{F}_T$. Let w be the largest distance between vertices in V_w. Consider the forest F_w that is constructed by removing all edges in G that are heavier than w. V_w induces a connected component C_w in F_w since each pairwise distance between vertices in V_w is not larger than w. Since the distance between each vertex in V_w and each vertex in $V_T \backslash V_w$ is larger than w, it follows that C_w does not contain any vertex that is not in V_w. Since the common laminar family $\mathcal{F}_C$ contains the vertex set of each component in F, it follows that $V_w \subset \mathcal{F}_C$. Since this is true for each vertex set in $\mathcal{F}_T$, it follows that $\mathcal{F}_T = \mathcal{F}_C$. □

Note that the laminar family representation $\mathcal{F}_T$ of a rooted tree, and the corresponding common laminar family $\mathcal{F}_C$, are not equivalent in general. See Figure 4.7 for an example.

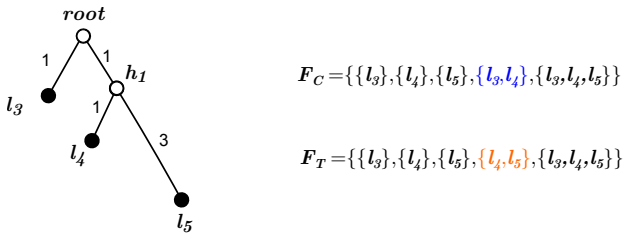


Figure 4.7: The equivalence between the laminar family representation $\mathcal{F}_T$ of a rooted phylogenetic tree, and the common laminar family $\mathcal{F}_C$, is not true in general.

4.4.6 Computing the common laminar family and the MST union graph

In this subsection we present an algorithm for constructing the common laminar family and the MST union graph. The MST union graph of a graph G is the subgraph of G that contains all the edges that are present in at least one MST of G.

Algorithm 4: Construct the common laminar family $\mathcal{F}_C$ and the MST union graph G_U

Input: $G = (V_G, E_G)$
Initialize:
$M = (V_M, E_M) \leftarrow$ singleton graph over V_G
$G_U = (V_U, E_U) \leftarrow$ singleton graph over V_G
$\mathcal{F}_C \leftarrow V_G$
$E_{G\leq} \leftarrow$ edges in E_G that are sorted in order of increasing weight
$w_{\text{previous}} \leftarrow$ weight of the lightest edge in E_G
$V_w \leftarrow \varnothing$
Functions:
$C_M(v)$: Returns the vertex set of the component of M containing v
$F_M(v)$: Returns id of the component of graph M containing v
$U_M(u, v)$: Adds edge $\{u, v\}$ to E_M and updates component ids
for $\{u, v\}$ in $E_{G\leq}$
  $w_{\text{current}} \leftarrow$ weight of $\{u, v\}$
  if $w_{\text{current}} > w_{\text{previous}}$
    for $\{u, v\}$ in E_w
      if $F_M(u) \neq F_M(v)$
        $U_M(u, v)$
    $E_w \leftarrow \varnothing$
    for v in V_w
      $\mathcal{F}_C \leftarrow \mathcal{F}_C \cup C_M(v)$
    $V_w \leftarrow \varnothing$
  else if $F_M(u) \neq F_M(v)$
    $E_U \leftarrow E_U \cup \{\{u, v\}\}$ // ensures that G_U contains all the edges that are present in at least one MST of G
    $E_w \leftarrow E_w \cup \{\{u, v\}\}$
    $V_w \leftarrow V_w \cup \{u\}$
  $w_{\text{previous}} \leftarrow w_{\text{current}}$
Output: $\mathcal{F}_C, G_U = (V_U, E_U)$

Lemma 5. *Given an edge-weighted graph $G = (V_G, E_G)$ with k distinct weight classes $W = \{w_1, w_2, \ldots, w_k\}$, the outputs $\mathcal{F}_C$ and G_U of Algorithm 4 are the common laminar family of G, and the MST union graph of G, respectively.*

Proof. Algorithm 4 adds edges to the singleton graph M in order of increasing weight, in such a way that M does not contain any cycles. From Kruskal (1956), we know that M is an MST of G.

Consider the forest F_i that is constructed by removing all edges in M that are heavier than w_i. By construction, $\mathcal{F}_C$ includes the vertex set of each component of F_i. Let C_i be the collection comprising the vertex set of each component of F_i. It follows that $\mathcal{F}_C = \left\{\cup_{i=1}^{k} C_i\right\} \cup V$. From Lemma 3, we know that $\mathcal{F}_C$ is the common laminar family of G.

E_U is constructed by adding the lightest edges that are incident to vertices in different components. The cut property of MSTs states that given a graph $G = (V, E)$, for each pair V_1, V_2 of disjoint sets such that $V_1 \cup V_2 = V$, each MST of G contains one of the lightest edges which have one endpoint in V_1 and the other endpoint in V_2. It follows that each edge in E_U is present in at least one MST of G. □

4.5 Selecting VMSTs with the minimum number of leaves

4.5.1 Implicitly selecting optimal surrogate vertices

Lemma 6. *We are given a phylogenetic tree T, the corresponding distance graph $G = (V, E)$. Let $\mathcal{F}_C$ be the common laminar family of G.* Let $G_U = (V_U, E_U)$ be the MST union graph of G. *Let h be a hidden vertex in T such that there is a leaf l in $S(h)$ that is adjacent to h. Let V_i be a vertex set in $\mathcal{F}$ and let w_i be the corresponding edge weight. Then the following is true:*

1. Let $N(v)$ be the set of all vertices that are adjacent to vertex v in G_U. Let $C(v)$ be a smallest sub-collection of $\mathcal{F}$ that covers $N(v)$ but not v. Among all MSTs, the maximum vertex degree $\delta_{\max}(v)$ of v is $|C(v)|$.

2. $\delta_{\max}(l) \leq \delta_{\max}(v)$ for each vertex v in $S(h)$

Proof. (i). Let $N(v) = \{j_1, j_2, \ldots, j_k\}$ be the neighbors of v in G_U. Let M be an MST of G. Let $C(v) = \{c_1, c_2, \ldots, c_m\}$ be a smallest sub-collection of $\mathcal{F}$ that covers $N(v)$ and does not include v.

Let $C(v)$ contain a set c_i that covers multiple vertices in $N(v)$. Let j_1 and j_2 be any two vertices in c_i. Let w_i be the heaviest weight on the path between j_1 and j_2 in M. The edges $\{v, j_1\}$ and $\{v, j_2\}$ are heavier than w_i. If they were not, then we would have $v \in c_i$. Since v, j_1 and j_2 are on a common cycle, each MST of G can only contain one of the two edges $\{v, j_1\}$, and $\{v, j_2\}$. It follows that, for each set $c_i \in C(v)$, each MST can contain at most one edge which is incident to v and to a vertex in c_i. Thus the maximum number of edges that can be incident to v in any MST is the number of vertex sets in $C(v)$, *i.e.*, $\delta_{\max}(v) = |C(v)|$.

(ii). Let $N(l)$ and $N(v)$ be the neighbors of l and v in G_U, respectively. Let $j \in N(l) \backslash S(h)$. The weight of the edge $\{j, l\} \in E_U$ is given by $d_T(j, l)$. $d_T(j, h) > d_T(v, h)$ since $j \notin S(h)$. Thus $d_T(l, j) > d_T(l, v)$, and consequently $v \in N(l)$. We have $d_T(j, l) = d_T(j, h) + d_T(h, l) = d_T(j, h) + d_T(h, v) = d_T(j, v)$. Consider the MST $M = (V_M, E_M)$ that contains the edges $\{l, v\}$ and $\{l, h\}$. Consider the spanning tree M' that is formed by removing $\{l, h\}$ from E_M and adding $\{v, h\}$. M' and M have the same sum of edge weights. Thus we also have $j \in N(v)$. Consequently $N(l) \subseteq N(v)$. Let $C(l)$ and $C(v)$ be the smallest sub-collections of $\mathcal{F}$ such that $C(l)$ covers $N(l)$ but does not contain l, and $C(v)$ covers $N(v)$ but does not contain v. $C(v)$ covers both $N(l)$ and $N(v)$ since $N(l) \subseteq N(v)$. Thus $|C(l)| \leq |C(v)|$. From part (i), we know that $|C(l)| = \delta_{\max}(l)$ and $|C(v)| = \delta_{\max}(v)$. Thus $\delta_{\max}(l) \leq \delta_{\max}(v)$. □

4.5.2 Computing a VMST with the minimum number of leaves

Theorem 2. *We are given a phylogenetic tree T and the corresponding distance graph G. Let M be the vertex order based MST that is computed using Algorithm 5. Among all VMSTs of G, M has the minimum number of leaves.*

Algorithm 5: Construct a minimum leaves VMST (MLVMST)

Input: $G = (V, E)$
$\mathcal{F}_C \leftarrow$the common laminar family of G
$\mathcal{F}_C^{\supseteq} \leftarrow$sets of $\mathcal{F}_C$ ordered in order of decreasing size
$G_U \leftarrow$the MST union graph of G
$\delta_{max} \leftarrow$empty array
for i in V
  $N(i) \leftarrow$neighbors of i in G_U
  $\delta_{max}(i) \leftarrow 0$
  for C in $\mathcal{F}_C^{\supseteq}$:
    if $C \cap N(i) \neq \varnothing$ and $C \cap \{i\} = \varnothing$
      $\delta_{max}(i) \leftarrow \delta_{max}(i) + 1$
      $N(i) \leftarrow N(i) \backslash C$
$<_* \leftarrow$ A total order over V such that $u_{<_*} < v_{<_*} \implies \delta_{max}(u) \leq \delta_{max}(v)$
$M_* \leftarrow$ VMST constructed by applying Algorithm 3 to $(G, <_*)$
Output: M_*

Proof. Let $S(h)$ be the set of vertices that are closest to h w.r.t the tree metric d_T that is associated with T. From Lemma 6(ii), we know that if there is a leaf l in $S(h)$ that is adjacent to h in T then, among all vertices in $S(h)$ $\delta_{\max}(l)$ is smallest. By construction of $<_*$, among all vertices in $S(h)$, the vertex rank $l_{<_*}$of l is the smallest. It follows that Algorithm 5 implicitly selects l as the surrogate vertex of h. Since each leaf in T is adjacent to at most one hidden vertex, the vertex order that is selected by Algorithm 5 maximizes the number of distinct leaves that are selected as surrogate vertices. M is constructed by contracting the path in T between each hidden vertex and the corresponding surrogate vertex. Contracting the path between a hidden vertex and the corresponding surrogate vertex increases the degree of the surrogate vertex. Thus, among all vertex order based MSTs, M has the minimum number of leaves. □

4.5.3 Implementation details and time complexity analysis

Algorithm 5 takes as input an edge-weighted graph $G = (V, E)$ and performs the following actions. First, the common laminar family $\mathcal{F}_C$ and the MST union graph G_U are constructed by applying Algorithm 4 to G. Subsequently, a vertex order $<_V$ is computed on the basis of $\mathcal{F}_C$ and G_U. Finally, a VMST is constructed by applying Algorithm 3 to $(G, <_V)$.

Algorithms 3 and 4 are variants of Kruskal's algorithm and were implemented using a disjoint-set data structure with balanced Union, and Find with path compression (Tarjan, 1975). The functions F_M and U_M correspond to a Find operation and a Union operation, respectively. A disjoint-set data structure can be represented as a forest with self-loops and directed edges. Each vertex points to its parent. The root of a component points to itself. A Find operation on a vertex v deletes the edge (v, v_{p_f}) that enters its former parent v_{p_f} and adds the edge (v, v_r) that enters the root of the component that contains v. A Union operation takes as input the roots of two components and creates an edge that exits the root of the smaller component and enters the root of the larger component, breaking ties arbitrarily. The function $C_M(u)$ is designed to return the set of vertices that are in the same component as u. C_M is implemented as follows. We store the vertex set of a component in the root of the component. Each time we perform a union operation $U_M(r_1, r_2)$ we combine the vertex sets and store the combined vertex set in the root of the component containing $r_1$and r_2.

The main steps of Algorithms 3 and 4 are (i), sorting $O(n^2)$ edges and, (ii), performing $O(n^2)$ Find operations and $O(n)$ Union operations, where n is the number of vertices in V. Step (i) can be done using mergesort in time $O(n^2 \log n^2)$, which simplifies to $O(n^2 \log n)$. Step (ii) takes time $O(n^2 \alpha(n^2, n))$ where α is the inverse of Ackermann's function (Tarjan, 1975). Since $\alpha(n^2, n) < \log n$, both the algorithms complete their computations in time $O(n^2 \log n)$.

In addition to calling Algorithms 3 and 4, Algorithm 5 sorts the sets in $\mathcal{F}_C$ and computes $\delta_{\max}$ for each vertex in V. $\mathcal{F}_C$ has $O(n)$ sets which can be sorted using mergesort in time $O(n \log n)$. For each vertex, $\delta_{\max}$ can be computed in time $O(n)$.

Thus the total time complexity of Algorithm 5 is $O(n^2 \log n)$.

4.6 Summary and Outlook

The current chapter identified the conditions under which MSTs constructed using tree-distances d_T share a topological relationship with phylogenetic trees T. The topological relationship that was introduced by Choi et al. (2011) states that MSTs can be constructed by contracting paths in phylogenetic trees between hidden vertices and their corresponding surrogate vertices. We showed that the indeterminacy in the proof by Choi et al. (2011) occurred because surrogate vertices were not properly defined in the case that there are multiple labeled vertices that could each be the surrogate vertex of a hidden vertex. We removed this indeterminacy by ensuring that surrogate vertices are uniquely defined on the basis of a vertex order over labeled vertices. Subsequently we provided an algorithm for constructing vertex-order based MSTs (VMSTs) that are guaranteed to share the topological relationship with phylogenetic trees. We related the number of leaves in a VMST to the number of non-trivial splits, which showed that VMSTs with the minimum number of leaves contained the maximum amount of information about phylogenetic trees. Finally we provided a polynomial-time algorithm for constructing VMSTs with the fewest number of leaves.

The proofs in this chapter required the use of tree-distances. Empirical estimates of evolutionary distances such as the Hamming distance, or model-based maximum likelihood distances are not additive in general. In the following Chapter we present an MST-based framework called MST-backbone that constrains the search for maximum-likelihood phylogenetic trees. In the following chapter we do not assume that distances are additive, and we do not make use of VMSTs in MST-backbone.

Chapter 5

Structural EM under the general Markov model via an MST backbone

The work that is presented in this chapter is unpublished.

The current approach to inferring model-based phylogenetic trees involves searching through tree space via tree-modification operations. The size of tree space is exponential in number of leaves. Consequently, popular software for phylogenetic tree inference make simplifying model assumptions about sequence evolution in order to reduce the number of free parameters, and facilitate fast search through tree space. The most commonly adopted assumption are stationarity and homogeneity. The stationarity assumption is violated by empirical observations of large variation in GC content (Agashe and Shankar, 2014). Currently available methods for inferring trees under non-stationary Markov models are limited to small data sets comprising less than 100 species due to high computational cost (Foster, 2004). The current chapter introduces a minimum spanning tree (MST) framework called MST-backbone that constrains the search for model-based phylogenetic trees. We extend the structural expectation-maximization (SEM) framework for phylogenetic tree inference (Friedman et al., 2002) in order to enable searching through tree space for maximum-likelihood trees under the general Markov model (GM). The GM model is a non-stationary, non-homogeneous and non-reversible Markov model that allows GC content to evolve through evolutionary history. We show on simulated data that it is possible to reconstruct large phylogenetic trees without loss of accuracy. We validated our method on six empirical data sets. Additionally, we compared our method with IQ-TREE, a phylogeny inference software that implements the largest selection of time-reversivle and irreversible stationary homogenous CT-HMM. We found that the unrooted topology of trees reconstructed by MST-backbone(SEM-GM) and IQ-TREE were realistic for five data sets. The location of the root as inferred by the GM model was accurate for two experimental phylogeny data sets but showed signs of overfitting for two virus data sets. We found that trees that are rooted under the UNREST model using MST-backbone(SEM-GM)+UNR, and IQ-TREE are realistic for four data sets.

To the best of our knowledge, there is currently no method that performs tree search under the general Markov model (GM; Barry and Hartigan (1987)). We extend the structural expectation-maximization (EM) framework by Friedman et al. (2002) in order to perform tree search under the GM model. We refer to this method as SEM-GM. In order to improve the scalability of SEM-GM we designed an easily implementable threshold-based divide-and-conquer framework called MST-backbone. We refer to the MST constrained tree-search method as MST-backbone(SEM-GM).

The structure of this chapter is as follows. SEM-GM and MST-backbone are described in Section 5.1 and Section 5.2, respectively. We performed a comparative analysis of MST-backbone(SEM-GM) with three popular software packages: FastTree v2.1.10 (Price et al., 2010), RAxML-NG v0.8.1 (Kozlov et al., 2019), and IQ-TREE v1.6.1 (Nguyen et al., 2015) using sequences that were simulated under non-stationary Markov models. We validated MST-backbone(SEM-GM) on empirical data and discovered that the location of the root was unrealistic for a majority of data sets. Subsequently, we performed model selection using BIC and

found that the UNREST model was selected for five empirical data sets. We compared the trees rooted under UNREST (Yang, 1994b) with trees inferred by IQ-TREE under non-reversible stationary and homogeneous Lie Markov models. Results of the comparative analysis on simulated data is described in Section 5.4. The comparative analysis with IQ-TREE on empirical data is described in Section 5.5. The chapter concludes with a summary of results, and an outlook on how the methods described in this chapter might impact current research in phylogeny inference.

5.1 A structural EM algorithm for the general Markov model

Expectation-maximization algorithms (EM) are a class of algorithms that are commonly used to infer the parameters of models with hidden variables (Dempster et al., 1977). Friedman (1997) designed an EM algorithm called structural EM for inferring Bayesian networks with hidden vertices. Friedman et al. (2002) applied SEM to infer phylogenetic trees under the GTR model. In this chapter we adapt SEM to the GM model, and refer to the method as SEM-GM.

The search problem of finding maximum likelihood phylogenetic trees given sequences of extant species (leaf vertices) is *NP-hard* (Chickering, 1996; Roch, 2006; Chor and Tuller, 2006), and the corresponding decision problem is *NP-complete*. If sequences of all species (extinct and extant) were available then the decision problem that corresponds to the search problem of finding maximum likelihood phylogenetic trees is P because maximum likelihood fully labeled phylogenetic trees can be found in polynomial-time (Chow and Liu, 1968).

The general principle of EM algorithms is as follows. If there are no hidden variables then observed statistics are sufficient to compute optimal estimates of model parameters using closed-form solutions. If there are hidden variables then, given suboptimal estimates of model parameters, it is possible to compute expected statistics over hidden variables that are sufficient to optimize parameters using closed-form solutions. In the following Subsection, we consider the case where there are no hidden variables, i.e, we are interested in finding maximum likelihood fully labeled phylogenetic trees. Subsequently, we show how to compute expected statistics such that we can simplify the problem of finding maximum likelihood leaf-labeled phylogentic trees.

5.1.1 Inferring maximum likelihood fully labeled phylogenetic trees

Chow and Liu (1968) showed that the undirected version of a maximum-likelihood Markov model M on a fully labeled phylogenetic tree $T_{\text{full}}^{\text{MLE}} = (V_{\text{full}}, E_{\text{full}})$ is a maximum mutual information spanning tree $T_{\text{full}}^{\text{MI}}$ of the edge-weighted complete graph $G_{\text{all}} = (V_{\text{full}}, E_{\text{all}})$ over V_{full} with edges in E_{all} weighted using mutual information scores. A maximum mutual information spanning tree is a maximum weight spanning tree of G_{all}. The mutual information score $I(\mathcal{X}_u; \mathcal{X}_v)$ for any edge $\{u, v\}$ in E_{all} is computed as

$$I(\mathcal{X}_u; \mathcal{X}_v) = \sum_x \sum_y P(\mathcal{X}_u = x, \mathcal{X}_v = y) \log \left(\frac{P(\mathcal{X}_u = x, \mathcal{X}_v = y)}{P(\mathcal{X}_u = x) P(\mathcal{X}_v = y)} \right)$$

where the entries in the probability distributions $P(\mathcal{X}_u)$ and $P(\mathcal{X}_u, \mathcal{X}_v)$ are estimated using observed sequences as follows. Let $\overline{C_u}$ be is the fraction of sites at which the observed sequence $\text{seq}(u)$ equals x , and let $\overline{C_{(u,v)}}$ be the fraction of sites at which $\text{seq}(u)$ equals x and $\text{seq}(v)$ equals y. The observed count matrices $\overline{C_u}$ and $\overline{C_{(u,v)}}$ are computed as

$$\overline{C_u}(x) = \frac{1}{k} \sum_{i=1}^{k} \delta(\mathcal{X}_u^i, x), \tag{5.1}$$

and

$$\overline{C_{(u,v)}}(x, y) = \frac{1}{k} \sum_{i=1}^{k} \delta(\mathcal{X}_u^i, x) \times \delta(\mathcal{X}_v^i, y) \tag{5.2}$$

where $\delta(x, y)$ is the Kroenecker delta function that is one if x equals y, and is zero otherwise. $P(\mathcal{X}_u = x)$ is estimated as $\overline{C_u}(x)$, and $P(\mathcal{X}_u = x, \mathcal{X}_v = x)$ is estimated as $\overline{C_{(u,v)}}(x, y)$.

Consider a complete graph G^{MI} over all labeled vertices whose edges are weighted with mutual information scores. A maximum mutual information spanning tree T^{MI} of G^{MI} can be computed in polynomial time using Prim's algorithm (Prim, 1957). Given T^{MI}, the maximum-likelihood estimate of the fully labeled tree $T^{\text{MLE}}_{\text{full}}$ can computed in polynomial time as

$$T^{\text{MLE}}_{\text{full}} = \arg \max_{\rho \in V_{\text{full}}} \prod_{i=1}^{k} \pi^{\text{MLE}}_{\rho}(\mathcal{X}^i_{\rho}) \prod_{(u,v) \in E_{\rho}} P^{\text{MLE}}_{(u,v)}(\mathcal{X}^i_u, \mathcal{X}^i_v) \tag{5.3}$$

where edges E_ρ are directed away from the vertex ρ in V_{full} that is selected as the root, and the maximum likelihood estimate of model parameters M are given by the following closed-form solutions (Koller and Friedman, 2009):

$$\pi^{\text{MLE}}_{\rho}(x) = \overline{C}_{\rho}(x), \text{ and} \tag{5.4}$$

$$P^{\text{MLE}}_{(u,v)}(x, y) = \frac{\overline{C}_{(u,v)}(x, y)}{\overline{C}_u(x)} \tag{5.5}$$

In the following Subsection we describe how to perform the expectation step.

5.1.2 Expectation (E) step

Given a hidden Markov model M on a rooted leaf-labeled phylogenetic tree $T = (V_{T_\rho}, E_{T_\rho})$, let $\mathcal{L}_{T_\rho}$ and $\mathcal{H}_{T_\rho}$ be the set of labeled vertices and hidden vertices, respectively. The expected counts $E_M\left[\overline{C}_u(x)\right]$ of a hidden vertex u can be computed as follows (Koller and Friedman, 2009).

$$E_M\left[\overline{C}_u(x)\right] = \sum_{i=1}^{k} P(\mathcal{X}^i_u = x), \tag{5.6}$$

where $P(\mathcal{X}^i_u)$ is the marginal probability

$$P(\mathcal{X}^i_u) = \sum_{\mathcal{X}^i_h : h \in \mathcal{H}_{T_\rho} \setminus \{u\}} P(\{\mathcal{X}^i_v : v \in V_{T_\rho}\} | M)$$

Similarly, the expected counts $E_M\left[\overline{C}_{(u,v)}(x, y)\right]$ for any vertex pair u, v can be computed as follows

$$E_M\left[\overline{C}_{(u,v)}(x, y)\right] = \sum_{i=1}^{k} P(\mathcal{X}^i_u = x, \mathcal{X}^i_v = y), \tag{5.7}$$

where $P(\mathcal{X}^i_u, \mathcal{X}^i_v)$ is the marginal probability

$$P(\mathcal{X}^i_u, \mathcal{X}^i_v) = \sum_{\mathcal{X}^i_h : h \in \mathcal{H}_{T_\rho} \setminus \{u,v\}} P(\{\mathcal{X}^i_v : v \in V_{T_\rho}\} | M)$$

$P(\mathcal{X}^i_u, \mathcal{X}^i_v)$ for adjacent vertices u and v can be computed efficiently using belief propagation as described in Section 2.6.3.2. $P(\mathcal{X}^i_u, \mathcal{X}^i_v)$ for non-adjacent vertices u, v is computed in order of increasing unweighted path length from u to v on T_ρ as follows: consider a path $(v_1, v_2, \ldots, v_{n-1}, v_n)$ in the undirected version of T_ρ such that $P(\mathcal{X}^i_{v_1}, \mathcal{X}^i_{v_{n-1}})$ is known and we are interested in computing the marginal probability $P(\mathcal{X}^i_{v_1}, \mathcal{X}^i_{v_n})$. Note that variables $\mathcal{X}^i_{v_n}$ and $\mathcal{X}^i_{v_1}$ are independent if conditioned on $\mathcal{X}^i_{v_{n-1}}$. Thus

$P(\mathcal{X}^i_{v_n}|\mathcal{X}^i_{v_{n-1}}, \mathcal{X}^i_{v_1})$ equals $P(\mathcal{X}^i_{v_n}|\mathcal{X}^i_{v_{n-1}})$. This enables us to decompose the joint marginal probability $P(\mathcal{X}^i_{v_1}, \mathcal{X}^i_{v_{n-1}}, \mathcal{X}^i_{v_n})$ as $P(\mathcal{X}^i_{v_1}, \mathcal{X}^i_{v_{n-1}})P(\mathcal{X}^i_{v_n}|\mathcal{X}^i_{v_{n-1}})$.

The marginal probability $P(\mathcal{X}^i_{v_1}, \mathcal{X}^i_{v_n})$ can be computed as follows:

$$P(\mathcal{X}^i_{v_1}, \mathcal{X}^i_{v_n}) = \sum_{\mathcal{X}^i_{v_{n-1}}} P(\mathcal{X}^i_{v_1}, \mathcal{X}^i_{v_{n-1}}, \mathcal{X}^i_{v_n}) = \sum_{\mathcal{X}^i_{v_{n-1}}} P(\mathcal{X}^i_{v_1}, \mathcal{X}^i_{v_{n-1}})P(\mathcal{X}^i_{v_n}|\mathcal{X}^i_{v_{n-1}}) \tag{5.8}$$

where $P(\mathcal{X}^i_{v_n}|\mathcal{X}^i_{v_{n-1}})$ is computed from the joint probability distribution $P(\mathcal{X}^i_{v_n}, \mathcal{X}^i_{v_{n-1}})$.

5.1.2.1 Maximization (M) step

The maximization step of SEM-GM is similar to the case of fully labeled phylogenetic trees with the difference being that observed count matrices are substituted with expected count matrices. The log-likelihood score increases subsequent to each iteration of an expectation step and a maximization step. The log-likelihood score is said to converge if successive increment of the log-likelihood score is smaller than a preselected log-likelihood threshold (known as convergence threshold). We used a convergence threshold of 10^{-2} log-likelihood units.

5.1.3 Transformation into a bifurcating tree

The rooted phylogenetic tree T_ρ that is computed by the maximization step is not necessarily a bifurcating tree. In the case that T_ρ is not a bifurcating tree we transform it into a bifurcating tree T_{bi}, and compute a GM model M_{bi} on T_{bi} using the steps described in Algorithm 6, such that the log-likelihood score remains unchanged.

The proof of correctness of Algorithm 6 is provided below.

Lemma 7. *The output of Algorithm 6 is a rooted bifurcating phylogenetic tree such that log-likelihood remains unchanged.*

Proof. The removal of edges incident to hidden vertices in cases (i) through (iii) results in the construction of the singleton hidden vertices that are used in cases (iv) through (vi).

We use conditional likelihood vectors in order to show that the likelihood score for any site i remains unchanged subsequent to the operations applied for each case considered by Algorithm 6. Note that $P_{(u,v)}$ denotes the conditional probability $P(\mathcal{X}_v|\mathcal{X}_u)$

Case (i)**:** T_ρ contains a hidden leaf h.

Let $D(v)$ be the set of children of the parent v of h. The conditional likelihood vector L^i_v is computed as follows

$$\begin{aligned}
L^i_v(x) &= \left(\sum_y P_{(v,h)}(y|x)L^i_h(y)\right) \prod_{d\in D(v)\setminus\{h\}} \left(\sum_z P_{(v,d)}(z|x)L^i_d(z)\right) \\
&= \left(\sum_y P_{(v,h)}(y|x)\right) \prod_{d\in D(v)\setminus\{h\}} \left(\sum_z P_{(v,d)}(z|x)L^i_d(z)\right) \qquad (L^i_h(y) \text{ equals one for all } y \text{ because } h \text{ is not observed}) \\
&= \prod_{d\in D(v)\setminus\{h\}} \left(\sum_z P_{(v,d)}(z|x)L^i_d(z)\right) \qquad (\text{Each row of } P_{(v,h)} \text{ sums to one})
\end{aligned}$$

Case (ii)**:** T_ρ contain a hidden vertex h with in-degree one and out-degree one.

Let u and v be the parent and child, respectively, of h. The conditional likelihood vector L^i_u is computed as follows

Algorithm 6: Transform non-canonical tree to leaf-labeled bifurcating tree

Input: A non-bifurcating tree $T_\rho = (V_{T_\rho}, E_{T_\rho})$, and a GM model $M = (\pi_\rho, \mathbf{P} = \{P_e : e \in E_{T_\rho}\})$
While T_ρ is a non-bifurcating tree **do:**

Case (i)**:** T_ρ contains a hidden leaf h
Let v be the parent of h
Remove edge (v, h), and matrix $P_{(v,h)}$

Case (ii)**:** T_ρ contain a hidden vertex h with in-degree one and out-degree one
Let u and v be the parent and child, respectively, of h
Remove edges (u, h) and (h, v), and add edge (u, v)
Remove matrices $P_{(u,h)}$ and $P_{(h,v)}$, and add matrix $P_{(u,v)}$ such that $P_{(u,v)} = P_{(u,h)}P_{(h,v)}$

Case (iii)**:** The root is a hidden vertex with out-degree one
Let π_ρ^{cur} denote the current root probability distribution
Compute new root probability distribution π_ρ^{new} as $\pi_\rho^{\text{new}}(y) = \sum_x \pi_\rho^{\text{cur}}(x) P_{(\rho,v)}(x, y)$
,where v is the child of ρ
Remove edge (ρ, v) and matrix $P_{(\rho,v)}$
Set v as the new root of T_ρ

Case (iv)**:** T_ρ contains a non-leaf labeled vertex l
Let $D(l)$ be the children of l, and let h be a singleton vertex.
Add edge (h, l), and add matrix $P_{(h,l)} = I$ (identity matrix)
for each vertex v in $D(l)$ **do**
  Remove edge (l, v), and add edge (h, v)
  Remove matrix and $P_{(l,v)}$, and add matrix $P_{(h,v)} = P_{(l,v)}$
if l has a parent u **then**
  Add edge (u, h) and matrix $P_{(u,h)} = P_{(u,l)}$
  Remove edge (u, l) and matrix $P_{(u,l)}$
else set h as the root

Case (v)**:** T_ρ contains a hidden vertex h_1 with out-degree greater than two
Let u and v be a two children of h_1 selected at random, and let h_2 be a singleton vertex.
Remove edges (h_1, u) and (h_1, v), and add edges (h_2, h_1), (h_2, u), and (h_2, v)
Remove matrices $P_{(h_1,u)}$ and $P_{(h_1,v)}$, and add matrices $P_{(h_2,u)}$, $P_{(h_2,v)}$, and $P_{(h_2,h_1)}$such that $P_{(h_2,u)} = P_{(h_1,u)}$, $P_{(h_2,v)} = P_{(h_1,v)}$, and $P_{(h_2,h_1)} = I$ (identity matrix)

Set T_{bi} as $T_{\text{bi}} \leftarrow T_\rho$, and set M_{bi} as $M_{\text{bi}} \leftarrow M$
Output: T_{bi} and the GM model M_{bi} on T_{bi}

$$
\begin{aligned}
L_u^i(x) &= \left(\sum_y P_{(u,h)}(y|x) L_h^i(y)\right) \prod_{d \in D(u)\setminus\{h\}} \left(\sum_z P_{(u,d)}(z|x) L_d^i(z)\right) \\
&= \left(\sum_y P_{(u,h)}(y|x) \sum_w P_{(h,v)}(w|y) L_v^i(w)\right) \prod_{d \in D(u)\setminus\{h\}} \left(\sum_z P_{(u,d)}(z|x) L_d^i(z)\right) \\
&= \left(\sum_w \left(\sum_y P_{(h,v)}(w|y) P_{(u,h)}(y|x)\right) L_v^i(w)\right) \prod_{d \in D(u)\setminus\{h\}} \left(\sum_z P_{(u,d)}(z|x) L_d^i(z)\right) \\
&= \left(\sum_w P_{(u,v)}(w|x) L_v^i(w)\right) \prod_{d \in D(u)\setminus\{h\}} \left(\sum_z P_{(u,d)}(z|x) L_d^i(z)\right) \qquad (\text{where } P_{(u,v)} = P_{(u,h)} P_{(h,v)})
\end{aligned}
$$

Case (iii)**:** The root is a hidden vertex with out-degree one.
Let π_ρ^{cur} denote the current root probability distribution.
The likelihood L^i for site i is computed as

$$
\begin{aligned}
L^i &= \left(\sum_x \pi_\rho^{\text{cur}}(x) L_\rho^i(x)\right) \\
&= \left(\sum_x \pi_\rho^{\text{cur}}(x) \sum_y P_{(\rho,v)}(y|x) L_v^i(y)\right) \\
&= \left(\sum_y \sum_x \pi_\rho^{\text{cur}}(x) P_{(\rho,v)}(y|x) L_v^i(y)\right) \\
&= \left(\sum_y \pi_\rho^{\text{new}}(y) L_v^i(y)\right) \qquad (\text{where } \pi_\rho^{\text{new}}(y) = \sum_x \pi_\rho^{\text{cur}}(x) P_{(\rho,v)}(y|x))
\end{aligned}
$$

Case (iv)**:** T_ρ contains a non-leaf labeled vertex l.
The conditional likelihood vector L_l^i of a labeled vertex l with children $D(l)$ is defined as

$$
\begin{aligned}
L_l^i(x) &= \mathcal{X}_l^i(x) \prod_{d \in D(l)} \left(\sum_z P_{(l,d)}(z|x) L_d^i(z)\right) \\
&= \mathcal{X}_l^i(x) \prod_{d \in D(h)\setminus\{l\}} \left(\sum_z P_{(h,d)}(z|x) L_d^i(z)\right) \qquad (5.9)
\end{aligned}
$$

where h is the hidden vertex that is referred to in the operations defined for case (iv).

Consider the conditional likelihood vector L_h^i of the hidden vertex h.

$$
\begin{aligned}
L_h^i(x) &= \prod_{d\in D(h)} \left(\sum_z P_{(h,d)}(z|x) L_d^i(z) \right) \\
&= \left(\sum_y P_{(h,l)}(y|x) L_l^i(y) \right) \prod_{d\in D(h)\setminus\{l\}} \left(\sum_z P_{(h,d)}(z|x) L_d^i(z) \right) \\
&= \left(\sum_y P_{(h,l)}(y|x) \mathcal{X}_l^i(y) \right) \prod_{d\in D(h)\setminus\{l\}} \left(\sum_z P_{(h,d)}(z|x) L_d^i(z) \right) \qquad \text{(because } L_l^i = \mathcal{X}_l^i \text{ for labeled leaves)} \\
&= P_{(h,l)}(x|x) \mathcal{X}_l^i(x) \prod_{d\in D(h)\setminus\{l\}} \left(\sum_z P_{(h,d)}(z|x) L_d^i(z) \right) \qquad \text{(because } P_{(h,l)} \text{ is the identity matrix)} \\
&= \mathcal{X}_l^i(x) \prod_{d\in D(h)\setminus\{l\}} \left(\sum_z P_{(h,d)}(z|x) L_d^i(z) \right) \qquad (5.10)
\end{aligned}
$$

The conditional likelihood vectors L_h^i and L_l^i are unchanged (see equation 5.9 and equation 5.10)

Case (v)**:** T_ρ contains a hidden vertex h_1 with out-degree greater than two. Let D_1 be the set of all children of h_1 prior to the operations performed in case (v). Let D_{uv} be $D_1\setminus\{u,v\}$

The conditional likelihood vector L_u^i prior to transformation operations is given by

$$
\begin{aligned}
L_{h_1}^i(x) &= \left(\sum_y P_{(h_1,u)}(y|x) L_u^i(y) \right) \left(\sum_y P_{(h_1,v)}(y|x) L_v^i(y) \right) \prod_{d\in D_{uv}} \left(\sum_z P_{(h_1,d)}(z|x) L_d^i(z) \right) \\
&= \left(\sum_y P_{(h_2,u)}(y|x) L_u^i(y) \right) \left(\sum_y P_{(h_2,v)}(y|x) L_v^i(y) \right) \prod_{d\in D_{uv}} \left(\sum_z P_{(h_1,d)}(z|x) L_d^i(z) \right) \\
&\qquad \text{(because } P_{(h_2,u)} = P_{(h_1,u)}\text{, and } P_{(h_2,v)} = P_{(h_1,v)}\text{)} \\
&= L_{h_2}^i(x) \prod_{d\in D_{uv}} \left(\sum_z P_{(h_1,d)}(z|x) L_d^i(z) \right) \\
&= \left(\sum_y P_{(h_2,h_1)}(y|x) L_{h_2}^i(y) \right) \prod_{d\in D_{uv}} \left(\sum_z P_{(h_1,d)}(z|x) L_d^i(z) \right) \qquad \text{(because } P_{(h_2,h_1)} \text{ is the identity matrix)}
\end{aligned}
$$

It follows that the log-likelihood score remains unchanged after the transformation operation. The algorithm terminates only if none of the cases apply, which will happen only if T_ρ is a leaf-labeled phylogenetic tree. □

5.1.4 Initial estimate of the general Markov model on a rooted phylogenetic tree

The expectation step requires an initial estimate T_ρ^0 of the rooted phylogenetic tree, and initial estimates of the parameters of a general Markov model M^0 on T_ρ^0. We compute the initial estimates as follows. An unrooted phylogenetic tree $T^{\text{NJ}} = (V_{T^{\text{NJ}}}, E_{T^{\text{NJ}}})$ is constructed by applying neighbor-joining (Saitou and Nei, 1987) to normalized Hamming distances. T^{NJ} is rooted along the midpoint of an edge that is selected at random in order to construct the initial estimate of the rooted tree T_ρ^0. The parameters of a general Markov model $M^0 = (\pi_\rho^0, \mathbf{P}^0 = \{P_e^0 : e \in E_{T_\rho^0}\})$ on T_ρ^0 are estimated as follows.

Given an assignment of characters to $\mathcal{X}_H$, the parsimony score $\text{pars}_{T_\rho}(\mathcal{X}_\mathcal{H})$ is the sum of character changes over edges

$$\text{pars}_{T_\rho}(\mathcal{X}_\mathcal{H}) = \sum_{i=1}^{k} \sum_{(u,v)\in E_{T_\rho}} \mathbb{1}(\mathcal{X}_u^i \neq \mathcal{X}_v^i)$$

We compute a character set $\mathcal{X}_\mathcal{H}$ that minimizes the parsimony score using Fitch-Hartigan's algorithm (Fitch, 1971; Hartigan, 1973). Given an assignment of characters that minimizes the parsimony score, we compute counts for each hidden vertex, and each vertex pair by treating each vertex as a labeled vertex. Subsequently, we compute the initial estimate of root probability using equation 5.4, and we compute the initial estimate of each transition matrix using equation 5.5.

5.2 MST-backbone: a divide-and-conquer framework for constraining search through tree space

Structural EM is a computationally expensive procedure (Friedman et al., 2002). We designed an MST-based framework called MST-backbone for constraining the search through tree space. The design of MST-backbone is inspired by a topological relationship between MSTs and unrooted phylogenetic trees (Choi et al., 2011) which is described below. Given an unrooted phylogenetic tree T and distances d_T that are additive in T, let M be an MST that is computed using d_T. The topological relationship can be described in terms of splits as follows. Each edge of M induces a split in T. It follows directly that each vertex set V_s that induces a subtree of M is the leaf-set of a subtree in T.

The correspondence between MSTs and phylogenetic trees holds for a subset of all possible MSTs (Kalaghatgi and Lengauer, 2017). Additionally, the topological correspondence holds only if MSTs are computed using tree distances. We do not assume that distances are additive in this Chapter. MST-backbone is a divide-and-conquer method that builds a global phylogenetic tree T by combining local phylogenetic trees. The main steps of MST-backbone are (i) computing a minimum spanning tree (MST); (ii) selecting smallest mutually independent vertex sets V_s and V_e comprising more than s vertices each such that (a) V_s induces a subtree in M, (b) $V_s \cup V_e$ induces a connected subgraph in M; (iii) computing a local phylogenetic tree t over $V_s \cup V_e$; (iv) updating the global phylogenetic tree T by adding edges in subtrees of t that are induced by V_s; (v) updating the MST; and (vi) rooting the global unrooted phylogenetic tree (see Figure 5.1 for an illustration).

Each step of MST-backbone is explained in detail below. See Algorithm 7 for an overview of MST-backbone.

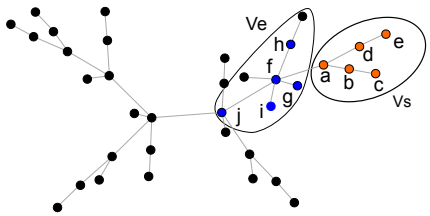
Initialization A minimum spanning tree (MST) M is computed using the Hamming distance for each sequence pair in $\mathcal{X}_\mathcal{L}$, where $\mathcal{L}$ represents the set of labeled vertices (species). We used Prim's algorithm (Prim, 1957) for computing the initial MST. The global phylogenetic tree $T = (V_T, E_T)$ is initialized by adding $\mathcal{L}$ to V_T, and setting E_T to the empty set $\varnothing$.

Selecting vertices of the MST Given a subtree size threshold s, select a smallest subtree $\tau_s = (V_s, E_s)$ of M comprising more than s sequences. Subsequently, perform a breadth-first-search (BFS) on M starting at the root of τ_s, and selecting s vertices V_e such that V_e and V_s are mutually exclusive.

Computing local phylogenetic trees An ML phylogenetic tree T_ρ^l over $V_s \cup V_e$ is inferred using SEM-GM. A maximum a posteriori (MAP) sequence $\text{seq}^{\text{MAP}}(h)$ for each hidden vertex h is computed as

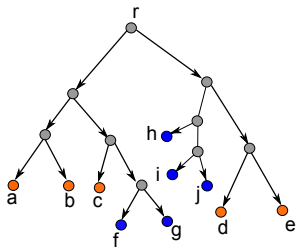
$$\mathcal{X}_h^{\text{MAP},i} = \underset{x}{\text{argmax}}\, P(\mathcal{X}_h^i = x)$$

(i)



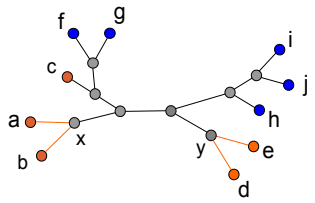
Select a vertex set *Vs* comprising *s* vertices such *Vs* induces a subtree in the MST. Start a breadth-first-search at the root of the subtree that is induced by *Vs* (vertex labeled *a*), and stop if *s* vertices have been visited such that *Ve* and *Vs* have no vertices in common.

(ii)



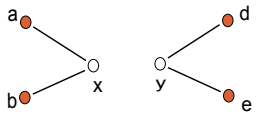
Fit a general Markov (GM) model on a rooted phylogenetic tree using SEM-GM. *r* indicates the root. Grey vertices represent the *maximum a posteriori* (MAP) estimate of ancestral sequences.

(iii)



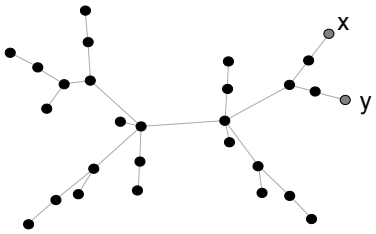
Suppress the root, and select the forest F comprising all non-singleton subtrees that are induced by the vertices *Vs*. Edges of selected subtrees are highlighted in orange. Vertices x and y are the roots of the selected subtrees.

(iv)



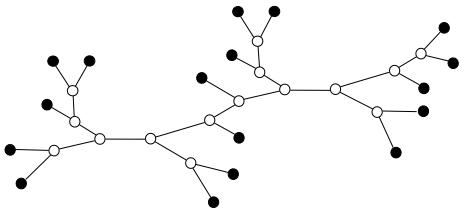
Update the global phylogenetic tree *T* by adding undirected edges in selcted subtrees.

(v)



Update the MST by removing vertices that are leaves in *F*, and adding the roots of selected subtrees (*x* and *y*)

(vi)



Iterate over steps (i) through (v), and stop if T is a connected graph. Root *T* at a vertex by fitting a GM model via SEM-GM such that the undirected topology is constrained to be *T*.

Figure 5.1: An illustration of the main steps of MST-backbone

Algorithm 7: Improving the scalability of SEM-GM using MST-backbone

Input: Distances d for species in $\mathcal{L}$, and a subtree threshold size s
Initialize: A global phylogenetic tree $T = (V_T, E_T)$ as $V_T \leftarrow \mathcal{L}$ and $E_T \leftarrow \varnothing$
$M = (V_M, E_M) \leftarrow$ a minimum spanning tree (MST) computed using d
While T is not connected **do**
  If M contains a subtree $\tau_s = (V_s, E_s)$ with more than s vertices, and $|V_M| - |V_s| > s$ **then**
    Select a smallest subtree τ_s of M containing more than s vertices
    Select s vertices V_e of M that are visited using a BFS starting at the root of τ_s such that $V_e \cap V_s = \varnothing$
    Compute a ML phylogenetic tree t_ρ over vertices $V_s \cup V_e$ by performing tree search using SEM-GM
    Construct t by suppressing the root of t_ρ, and select the largest non-singleton subtrees $\mathcal{T}_s$ of t that are induced by V_s
    Add edges of each selected subtree to global phylogenetic tree T
    Update M by removing leaves of subtrees, and adding the root of each subtree.
  Else
    Compute a ML phylogenetic tree t_ρ over all vertices in V_M using SEM-GM
    Construct an unrooted phylogenetic tree t by suppressing the root in t_ρ, and add all edges in t to T
Construct a rooted tree T_ρ using SEM-GM such that the undirected topology of T_ρ is constrained to be T.
Output: T_ρ

where $P(\mathcal{X}_h^i = x)$ is the marginal probability for observing character x at site i for the sequence seq(h) that is represented by vertex h, and $\mathcal{X}_h^{\text{MAP},i}$ is the character at position i in $\text{seq}^{\text{MAP}}(h)$.

The location of the root as inferred in T_ρ^l is not necessarily an optimal location of the root in the global phylogenetic tree. An unrooted phylogenetic tree T^l is constructed by suppressing the root in T_ρ^l. Subsequently, the subforest $\mathcal{F}_s$ of T^l is selected such that (i) the leaves of each subtree in $\mathcal{F}_s$ are a subset of V_s, and (ii) no component of $\mathcal{F}_s$ is a singleton vertex.

Updating the global phylogenetic tree The global phylogenetic tree $T^g = (V_{T_g}, E_{T_g})$ is updated as follows. Non-leaf vertices of $\mathcal{F}_s$ are added to V_{T_g}. All edges of $\mathcal{F}_s$ are added to E_{T_g}.

Updating the MST Vertices that are leaves in $\mathcal{F}_s$ are removed from V_M. The root of each subtree in $\mathcal{T}_s$ is added to V_M. MAP sequences of each root are used to compute the Hamming distances $d(r, v)$ for each root r in $\mathcal{F}_s$, and each vertex v in $V_M \cap \{V_s \cup V_e\}$. A new MST is computed using Prim's algorithm using the newly computed distances.

Rooting the global phylogenetic tree The global phylogenetic tree T^g is rooted using structural EM such that the undirected topology of the rooted tree T_ρ^g is restricted to be identical to T^g. Restricted SEM-GM (rSEM-GM) is performed as follows: (i) Root $T^g = (V_g, E_g)$ at a hidden vertex in V_g that is selected at random, (ii) Estimate MP sequences for hidden vertices, and initialize the parameters of a GM model (see Subsection 5.1.4), (iii) root T^g at a hidden vertex in V_g by maximizing expected log-likelihood score, (iv) compute MLE of GM model parameters (equations 5.4 and 5.5). Steps (iii) and (iv) are performed iteratively until the log-likelihood score converged.

5.3 Model selection

We implemented a model selection framework that selects the optimal number of rate matrices of a CT-HMM using BIC. Time-reversible models were not included in the model selection framework because we wanted to investigate whether or not phylogenetic trees that are rooted under CT-HMM are biologically meaningful. We validated phylogenetic trees using non-genetic information pertaining to evolutionary relationships. Model selection was performed as follows.

A rooted phylogenetic tree $T_\rho = (V_\rho, E_\rho)$ is inferred using MST-backbone(SEM-GM). A maximum a posteriori (MAP) estimate of ancestral sequence $\text{seq}^{\text{MAP}}(h)$ is inferred for each hidden vertex h in V_ρ. The edge length t_e of any edge $e = (u, v)$ in E_ρ is defined as the Hamming distance between sequences $\text{seq}^{\text{MAP}}(u)$, and $\text{seq}^{\text{MAP}}(v)$. The change in base frequency $\Delta_f(u, v)$ for each edge (u, v) in E_ρ is computed as follows:

$$\Delta_f(u,v) = \sum_{x \in \{A,C,G,T\}} |f_u(x) - f_v(x)| \tag{5.11}$$

where $f_u(x)$ is the fraction of characters in $\text{seq}^{\text{MAP}}(u)$ that are x. We construct CT-HMM on the basis of a base frequency threshold ϵ_f as defined below. Given a base frequency threshold ϵ_f, vertex-specific rate categories are assigned to each vertex in V_ρ as follows. The rate category of each vertex v in V_ρ is denoted by v_{cat}. The rate category ρ_{cat} of the root is set to zero. Vertices are visited by performing a preorder tree traversal. Each non-root vertex c that is visited is assigned the rate category of its parent p if $\Delta_f(p, c)$ is not larger than ϵ_f, otherwise c_{cat} is set to $p_{\text{cat}}+1$. A distinct UNR rate matrix Q^i is defined for each rate category i. The rate category $Q_{(p,c)}$ for each edge (p, c) in E_ρ is defined as $Q^{c_{\text{cat}}}$. The root probability distribution π_ρ is defined as the stationary distribution of the rate matrix $Q^{\rho_{\text{cat}}}$. The number of free parameters equals $(11 \times r) + |E_\rho|$, where r is the number of rate catergories.

Parameter estimation is performed by optimizing edge lengths, and rate matrices, iteratively until the log-likelihood score converges, using a convergence threshold of 10^{-2} log-likelihood units. Edge lengths are optimized using Newton-Raphson (details provided in Section B.1). Note that the transition matrix P_e for each e in E_ρ is computed as the matrix exponential $P_e = e^{Q_e t_e}$. It is necessary to constrain the elements of Q_e because it is possible to scale Q_e and t_e such that the product $Q_e t_e$ remains unchanged. The rate matrix Q^i for each rate category i is optimized using a simplex method called Nelder-Mead (Nelder and Mead, 1965) subject to the restriction that a non-diagonal element of Q^i was constrained to be one. Subsequently, each rate matrix is scaled in order to construct a normalized rate matrix. Threshold ϵ_f is initially set to the largest observed change in base composition. For each subsequent iteration, ϵ_f is set to the largest observed change in base composition that is smaller than the value of ϵ_f for the previous iteration. Model selection is terminated if BIC increases between successive iterations. We refer to the model selection framework described above as UNRmodelSelector in the following text.

5.4 Comparative analysis on simulated data

Current software performs tree-search under stationary and homogeneous Markov models. Empirical studies suggest that violation of the stationarity assumption leads to the construction of phylogenetic trees where species are incorrectly grouped close to each other due to similarity in base composition (Foster and Hickey, 1999; Nabholz et al., 2011). We compared the reconstruction accuracy of MST-backbone(SEM-GM) with three widely used software: FastTree v2.1.10 (Price et al., 2010), RAxML-NG v0.8.1 (Kozlov et al., 2019), and IQ-TREE v1.6.1 (Nguyen et al., 2015) using sequences simulated under non-stationary Markov models.

Phylogenetic trees that were used for simulating sequence evolution were generated using the R package apTreeshape v1.4.5 (Bortolussi et al., 2006). Rooted phylogenetic trees were generated by sampling from the uniform distribution over rooted trees. A general Markov model was constructed for each phylogenetic tree as follows. The root probability distribution π_ρ was generated by sampling each element of π_ρ from the uniform distribution $\text{U}(0, 1)$ and scaling such that the sum of the entries in π_ρ was equal to one. Transition matrices P were generated as follows. Each diagonal element of P was sampled from the uniform distribution $\text{U}(p_{\min}, 1)$, where $p_{\min}$ was varied from 0.99 to 0.7. Smaller values of $p_{\min}$ result in greater change in GC

Table 5.1: Percentage of simulated sequences that reject the null hypothesis of homogeneity in base composition. Median and inter-quartile range (shown in parentheses) are listed below.

$p_{\min}$	Median edge length	$p < 0.05$ (%)
0.7	0.15 (0.064)	89.8 (0.8)
0.8	0.1 (0.043)	83.2 (2.5)
0.9	0.05 (0.022)	68.0 (8.3)
0.95	0.025 (0.012)	53.5 (19.9)
0.99	0.005 (0.004)	19.9 (34.8)
0.995	0.0025 (0.0014)	2.5 (19.1)

content. Each non-diagonal element of P was sampled from the uniform distribution $\mathrm{U}(0, 1)$ and scaled such that the sum of elements of each row of P was equal to one.

In order to check for systematic error caused by using stationary homogeneous Markov models, we simulated non-stationary non-homogeneous sequence evolution. We varied the amount of sequence change per edge by setting $p_{\min}$ to 0.995, 0.99, 0.95, 0.9, 0.8, and 0.7. We measured edge length for each setting of $p_{\min}$ in order to facilitate comparison with simulation experiments that are usually performed using continuous-time Markov models (see Table 5.1). The length of each edge (u, v) was computed as the normalized Hamming distance between simulated sequences $\mathrm{seq}(u)$ and $\mathrm{seq}(v)$. The largest setting of $p_{\min}$ (0.995) corresponds to short edges (0.0025 subs/site). We included this setting because RAxML-NG and IQ-TREE perform extensive search and are good at recovering short edges. The smallest value of $p_{\min}$ (0.70) corresponds to large edges (0.15 subs/site). We included this setting in order to generate sequences with large change in base composition. In order to compare scalability we set $p_{\min}$ to 0.99, and varied the number of leaves from 1000 to 5000 in increments of 1000 leaves. Sequence length was set to 1000 base pairs which is comparable to the number of columns in the empirical alignments (ranging from 128 bp to 2214 bp, see Table 5.5) that we analyzed.

Chi-square test for significance in variation of base composition was performed for each simulation scenario using a p-value cut-off of 0.05 (implemented in the stats package of SciPy (Virtanen et al., 2020)). The percentage of sequences that exhibited significant deviation in base composition when compared with the average base composition ranged from 89.8% for $p_{\min}$ of 0.7 to 2.5% for $p_{\min}$ of 0.995 (see Table 5.1). The number of sequences that reject the null hypothesis are large (19.9% for average edge length of 0.005 subs/site) because the size of the tree is not taken into account when performing the test. The large inter-quartile-range of 19.9% for $p_{\min}$ of 0.95 and 34.8% for $p_{\min}$ of 0.99 can probably be attributed to variance in the imbalance of trees that are sampled from the uniform distribution over rooted trees. It may be possible that the number of sequences that show significant deviation in base composition increases with increasing imbalance of the model tree. The chi-square test that we have used is commonly used by practitioners of phylogeny inference (Nguyen et al., 2015).

5.4.1 Measures of reconstruction accuracy

We measured the extent to which the rooted topology, and the unrooted topology of the simulated trees were recovered using the following metrics. A rooted phylogenetic tree T_ρ specifies a hierarchical clustering $\mathcal{C}(T_\rho)$ over labeled vertices as follows.

$$\mathcal{C}(T_\rho) = \{\mathcal{L}_{\tau_v} : v \in V_{T_\rho}\}$$

where $\mathcal{L}_{\tau_v}$ is the set of labeled vertices in the subtree τ_v in T_ρ that is rooted at v. We measured reconstruction accuracy using recall values. $\mathrm{Re}_C(T^\rho, \hat{T^\rho})$ is the fraction of clusters in the model tree that are present in the estimated tree. $\mathrm{Re}_C(T^\rho, \hat{T^\rho})$ is computed as

Table 5.2: Accuracy ($\text{Re}_S^{\text{nontriv}}$) with which the unrooted topology was recovered. Median and inter-quartile range (shown in parentheses) of recall values are listed below, and were computed using 100 replicates.

$p_{\min}$	MST-backbone(SEM-GM)	FastTree	RAxML-NG	IQ-TREE
0.7	0.88 (0.014)	0.81 (0.035)	0.75 (0.04)	0.82 (0.061)
0.8	0.95 (0.009)	0.87 (0.044)	0.9 (0.03)	0.93 (0.063)
0.9	0.99 (0.004)	1.0 (0.0)	1.0 (0.0)	1.0 (0.034)
0.95	1.0 (0.001)	1.0 (0.0)	1 (0.0)	1.0 (0.036)
0.99	0.98 (0.0045)	0.99 (0.005)	0.98 (0.004)	0.98 (0.034)
0.995	0.92 (0.013)	0.92 (0.013)	0.92 (0.013)	0.92 (0.014)

$$\text{Re}_C(T^\rho, \hat{T^\rho}) = \frac{|\mathcal{C}(T_\rho) \cap \mathcal{C}(\hat{T}_\rho)|}{|\mathcal{C}(T_\rho)|} \tag{5.12}$$

$\text{Re}_S(T^\rho, \hat{T^\rho})$ is the fraction of splits in the model tree that are present in the estimated tree. $\text{Re}_S(T^\rho, \hat{T^\rho})$ is computed as

$$\text{Re}_S(T, \hat{T}) = \frac{|\mathcal{S}(T) \cap \mathcal{S}(\hat{T})|}{|\mathcal{S}(T)|} \tag{5.13}$$

where $\mathcal{S}(T)$ is the set of splits in the unrooted phylogenetic tree T that is constructed by suppressing the root of simulated tree T_ρ. A split is said to be a trivial split if the smallest side of the split contains one labeled vertex. A singleton cluster is said to be a trivial cluster. $\text{Re}_C^{\text{nontriv}}$ and $\text{Re}_S^{\text{nontriv}}$ are recall values that have been computed using nontrivial clusters and nontrivial splits, respectively. Re_C^{all} and Re_S^{all} are recall values that have been computed using all clusters, and all splits, respectively.

5.4.2 Systematic error due to model misspecification

A Markov process is an information destroying process. The amount of sequence information that is lost is proportional to edge length. A general trend in recall values is that each method has high recall values for $p_{\min}$ values that range from 0.9 to 0.99, and the recall values decrease as $p_{\min}$ is lowered to 0.8, and 0.7, and recall values decrease as $p_{\min}$ is increased to 0.995 (see Table 5.2). The relatively lower recall values for $p_{\min}$ of 0.7 and 0.8 are probably because of greater information loss over each edge. The reduction in recall value as $p_{\min}$ is increased to 0.995 is probably because there are some edges where no change takes place.

That said, MST-backbone(SEM-GM) outperformed competing methods at reconstructing the unrooted topology for $p_{\min}$ values of 0.7 and 0.8. The relatively better performance of MST-backbone(SEM-GM) for small values of $p_{\min}$ is probably because of large amount of change in base composition that takes place across each edge. The high Re_C of all methods for simulation scenarios where $p_{\min}$ ranges from 0.9 to 0.99 seems odd at first glance given that the number of sequences that deviate in base composition ranges from 20% to 68%. The results suggest that edge length is a better predictor of reconstruction accuracy than the total number of sequences that exhibit significant deviation in base composition.

MST-backbone(SEM-GM) + rSEM-GM has similar Re_C values when compared with IQ-TREE except for the marginally worse performance of MST-backbone(SEM-GM) + rSEM-GM at $p_{\min}$ of 0.95 (Re_C of 0.94 vs 0.95), and the marginally better performance of MST-backbone(SEM-GM) + rSEM-GM at $p_{\min}$ of 0.7 (Re_C of 0.83 vs 0.81).

Table 5.3: Accuracy ($\text{Re}_C^{\text{nontriv}}$) with which the rooted topology was recovered. Median and inter-quartile range (shown in parentheses) of recall values are listed below, and were computed using 100 replicates.

$p_{\min}$	average edge length	MST-backbone(SEM-GM) + rSEM-GM	IQ-TREE
0.7	0.15 (0.064)	0.83 (0.033)	0.81 (0.042)
0.8	0.1 (0.043)	0.9 (0.045)	0.9 (0.046)
0.9	0.05 (0.022)	0.93 (0.038)	0.95 (0.035)
0.95	0.025 (0.012)	0.94 (0.037)	0.95 (0.038)
0.99	0.005 (0.004)	0.93 (0.035)	0.93 (0.035)
0.995	0.0025 (0.0014)	0.87 (0.044)	0.87 (0.034)

5.4.2.1 Scalability

We compare the worst-case time complexity and the CPU time of the algorithms implemented in MST-backbone(SEM-GM) and rSEM-GM with the algorithms implemented by FastTree, RAxML-NG, and IQ-TREE. A comparison of CPU times is shown in Figure 5.2. The number of rate categories was set to one because simulated sequences were generated using a common rate category across sites. All methods were run using a single thread in order to facilitate a fair comparison with our method which has not been developed for distributed computing.

MST-backbone(SEM-GM) computes Hamming distances in time $O(n^2k)$ where n is the number of input sequences and k is the number of alignment columns. Maximum parsimony spanning trees (MSTs) of the complete graph $G = (V, E)$ over input sequences are computed using Prim's algorithm, which as implemented in the boost graph library (Siek et al., 2000), takes time $O(|E| \log |V|)$. The number of edges $|E|$ is $n(n-1)/2$. Thus the total time complexity of computing the initial MST is bounded from above by $O(n^2 \log n)$. The time complexity of updating the MST is bounded from above by $O(n \log n)$. The total time complexity of computing the MST and updating the MST is $O(n^2 \log n) + O(\frac{n}{n_{\min}} \times n \log n) = O(n^2 \log n)$, where $n_{\min}$ is the size of smallest local phylogenetic tree. The time complexity of each iteration of SEM-GM is dominated by the time required to compute expected counts which is bounded from above by $O(n_l^2 a^3 k_l)$ (Friedman et al., 2002), where n_l is the number leaves in the local phylogenetic tree, a is the size of the alphabet, and k_l is the number of distinct columns in the alignment comprising the sequences represented by the leaves in the local phylogenetic tree. Thus the total time complexity of steps involving SEM-GM is $O\left(n \times \frac{n_{\max}^2 a^3 k_l}{n_{\min}}\right)$, where $n_{\max}$ is the size of the largest local phylogenetic tree. The cumulative time required to compute the local phylogenetic trees grows linearly in the number of input sequences under the assumption that $n_{\max}$ is substantially smaller than the total number of sequences.

The total time complexity of MST-backbone(SEM-GM) is dominated by time $O(\max\{n^2k, n^2 \log n\})$. However, the CPU time that is taken by MST-backbone(SEM-GM) scales linearly with the number of input sequences n (see Figure 5.2 B). This is because CPU time is dominated by the time required to perform SEM-GM which involves computationally expensive floating-point arithmetic, whereas the computation of Hamming distances, and the computation of MSTs involves relatively cheaper integer type based arithmetic. The global unrooted phylogenetic tree T is rooted using restricted SEM-GM. The time complexity of each iteration of restricted SEM-GM is bounded from above by the $O(na^2k)$ steps that are required to compute expected counts via belief propagation on clique trees (Koller and Friedman, 2009), where n is the number of leaves. The CPU time taken to root the global phylogenetic tree via restricted SEM-GM scales quadratically in the number of leaves (see Figure 5.2 A). This may be because the number of iterations of restricted SEM-GM that are required for the convergence of the log-likelihood score scales linearly with the number of leaves.

FastTree employs a large list of heuristics in order to quickly perform tree search (Price et al., 2010). The main heuristics employed by FastTree are (i) extensive use of tree length score instead of likelihood

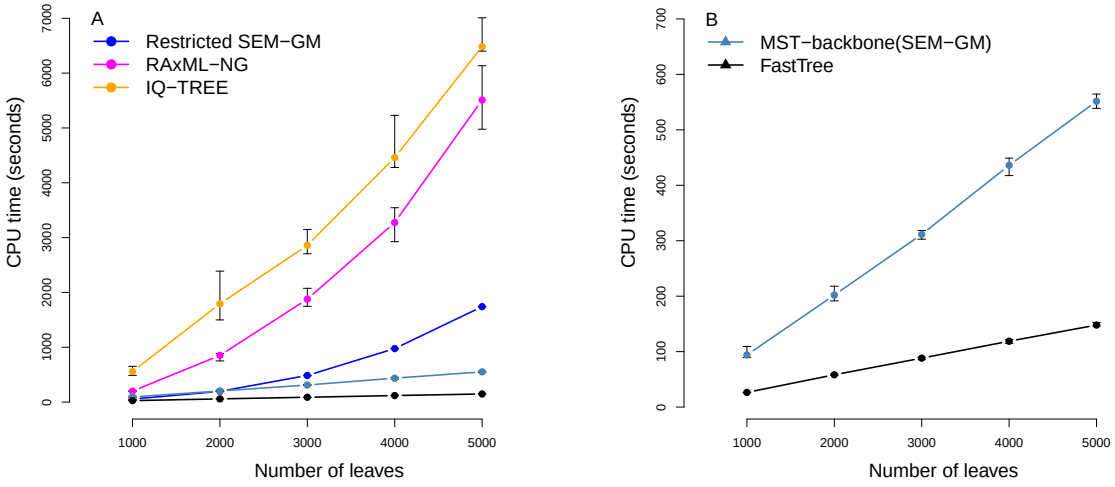


Figure 5.2: A comparison of CPU times used by MST-backbone(SEM-GM), RAxML-NG, and IQ-TREE is shown in panel A. MST-backbone(SEM-GM) + rSEM-GM is the time taken to root the global unrooted phylogenetic tree that is computed by MST-backbone(SEM-GM). Error bars represent inter-quartile range computed using 20 replicates.

score, (*ii*) restriction of SPR moves to moves that can be performed in linear time, (*iii*) optimization of edge lengths via ML using Brent's line search, an operation that takes as input the conditional likelihood vectors of the vertices that are incident to the edge under consideration, *i.e.*, Brent's line search does not require a tree traversal, (*iv*) optimizing the time-reversible rate matrix only once. The operation that optimizes rate matrices necessarily involves tree traversal and computationally expensive floating-point arithmetic.

RAxML-NG and IQ-TREE are orders of magnitude slower than MST-backbone(SEM-GM) and FastTree because they optimize edge lengths using Newton-Raphson, an operation that involves $O(n)$ floating-point arithmetic operations where n is the number of leaves in the global tree. Additionally, IQ-TREE and RAxML-NG repeatedly optimize the rate matrix. That said, the CPU times used by RAxML-NG and IQ-TREE do not increase exponentially because the SPR moves that are employed by these programs are restricted in order to avoid evaluating the likelihood score for trees that are unlikely to improve the likelihood score. MST-backbone(SEM-GM)+rSEM-GM is substantially faster than RAxML-NG and IQ-TREE. This is probably because the log-likelihood score of the general Markov model convergences faster via EM in comparison to the slower convergence of the log-likelihood score of CT-HMMs which are optimized via numerical optimization techniques such as Newton-Raphson and BFGS.

5.4.3 Subtree size threshold

MST-backbone(SEM-GM) is a threshold-based framework for constraining search through phylogenetic tree space. We measured the effect of varying subtree size threshold on reconstruction accuracy using simulated data with $p_{\min}$ set to 0.99. We varied subtree size from 10 to 40 and measured $\text{Re}_S^{\text{nontriv}}$ and $\text{Re}_C^{\text{nontriv}}$. Median values of $\text{Re}_S^{\text{nontriv}}$ and $\text{Re}_C^{\text{nontriv}}$ were 0.98 and 0.93, respectively (see Table 5.4). There was no significant change in either $\text{Re}_S^{\text{nontriv}}$ or $\text{Re}_C^{\text{nontriv}}$ across subtree size thresholds, suggesting that any

Table 5.4: Comparison of recall values for MST-backbone(SEM-GM) at different subtree sizes. Sequences were simulated by setting $p_{\min}$ to 0.99. Median and inter-quartile range (in parentheses) values are reported below, and were computed using 100 replicates.

subtree size	$\mathrm{Re}_S^{\mathrm{nontriv}}$	$\mathrm{Re}_C^{\mathrm{nontriv}}$
10	0.98 (0.0045)	0.93 (0.035)
20	0.98 (0.0055)	0.93 (0.038)
30	0.98 (0.005)	0.93 (0.038)
40	0.98 (0.006)	0.93 (0.041)

reasonable threshold can be selected when implementing MST-backbone.

5.5 Validation on empirical data

Two datasets were selected where GC content varied across species. The first data set comprised beetle mitochondrial gene sequences exhibiting large variation in GC content (Sheffield et al., 2009). The second data set comprised 16S ribosomal RNA (16S rRNA) sequences of bacteria, archaea, and eukaryotes (Hug et al., 2016). Ribosomes are essential RNA-protein complexes that are used by all known forms of life to synthesize proteins. We downloaded 1425 16S rRNA sequences from the supplementary material provided by Hug et al. (2016).

The true evolutionary history of genes is not known in general. We selected two experimental phylogeny data sets where gene sequences were evolved *in vitro* according to a specified phylogenetic tree (Sanson et al., 2002; Randall et al., 2016). An experimental phylogeny setup performs *in vitro* simulation of sequence evolution along the edges of a phylogenetic tree. Sequences are evolved in flasks using error-prone polymerase chain reaction (PCR). Flasks can be treated as species in the context of experimental phylogenetic trees. Evolutionary relationships are represented by a fully labeled phylogenetic tree over flasks. Sequences from an ancestral flask are sampled subsequent to PCR runs, and sampled sequences are used to start PCR runs in the flasks that descend immediately from the ancestral flask. We analyzed sequences from the experimental phylogeny data sets that were generated by Sanson et al. (2002) and Randall et al. (2016). The sequences for Randall et al. (2016) were obtained directly from the authors. The sequences for Sanson et al. (2002) were downloaded using Genbank ids that were provided by the authors (Sanson et al., 2002). The experimental phylogeny for Randall et al. (2016) comprised 19 leaves and 330 ancestors. The experimental phylogeny for Sanson et al. (2002) comprised 16 leaves and 15 ancestors. In the following we use Randall2016 and Sanson2002, respectively, to refer to sequences obtained from Randall et al. (2016) and Sanson et al. (2002).

Additionally, we selected two virus data sets (HIV and Influenza A H3N2) for which the collection times of viruses are known. Rapidly evolving pathogens such as the Influenza virus facilitate the observation of molecular evolution on a time scale of years. We downloaded all Influenza A H3N2 virus sequences from the GISAID data base (Shu and McCauley, 2017) whose collection times ranged from 1968 to 2017. Subsequently, we discarded all duplicate sequences, and created a smaller data set by sampling at random at most five sequences per year of collection. The resulting data set comprised 156 sequences. The high mutation rate of viruses such as HIV enables the reconstruction of transmission networks from phylogenetic trees (Ratmann et al., 2019). We validated MST-backbone(SEM-GM) using 181 HIV *env* gene sequences that were sampled from 11 individuals that were involved in a partially known transmission network (see Figure 5.6 A). The direction of transmission involving individuals A and B is not known. The HIV sequences were made available by Vrancken et al. (2014) on the HIV Los Alamos National Laboratory database (HIVLANL).

We performed multiple sequence alignment using MAFFTv7.3.3 (Katoh et al., 2002; Katoh and Standley, 2013) and removed all alignment columns that contained gaps or ambiguous characters because MST-backbone(SEM-GM) is a prototype method that is not designed to handle gaps or ambiguous characters. The size of the alignment constructed using MAFFT and the size of the trimmed alignment is shown in Table

Table 5.5: Results of chi-square test, and alignment size

Gene type	$p < 0.05$ (%)	Original alignment (bp)	Trimmed alignment (bp)
16S rRNA	19.3	1947	971
mt ATP6	33.33	750	652
mt ATP8	83.33	218	128
mt COX1	11.11	1549	1530
mt COX2	44.44	689	659
mt COX3	44.44	792	712
mt CYTB	11.11	1156	1085
mt ND1	50.0	988	902
mt ND2	5.56	1099	890
mt ND3	50.0	361	337
mt ND4	16.67	1391	1249
mt ND4L	77.78	306	234
mt ND5	16.67	1773	1649
mt ND6	38.89	561	308
ExpPhylo Randall2016	0	678	678
ExpPhylo Sanson2002	0	2236	2214
H3N2	0	1701	1701
HIV	0	2873	1357

5.5. We used trimmed alignment to quantify the extent to which empirical data violated the stationarity assumption using a chi-square test with a p-value cutoff of 0.05 (see Table 5.5).

5.5.1 Test for violation of stationarity assumption

19.3% of 16S rRNA sequences showed significant deviation in base composition. All of the mitochondrial genes exhibited large variation in base composition. More than 70% of two mitochondrial gene sequences, ATP8 and ND4L, had significantly different base composition in comparison to the average base composition. None of the experimental phylogeny data sets, and none of the virus data sets exhibited any significant variation in base composition. We performed model selection to select simpler non-reversible models for each data set because the general Markov model has a large number of free parameters. The results of model selection are shown in the following Subsection.

5.5.2 Results of model selection

The experimental phylogeny data sets comprise leaf sequences and ancestral sequences. We wanted to compare the effect of including ancestral sequences in the experimental phylogeny data sets on reconstruction accuracy. We constructed two alignments for each experimental phylogeny data set, one containing leaf sequences and ancestral sequences, and one containing only leaf sequences.

Our model selection framework selected one UNREST (UNR) matrix for all gene alignments except for three mitochondrial genes, COX1, CYTB and ND1 where two UNR matrices were selected. Additionally, we performed model selection using IQ-TREE because IQ-TREE implements the largest number of time-reversible and non-reversible Markov models. The results of model selection are presented in Table 5.6. The Lie Markov models that are implemented in IQ-TREE are stationary and homogeneous. The stationary and homogeneous version of model 12.12 is the UNREST model. We refer to model 12.12 as UNR in the following text. IQ-TREE reports the results of model selection using AIC, AIC corrected for small sample

Table 5.6: Models selected by us and by IQ-TREE. The Lie Markov models that are implemented in IQ-TREE are stationary and homogeneous. The stationary and homogeneous version of 12.12 is UNREST. Models shown in bold are time-reversible.

Data type	Number of UNREST matrices selected using BIC	Model selected using IQ-TREE		
		AIC	AICc	BIC
16S rRNA	1	12.12+R_7	12.12+R_7	12.12+R_6
ATP6	1	RY8.18+I+Γ_4	RY8.18+I+Γ_4	RY8.18+I+Γ_4
ATP8	1	WS8.18+I+Γ_4	WS8.18+I+Γ_4	WS8.18+I+Γ_4
COX1	2	MK10.34+R_2	MK10.34+R_2	RY8.18+R_3
COX2	1	12.12+R_2	12.12+R_3	**GTR**+F+R_3
COX3	1	12.12+I+Γ_4	12.12+I+Γ_4	WS10.34+I+Γ_4
CYTB	2	12.12+I+Γ_4	12.12+I+Γ_4	12.12+Γ_4
ND1	2	RY10.12+R_3	RY10.12+R_3	RY8.18+I+Γ_4
ND2	1	12.12+I+Γ_4	12.12+I+Γ_4	12.12+I+Γ_4
ND3	1	MK10.34+Γ_4	MK10.34+Γ_4	**TIM**+F+Γ_4
ND4	1	MK10.34+I+Γ_4	MK10.34+I+Γ_4	RY8.18+I+Γ_4
ND4L	1	MK10.34+R_3	**TVM**+F+R_3	**K3Pu**+F+I+Γ_4
ND5	1	RY10.12+I+Γ_4	RY10.12+I+Γ_4	RY8.10a+I+Γ_4
ND6	1	MK10.34+Γ_4	MK10.34+Γ_4	RY8.18+Γ_4
Sanson2002All	1	WS6.6+Γ_4	WS6.6+Γ_4	**TVMe**+Γ_4
Sanson2002Leaf	1	**TVMe**	**TVMe**	**TVMe**
Randall2016All	1	12.12+I+Γ_4	**JC**	**TVMe**+Γ_4
Randall2016Leaf	1	12.12+R_2	12.12+R_2	WS10.12+I+Γ_4
H3N2	1	12.12+R_2	12.12+Γ_4	RY8.17+Γ_4
HIV	1	**TVM**+F+R_4	**TVM**+F+R_4	**TVM**+F+R_4

size (AICc), and BIC. AIC and BIC are defined in equation 2.8 and equation 2.9, respectively. AICc is computed as

$$\text{AICc} = \text{AIC} + \frac{2m^2 + 2m}{k - m - 1}$$

where m is the number of free parameters, and k is the sample size which is equal to the number of alignment columns for phylogeny inference.

Non-reversible models were selected by IQ-TREE for all but two empirical datasets on the basis of AIC. The symbols I,Γ,and R in the mixture models that are selected by IQ-TREE account for heterogeneity of substitution rate across sites (see Subsection 2.4.3). The symbol F indicates that the stationary distribution is set to the sample estimate of the base composition of the sequence alignment. The time reversible models that were selected by IQ-TREE are (*i*) the GTR model, (*ii*) the transition model (TIM; Posada et al. (2003)), (*iii*) the Jukes-Cantor model (JC, Jukes and Cantor (1969)), (*iv*) the transversion model (TVM; Posada et al. (2003)), and (*v*) the transversion model with equal base frequency (TVMe). The rate matrix of the transition model allows a different rate for each transition and a two rates for transversions.

We wanted to compare the trees constructed by MST-backbone(SEM-GM) and subsequently rooted by UNR — hereafter called MST-backbone(SEM-GM) + UNR — , with rooted trees inferred using IQ-TREE. We inferred rooted trees using IQ-TREE via the non-reversible models that were selected on the basis of AIC. There were two datasets, Sanson2002_leaf and HIV, where IQ-TREE selected time-reversible models. We inferred rooted trees for Sanson2002_leaf and HIV using IQ-TREE via the UNREST model allowing for two free rate categories, *i.e.*, UNR $+R_2$. We chose two rate categories because the genes included in HIV and Sanson2002_leaf are protein-coding genes; nucleotides in the third codon position evolve faster than the nucleotides in the first codon position and the second codon position due to degeneracy in the genetic code at the third codon position (Bofkin and Goldman, 2006).

We compared the CPU times used by MST-backbone(SEM-GM) and IQ-TREE to infer phylogenies. Additionally we compared the CPU times used by UNRmodelSelector and IQ-TREE to perform model selection (see Table 5.7). The most notable difference is the time required for IQ-TREE to perform model selection in comparison to the the time required for UNRmodelSelector to select an optimal number of rate matrices on the basis of the BIC score. IQ-TREE took less than a second to perform model selection whereas UNRmodelSelector took around 25 days for the largest dataset (Randall2016 all) comprising 349 sequences, for a single bootstrap replicate. The reason for the drastic time difference is because IQ-TREE employs a fast EM method to perform model selection (Kalyaanamoorthy et al., 2017), whereas UNRmodelSelector optimizes branch lengths and rate matrices by optimizing the log-likelihood score instead of the expected log-likelihood score. Our initial attempt to optimize rate matrices and branch lengths using EM met with issues involving lack of convergence of the log-likelihood score. MST-backbone (SEM-GM) inferred unrooted phylogenies 12 times faster (on average) than IQ-TREE took to infer a rooted phylogeny under the model selected selected by IQ-TREE.

5.5.2.1 Beetle mitochondrial genomes

Jermiin et al. (2004) used simulated data to establish that phylogeny inference under stationary models of gene evolution may lead to systematic error, a result that is in agreement with our simulation based comparative analysis. It is difficult to make decisive statements regarding systematic error using empirical data, except in the case of experimental phylogenies, because the true evolutionary history of species has not been observed and is inferred via comparative analysis. Sheffield et al. (2009) used evolutionary relationships among beetles that were established on the basis of morphological similarity in order to check for systematic error in phylogenies inferred using beetle mitochondrial genes that exhibited large variation in GC content. The established relationships among the beetles include (*i*) the monophyly of six species in the infraorder *Cucujiformia*, (*ii*) the monophyly of four species in the superfamily *Elateroidea*, and (*iii*) sister relationship

Table 5.7: A comparison of CPU times used by MST-backbone(SEM-GM), UNRmodelSelector, and IQ-TREE. MSTB(SEM-GM) is short for MST-backbone (SEM-GM). UNR ms is short for UNRmodelSelector. The columns indexed by IQ-TREE inf and IQTREE ms are populated with the times used to infer a rooted phylogeny and perform model selection, respectively. The times shown below are the average times taken for one bootstrap alignment.

Data set	aln cols (distinct)	leaves	MSTB(SEM-GM)	UNR ms	IQ-TREE inf	IQ-TREE ms
16S rRNA	971 (739)	100	0h:2m:17s	4d:22h:11m:46s	0h:4m:9s	0h:0m:9s
mt ATP8	128 (99)	18	0h:0m:1s	1h:07m:51s	0h:9m:0.1s	0h:0m:0.1s
mt ND4L	234 (186)	18	0h:0m:1s	3h:30m:15s	0h:0m: 8s	0h:0m:0.01s
mt ND6	308 (249)	18	0h:0m:3s	0h:26m:01s	0h:0m:8s	0h:0m:0.03s
mt ND3	337 (243)	18	0h:0m:2s	1h:12m:00s	0h:0m:11s	0h:0m:0.04s
mt ATP6	652 (426)	18	0h:0m:3s	1h:25m:05s	0h:0m:41s	0h:0m:15s
mt COX2	659 (405)	18	0h:0m:3s	2h:54m:37s	0h:0m:17s	0h:0m:3s
mt COX3	712 (417)	18	0h:0m:4s	7h:43m:32s	0h:0m:33s	0h:0m:23s
mt ND2	890 (674)	18	0h:0m:6s	2h:05m:26s	0h:0m:29s	0h:0m:3s
mt ND1	902 (557)	18	0h:0m:3s	3h:24m:23s	0h:0m:18s	0h:0m:4s
mt CYTB	1085 (646)	18	0h:0m:8s	10h:08m:09s	0h:0m:19s	0h:0m:33s
mt ND4	1249 (832)	18	0h:0m:7s	7h:27m:16s	0h:0m:17s	0h:0m:35s
mt COX1	1530 (759)	18	0h:0m:8s	11h:25m:47s	0h:0m:15s	0h:0m:14s
mt ND5	1649 (1086)	18	0h:0m:6s	9h:19m:10s	0h:0m:19s	0h:0m:5s
Randall2016 all	678 (341)	349	0h:0m:4s	25d:5h:39m:44s	0h:5m:5s	0h:0m:0.1s
Randall2016 leaf	678 (293)	19	0h:0m:2s	0h:51m:27s	0h:0m:5s	0h:0m:0.01s
Sanson2002 all	2214 (98)	31	0h:0m:1s	0h:38m:05s	0h:0m:2s	0h:0m:0.01s
Sanson2002 leaf	2214 (96)	16	0h:0m:1s	0h:06m:11s	0h:0m:2s	0h:0m:0.01s
H3N2	1701 (588)	156	0h:0m:14s	4d:19h:39m:32s	0h:5m:51s	0h:0m:0.3s
HIV	1357 (488)	181	0h:0m:52s	9d:6h:29m:44s	0h:10m:36s	0h:0m:2s

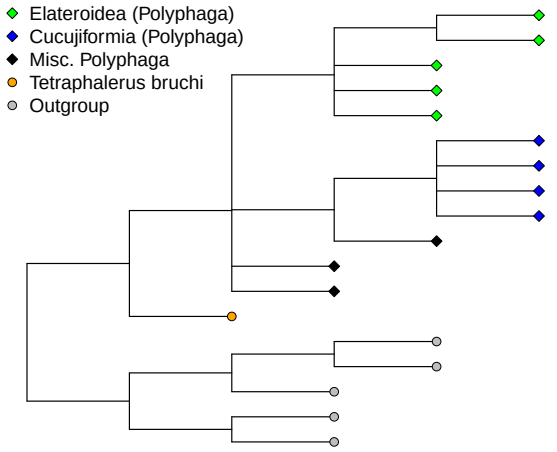


Figure 5.3: The established phylogenetic tree of the beetles that were analyzed in this study. Edge lengths are set to a common value for better illustrating evolutionary relationships, and are not biologically meaningful.

Table 5.8: Number (and name) of established evolutionary relationships that are compatible with inferred gene trees.

Data set	rSEM-GM	UNRmodelSelector	IQ-TREE
ATP6	0/3	0/3	0/3
ATP8	0/3	0/3	0/3
COX1	0/3	0/3	0/3
COX2	1/3 (*Cucujiformia*)	1/3 (*Cucujiformia*)	1/3 (*Cucujiformia*)
COX3	0/3	0/3	0/3
CYTB	0/3	0/3	0/3
ND1	0/3	0/3	1/3 (*Cucujiformia*)
ND2	0/3	0/3	1/3 (*Elateroidea*)
ND3	0/3	0/3	0/3
ND4	0/3	0/3	0/3
ND4L	0/3	0/3	0/3
ND5	0/3	0/3	1/3 (*Cucujiformia*)
ND6	0/3	0/3	0/3

between the suborders *Polyphaga* and *Archostemata* constituting, 12 species and 1 species, respectively, (see Figure 5.3).

Sheffield et al. (2009) constructed a concatenated alignment of 13 mitochondrial genes and inferred phylogenetic trees using a software that performs Bayesian inference via MCMC sampling using time-reversible models (MrBayes v3 Ronquist and Huelsenbeck (2003) and PhyloBayes (Blanquart and Lartillot, 2006),), and four software that perform Bayesian inference via MCMC sampling using nonstationary Markov models: p4 (Foster, 2004), PHASE (Gowri-Shankar and Rattray, 2007), and nhPhyML (Guindon et al., 2010). Sheffield et al. (2009) found that the consensus tree inferred by MrBayes and PhyloBayes violated all established evolutionary relationships among beetles. The consensus trees inferred by p4 and PHASE were in agreement with all established relationships. Trees inferred by nhPhyML agreed with the established relationships varied depending on the input tree. The authors used the tree inferred by p4, and the tree inferred by neighbor joining using LogDet distances as input trees. The bootstrap consensus tree that was inferred by nhPhyML using the tree inferred by p4 as the starting tree (nhPhyML-p4) was compatible with all established relationships. However, the consensus tree inferred by nhPhyML using a neighbor-joining tree computed using LogDet distances as the starting tree (nhPhyML-NJLogDet) was not compatible with any established relationship.

Genes have individual evolutionary histories that are not necessarily identical. Additionally, individual genes may evolve at different rates. The conflicting results of PhyloBayes, p4 and phase may have resulted due to use of a concatenated alignment. We inferred a separate tree for each mitochondrial gene. We inferred phylogenetic trees using MST-backbone(SEM-GM)+rSEM-GM, MST-backbone(SEM-GM)+UNR. The results on simulated data suggest that the number of sequences that violate the assumption of homogeneity in base composition are not a reliable indicator of systematic error. Consequently the rooted trees inferred using IQ-TREE might be accurate estimators of mitochondrial gene relationships even though there is substantial variation of GC content among gene sequences. We used IQ-TREE to infer a rooted tree for each mitochondrial gene using the non-reversible mixture model that was selected via AIC (see Table 5.6).

Four out of thirteen gene trees that were inferred using IQ-TREE were compatible with one out of three established relationships each. One out of thirteen gene trees that were inferred by us was compatible with one out of three established relationships. The gene trees that were inferred by IQ-TREE for *COX2*, *ND1*, and *ND5* were compatible with the monophyly of *Cucujiformia*, and the gene tree that was inferred by IQ-TREE for *ND2* was compatible with the monophyly of *Elateroidea*. The gene tree for *COX2* that was inferred by MST-backbone(SEM-GM) and rooted using rSEM-GM was compatible with the monophyly

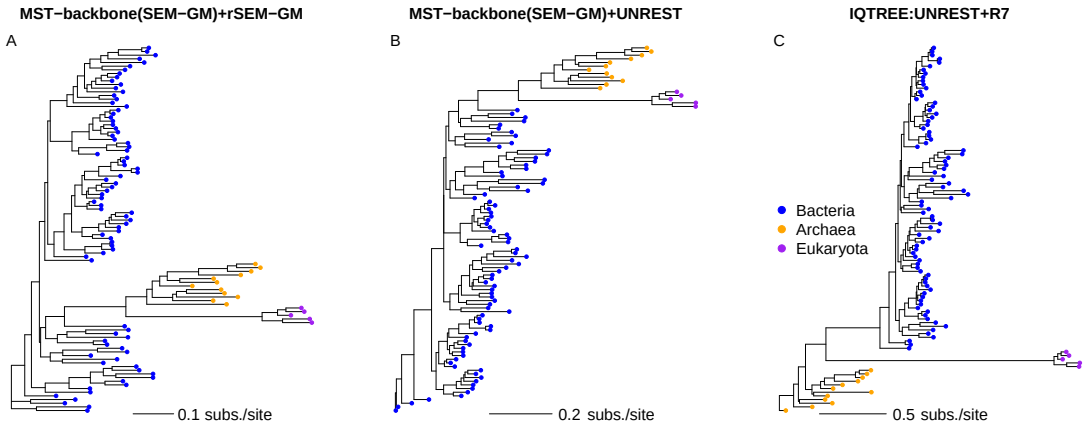


Figure 5.4: Phylogenetic trees of 16S rRNA gene that were (*i*) inferred using MST-backbone(SEM-GM) and rooted using restricted SEM-GM (panel A), (*ii*) inferred using MST-backbone(SEM-GM) and rooted using a single UNR matrix (panel B), (*iii*) inferred using IQ-TREE under the model UNR $+R_7$ (panel C).

of *Cucujiformia*. The gene tree for COX2 that was rooted under the CT-HMM that was selected using UNRmodelSelector was compatible with the monophyly of *Cucujiformia*. None of the gene trees that were inferred by us and none of the gene trees that were inferred using IQ-TREE were compatible with all established relationships. The poor empirical support for IQ-TREE may be because IQ-TREE performs tree search under stationary models. Sheffield et al. (2009) found that Mr. Bayes and PhyloBayes, software that perform Bayesian inference under stationary models, were also unable to reconstruct trees that supported all established relationships. The poor performance of MST-backbone(SEM-GM) could be because of lack of extensive search through tree space.

5.5.2.2 16S RNA

There are three domains of life: the prokaryotic domains — archaebacteria (archaea) and eubacteria (bacteria) — and the eukaryota (Woese et al., 1990). We used the three domain classification to validate phylogenetic trees inferred using MST-backbone(SEM-GM), and IQ-TREE. The bootstrap support for three domain classification was greater than 95% for the unrooted trees inferred by MST-backbone (SEM-GM) and IQ-TREE (see Figure 5.5). All consensus trees discussed hereafter have been constructed using sumtrees.py v4.4.0 (Sukumaran and Holder, 2015, 2010) by collapsing all edges with bootstrap support smaller than 70%. The phylogenetic tree that was inferred by MST-backbone(SEM-GM) + rSEM-GM was rooted among bacteria. We rooted the tree that was inferred using MST-backbone(SEM-GM) using a single UNR matrix which was the model that was selected by UNRmodelSelector. The rooted phylogenetic tree that was inferred using MST-backbone(SEM-GM) + UNR was placed at a bacterium. Subsequently, we checked the placement of the root as inferred by IQ-TREE under the UNR+R_7 model that was selected by IQ-TREE using AIC. The phylogenetic tree that was inferred by IQ-TREE was rooted among archaea. It is generally accepted that the two prokaryotic domains, bacteria and archaea, have evolved independently from a common ancestral cell and thus the root should be ancestral to the lca of bacteria, and the lca of archaea and eukaryota (Lake et al., 2009). It may be possible that horizontal gene transfer of the ribosomal RNA gene makes it difficult to obtain a realistic location of the root.

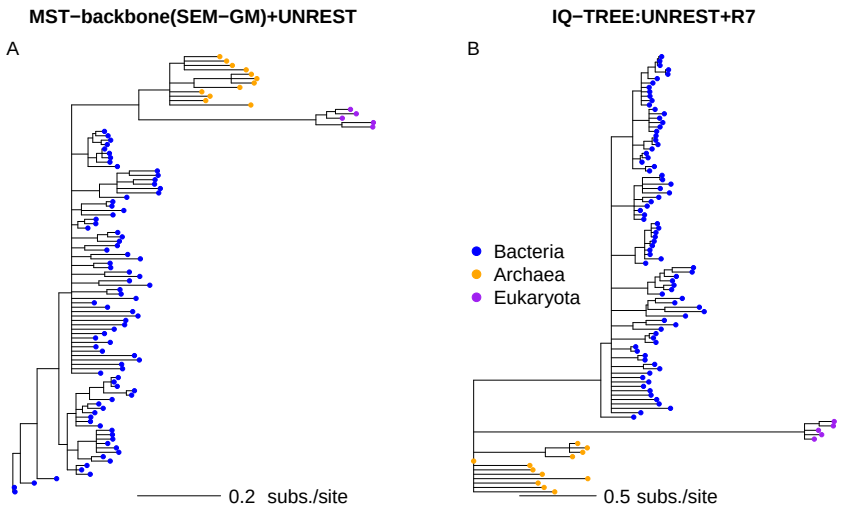


Figure 5.5: Bootstrap consensus phylogenetic trees of 16S rRNA genes inferred using MST-backbone(SEM-GM) and rooted under a single UNR matrix (panel A), and the bootstrap consensus phylogenetic tree inferred using IQ-TREE under the UNR + R_7 model (panel B). All edges with bootstrap support less than 70% are contracted.

5.5.2.3 Experimental phylogeny

Experimental phylogenies are ideal hypotheses for falsifying inferred evolutionary relationships because true evolutionary relationships are part of experimental design. We measured reconstruction accuracy of phylogenetic trees that were inferred using all sequences using recall computed with all splits, *i.e.*, Re_S^{all}, and recall computed with all clusters, *i.e.*, Re_C^{all}. The reconstruction accuracy of phylogenetic trees that were inferred using leaf sequences was measured using recall computed using all non-trivial splits, *i.e.*, $\text{Re}_S^{\text{nontriv}}$, and recall computed using all non-trivial clusters, *i.e.*, $\text{Re}_C^{\text{nontriv}}$. We measured recall values for the complete alignment and bootstrapped alignments.

MST-backbone(SEM-GM) recovered almost all the splits in the experimental phylogeny if all sequences were used (Re_S^{all} values of 0.99 and 1.0, for Randall2016 and Sanson2002, respectively, see Table 5.9), with a reduction in accuracy if only leaf sequences were used ($\text{Re}_S^{\text{nontriv}}$ values of 0.86 and 1.0, for Randall2016 and Sanson2002, respectively). IQ-TREE recovered splits with Re_S^{all} values of 0.98 and 1.0 for Randall2016All and Sanson2002All, respectively., with a reduction in accuracy if only leaf sequences were used ($\text{Re}_S^{\text{nontriv}}$ values of 0.94 and 1.0, for Randall2016 and Sanson2002, respectively). IQ-TREE recovered splits with significantly lower recall values ($p < 0.01$) than MST-backbone(SEM-GM) for Randall2016All and Sanson2002Leaf, where significance was calculated using recall values for bootstrap data sets. IQ-TREE recovered splits with significantly higher recall values for Randall2016Leaf ($p < 0.01$).

Phylogenetic trees that were inferred by MST-backbone(SEM-GM)+rSEM-GM recovered clusters with high Re_C^{all} values of 0.94 and 0.93, for Randall2016 and Sanson2002, respectively, and relatively lower $\text{Re}_C^{\text{nontriv}}$ values of 0.82 and 0.79, for Randall2016 and Sanson2002, respectively. It appears that it is possible to infer rooted phylogenetic trees under the general Markov model. We wanted to check if recall values would change substantially if we rooted trees under simpler CT-HMM. Phylogenetic trees were inferred using MST-backbone(SEM-GM)+UNR. Additionally, phylogenetic trees were inferred under the non-reversible model that was selected by IQ-TREE using AIC. IQ-TREE selected a time-reversible model (TVMe) for the Sanson2002Leaf data set. We inferred rooted trees using IQ-TREE for Sanson200Leaf under the model UNR+R_2.

Table 5.9: Recall for experimental phylogeny data sets for complete alignment with recall at the 25^{th} percentile and the 75^{th} percentile for bootstrap alignments shown in parentheses. MSTB(SEM-GM) is short for MST-backbone(SEM-GM). Phylogenetic trees were inferred using IQ-TREE using the model selected via AIC, except for Sanson2002Leaf where the model UNR+R_2 was used.

Data set	MSTB(SEM-GM)	rSEM-GM	UNR	IQ-TREE	
	Splits	Clusters	Clusters	Splits	Clusters
Randall2016All	0.99 (0.82–0.85)	0.94 (0.76–0.79)	0.96 (0.76–0.81)	0.98 (0.82–0.84)	0.96 (0.76–0.88)
Randall2016Leaf	0.86 (0.75–0.87)	0.82 (0.59–0.76)	0.76 (0.59–0.71)	0.94 (0.81–0.94)	0.88 (0.76–0.88)
Sanson2002All	1.0 (0.97–1.0)	0.93 (0.87–0.93)	0.9 (0.87–0.9)	1.0 (0.97–1.0)	0.9 (0.87–0.9)
Sanson2002Leaf	1.0 (1.0–1.0)	0.79 (0.79–0.79)	0.79 (0.79–0.79)	1.0 (0.92–1.0)	0.79 (0.71–0.79)

IQ-TREE recovered clusters with higher Re_C^{all} and $\text{Re}_C^{\text{nontriv}}$values on bootstrapped data sets when compared with rooting via rSEM-GM and rooting under UNR for Randall2016. and relatively lower $\text{Re}_C^{\text{nontriv}}$ values of 0.82 and 0.79, for Randall2016 and Sanson2002, respectively. All methods for constructing rooted trees had similar recall values on bootstrapped alignments for the Sanson2002 dataset, although Re_C^{all} for MST-backbone(SEM-GM)+rSEMGM was 0.93 compared to Re_C^{all} values of 0.9 and 0.9 for MST-backbone(SEM-GM)+UNR, and rooting using IQ-TREE. Recall values are lower for bootstrap alignments compared to the original alignment because there are fewer distinct site patterns that contain information for rooting trees.

5.5.2.4 HIV transmission network

HIV spreads through inter-personal contact making it possible to use transmission history to falsify inferred evolutionary relationships. A rooted pathogen phylogenetic tree is said to be compatible with a transmission event if pathogens from the recipient have descended from pathogens of the transmitter. We inferred phylogenetic trees using 181 HIV *env* gene sequences that were sampled from 11 individuals that were part of a transmission network (see Figure 5.6).

The phylogenetic tree for HIV that was inferred by MST-backbone(SEM-GM)+rSEM-GM was not rooted realistically because it was rooted at a sequence from individual C which is not compatible with the transmission from B to C (see Figure 5.6). We rooted the tree that was inferred using MST-backbone(SEM-GM) with a single UNR matrix, and found that the tree was compatible with nine out of ten transmission events (see Figure 5.6 A). The transmission event $B \rightarrow I$ was not compatible with the phylogenetic tree. We performed model selection using IQ-TREE and found that IQ-TREE selected the time-reversible model TVM+F+R_4. We inferred phylogenetic trees using IQ-TREE with the UNR +R_2 model because we wanted to check if rooted trees inferred by IQ-TREE with the UNR + R_2 model were compatible with transmission history. The rooted tree that was inferred by IQ-TREE was compatible with all transmission events (see Figure 5.6 B). A reason why the phylogenetic tree that was inferred by MST-backbone(SEM-GM) and rooted with a single UNR matrix was not compatible with the transmission $B \rightarrow I$ could be because the initial tree for SEM-GM, which is the neighbor-joining tree, is suboptimal, and that SEM-GM gets stuck in a local optima. The HIV tree that was inferred using FJ-BIC, which is a modification of neighbor-joining, was not compatible with $B \rightarrow I$ (see Figure 3.4 in Chapter 3). The consensus tree for MST-backbone(SEM-GM)+UNR and IQ-TREE had poor bootstrap support. Out of a total of 179 clusters, only 48 clusters for MST-backbone(SEM-GM)+UNR were supported, and only 44 clusters for IQ-TREE were supported (see Figure 5.8).

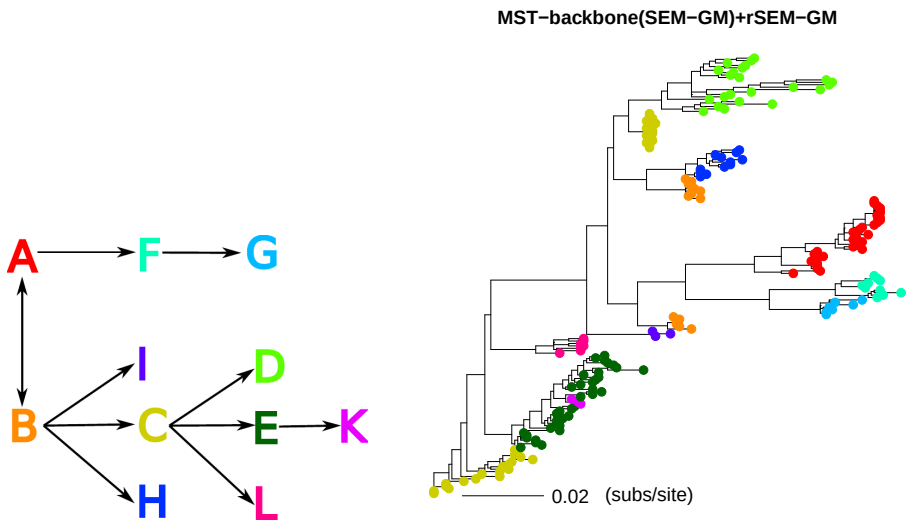


Figure 5.6: Panel A: The known HIV transmission network of 11 individuals. The direction of transmission is known for all transmission events except for the transmission between patients A and B. Panel B: A phylogenetic tree inferred via MST-backbone(SEM-GM)+rSEM-GM.

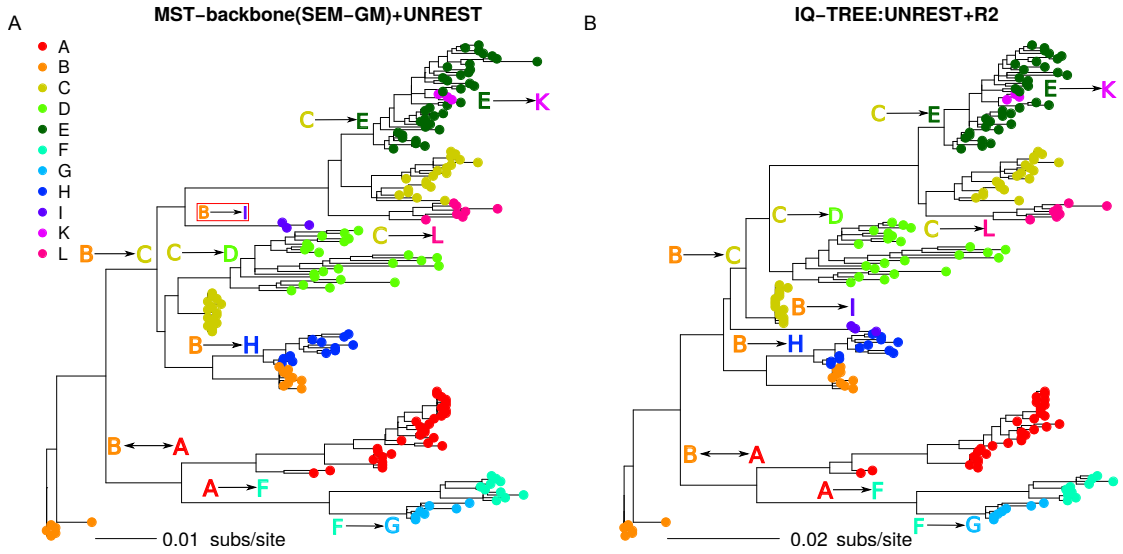


Figure 5.7: Rooted HIV phylogenetic trees labeled with transmission edges. Panel A shows trees inferred using MST-backbone(SEM-GM)+UNR. Panel B shows trees inferred using IQ-TREE. The transmission $B \to I$ is not compatible with the tree inferred using MST-backbone(SEM-GM)+UNR.

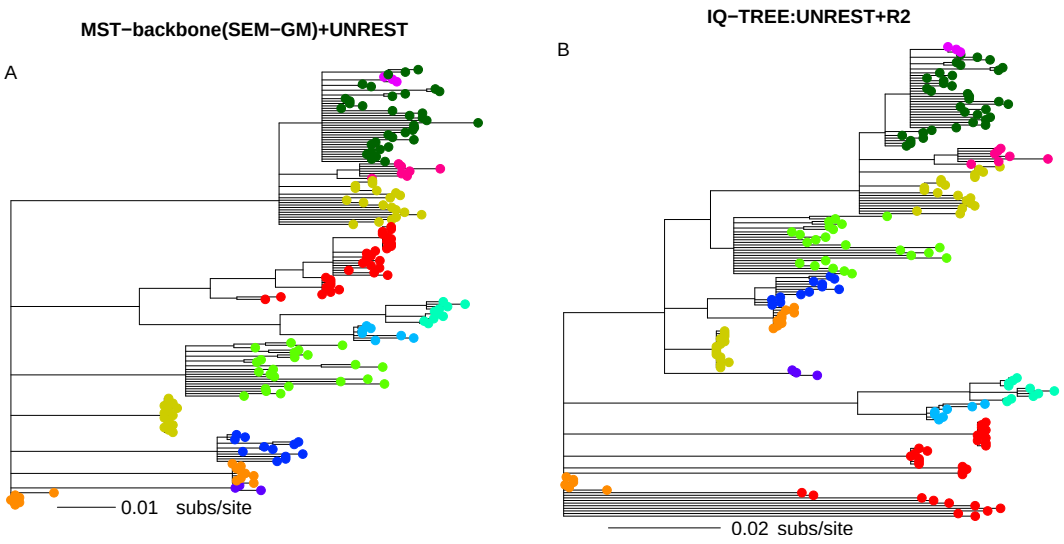


Figure 5.8: Bootstrap consensus trees for HIV as inferred by MST-backbone(SEM-GM)+UNR (panel A), and IQ-TREE with the model UNR + R_2 (panel B).

5.5.2.5 Influenza A H3N2

Under the assumption of a strict molecular clock, the number of character changes that accumulate in a sequence are proportional to the collection time of the sequence. The Influenza A H3N2 virus exhibits a strict molecular-clock-like evolution (Gojobori et al., 1990). We validated inferred phylogenetic trees by measuring the Pearson's correlation coefficient of collection times with weighted root-to-leaf path lengths in inferred phylogenetic trees.

The phylogenetic tree that was inferred under the GM model had a Pearson's correlation coefficient of -0.92 indicating that the location of the root was not realistic (see Figure 5.9A). We performed model selection as described in Section 5.3 in order to select the optimal number of distinct rate matrices. The UNR model was selected as the optimal model. Collection times were highly correlated with root-to-leaf path lengths in the phylogenetic tree that inferred with MST-backbone(SEM-GM)+UNR model with a Pearson's correlation coefficient of 0.99 suggesting that the inferred phylogenetic tree accurately represents the evolutionary history of the Influenza A H3N2 virus (see Figure 5.9B). The phylogenetic tree inferred using IQ-TREE: UNR + R_2 seemed realistic because collection times were correlated with root-to-leaf path lengths with Pearson's correlation coefficient of 0.99. Collection times were highly correlated with root-to-leaf path lengths in the consensus phylogenetic tree for MST-backbone(SEM-GM)+UNR (Pearson's ρ of 0.99) and IQ-TREE: UNR + R_2 (Pearson's ρ of 0.99) (see Figure 5.10).

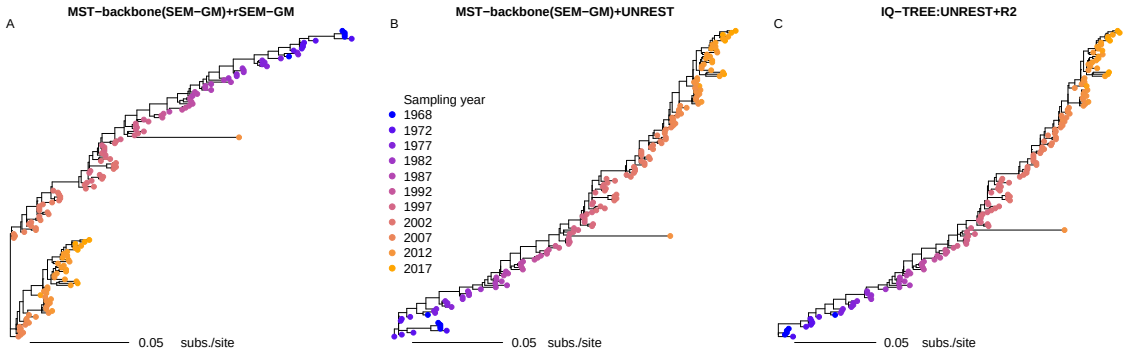


Figure 5.9: Phylogenetic trees for Influenza H3N2 as inferred by MST-backbone(SEM-GM)+rSEM-GM (panel A), MST-backbone(SEM-GM)+UNR (panel B), and IQTREE:UNR+R_2 (panel C). The leaves of each phylogenetic tree have been colored according to year of sampling.

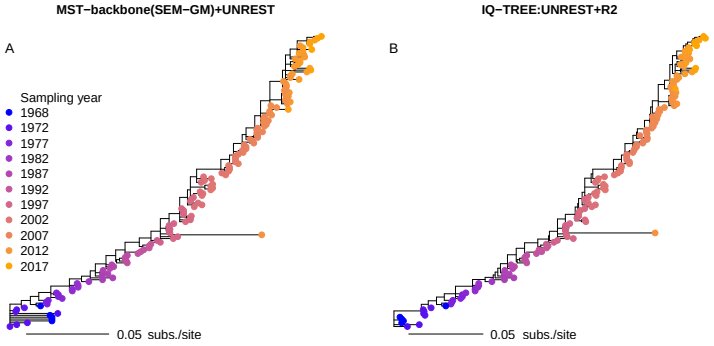


Figure 5.10: Bootstrap consensus trees for H3N2 as inferred using MST-backbone(SEM-GM)+UNR (panel A), and IQ-TREE with UNR+R_2(panel B). The leaves of each phylogenetic tree have been colored according to year of sampling.

5.6 Summary and Outlook

Time-reversible hidden Markov models such as the GTR model are widely used for inferring phylogenetic trees. The GTR model is widely used because homogeneous Markov models enable fast search through tree space by allowing the reuse of conditional likelihood vectors of roots of subtrees that are not changed due to tree modification operations. The GTR model makes restrictive assumptions about the nature of sequence evolution such as stationarity of base composition. Comparison of GC content across species has established that base composition have evolved over the course of evolution. Phylogenetic tree inference under more realistic non-stationary non-homogenous Markov models is limited to small number of species due to the computational burden of optimizing the parameters of non-stationary non-homogeneous Markov models.

Inspired by the topological correspondence between MSTs and phylogenetic trees, we designed a threshold-based framework for inferring phylogenetic trees called MST-backbone. We developed a method called SEM-GM for performing tree search under the GM model.

MST-backbone(SEM-GM) demonstrated higher recall in comparison to RAxML-NG and IQTREE for simulated data that were evolved under non-stationary non-homogeneous Markov models with average edge length greater than 0.1 substitutions per site. The recall values for MST-backbone(SEM-GM) did not significantly vary with threshold suggesting that MST-backbone(SEM-GM) is effectively threshold-free (see Table 5.4). The CPU time taken to infer a global unrooted phylogenetic tree via MST-backbone(SEM-GM) and FastTree increased linearly with number of leaves whereas the CPU time for MST-backbone(SEM-GM)+rSEM-GM, RAxML-NG and IQTREE increased quadratically with number of leaves.

Empirical phylogenetic trees rooted under the GM model were realistic for experimental phylogeny data sets. The topology of the unrooted phylogenetic tree was realistic for all data sets except the beetle mitochondrial data set by Sheffield et al. (2009). The location of the root was not realistic for the Influenza trees and the HIV trees that were inferred under the GM model. Realistic rooted trees were inferred for the Influenza data set and the HIV data set if trees were rooted under the UNR model (either using MST-backbone(SEM-GM)+UNR, or IQTREE) suggesting that the GM model may be over-parameterized for empirical data sets that contain sequences with limited variation in GC content.

Sheffield et al. (2009) reported that phylogenetic trees inferred under non-stationary non-homogeneous HMM recovered established evolutionary relationships whereas phylogenetic trees inferred under the GTR model did not. The phylogenetic trees inferred by MST-backbone(SEM-GM) did not support the established relationships. Although it is possible that the lack of empirical support for the trees inferred by MST-backbone(SEM-GM) is because the GM model is over parameterized, note that the consensus phylogenetic tree inferred by PhyloBayes under a non-stationary non-homogeneous Markov model did not recover any of the established evolutionary relationships.

Chapter 6

Conclusions

Phylogenetic trees are essential for better understanding the molecular basis of phenotypes via comparative analysis. In practice, phylogenetic trees are inferred by solving a combinatorial optimization involving searching through tree space, and parameter optimization. A commonly used strategy for finding optimal phylogenetic trees is to search through the set of phylogenetic trees via tree modification operations (Stamatakis, 2014). Parameter optimization involves CPU-intensive operations that optimize the parameters of the hidden Markov model (HMM) that is used to model sequence evolution. Homogeneous Markov models such as the general time-reversible model (GTR) are widely used in order to reduce the computational burden of parameter optimization.

The GTR model assumes that base composition (including GC content) remains constant throughout evolutionary history. The widespread variation of GC content across species indicates that the stationarity assumption of the GTR model is violated in practice. Complex Markov models such as the general Markov model (GM) by Barry and Hartigan (1987) allow GC content to vary across species, and may be more realistic than the GTR model. Additionally the GM model allows inferring rooted phylogenetic trees, whereas the trees inferred by the GTR model are unrooted.

We implemented a method called SEM-GM for performing tree search under the GM model. SEM-GM adapts the structural expectation maximization framework by Friedman (1997) to the GM model. We implemented a minimum spanning tree framework called MST-backbone in order to improve the scalability of SEM-GM. We validated our method extensively using multiple empirical data sets. We found that the experimental phylogenetic trees were accurately reconstructed via MST-backbone(SEM-GM)+rSEM-GM. The rooted topology of the trees rooted under the GM model seemed to be incorrect for the pathogen datasets. We found that the pathogen trees that were rooted under the UNR model were realistic, however the location of the root was not robust across bootstrap replicates for the HIV. The unrooted ribosomal phylogenetic tree supported the expected monophyly of bacteria, archaea, and eukaryotes. Sheffield et al. (2009) report that the phylogenetic trees of beetle mitochondrial that were inferred under the GTR model were incorrect whereas phylogenetic trees inferred under non-stationary Markov models were correct w.r.t. independently established evolutionary relationships. The phylogenetic trees that were inferred by MST-backbone(SEM-GM) did not support any established evolutionary relationship. Note that the beetle mitochondrial phylogenetic trees that inferred by PhyloBayes, a method that searches for optimal phylogenetic trees under non-stationary Markov models, also did not support any established evolutionary relationship (Sheffield et al., 2009).

The design of MST-backbone was inspired by the topological relationship between minimum spanning trees (MST) and phylogenetic trees proposed by Choi et al. (2011). Choi et al. claimed that given distances that are additive in a phylogenetic tree, an MST that is constructed using the tree-distances shares a topological relationship with the phylogenetic tree. In Kalaghatgi and Lengauer (2017) we showed that MSTs constructed using tree-distances do not necessarily share the topological relationship introduced by Choi et al. We introduced so-called vertex order based MSTs (VMSTs) that are guaranteed to share a topological relationship with phylogenetic trees. We related the number of leaves in a minimum spanning tree to the number of non-trivial splits of a phylogenetic tree, showing the a VMST with the fewest number

of leaves contained the maximum amount of non-trivial split information about a phylogenetic tree.

Rapidly evolving pathogens such as Influenza and HIV enable the study of molecular evolution over short time scales. Pathogens collected at multiple time points from infected individuals may contain ancestor-descendant pairs. The standard model of evolutionary relationships is a leaf-labeled phylogenetic tree that does not allow species to be placed at ancestral vertices. In Kalaghatgi et al. (2016a) we developed a method called family-joining (FJ) for modeling ancestor-descendant relationships using generally labeled trees. FJ constructs generally labeled trees by contracting short edges using a threshold that is selected using BIC. FJ was validated using HIV sequences sampled from individuals that were part of a common transmission network. The HIV tree that was inferred by FJ was rooted under a strict molecular clock. The rooted HIV tree was compatible with nine out of ten transmission events.

In conclusion we state that minimum spanning trees enable large scale inference of phylogenetic trees under non-stationary Markov models such as the general Markov model. The unrooted topology can be recovered using MST-backone(SEM-GM) but it may be necessary to perform model selection under simpler non-reversible CT-HMM in order to recover a realistic location of the root. The GTR model need not be used for the sake of computational ease.

Bibliography

D. G. Adkins. *A Minimum Spanning Tree Framework for Inferring Phylogenies*. PhD thesis, University of California at Berkeley, 2010. URL `https://www2.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-157.pdf`.

D. Agashe and N. Shankar. The evolution of bacterial DNA base composition. *Journal of Experimental Zoology (Molecular and Devolpmental Evolution)*, 322B:517–528, 2014.

H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19: 716–723, 1974.

B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, et al. *Molecular Biology of the Cell, Fourth Edition*. Garland Science, 4 edition, 2002.

E. S. Allman and J. A. Rhodes. Phylogenetic Invariants. In O. Gascuel and M. Steel, editors, *New Mathematical Models of Evolution*, pages 108–147. Oxford University, 2007.

E. S. Allman and J. A. Rhodes. Phylogenetic ideals and varieties for the general Markov model. *Advances in Applied Mathematics*, 40:127–148, 2008.

E. S. Allman, L. S. Kubatko, and J. A. Rhodes. Split Scores: A Tool to Quantify Phylogenetic Signal in Genome-Scale Data. *Systematic Biology*, 66:620–636, 2017.

D. I. Andersson, J. Jerlström-Hultqvist, and J. Näsvall. Evolution of new functions de novo and from preexisting genes. *Cold Spring Harbor perspectives in biology*, 7:1–18, 2015.

D. Barry and J. Hartigan. Statistical Analysis of Hominoid molecular evolution. *Statistical Science*, 2: 191–210, 1987.

D. J. Begun, A. K. Holloway, K. Stevens, L. W. Hillier, Y. Poh, et al. Population genomics: whole-genome analysis of polymorphism and divergence in Drosophila simulans. *PLoS Biology*, 5:e310, nov 2007.

R. Betancur-R, C. Li, T. A. Munroe, J. A. Ballesteros, and G. Ortí. Addressing gene tree discordance and non-stationarity to resolve a multi-locus phylogeny of the flatfishes (Teleostei: Pleuronectiformes). *Systematic Biology*, 62:763–85, 2013.

B. Bettisworth and A. Stamatakis. RootDigger: a root placement program for phylogenetic trees. *bioRxiv*, page 2020.02.13.935304, 2020.

S. Blanquart and N. Lartillot. A Bayesian compound stochastic process for modeling nonstationary and nonhomogeneous sequence evolution. *Molecular Biology and Evolution*, 23:2058–2071, 2006.

L. Bofkin and N. Goldman. Variation in Evolutionary Processes at Different Codon Positions. *Molecular Biology and Evolution*, 24:513–521, 2006.

N. Bortolussi, E. Durand, M. Blum, and O. François. apTreeshape: Statistical analysis of phylogenetic tree shape. *Bioinformatics*, 22:363–364, 2006.

R. Bouckaert, J. Heled, D. Kühnert, T. Vaughan, C. H. Wu, et al. BEAST 2: A Software Platform for Bayesian Evolutionary Analysis. *PLoS Computational Biology*, 10:1–6, 2014.

B. Boussau and M. Gouy. Efficient likelihood computations with nonreversible models of evolution. *Systematic Biology*, 55:756–68, 2006.

D. Bryant. *Building Trees, Hunting for Trees, and Comparing Trees–Theory and methods in phylogenetic analysis*. PhD thesis, University of Canterbury, Christchurch, New Zealand., 1997.

D. Bryant, N. Galtier, and M.-A. Poursat. Likelihood calculation in molecular phylogenetics. In O. Gascuel, editor, *Mathematics of Evolution and Phylogeny*, pages 33–62. Oxford University Press, Oxford, 2005.

P. Buneman. The recovery of trees from measures of dissimilarity. In F. R. Hodson, D. G. Kendall, and P. Tautu, editors, *Mathematics in the Archaeological and Historical Sciences*, pages 387–395. Edinburgh University Press, Edinburgh, UK, 1971.

J. E. Campbell. On a law of combination of operators (second paper). *Proceedings of the London Mathematical Society*, 28:381–390, 1987.

O. Cappé, E. Moulines, and T. Ryden. *Inference in Hidden Markov Models*. Springer-Verlag, Berlin, Heidelberg, 2005.

J. M. Chen, D. N. Cooper, N. Chuzhanova, C. Férec, and G. P. Patrinos. Gene conversion: Mechanisms, evolution and human disease. *Nature Reviews Genetics*, 8:762–775, 2007.

D. M. Chickering. Learning Bayesian networks is NP-complete. In D. Fisher and H.-J. Lenz, editors, *Learning from Data: Artificial Intelligence and Statistics*, pages 121–130. Springer-Verlag, 1996.

M. J. Choi, V. Y. F. Tan, A. Anandkumar, and A. S. Willsky. Learning latent tree graphical models. *Journal of Machine Learning Research*, 12:1771–1812, 2011.

B. Chor and T. Tuller. Finding a maximum likelihood tree is hard. *Journal of the ACM*, 53:722–744, 2006.

C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14:462–467, 1968.

D. H. Colless. [Review of] Phylogenetics: the theory and practice of phylogenetic systematics. *Systematic Zoology*, 31:100–104, 1982.

D. Darriba, G. L. Taboada, R. Doallo, and D. Posada. jModelTest 2: more models, new heuristics and parallel computing. *Nature methods*, 9:772, 2012.

D. Darriba, D. Posada, A. M. Kozlov, A. Stamatakis, B. Morel, et al. ModelTest-NG: A New and Scalable Tool for the Selection of DNA and Protein Evolutionary Models. *Molecular Biology and Evolution*, 37: 291–294, 2020.

C. Darwin. *On the origin of species by means of natural selection, or the preservation of favoured races in the struggle for life*. John Murray, London, 1859.

C. Darwin. *The Life and Letters of Charles Darwin: Including an Autobiographical Chapter*, volume 1 of *Cambridge Library Collection — Darwin, Evolution and Genetics*. Cambridge University Press, 1887.

K. de Queiroz. Ernst Mayr and the modern concept of species. *Proceedings of the National Academy of Sciences of the United States of America*, 102:6600–7, 2005.

A. Dempster, N. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B Methodological*, 39:1–38, 1977.

R. Desper and O. Gascuel. Fast and accurate phylogeny reconstruction algorithms based on the minimum-evolution principle. *Journal of Computational Biology: a Journal of Computational Molecular Cell Biology*, 9:687–705, 2002.

M. dos Reis, P. C. J. Donoghue, and Z. Yang. Bayesian molecular clock dating of species divergences in the genomics era. *Nature Reviews Genetics*, 17:71–80, 2016.

A. J. Drummond, M. A. Suchard, D. Xie, and A. Rambaut. Bayesian phylogenetics with BEAUti and the BEAST 1.7. *Molecular Biology and Evolution*, 29:1969–1973, 2012.

L. Duret and N. Galtier. Biased Gene Conversion and the Evolution of Mammalian Genomic Landscapes. *Annual Review of Genomics and Human Genetics*, 10:285–311, 2009.

C. D. Epp. Definition of a gene. *Nature*, 389:537, 1997.

P. L. Erdös, M. A. Steel, L. Székely, and T. J. Warnow. A few logs suffice to build (almost) all trees (I). *Random Structures and Algorithms*, 14:153–184, 1999a.

P. L. Erdös, M. A. Steel, L. Székely, and T. J. Warnow. A few logs suffice to build (almost) all trees: Part II. *Theoretical Computer Science*, 221:77–118, 1999b.

N. Eriksson. Tree Construction using Singular Value Decomposition. In L. Pachter and B. Sturmfels, editors, *Algebraic Statistics for Computational Biology*, pages 347–358. Cambridge University Press, Berkeley, 2005.

J. Felsenstein. Cases in which Parsimony or Compatibility Methods Will be Positively Misleading. *Systematic Zoology*, 27:401–410, 1978.

J. Felsenstein. Evolutionary trees from DNA sequences: A maximum likelihood approach. *Journal of Molecular Evolution*, pages 368–376, 1981.

J. Felsenstein. *Inferring Phylogenies*. Sinauer Associates, Sunderlands, Massachusetts, 2003.

J. Fernández-Sánchez, J. G. Sumner, P. D. Jarvis, and M. D. Woodhams. Lie Markov models with purine/pyrimidine symmetry. *Journal of Mathematical Biology*, 70:855–891, 2015.

W. M. Fitch. Toward Defining the Course of Evolution: Minimum Change for a Specific Tree Topology. *Systematic Zoology*, 20:406–416, 1971.

R. Fletcher. *Practical Methods of Optimization*. John Wiley and Sons, New York, 2 edition, 1987.

A. D. Foote, Y. Liu, G. W. C. Thomas, T. Vinař, J. Alföldi, et al. Convergent evolution of the genomes of marine mammals. *Nature Genetics*, 47:272–5, 2015.

P. G. Foster and D. A. Hickey. Compositional bias may affect both DNA-based and protein-based phylogenetic reconstructions. *Journal of Molecular Evolution*, 48:284–90, 1999.

P. G. Foster. Modeling compositional heterogeneity. *Systematic Biology*, 53:485–495, 2004.

N. Friedman. Learning Belief Networks in the Presence of Missing Values and Hidden Variables. *International Conference on Machine Learning*, pages 125–133, 1997.

N. Friedman, M. Ninio, I. Pe'er, and T. Pupko. A structural EM algorithm for phylogenetic inference. *Journal of Computational Biology: A Journal of Computational Molecular Cell Biology*, 9:331–353, 2002.

N. Galtier, M. Gouy, and C. Gautier. SEAVIEW and PHYLO_WIN: two graphic tools for sequence alignment and molecular phylogeny. *Computer applications in the biosciences : CABIOS*, 12:543–8, 1996.

N. Galtier and M. Gouy. Inferring pattern and process: maximum-likelihood implementation of a nonhomogeneous model of DNA sequence evolution for phylogenetic analysis. *Molecular Biology and Evolution*, 15:871–879, 1998.

A. Gavryushkina, D. Welch, T. Stadler, and A. J. Drummond. Bayesian inference of sampled ancestor trees for epidemiology and fossil calibration. *PLoS Computational Biology*, 10:e1003919, 2014.

T. Gojobori, E. N. Moriyama, and M. Kimura. Molecular clock of viral evolution, and the neutral theory. *Proceedings of the National Academy of Sciences of the United States of America*, 87:10015–8, 1990.

P. Goloboff. Analyzing large data sets in reasonable times: solution for composite optima. *Cladistics*, 15: 415–428, 1999.

G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 3 edition, 1996.

V. Gowri-Shankar and M. Rattray. A reversible jump method for Bayesian phylogenetic inference with a nonhomogeneous substitution model. *Molecular Biology and Evolution*, 24:1286–99, 2007.

S. W. Graham, R. G. Olmstead, and S. C. H. Barrett. Rooting phylogenetic trees with distant outgroups: a case study from the commelinoid monocots. *Molecular Biology and Evolution*, 19:1769–81, 2002.

G. Guennebaud and J. Benoit. Eigen v3. 2010. URL `http://eigen.tuxfamily.org`.

S. Guindon, J. F. Dufayard, V. Lefort, M. Anisimova, W. Hordijk, et al. New algorithms and methods to estimate maximum-likelihood phylogenies: Assessing the performance of PhyML 3.0. *Systematic Biology*, 59:307–321, 2010.

J. A. Hartigan. Minimum mutation fits to a given tree. *Biometrics*, 29:53–65, 1973.

M. Hasegawa, H. Kishino, and T. A. Yano. Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. *Journal of Molecular Evolution*, 22:160–174, 1985.

S. B. Hedges, J. Dudley, and S. Kumar. TimeTree: a public knowledge-base of divergence times among organisms. *Bioinformatics*, 22:2971–2972, 2006.

S. B. Hedges, J. Marin, M. Suleski, M. Paymer, and S. Kumar. Tree of life reveals clock-like speciation and diversification. *Molecular Biology and Evolution*, 32:835–845, 2015.

HIVLANL. The hiv sequence database that is hosted at the los alamos national laboratory. URL `http://www.hiv.lanl.gov/`. Accessed: 22-03-2019.

A. Hodgkinson and A. Eyre-Walker. Human triallelic sites: evidence for a new mutational mechanism? *Genetics*, 184:233–41, jan 2010.

S. Hohna, M. J. Landis, T. A. Heath, B. Boussau, N. Lartillot, et al. RevBayes: Bayesian phylogenetic inference using graphical models and an interactive model-specification language. *Systematic Biology*, 65: 726–736, 2016.

M. T. Holder, D. J. Zwickl, and C. Dessimoz. Evaluating the robustness of phylogenetic methods to among-site variability in substitution processes. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 363:4013–21, 2008.

W. Hordijk and O. Gascuel. Improving the efficiency of SPR moves in phylogenetic tree search methods based on maximum likelihood. *Bioinformatics*, 21:4338–4347, 2005.

D. C. Hoyle and P. G. Higgs. Factors affecting the errors in the estimation of evolutionary distances between sequences. *Molecular Biology and Evolution*, 20:1–9, 2003.

F. Huang, N. U. N, I. Perros, Robert, J. Sun, et al. Scalable latent tree model and its application to health analytics. *arXiv preprint*, arXiv:1406.4566, 2014.

J. P. Huelsenbeck, J. P. Bollback, and A. M. Levine. Inferring the root of a phylogenetic tree. *Systematic Biology*, 51:32–43, 2002.

L. A. Hug, B. J. Baker, K. Anantharaman, C. T. Brown, A. J. Probst, et al. A new view of the tree of life. *Nature Microbiology*, 1:1–6, 2016.

D. H. Huson, S. M. Nettles, and T. J. Warnow. Disk-covering, a fast-converging method for phylogenetic tree reconstruction. *Journal of Computational Biology: a journal of Computational Molecular Cell Biology*, 6: 369–86, 1999.

V. Jayaswal, L. S. Jermiin, and J. Robinson. Estimation of phylogeny using a general Markov model. *Evolutionary Bioinformatics Online*, 1:62–80, 2005.

L. Jermiin, S. Y. Ho, F. Ababneh, J. Robinson, and A. W. Larkum. The biasing effect of compositional heterogeneity on phylogenetic estimates may be underestimated. *Systematic Biology*, 53:638–43, 2004.

N. A. Johnson. Hybrid incompatibility and speciation. *Nature Education*, 1:20, 2008.

T. Jombart, R. M. Eggo, P. J. Dodd, and F. Balloux. Reconstructing disease outbreaks from genetic data: a graph approach. *Heredity*, 106:383–390, 2011.

T. H. Jukes and C. R. Cantor. Evolution of protein molecules. In *Mammalian Protein Metabolism*, pages 21–132. Elsevier, 1969.

P. Kalaghatgi and T. Lengauer. Computing phylogenetic trees using topologically related minimum spanning trees. *Journal of Graph Algorithms and Applications*, 21:1003–1025, 2017.

P. Kalaghatgi, N. Pfeifer, and T. Lengauer. Family-joining: a fast distance-based method for constructing generally labeled trees. *Molecular Biology and Evolution*, 33:2720–2734, 2016a.

P. Kalaghatgi, A. M. Sikorski, E. Knops, D. Rupp, S. Sierra, et al. Geno2pheno[HCV] - A web-based interpretation system to support Hepatitis C treatment decisions in the era of direct-acting antiviral agents. *PLoS ONE*, 11:e0155869, 2016b.

S. Kalyaanamoorthy, B. Q. Minh, T. K. F. Wong, A. von Haeseler, and L. S. Jermiin. ModelFinder: fast model selection for accurate phylogenetic estimates. *Nature methods*, 14:587–589, 2017.

K. Katoh and D. M. Standley. MAFFT multiple sequence alignment software version 7: improvements in performance and usability. *Molecular Biology and Evolution*, 30:772–80, 2013.

K. Katoh, K. Misawa, K.-i. Kuma, and T. Miyata. MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Research*, 30:3059–66, 2002.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques – Adaptive Computation and Machine Learning*. The MIT Press, 2009.

E. V. Koonin. Orthologs, Paralogs, and Evolutionary Genomics. *Annual Review of Genetics*, 39:309–338, 2005.

A. M. Kozlov, D. Darriba, T. Flouri, B. Morel, and A. Stamatakis. RAxML-NG: A fast, scalable and user-friendly tool for maximum likelihood phylogenetic inference. *Bioinformatics*, 35:4453–4455, 2019.

J. B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, page 48, 1956.

S. Kumar and S. B. Hedges. A molecular timescale for vertebrate evolution. *Nature*, 392:917–20, 1998.

S. Kumar and S. B. Hedges. TimeTree2 : species divergence times on the iPhone. *Bioinformatics*, 27: 2023–2024, 2011.

J. A. Lake, R. G. Skophammer, C. W. Herbold, and J. A. Servin. Genome beginnings: rooting the tree of life. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 364:2177–2185, 2009.

C. Lanave, G. Preparata, C. Saccone, and G. Serio. A new method for calculating evolutionary substitution rates. *Journal of Molecular Evolution*, 20:86–93, 1984.

T. Le, A. Sy, E. K. Molloy, Q. R. Zhang, S. Rao, et al. Using INC Within Divide-and-Conquer Phylogeny Estimation. In I. Holmes, C. Martín-Vide, and M. A. Vega-Rodríguez, editors, *Algorithms for Computational Biology*, pages 167–178, Cham, 2019. Springer International Publishing.

V. Lefort, J. E. Longueville, and O. Gascuel. SMS: Smart Model Selection in PhyML. *Molecular Biology and Evolution*, 34:2422–2424, 2017.

P. Lemey, I. Derdelinckx, A. Rambaut, K. Van Laethem, S. Dumont, et al. Molecular Footprint of Drug-Selective Pressure in a Human Immunodeficiency Virus Transmission Chain. *Journal of Virology*, 79: 11981–11989, 2005.

W. H. Li. So, what about the molecular clock hypothesis? *Current opinion in genetics & development*, 3: 896–901, 1993.

J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In N. J. Le Cam LM, editor, *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, pages 281–297. University of California Press, Berkeley, 1967.

U. Mai, E. Sayyari, and S. Mirarab. Minimum variance rooting of phylogenetic trees and implications for species tree reconstruction. *PloS ONE*, 12:e0182238, 2017.

E. Margoliash. Primary structure and evolution of cytochrome c. *Proceedings of the National Academy of Sciences of the United States of America*, 50:672–9, 1963.

B. Nabholz, A. Künstner, R. Wang, E. D. Jarvis, and H. Ellegren. Dynamic evolution of base composition: causes and consequences in avian phylogenomics. *Molecular Biology and Evolution*, 28:2197–210, 2011.

S. Naser-Khdour, B. Q. Minh, W. Zhang, E. A. Stone, and R. Lanfear. The Prevalence and Impact of Model Violations in Phylogenetic Analysis. *Genome Biology and Evolution*, 11:3341–3352, 2019.

J. A. Nelder and R. Mead. A Simplex Method for Function Minimization. *The Computer Journal*, 7:308–313, 1965.

L. T. Nguyen, H. A. Schmidt, A. von Haeseler, and B. Q. Minh. IQ-TREE: A fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. *Molecular Biology and Evolution*, 32:268–274, 2015.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, NY, USA, second edition, 2006.

L. Pachter and B. Sturmfels. *Algebraic statistics for computational biology*. Cambridge University Press, 2005.

J. Pearl. Reverend Bayes on Inference Engines: A Distributed Hierarchical Approach. In *Proceedings of the Second AAAI Conference on Artificial Intelligence*, pages 133–136, 1982.

M. Plummer, N. Best, K. Cowles, and K. Vines. CODA: Convergence Diagnosis and Output Analysis for MCMC. *R News*, 6:7–11, 2006.

D. Posada, A. Baxevanis, D. Davison, R. Page, G. Petsko, et al. Using Modeltest and PAUP* to select a model of nucleotide substitution. In *Current Protocols in Bioinformatics*, chapter 6, pages 6.5.1–6.5.14. 2003.

M. J. D. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7:155–162, 1964.

L. Pozzi, J. A. Hodgson, A. S. Burrell, K. N. Sterner, R. L. Raaum, et al. Primate phylogenetic relationships and divergence dates inferred from complete mitochondrial genomes. *Molecular Phylogenetics and Evolution*, 75:165–183, 2014.

M. N. Price, P. S. Dehal, and A. P. Arkin. FastTree 2 – Approximately Maximum-Likelihood Trees for Large Alignments. *PLoS ONE*, 5:e9490, mar 2010. ISSN 1932-6203.

R. C. Prim. Shortest Connection Networks And Some Generalizations. *Bell System Technical Journal*, 36: 1389–1401, 1957.

A. Rambaut and N. Grassly. Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Computer Application in the Biosciences*, 13:235–238, 1997.

A. Rambaut. Estimating the rate of molecular evolution: incorporating non-contemporaneous sequences into maximum likelihood phylogenies. *Bioinformatics*, 16:395–9, 2000.

A. Rambaut and L. Bromham. Estimating divergence dates from molecular sequences. *Molecular Biology and Evolution*, 15:442–8, 1998.

A. Rambaut, T. T. Lam, L. Max Carvalho, and O. G. Pybus. Exploring the temporal structure of heterochronous sequences using TempEst (formerly Path-O-Gen). *Virus Evolution*, 2:vew007, 2016.

R. N. Randall, C. E. Radford, K. A. Roof, D. K. Natarajan, and E. A. Gaucher. An experimental phylogeny to benchmark ancestral sequence reconstruction. *Nature Communications*, 7:12847, 2016.

O. Ratmann, M. K. Grabowski, M. Hall, T. Golubchik, C. Wymant, et al. Inferring HIV-1 transmission networks and sources of epidemic spread in Africa with deep-sequence phylogenetic analysis. *Nature Communications*, 10, 2019.

R. Ravi and M. Singh. Delegate and conquer: an LP-based approximation algorithm for minimum degree MSTs. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming*, pages 169–180. 2006.

D. F. Robinson and L. R. Foulds. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53:131–147, 1981.

S. Roch. A short proof that phylogenetic tree reconstruction by maximum likelihood is hard. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3:92–94, 2006.

F. Ronquist and J. P. Huelsenbeck. MrBayes 3: Bayesian phylogenetic inference under mixed models. *Bioinformatics*, pages 1572–1574, 2003.

U. W. Roshan, B. M. Moret, T. Warnow, and T. L. Williams. Rec-I-DCM3: a fast algorithmic technique for reconstructing large phylogenetic trees. In *Proceedings. IEEE Computational Systems Bioinformatics Conference*, pages 98–109, 2004.

A. RoyChoudhury. Consistency of the maximum likelihood estimator of evolutionary tree. *arXiv preprint*, (arXiv:1405.0760), 2014.

N. Saitou and M. Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4:406–25, 1987.

G. Salamon and G. Wiener. On finding spanning trees with few leaves. *Information Processing Letters*, 105: 164–169, 2008.

G. F. Sanson, S. Y. Kawashita, A. Brunstein, and M. R. Briones. Experimental phylogeny of neutrally evolving DNA sequences generated by a bifurcate series of nested polymerase chain reactions. *Molecular Biology and Evolution*, 19:170–178, 2002.

G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

C. Semple and M. Steel. *Phylogenetics*, volume 24 of *Oxford Lecture Series In Mathematics And Its Applications*. Oxford University Press, 2003.

N. C. Sheffield, H. Song, S. L. Cameron, and M. F. Whiting. Nonstationary evolution and compositional heterogeneity in beetle mitochondrial phylogenomics. *Systematic Biology*, 58:381–94, 2009.

Y. Shu and J. McCauley. GISAID: Global initiative on sharing all influenza data - from vision to reality. *Euro surveillance*, 22, 2017.

J. Siek, L.-Q. Lee, and A. Lumsdaine. Boost graph library. http://www.boost.org/libs/graph/, 2000.

K. St. John, T. Warnow, B. M. E. Moret, and L. Vawter. Performance study of phylogenetic methods: (Unweighted) quartet methods and neighbor-joining. *Journal of Algorithms*, 48:173–193, 2003.

A. Stamatakis. RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics*, 22:2688–2690, 2006.

A. Stamatakis. RAxML version 8: A tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, 30:1312–1313, 2014.

M. Steel. *Phylogeny: discrete and random processes in evolution*. Society for Industrial and Applied Mathematics, Philadelphia, 2016.

M. Steel. Recovering a tree from the leaf colourations it generates under a Markov model. *Applied Mathematics Letters*, 2:19–23, 1994.

M. Steel, D. Huson, and P. J. Lockhart. Invariable sites models and their use in phylogeny reconstruction. *Systematic Biology*, 49:225–232, 2000.

K. Strimmer and A. von Haeseler. Quartet puzzling: A quartet maximum-likelihood method for reconstructing tree topologies. *Molecular Biology and Evolution*, 13:964–969, 1996.

J. A. Studier and K. J. Keppler. A note on the neighbor-joining algorithm of Saitou and Nei. *Molecular Biology and Evolution*, 5:729–31, 1988.

M. A. Suchard, P. Lemey, G. Baele, D. L. Ayres, A. J. Drummond, et al. Bayesian phylogenetic and phylodynamic data integration using BEAST 1.10. *Virus Evolution*, 4:1–5, 2018.

J. Sukumaran and M. T. Holder. DendroPy: a Python library for phylogenetic computing. 26:1569–71, 2010.

J. Sukumaran and M. T. Holder. SumTrees: Phylogenetic Tree Summarization. 4.4.0, 2015. URL `https://github.com/jeetsukumaran/DendroPy`.

J. Sullivan, Z. Abdo, P. Joyce, and D. L. Swofford. Evaluating the performance of a successive-approximations approach to parameter optimization in maximum-likelihood phylogeny estimation. *Molecular Biology and Evolution*, 22:1386–92, 2005.

J. G. Sumner, J. Fernández-Sánchez, and P. D. Jarvis. Lie Markov models. *Journal of Theoretical Biology*, 298:16–31, 2012.

K. Tamura and M. Nei. Estimation of the number of nucleotide substitutions in the control region of mitochondrial DNA in humans and chimpanzees. *Molecular Biology and Evolution*, 10:512–26, 1993.

R. E. Tarjan. Efficiency of a Good But Not Linear Set Union Algorithm. *Journal of the Association for Computing Machinery*, 22:215–225, 1975.

R. E. Tarjan. *Data Structures and Network Algorithms*. SIAM, Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, USA, 1992.

S. Tavare. Some probabilistic and statistical problems in the analysis of DNA sequences. In R. M. Miura, editor, *Lectures on Mathematics in the Life Sciences*, volume 17, pages 57–86. Providence, R.I. American Mathematical Society, 1986.

P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 2020.

B. Vrancken, A. Rambaut, M. a. Suchard, A. Drummond, G. Baele, et al. The Genealogical Population Dynamics of HIV-1 in a Large Transmission Chain: Bridging within and among Host Evolutionary Rates. *PLoS Computational Biology*, 10:e1003505, 2014.

P. J. Waddell and M. A. Steel. General time-reversible distances with unequal rates across sites: mixing gamma and inverse Gaussian distributions with invariant sites. *Molecular Phylogenetics and Evolution*, 8: 398–414, 1997.

W. Wagner. Problems in the classification of ferns. In *Recent Advances in Botany*, volume 1, pages 841–844. Univ. of Toronto press Toronto, Can UnderwritCanada., 1961.

C. Wikimedia. File difference_dna_rna-en.svg — wikimedia commons the free media repository, 2017. URL `https://commons.wikimedia.org/wiki/File:Difference_DNA_RNA-EN.svg`. [Online; accessed 20-February-2020].

T. A. Williams, S. E. Heaps, S. Cherlin, T. M. W. Nye, R. J. Boys, et al. New substitution models for rooting phylogenetic trees. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 370:20140336, 2015.

C. R. Woese, O. Kandler, and M. L. Wheelis. Towards a natural system of organisms: proposal for the domains Archaea, Bacteria, and Eucarya. *Proceedings of the National Academy of Sciences of the United States of America*, 87:4576–9, 1990.

M. D. Woodhams, J. Fernández-Sánchez, and J. G. Sumner. A New Hierarchy of Phylogenetic Models Consistent with Heterogeneous Substitution Rates. *Systematic Biology*, 64:638–650, 2015.

Z. Yang and D. Roberts. On the use of nucleic acid sequences to infer early branchings in the tree of life. *Molecular Biology and Evolution*, 12:451–8, 1995.

Z. Yang. Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods. *Journal of Molecular Evolution*, 39:306–14, 1994a.

Z. Yang. Estimating the pattern of nucleotide substitution. *Journal of Molecular Evolution*, 39:105–111, 1994b.

Z. Yang. A space-time process model for the evolution of DNA sequences. *Genetics*, 139:993–1005, 1995.

Z. Yang. Maximum likelihood estimation on large phylogenies and analysis of adaptive evolution in human influenza virus A. *Journal of Molecular Evolution*, 51:423–32, 2000.

Q. Zhang, S. Rao, and T. Warnow. Constrained incremental tree building: new absolute fast converging phylogeny estimation methods with improved scalability and accuracy. *Algorithms for Molecular Biology*, 14:2, 2019.

E. Zukerkandl and L. Pauling. Evolutionary Divergence and Convergence in Proteins. In V. Bryson and H. Vogel, editors, *Evolving Genes and Proteins*, pages 97–166. New York, 1965.

E. Zukerkandl and L. Pauling. Molecular disease, evolution, and genic heterogeneity. In M. Kasha and B. Pullman, editors, *Horizons in Biochemistry*, pages 189–225. Academic press, New York, 1962.

Appendix A

Supplementary material for Chapter 3

A.1 OLS estimate of edge length for generally labeled trees

In what follows we show that the edge length formula, equation A.1 that was derived by Bryant (1997) for leaf-labeled trees is also applicable for generally labeled trees. We follow the terminology that was defined in Chapter 3.

A.1.1 Internal edges

Consider the internal edge $e_0 = \{\alpha, \beta\}$ shown in Figure A.1 such that edges $e_1, \dots e_k$ are incident to α but are not incident to β, and edges $e_{k+1} \dots e_m$ are incident to β but are not incident to α. Let $\mathcal{L}_\alpha|\mathcal{L}_\beta$ be the split that is induced by $\{\alpha, \beta\}$ such that $\mathcal{L}_\alpha$ is closer to α in comparison to β. Let n_α be the cardinality of $\mathcal{L}_\alpha$, and let n_β be the cardinality of $\mathcal{L}_\beta$.

For each edge e_i, define $W_i = \sum_{x \in A_i, y \in B_i} p_{xy}$ where A_i and B_i are the sides of the split defined by edge e_i. The notation p_{xy} is used instead of $p^w_T(x, y)$ to denote the weighted path length of the path from x to y where edge lengths are determined by OLS. It turns out that $W_i = \delta_i^T \mathbf{d}^c$.

For each edge e_i such that $1 \leq i \leq k$, let C_i be the side of the split induced by e_i that is closer to α in comparison to β. For each edge e_i such that $k+1 \leq i \leq m$, let C_i be the side of the split induced by e_i that is closer to β in comparison to α. Let n_i be the cardinality of C_i. Define

$$Y_i = \begin{cases} \sum_{x \in C_i} p_{\alpha x}, & \text{if } 1 \leq i \leq k \\ \sum_{x \in C_i} p_{\beta x}, & \text{if } k+1 \leq i \leq m \end{cases}$$

If both α and β are not labeled (Case 1 in Figure A.1) it can be shown that Bryant (1997)

$$\underline{W} = (nI - 2N)\underline{Y} + NU\underline{Y} + t_{e_0} N \underline{v}$$

where N is the $m \times m$ diagonal matrix with $(n_1, n_2, \dots, n_m)$ on the diagonal, I is the identity matrix, $\underline{Y} = (Y_1, Y, \dots, Y_m)^T$, U is the $m \times m$ matrix of ones, $\underline{v}$ is the vector with n_β in positions 1 to k followed

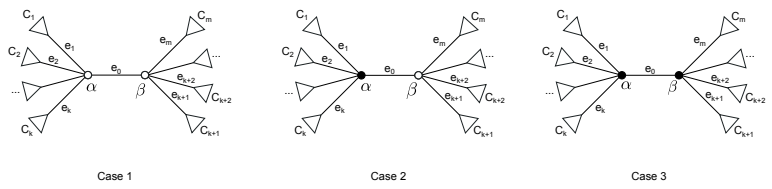


Figure A.1: The three cases for the internal edge e_0. Case 1: Both α and β are not labeled. Case 2: Only α is labeled. Case 3: Both α and β are labeled. The triangles represent subtrees.

by n_α in positions $k+1$ to m, $\underline{W} = (W_1, W_2, \ldots, W_m)^T$, n is the total number of labeled vertices, and t_{e_0} is the edge length of the edge e_0

Similarly for the internal edge e_0,

$$W_0 = \underline{v}^T \underline{Y} + n_\alpha n_\beta t_{e_0}$$

Letting $X = (nN^{-1} - 2I + U)$ and substituting Y gives the following edge length estimate.

$$t_{e_0} = \frac{W_0 - \underline{v}^T X^{-1} N^{-1} \underline{W}}{n_\alpha n_\beta - \underline{v}^T X^{-1} \underline{v}}$$

For cases where only α and both α and β are labeled, respectively, the derivation of the equations are similar to that described in Bryant (1997) and is described below.

Case 2: α is labeled and β is not labeled

For edges e_i incident to α, $i = 1 \ldots k$, we have

$$\begin{aligned}
W_i &= \sum_{x \in A_i} \sum_{y \in B_i} p_{xy} \\
&= \sum_{j=1, j \neq i}^{m} \sum_{x \in C_i} \sum_{y \in C_j} p_{xy} + \sum_{x \in C_i} p_{\alpha x} \\
&= \sum_{j=1, j \neq i}^{k} \sum_{x \in C_i} \sum_{y \in C_j} (p_{\alpha x} + p_{\alpha y}) + \sum_{j=k+1}^{m} \sum_{x \in C_i} \sum_{y \in C_j} (p_{\alpha x} + t_{e_0} + p_{\beta y}) + \sum_{x \in C_i} p_{\alpha x} \\
&= \sum_{j=1, j \neq i}^{k} [n_j Y_i + n_i Y_j] + \sum_{j=k+1}^{m} [n_j Y_i + n_i Y_j + n_i n_j t_{e_0}] + Y_i \\
&= (n - n_i - 1) Y_i + n_i (Y_1 + \ldots + Y_{i-1} + Y_{i+1} + \ldots + Y_m) + n_i n_\beta t_{e_0} + Y_i \\
&= (n - 2n_i) Y_i + n_i \sum_{j=1}^{m} Y_j + n_i n_\beta t_{e_0}
\end{aligned}$$

For edges e_i incident to β, $i = k+1 \ldots m$, we have

$$\begin{aligned}
W_i &= \sum_{x \in A_i} \sum_{y \in B_i} p_{xy} \\
&= \sum_{j=1, j \neq i}^{m} \sum_{x \in C_i} \sum_{y \in C_j} p_{xy} + \sum_{x \in C_i} p_{\alpha x} \\
&= \sum_{j=1}^{k} \sum_{x \in C_i} \sum_{y \in C_j} (p_{\beta x} + t_{e_0} + p_{\alpha y}) + \sum_{j=k+1, j \neq i}^{m} \sum_{x \in C_i} \sum_{y \in C_j} (p_{\beta x} + p_{\beta y}) + \sum_{x \in C_i} (p_{\beta x} + t_{e_0}) \\
&= (\sum_{j=1}^{k} n_j Y_i + n_i Y_j + n_i n_j t_{e_0}) + (\sum_{j=k+1, j \neq i}^{m} n_j Y_i + n_i Y_j) + Y_i + n_i t_{e_0} \\
&= (n - n_i - 1) Y_i + n_i (Y_1 + \ldots + Y_{i-1} + Y_{i+1} + \ldots + Y_m) + n_i (n_\alpha - 1) t_{e_0} + Y_i + n_i t_{e_0} \\
&= (n - 2n_i) Y_i + n_i \sum_{j=1}^{m} Y_j + n_i n_\alpha t_{e_0}
\end{aligned}$$

In matrix form,

$$\begin{aligned}
&\underline{W} = (nI - 2N) Y + NU \underline{Y} + t_{e_0} N \underline{v} \\
&\Leftrightarrow N(nN^{-1} - 2I + U) \underline{Y} = \underline{W} - t_{e_0} N \underline{v}
\end{aligned}$$

Setting $X = (nN^{-1} - 2I + U)$ and rearranging, we get

$$\underline{Y} = X^{-1}N^{-1}\underline{W} - t_{e_0}X^{-1}\underline{v}$$

For the internal edge e_0 we have

$$\begin{aligned}
W_0 &= \sum_{i=1}^{k}\sum_{j=k+1}^{m}\sum_{x\in C_i, y\in C_j} p_{xy} + \sum_{j=k+1}^{m}\sum_{x\in C_j}(t_{e_0} + p_{\beta x}) \\
&= (\sum_{i=1}^{k}\sum_{j=k+1}^{m}\sum_{x\in C_i, y\in C_j} p_{\alpha x} + t_{e_0} + p_{\beta y}) + n_\beta t_{e_0} + \sum_{j=k+1}^{m} Y_j \\
&= (\sum_{i=1}^{k}\sum_{j=k+1}^{m} n_j Y_i + n_i n_j t_{e_0} + n_i Y_j) + n_\beta t_{e_0} + \sum_{j=k+1}^{m} Y_j \\
&= \sum_{i=1}^{k} n_\beta Y_i + \sum_{j=k+1}^{m}(n_\alpha - 1)Y_j + (n_\alpha - 1)n_\beta t_{e_0} + n_\beta t_{e_0} + \sum_{j=k+1}^{m} Y_j \\
&= \underline{v}^T\underline{Y} + n_\alpha n_\beta t_{e_0}
\end{aligned}$$

After substituting Y and rearranging we get,

$$t_{e_0} = \frac{W_0 - \underline{v}^T X^{-1}N^{-1}\underline{W}}{n_\alpha n_\beta - \underline{v}^T X^{-1}\underline{v}} \tag{A.1}$$

Case 3: α and β are labeled

For edges e_i incident to α, $i = 1 \ldots k$, we have

$$\begin{aligned}
W_i &= \sum_{x\in A_i}\sum_{y\in B_i} p_{xy} \\
&= \left[\sum_{j=1, j\neq i}^{m}\sum_{x\in C_i}\sum_{y\in C_j} p_{xy}\right] + \sum_{x\in C_i} p_{\alpha x} + \sum_{x\in C_i} p_{\beta x} \\
&= \left[\sum_{j=1, j\neq i}^{k}\sum_{x\in C_i}\sum_{y\in C_j} p_{\alpha x} + p_{\alpha y}\right] + \left[\sum_{j=k+1}^{m}\sum_{x\in C_i}\sum_{y\in C_j} p_{\alpha x} + t_{e_0} + p_{\beta y}\right] + 2\sum_{x\in C_i} p_{\alpha x} + n_i t_{e_0} \\
&= \left[\sum_{j=1, j\neq i}^{k} n_j Y_i + n_i Y_j\right] + \left[\sum_{j=k+1}^{m} n_j Y_i + n_i Y_j + n_i n_j t_{e_0}\right] + 2Y_i + n_i t_{e_0} \\
&= (n - n_i - 2)Y_i + n_i(Y_1 + \ldots + Y_{i-1} + Y_{i+1} + \ldots + Y_m) + n_i t_{e_0}(1 + \sum_{j=k+1}^{m} n_j) + 2Y_i \\
&= (n - 2n_i)Y_i + n_i\sum_{j=1}^{m} Y_j + n_i n_\beta t_{e_0}
\end{aligned}$$

By symmetry, for edges e_i incident to β, $i = k+1 \ldots m$, we have,

$$W_i = (n - 2n_i)Y_i + n_i\sum_{j=1}^{m} Y_j + n_i n_\alpha t_{e_0}$$

In matrix form,

$$\underline{W} = (nI - 2N)\underline{Y} + NU\underline{Y} + t_{e_0}N\underline{v}$$

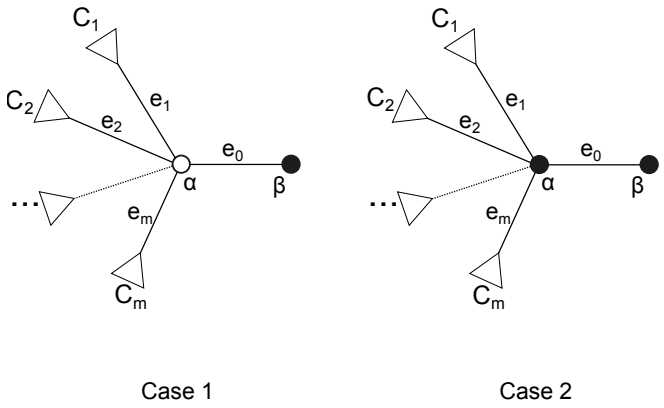


Figure A.2: The two cases for the terminal edge e_0. α is not labeled in case 1, and is labeled in case 2. The triangles represent subtrees.

For the internal edge e_0 we have

$$\begin{aligned}
W_0 &= \sum_{i=1}^{k}\sum_{j=k+1}^{m}\sum_{x\in C_i, y\in C_j} p_{xy} + \left[\sum_{j=1}^{k}\sum_{x\in C_j} t_{e_0} + p_{\alpha x}\right] + \left[\sum_{j=k+1}^{m}\sum_{x\in C_j} t_{e_0} + p_{\beta x}\right] + t_{e_0} \\
&= \left[\sum_{i=1}^{k}\sum_{j=k+1}^{m}\sum_{x\in C_i, y\in C_j} p_{\alpha x} + t_{e_0} + p_{\beta y}\right] + (n_\alpha + n_\beta - 1)t_{e_0} + \sum_{j=1}^{m} Y_j \\
&= \left[\sum_{i=1}^{k}\sum_{j=k+1}^{m} n_j Y_i + n_i n_j t_{e_0} + n_i Y_j\right] + (n_\alpha + n_\beta - 1)t_{e_0} + \sum_{j=1}^{m} Y_j \\
&= (n_\beta - 1)\sum_{i=1}^{k} Y_i + (n_\alpha - 1)\sum_{j=k+1}^{m} Y_j + (n_\alpha - 1)(n_\beta - 1)t_{e_0} + (n_\alpha + n_\beta - 1)t_{e_0} + \sum_{j=1}^{m} Y_j \\
&= n_\beta \sum_{i=1}^{k} Y_i + n_\alpha \sum_{i=k+1}^{m} Y_i + n_\alpha n_\beta t_{e_0} \\
&= \underline{v}^T \underline{Y} + n_\alpha n_\beta t_{e_0}
\end{aligned}$$

After substituting $\underline{Y}$ and rearranging we get,

$$t_{e_0} = \frac{W_0 - \underline{v}^T X^{-1} N^{-1} \underline{W}}{n_\alpha n_\beta - \underline{v}^T X^{-1} \underline{v}}$$

m

A.1.2 Terminal edges

Consider the terminal edge e_0 shown in Figure A.2 with adjacent edges $e_1, e_2 \ldots e_m$. e_0 is incident to the vertices α and β. The respective sizes of the sides of the split defined by e_0 are n_α and n_β. Since e_0 is a terminal edge the leaf β is labeled. There are two cases to consider depending on if α is labeled or not labeled.

If α is not labeled (Case 1 in Figure A.2), the edge length formula given by Bryant (1997) is

$$t_{e_0} = \frac{W_0 - \underline{v}^T X^{-1} N^{-1} \underline{W}}{n_\alpha n_\beta - \underline{v}^T X^{-1} \underline{v}}$$

where $n_\alpha = (n-1)$, $n_\beta = 1$ and $k = m$. If α is labeled (Case 2 in Figure A.2), the edge length formula can be derived as follows.

For edges e_i incident to α we have,

$$\begin{aligned}
W_i &= \sum_{x\in A_i}\sum_{y\in B_i} p_{xy} \\
&= \sum_{j=1,j\neq i}^{m}\sum_{x\in C_i}\sum_{y\in C_j} p_{xy} + \sum_{x\in C_i}(p_{\alpha x} + p_{\beta x}) \\
&= \sum_{j=1,j\neq i}^{m}\sum_{x\in C_i}\sum_{y\in C_j} (p_{\alpha x} + p_{\alpha y}) + \sum_{x\in C_i}(2p_{\alpha x} + t_{e_0}) \\
&= \sum_{j=1,j\neq i}^{m} [n_j Y_i + n_i Y_j] + 2Y_i + n_i t_{e_0} \\
&= (n - n_i - 2)Y_i + n_i \sum_{j=1,j\neq i}^{m} Y_j + 2Y_i + n_i t_{e_0} \\
&= (n - 2n_i)Y_i + n_i \sum_{j=1}^{m} Y_j + n_i t_{e_0}
\end{aligned}$$

In matrix form,

$$\underline{W} = (nI - 2N)\underline{Y} + NU\underline{Y} + t_{e_0} N\underline{v}$$

For the terminal edge e_0 we have,

$$\begin{aligned}
W_0 &= \sum_{i=1}^{m}\sum_{x\in C_i} p_{\beta x} + t_{e_0} \\
&= (\sum_{i=1}^{m}\sum_{x\in C_i} p_{\alpha x} + t_{e_0}) + t_{e_0} \\
&= \sum_{i=1}^{m} Y_i + (n-1)t_{e_0} \\
&= \underline{v}^T\underline{Y} + n_\alpha n_\beta t_{e_0}
\end{aligned}$$

where $n_\alpha = (n-1)$, $n_\beta = 1$ and $k = m$.

After substituting $\underline{Y}$ and rearranging we get,

$$t_{e_0} = \frac{W_0 - \underline{v}^T X^{-1} N^{-1}\underline{W}}{n_\alpha n_\beta - \underline{v}^T X^{-1}\underline{v}}$$

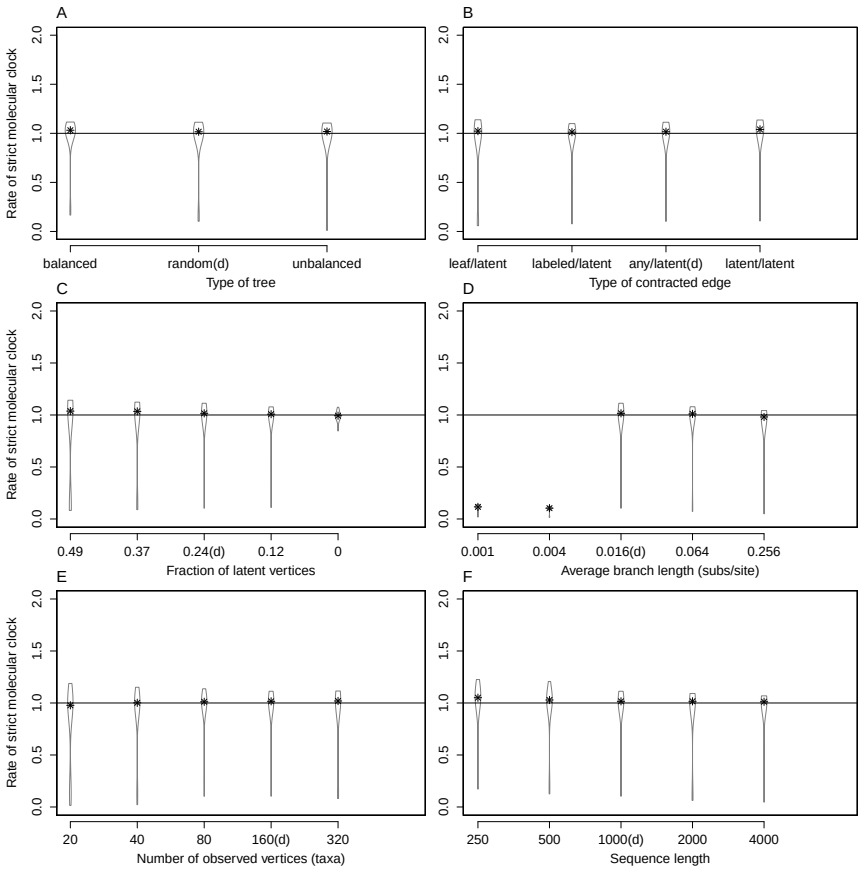


Figure A.3: Rate of the strict molecular clock that is estimated by SA. The true rate of the strict molecular clock is 1.0 subs./site/time in all simulation scenarios.

A.2 Molecular clock rate inferred by SA

A.3 Comparison of various FJ-based methods

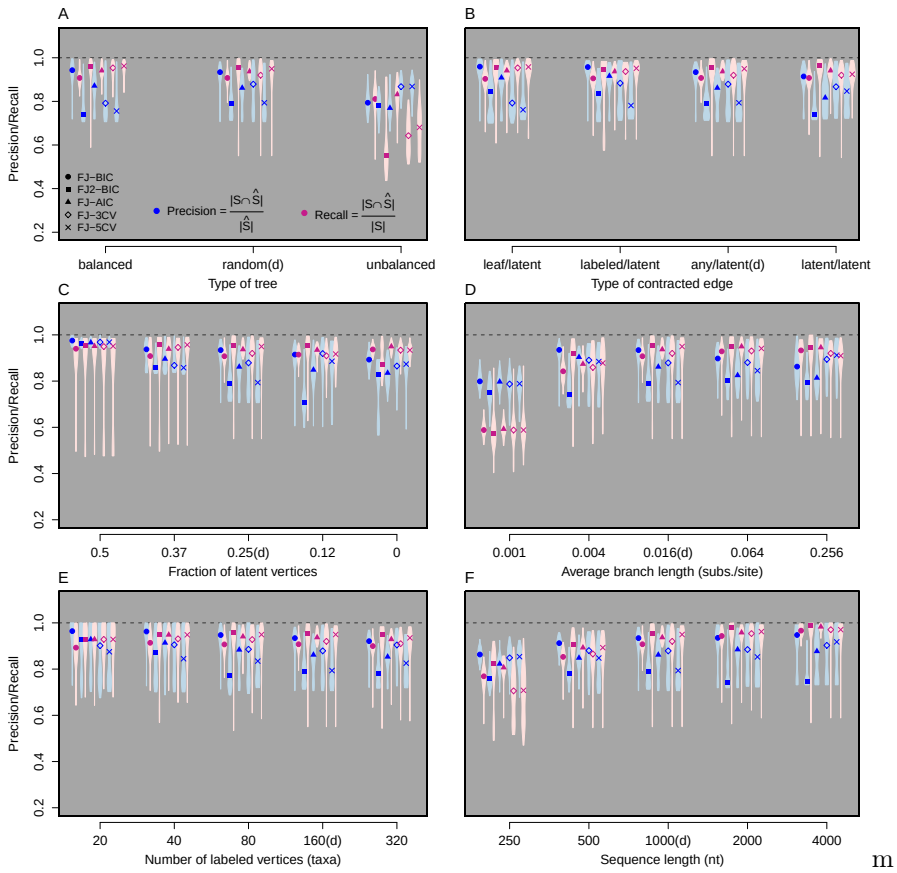


Figure A.4: A comparison of various FJ-based methods. FJ-BIC is the method that is presented in the main paper. FJ2-BIC checks if siblings have a parent using the criterion shown in equation 3.4 of the Chapter 3. FJ-AIC uses AIC for model selection. FJ-3CV and FJ-5CV performs model selection using 3-fold CV and 5-fold CV respectively.

Appendix B

Supplementary material for Chapter 5

B.1 Optimizing edge lengths

Given a continuous-time hidden Markov model $M_{\mathrm{CT}} = (\pi_\rho, \mathbf{Q}, \mathbf{t})$ on a rooted phylogenetic tree $T_\rho = (V_{T_\rho}, E_{T_\rho})$, M_{CT} is parameterized in terms of (i) a root probability distribution π_ρ, (ii) the set of rate matrices $\mathbf{Q} = \{Q_e : e \in E_{T_\rho}\}$, and (iii) the set of edge lengths $\mathbf{t} = \{t_e : e \in E_{T_\rho}\}$. The transition matrix P_e for edge e is computed as $P_e = e^{Q_e t_e}$.

Edge lengths were optimized with Newton-Raphson using a convergence threshold of 10^{-4} substitutions/site, as described below.

$$t_e^{\mathrm{u}} = t_e^{\mathrm{c}} - \left(\frac{\partial^2 \ell}{\partial t_e^2}\right)^{-1} \frac{\partial \ell}{\partial t_e}$$

, where ℓ is the log likelihood score, t_e^{u} and t_e^{c} are the updated edge length and the current edge length, respectively, of edge e. The partial derivatives $\dfrac{\partial \ell}{\partial t_e}$ and $\dfrac{\partial^2 \ell}{\partial t_e^2}$ are evaluated at $t_e = t_e^{\mathrm{c}}$.

The current Section lists the equations for computing the first and second order partial derivatives of the log likelihood score with respect to an edge length. Assuming i.i.d. we have

$$\ell = \sum_i w^i \log L^i$$

, where w^i is the number of times that the site pattern for site i is repeated, and L^i is the likelihood score for site i. L^i is computed as follows:

$$L^i = \sum_x \pi_\rho(x) L_\rho^i(x)$$

, where L_ρ^i is the conditional likelihood for site i that is computed recursively using the following equation that applies for each non-leaf vertex of T.

$$L_u^i(x) = \left(\sum_y P_{(u,v)}(y|x) L_v^i(y)\right)\left(\sum_z P_{(u,w)}(z|x) L_w^i(z)\right)$$

where v and w are the children of u, and u has two children.

$$P_{(u,v)} = e^{Q_{(u,v)} t_{(u,v)}}$$

The first derivative of log likelihood taken with respect to any edge length $t_{(a,b)}$ can be computed as follows

$$\frac{\partial l}{\partial t_{(a,b)}} = \sum_i \frac{w^i}{L^i} \times \frac{\partial L^i}{\partial t_{(a,b)}}$$

where

$$\frac{\partial L^i}{\partial t_{(a,b)}} = \sum_x \pi_\rho(x) \frac{\partial L^i_\rho(x)}{\partial t_{(a,b)}}$$

Let v and w be the children of u. Given an edge (a,b) the first derivative $\frac{\partial L^i_u(x)}{\partial t_{(a,b)}}$ can be calculated as follows.

$$\begin{aligned}\frac{\partial L^i_u(x)}{\partial t_{(a,b)}} = &\left(\sum_y \frac{\partial P_{(u,v)}(y|x)}{\partial t_{(a,b)}} L^i_v(y) + P_{(u,v)}(y|x)\frac{\partial L^i_v(y)}{\partial t_{(a,b)}}\right)\left(\sum_z P_{(u,w)}(z|x)L^i_w(z)\right) \\ &+ \left(\sum_y P_{(u,v)}(y|x)L^i_v(z)\right)\left(\sum_z \frac{\partial P_{(u,w)}(z|x)}{\partial t_{(a,b)}} L^i_w(z) + P_{(u,w)}(z|x)\frac{\partial L^i_w(z)}{\partial t_{(a,b)}}\right)\end{aligned}$$

where v and w are children of u.

$$\frac{\partial P_{(u,v)}(y|x)}{\partial t_{(a,b)}} = \begin{cases} Q_{(u,v)} e^{Q_{(u,v)} t_{(u,v)}} & \text{if } t_{(a,b)} = t_{(u,v)} \\ 0 & \text{otherwise} \end{cases}$$

$\frac{\partial L^i_u(x)}{\partial t_{(a,b)}}$ equals zero for any x if u is a leaf.

The expression for $\frac{\partial L^i_u(x)}{\partial t_{(a,b)}}$ simplifies as follows for any u that is not a leaf. Let b and c be the two children of a. Let $p_T(\rho, a)$ be the directed path in T from root ρ to a.

$$\frac{\partial L^i_u(x)}{\partial t_{(a,b)}} = \begin{cases} \left(\sum_y \left[Q_{(a,b)} e^{Q_{(a,b)} t_{(a,b)}}\right](y|x) L^i_b(y)\right)\left(\sum_z P_{(a,c)}(z|x) L^i_c(z)\right) & \text{if } u = a \\ \left(\sum_y P_{uv}(y|x)\frac{\partial L^i_v(y)}{\partial t_{(a,b)}}\right)\left(\sum_z P_{(u,w)}(z|x)L^i_w(z)\right) & \text{if } u \neq a, \text{ and } u,\ v \text{ are in } p_T(\rho,a) \\ \text{and } v \text{ are in the directed path from } \rho \text{ to } a0 & \text{otherwise} \end{cases} \tag{B.1}$$

The second derivative of log likelihood taken with respect to the edge length $t_{(a,b)}$ can be computed as follows

$$\frac{\partial^2 \ell}{\partial t^2_{(a,b)}} = \sum_i w^i \left(\frac{1}{L^i}\frac{\partial^2 L^i}{\partial t^2_{(a,b)}} - \left(\frac{1}{L^i}\frac{\partial L^i}{\partial t_{(a,b)}}\right)^2\right)$$

where

$$\frac{\partial^2 L^i}{\partial t^2_{(a,b)}} = \sum_x \pi_\rho(x) \frac{\partial^2 L^i_\rho(x)}{\partial t^2_{(a,b)}}$$

Let v and w be the children of u. Given an edge (a,b), such that b and c are the children of a, the second derivative $\frac{\partial^2 L^i_u(x)}{\partial t^2_{(a,b)}}$ of any u that is not a leaf can be calculated as

$$\frac{\partial^2 L_u^i(x)}{\partial^2 t_{(a,b)}} = \begin{cases} \left(\sum_y \left[Q_{(a,b)}^2 e^{Q_{(a,b)}t_{(a,b)}}\right](y|x)L_b^i(y)\right)\left(\sum_z P_{(a,c)}(z|x)L_c^i(z)\right) & \text{if } u \text{ equals } a \\ \left(\sum_y P_{(u,v)}(y|x)\frac{\partial^2 L_v^i(y)}{\partial^2 t_{(a,b)}}\right)\left(\sum_z P_{(u,w)}(z|x)L_w^i(z)\right) & \text{if } u \text{ is not } a, \text{ and } u, v \text{ are in } p_T(\rho, a) \\ 0 & \text{otherwise} \end{cases} \tag{B.2}$$

, where $p_T(\rho, a)$ is the directed path in T from ρ to a. $\frac{\partial^2 L_u^i(x)}{\partial t_{(a,b)}^2}$ equals zero for any x if u is a leaf.

B.1.1 Avoiding numerical underflow using scaling factors

Each element of a conditional likelihood vector is a product of fractions (see equation 2.7). The computation of entries of conditional likelihood vectors is susceptible to numerical underflow for large trees. We applied the commonly used technique of scaling each conditional likelihood vector with a small scaling factor, and storing the log transformed value of the scaling factor (Yang, 2000). We used a similar technique to scale the first derivative and the second derivative, respectively, of the conditional likelihood vector.

The scaling factor $\text{sf}(L_u^i)$ of any vector L_u^i is defined as the entry in $L^i(u)$ that has the largest absolute value. Scaling a vector involves dividing each element of the vector with the scaling factor of the vector. A log transformed factor $\text{lgsf}(u)$ is computed recursively for each vertex as follows.

$$\text{lgsf}(L_u^i) = \log(\text{sf}(L_u^i)) + \text{lgsf}(L_v^i) + \text{lgsf}(L_w^i)$$

, where v and w are the children of u. $\text{lgsf}(L_u^i)$ is zero if u is a leaf. L_u^i is computed using scaled conditional likelihood vectors as follows.

$$L_u^i(x) = \left(\sum_y P_{(u,v)}(y|x)\overline{L_v^i}(y)\right)\left(\sum_z P_{(u,w)}(z|x)\overline{L_w^i}(z)\right)$$

where $\overline{L_v^i}$ is the conditional likelihood vector that is obtained by scaling L_u^i.

Log likelihood ℓ is computed as follows.

$$\ell = \sum_i w^i \left(\text{lgsf}(L_\rho^i) + \log L^i\right)$$

, where

$$L^i = \sum_x \pi_\rho(x)\overline{L_\rho^i}(x)$$

The first derivative $\frac{\partial L_u^i(x)}{\partial t_{(a,b)}}$, and the second derivative $\frac{\partial^2 L_u^i(x)}{\partial^2 t_{(a,b)}}$ are computed recursively using equation B.1, and equation B.2, respectively, using scaled versions of conditional likelihood vectors, and derivatives of conditional likelihood vectors.

The first derivative of log likelihood w.r.t. to any edge length $t_{(a,b)}$ is computed as follows

$$\frac{\partial \ell}{\partial t_{(a,b)}} = \sum_i w^i \frac{\partial \ell^i}{\partial t_{(a,b)}}$$

, where

$$\frac{\partial \ell^i}{\partial t_{(a,b)}} = \exp\left(\text{lgsf}\left(\frac{\partial L_\rho^i}{\partial t_{(a,b)}}\right) - \text{lgsf}\left(L_\rho^i\right)\right) \times \frac{1}{L^i}\frac{\partial L^i}{\partial t_{(a,b)}} \tag{B.3}$$

The log transformed factor $\text{lgsf}\left(\frac{\partial L_u^i}{\partial t_{(a,b)}}\right)$ for the first derivative of conditional likelihood vector $\frac{\partial L_u^i}{\partial t_{(a,b)}}$ is computed as follows.

$$\text{lgsf}\left(\frac{\partial L_u^i}{\partial t_{(a,b)}}\right) = \begin{cases} \log\left(\text{sf}\left(\frac{\partial L_u^i}{\partial t_{(a,b)}}\right)\right) + \text{lgsf}(L_b^i) + \text{lgsf}(L_c^i) & \text{if } u \text{ equals } a \\ \log\left(\text{sf}\left(\frac{\partial L_u^i}{\partial t_{(a,b)}}\right)\right) + \text{lgsf}\left(\frac{\partial L_v^i}{\partial t_{(a,b)}}\right) + \text{lgsf}(L_w^i) & \text{if } u \text{ is not } a, \text{ and } u, v \text{ are in } p_T(\rho, a) \\ 0 & \text{otherwise} \end{cases}$$

The second derivative of log likelihood w.r.t. to any edge length $t_{(a,b)}$ is computed as follows

$$\frac{\partial^2 \ell}{\partial t_{(a,b)}^2} = \sum_i w^i \frac{\partial^2 \ell^i}{\partial t_{(a,b)}^2}$$

, where

$$\frac{\partial^2 \ell^i}{\partial t_{(a,b)}^2} = \left(\exp\left(\text{lgsf}\left(\frac{\partial^2 L_\rho^i}{\partial t_{(a,b)}^2}\right) - \text{lgsf}\left(L_\rho^i\right)\right) \times \frac{1}{L^i}\frac{\partial L^i}{\partial t_{(a,b)}}\right) - \left(\frac{\partial \ell^i}{\partial t_{(a,b)}}\right)^2 \tag{B.4}$$

The log transformed factor $\text{lgsf}\left(\frac{\partial^2 L_u^i}{\partial t_{(a,b)}^2}\right)$ for the second derivative of conditional likelihood vector $\frac{\partial^2 L_u^i}{\partial t_{(a,b)}^2}$ is computed as follows. Let b and c be the two children of a. Let $p_T(\rho, a)$ be the directed path from the root ρ to vertex a.

$$\frac{\partial^2 L_u^i(x)}{\partial^2 t_{(a,b)}} = \begin{cases} \log\left(\text{sf}\left(\frac{\partial^2 L_u^i}{\partial t_{(a,b)}^2}\right)\right) + \text{lgsf}(L_b^i) + \text{lgsf}(L_c^i) & \text{if } u \text{ equals } a \\ \log\left(\text{sf}\left(\frac{\partial^2 L_u^i}{\partial t_{(a,b)}^2}\right)\right) + \text{lgsf}\left(\frac{\partial^2 L_v^i}{\partial t_{(a,b)}^2}\right) + \text{lgsf}(L_w^i) & \text{if } u \text{ is not } a, \text{ and } u \text{ is in } p_T(\rho, a) \\ 0 & \text{otherwise} \end{cases}$$

The optimization procedures for estimating edge lengths were implemented in C++. We used *double* for storing the derivatives of conditional likelihood vectors. We found that the exponents in equation B.3 and equation B.4 were within the range of permissible values for *double* for all empirical data sets and simulated data sets that were analyzed. Additionally, we found that $L \times \frac{\partial \ell^i}{\partial t_{(a,b)}}$ and $L \times \frac{\partial^2 \ell^i}{\partial t_{(a,b)}^2}$ was within the range of permissible values for *double* for each site i, where L is the number of columns in the multiple sequence alignment. We concluded that we avoided numerical underflow and numerical overflow in the computation of the first derivatives and the second derivatives, respectively, of the log likelihood score for each data set.