



universität
wien

MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

**“A comparison between saturation and bootstrapping
to measure phylogenetic information”**

verfasst von / submitted by

Clemens Heiderer BSc

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien, 2024 / Vienna 2024

Studienkennzahl lt. Studienblatt /
degree programme code as it appears on
the student record sheet:

UA 066 875

Studienrichtung lt. Studienblatt /
degree programme as it appears on
the student record sheet:

Masterstudium Bioinformatik

Betreut von / Supervisor:

Univ.-Prof. Dr. Arndt von Haeseler

Contents

1	Introduction	6
2	The Evolutionary Model	9
3	A Test for Saturation	19
4	Bootstrap	26
5	Results and Discussion	30
5.1	Comparison between Saturation and UFBoot Support Values	30
5.2	Statistical Analysis	35
5.3	Bootstrapping Behavior in the Presence of Long Saturated Branches	43
5.4	Influence of Clades Separated by a Saturated Branch on Tree Reconstruction	54
5.4.1	Analysis of Average UFBoot Values	58
5.4.2	Robinson-Foulds Distances to Assess Left and Right Clade Independence	59
6	Conclusion	63
A	Appendix	70
A.1	Manuel:	70
A.1.1	correlation.py	70
A.1.2	Subtree Creator.py	71
A.1.3	nodestyle.py	72
A.2	GitHub	73
A.3	Statistics	74

Acknowledgements

To Carmen Viñas, I express my gratitude, as she was the first person who implemented Satute. My sincere thanks go to Christiane Elgert and Enes Sakalli for their dedicated contributions to the implementation of Satute throughout my thesis project. I'm thankful for the suggested corrections by Heiko Schmidt.

My main thanks go to Cassius Manuel the inventor and author of Satute, I particularly appreciate his lecture introducing me to Satute and his continual availability for Zoom consultations. His passion for the project and willingness to share his knowledge helped me a lot.

Lastly, I'm thankful for Arndt von Haeseler, his expertise, and his ideas in the meetings. My thanks go to Gudrun Koller. I thank god.

Abstract

Evolutionary relationships among species are often reconstructed using Maximum Likelihood (ML) trees based on genetic sequences from a multiple sequence alignment (MSA).

Saturation in a multiple sequence alignment (MSA) occurs when the number of substitutions between sequences reaches a point where they no longer reflect true evolutionary relationships accurately. Satute employs an innovative approach utilizing maximum likelihood and two likelihood vectors flanking the analyzing branch to detect saturated branches in multiple sequence alignments, dividing the tree at the tested branch into two subclades rooted by the branch. A likelihood vector is a mathematical representation, typically a one-dimensional array, that encapsulates the likelihood scores of observing various nucleotide patterns at a specific position in a sequence alignment. The method evaluates whether sufficient phylogenetic information remains in both subclades to classify the branch as informative.

It all began with a comparison of saturation values and UFBoot values. UFBoot values, a method used in phylogenetics to assess the confidence in branching patterns by resampling site-likelihoods, and saturation values both utilize maximum likelihood methods. This initial assumption, coupled with the discovery that saturation values and bootstrap values do not correlate, led us to the next question. Observing very long branches with UFBoot values of 100 prompted us to investigate why bootstrapping in general might assign misleading confidence values in these cases. The high UFBoot values for very long branches with neighboring short branches are due to the dominant influence these branches exert on the overall tree structure across numerous pseudoalignments, which will be similar to the original starting alignments, no matter the resampling.

We sought to understand the impact of a saturated branch with high UFBoot values on the neighboring clades to its left and right. Based on one limited experiment, using a single model tree, our surprising findings suggest that information from one clade in the MSA can contribute to a more accurate reconstruction of the other clade's topology despite being separated by a saturated branch. These findings challenge the assumption of complete independence between clades on either side of a saturated branch during Maximum Likelihood (ML) tree inference. Further research with more different models is warranted to solidify these findings.

Keywords: maximum likelihood (ML) trees, multiple sequence alignment (MSA), saturation, likelihood vectors, UFBoot, pattern, clade

Abstract

Die evolutionären Beziehungen zwischen den Arten werden häufig mithilfe von Maximum-Likelihood-Bäumen (ML) rekonstruiert, die auf genetischen Sequenzen aus einem Multiple-Sequence-Alignment (MSA) basieren.

Sättigung in einem Multiplen Sequenzalignment (MSA) tritt auf, wenn die Anzahl der Substitutionen zwischen Sequenzen einen Punkt erreicht, an dem sie die wahren evolutionären Beziehungen nicht mehr genau widerspiegeln. Satute verwendet einen innovativen Ansatz, bei dem die maximale Wahrscheinlichkeit und zwei Likelihood-Vektoren, die den analysierten Zweig flankieren, verwendet werden, um gesättigte Zweige im MSA zu erkennen, indem der Baum an dem getesteten Zweig in zwei durch den Zweig verwurzelte Subkladen aufgeteilt wird. Ein Likelihood-Vektor ist eine mathematische Darstellung, in der Regel ein eindimensionales Array, das die Wahrscheinlichkeitsbewertungen für die Beobachtung verschiedener Nukleotidmuster an einer bestimmten Position in einem Sequenzalignment enthält. Die Methode bewertet, ob genügend phylogenetische Informationen in beiden Unterkladen verbleiben, um den Zweig als informativ einzustufen.

Alles begann mit einem Vergleich von Sättigungswerten und UFBoot-Werten. UFBoot-Werte, die Konfidenzwerte von Zweigstrukturen durch Resampling von Site-Likelihoods erfassen, und Sättigungswerte basieren beide auf Maximum-Likelihood-Methoden. Diese anfängliche Annahme, zusammen mit der Feststellung, dass Sättigungswerte und Bootstrap-Werte nicht korrelieren, führte uns zur nächsten Frage. Die Beobachtung sehr langer Zweige mit UFBoot-Werten von 100 veranlasste uns zu der Frage, warum UFBoot in diesen Fällen irreführende Vertrauenswerte zuweist. Die hohen UFBoot-Werte für sehr lange Äste mit benachbarten kurzen Ästen resultieren aus dem dominanten Einfluss, den diese Äste auf die Gesamtstruktur des Baumes über zahlreiche Pseudoalignments ausüben, die den ursprünglichen Ausgangsalignments ähnlich sind, unabhängig vom Resampling.

Ziel der nächsten Untersuchung war, welchen Einfluss ein gesättigter Zweig mit hohen UFBoot-Werten auf die benachbarten Kladen links und rechts davon hat. Basierend auf einem begrenzten Experiment mit einem einzelnen Modellbaum deuten unsere überraschenden Ergebnisse darauf hin, dass Informationen von einem Klade in dem MSA zu einer genaueren Rekonstruktion der Topologie des anderen Klades beitragen können, obwohl sie durch einen gesättigten Zweig getrennt sind. Diese Ergebnisse stellen die Annahme der vollständigen Unabhängigkeit zwischen Kladen auf beiden Seiten eines gesättigten Zweigs während der Maximum-Likelihood (ML)-Baumrekonstruktion in Frage. Weitere Forschung mit verschiedenen Modellen ist erforderlich, um diese Ergebnisse zu festigen.

Schlagwörter: Maximum-Likelihood (ML) Bäume, Multiples Sequenzalignment (MSA), Sättigung, Likelihood-Vektoren, UFBoot, Sequenzmuster, Klade

List of Abbreviations

Glossary

α_S	significance level in Satute.	ML	maximum likelihood.
z_α	alpha-percentile denotes the critical value of the standard normal distribution in Satute.	MSA	multiple sequence alignment.
CC	correlation coefficient.	n_a	number of Alignments.
CSC	correct sister clades.	nt	nucleotide.
DIFF	the difference in the percentage of saturated branches between File 1 and File 2, performed on the same alignment.	p-value (w)	weighted p-value.
EXTB	average number of external single branches per tree.	p-value (w/o)	p-value without outliers.
Inf T	inferred trees.	r0	uppermost branch adjacent to the center within the right clade.
JB	junction branch.	r1	bottom branch next to the center within the right clade.
JBs	junction branches.	RC	rate category.
l0	uppermost branch adjacent to the center within the left clade.	RF	Robinson–Foulds.
l1	bottom branch next to the center within the left clade.	SB	saturated branches.
len	length.	SD	standard deviation.
MDIFF	the mean DIFF from all alignments.	seq	sequence.
		Subst. Mod.:	substitution model.
		t-value (w)	weighted t-value.
		t-value (w/o)	t-value without outliers.
		TV	tree variation.
		UFBoot	ultrafast bootstrap approximation.

1: Introduction

Maximum Likelihood methods and the substitution process modeled as a Markov process are fundamental in tree reconstruction, widely employed in programs like IQ-TREE2 (Minh et al., 2020). The ML method calculates the likelihood of observed data given a proposed phylogenetic tree and associated model parameters, including substitution rates, among-site rate variation, and nucleotide frequencies.

For a comprehensive understanding of these methods, foundational textbooks such as those by Felsenstein (Felsenstein, 2003) and Lemey et al. (Lemey et al., 2009) provide essential insights. Felsenstein's book "Inferring Phylogenies" is a seminal work in phylogenetic analysis, covering key concepts in tree reconstruction and statistical inference. Similarly, "The Phylogenetic Handbook" by Lemey, Salemi, and Vandamme offers a practical approach to phylogenetic analysis and hypothesis testing, making it a valuable resource for researchers in the field. In the chapters by Strimmer and von Haeseler on genetic distances and phylogenetic models, phylogenetic saturation is addressed. The chapter by von Haeseler and Schmidt focuses on phylogenetic inference using maximum likelihood methods.

Satute, a Python tool developed by Manuel, uses an innovative approach that employs maximum likelihood and two likelihood vectors flanking the analyzed branch to detect saturated branches, using a multiple sequence alignment as input.

A likelihood vector is typically a one-dimensional array representing the likelihood of observing different possible patterns (nucleotides or amino acids) at a specific position in a sequence alignment. Imagine a set of dice (representing possible nucleotide combinations) for a specific position. Each die face represents a different combination (e.g., AAAA, ACGT). The likelihood vector holds the values (likelihood scores), similar to the probabilities of each die face landing, but based on the evolutionary history depicted by the tree and the branch lengths.

Question

Von Haeseler proposed an alternative approach to consider whether a branch in a phylogenetic tree, with sequences on both sides of the clades, retains informative biological content. Whether the edge, estimated through maximum likelihood, in terms of either time or branch length, still contains meaningful evolutionary information.

Has evolution progressed to the extent that the branch no longer exhibits differentiation from an infinite evolutionary past, resulting in both sequences being merely coincidentally similar to each other? Further with the implementation of Satute, a question was if we can draw a correlation between the saturation values obtained from the implementation of Satute and the well-established bootstrapping values.

IQ-TREE2

IQ-TREE2 (Minh et al., 2019) which is used in conjunction with the Maximum Likelihood method is valued for robustness and memory efficiency besides speed and flexibility. IQ-TREE2 starts with selecting the best-fit model based on the corrected AIC (Akaike Information Criterion) (Akaike, 1974), and BIC (Bayesian Information Criterion) (Schwarz, 1978), and executes likelihood ratio tests. After estimating the optimal values of the model parameters, it chooses a random tree and performs the tree search heuristic hill climbing NNI (Nearest-Neighbor Interchange) (Collienne and Gavryushkin, 2021) for finding a locally optimal tree and improves it by SPR (Subtree Pruning and Regrafting) (Bordewich and Semple, 2005). The program computes the log-likelihood of sites, identifies the optimal tree using methods like hill climbing NNI (Nearest-Neighbor Interchange), and assigns posterior rate categories to each site according to the chosen evolutionary model.

History of Saturation Detection in Phylogenetic Alignments

Currently, there is no efficient and consistently reliable method available for ascertaining the reliability of an evolutionary tree (Jermiin et al., 2004). The impact of a decaying historical signal can lead to a declining ratio of the lengths of long branches to internal branches as time increases, which is a symptom of substitutional saturation. Incorrect branches may appear longer due to higher substitution rates along these edges (Ho and Jermiin, 2004). Therefore, it is important to identify sequences in an alignment that lack phylogenetic information due to saturation.

Consider a branch in an evolutionary tree where we observe a certain number of nucleotide substitutions between sequences. "Observed difference" refers to the differences detected when comparing nucleotide sequences at corresponding positions in an alignment. The Hamming distance is often used to quantify these observed differences, as it counts the number of positions where the nucleotides differ between two sequences. However, due to substitutional saturation—where multiple substitutions occur at the same site over time—the actual number of substitutions may be much higher than the Hamming distance. This leads to a situation where the observed genetic distance (the Hamming distance) significantly underestimates the true genetic distance (the total number of substitutions that have occurred) (Philippe et al., 2011). Consequently, such an alignment may no longer provide highly accurate information about the evolutionary history, due to the underestimation of the true genetic distance caused by substitutional saturation.

Xia (2003) introduced a saturation index based on entropy, which assesses the extent of phylogenetic information loss due to multiple substitutions at the same site over evolutionary time. As sequences approach full substitution saturation, the expected nucleotide frequencies at each site tend to align closely with the global nucleotide frequencies observed across all sequences in the dataset.

Global frequencies refer to the overall proportions of nucleotides (A, T, C, G) observed across all sequences in a dataset. They represent the aggregate composition of nucleotides in the entire set of sequences analyzed. If the global frequency of nucleotide A is 30%, it means

that nucleotide A is observed in 30% of the nucleotide positions across the entire alignment.

Jermiin saturation test employs the observed number of substitutions per site between two sequences. He introduces a loss of historical signal and a compositional distance which quantifies if the sequences are saturated or not (Jermiin and Misof, 2020).

The saturation index described here applies uniformly to each site and does not incorporate tree structure or any rate matrix, in contrast to Satute (Manuel, 2021).

(Duchêne et al., 2022) examines his test on sites in the phylogenetic data that provide significant evolutionary information. He excludes sites where the expected nucleotide frequencies are similar to the global frequencies found across all sequences.

The novel saturation model, Satute, developed by Manuel and von Haeseler, introduces a method for testing internal branches using partial likelihood vectors to represent branch nodes. Following the inference of an evolutionary model and phylogenetic tree, the alignment undergoes subdivision into two subalignments at each branch. This subdivision enables the assessment of whether these subalignments collectively provide sufficient shared information to support the existence of the branch. If the average values of these partial likelihood vectors approach zero, it indicates reduced phylogenetic reliability for that specific branch.

The Satute saturation test is implemented in Python and employs a direct z-test rather than simulation-based approaches. This methodology offers a deeper understanding of the phylogenetic information embedded within an alignment. Central to saturation test development is the assumption of independent mutations at sites, guided by a reversible rate matrix specified alongside the phylogenetic tree. We further demonstrate the application of spectral decomposition of the rate matrix to construct the z-test for evaluating branch saturation within the phylogenetic tree.

2: The Evolutionary Model

In the context of phylogenetic inference, we define the substitution probability matrix, denoted by $\mathbf{P}_{ij}(t)$, where i and j represent different nucleotides and t represents evolutionary time. This matrix $\mathbf{P}_{ij}(t)$ represents the probability that a base j will be found at a specific position after time t , given that there was a base i at that position at time 0.

It is constructed from the rate matrix $\mathbf{R}$ and the equilibrium frequencies $\boldsymbol{\pi}$. The rate matrix $\mathbf{R}$ describes the rates of nucleotide substitutions between different bases in a DNA sequence. For instance, r_{AC} denotes the rate at which adenine (A) mutates to cytosine (C), reflecting the relative frequency of this specific nucleotide change. The diagonal elements are set to zero, reflecting the concept of a nucleotide remaining unchanged. This simplification ensures that the sum of each row in $\mathbf{R}$ represents the total rate of a nucleotide undergoing any type of substitution and transitioning away from its original state.

The equilibrium frequencies $\boldsymbol{\pi}$ represent the long-term proportions of each base in the sequence, assuming a state of equilibrium.

The rate matrix $\mathbf{R}$ and the equilibrium frequencies $\boldsymbol{\pi}$ are combined to form the instantaneous rate matrix $\mathbf{Q}$.

$$\mathbf{R} = \begin{bmatrix} 0 & r_{AC} & r_{AG} & r_{AT} \\ r_{CA} & 0 & r_{CG} & r_{CT} \\ r_{GA} & r_{GC} & 0 & r_{GT} \\ r_{TA} & r_{TC} & r_{TG} & 0 \end{bmatrix} \quad \boldsymbol{\pi} = \begin{bmatrix} \pi_A \\ \pi_C \\ \pi_G \\ \pi_T \end{bmatrix} \quad (2.1)$$

$$\mathbf{Q} = \begin{bmatrix} * & \pi_A r_{AC} & \pi_A r_{AG} & \pi_A r_{AT} \\ \pi_C r_{CA} & * & \pi_C r_{CG} & \pi_C r_{CT} \\ \pi_G r_{GA} & \pi_G r_{GC} & * & \pi_G r_{GT} \\ \pi_T r_{TA} & \pi_T r_{TC} & \pi_T r_{TG} & * \end{bmatrix} \quad (2.2)$$

The rate matrix $\mathbf{Q}$ has off-diagonal entries $q_{ij} > 0$ that represent the instantaneous substitution rates at which nucleotide i is replaced by nucleotide j (where $i \neq j \in \{A, C, G, T\}$). Here, the symbol "*" indicates that the diagonal elements of the matrix $\mathbf{Q}$ are negative values, which satisfy the condition:

$$q_{ii} = - \sum_{j \neq i} q_{ij}, \quad (2.3)$$

ensuring that rows sum to zero, reflecting the stationary distribution of nucleotides. $\mathbf{Q}$ provides insights into the instantaneous rates at which a continuous-time Markov chain transitions between states (Meyn and Tweedie, 2012; Rubino and Sericola, 1989).

The substitution probability matrix

$$\mathbf{P}(t) = e^{\mathbf{Q}t} \quad (2.4)$$

utilizes the rate matrix $\mathbf{Q}$ and evolutionary time t . The exponential function models continuous-time Markov processes, describing probabilities of nucleotide substitutions over time and facilitating calculations of evolutionary processes along phylogenetic tree branches.

Matrix $\mathbf{P}$ is stochastic, featuring non-negative elements where each row forms a probability vector (Weng et al., 2011). This characteristic ensures that the total probability across all states is 1 (Yutsumura, 2024). In phylogenetics, $\mathbf{P}$ is crucial for understanding evolutionary dynamics and is integral to various models for predicting sequence likelihoods given a phylogenetic tree.

Site Rate Heterogeneity and the Gamma model

Not all parts of a genetic sequence mutate at the same pace. Site rate heterogeneity means that certain positions in the sequence are more susceptible to genetic mutations, while others may remain relatively stable.

Yang (1994) proposed the Γ model to accommodate rate heterogeneity, characterized by the shape parameter α . Typically, rate heterogeneity is introduced using Γ -distributed rates, modeled by a discrete Γ -distribution with a fixed number of rate categories, usually ranging from 4 to 8, such as +G4 (Schmidt, 2022). Approximating the continuous rate distribution with several rate classes, where each is assigned equal percentiles and probabilities, reduces computational burden (Jia et al., 2014).

Each category within this distribution is associated with a specific rate multiplier. This rate multiplier scales the substitution rates of sites assigned to that category relative to a baseline rate or average rate (Felsenstein, 2003). More rate categories enhance the fit of the discrete gamma distribution to the continuous gamma distribution (Jia et al., 2014).

The parameter α in the continuous gamma distribution describes the distribution of substitution rates across sites. Smaller values of α (< 1) indicate an L-shaped distribution, suggesting pronounced rate heterogeneity, with some sites evolving rapidly while others evolve slowly.

Conversely, larger values of α (> 1) result in a unimodal distribution, indicating rates are more concentrated around a central value. This implies reduced rate heterogeneity, with most sites evolving closer to the average rate (Nguyen et al., 2017).

Spectral Decomposition

A key step in our phylogenetic model is the spectral decomposition (Levin and Peres, 2017) of the instantaneous rate matrix $\mathbf{Q}$ into its eigenvalues and eigenvectors.

The requirement that each row sum to zero in the instantaneous rate matrix $\mathbf{Q}$ ensures that the total probability across all possible states at any given time is 1. This indicates that a nucleotide will either transition to another state or remain unchanged.

An evolutionary process is considered **stationary** if the initial state distribution converges to a unique equilibrium nucleotide distribution $\boldsymbol{\pi}$, which fulfills $\boldsymbol{\pi}^T \mathbf{Q} = \mathbf{0}$ and $\boldsymbol{\pi}^T \mathbf{P}(t) = \boldsymbol{\pi}$.

Statistical model methods for phylogenetic reconstruction are typically required to be time-reversible (Sumner et al., 2012). It assumes that the Markov process modeling evolution occurs in the same way if followed forward or backward in time (Hoff et al., 2016).

Reversibility, expressed as:

$$\pi_i q_{ij} = \pi_j q_{ji}, \quad (2.5)$$

for $i \neq j \in \{A, C, G, T\}$, ensures that the substitution process is symmetric over time. This equation, known as the detailed balance condition, specifies that in the equilibrium frequency of a nucleotide, the flow of probability from state i to state j must be equal to the flow from state j to state i .

This equation is a characteristic of continuous-time Markov chains, and it is a special case of time-reversibility (Mazumdar, 2012). It is crucial in molecular evolution models to ensure a balanced flow between nucleotide states, which contributes to the model's ability to reach and maintain a stationary distribution.

The rate matrix $\mathbf{Q}$ contains 9 free parameters (Weiss and von Haeseler, 2003): Due to the reversibility of the substitution process, it applies: $(\pi_A r_{AC} = \pi_C r_{CA})$, $(\pi_A r_{AG} = \pi_G r_{GA})$, $(\pi_A r_{AT} = \pi_T r_{TA})$, $(\pi_C r_{CG} = \pi_G r_{GC})$, $(\pi_C r_{CT} = \pi_T r_{TC})$, $(\pi_G r_{GT} = \pi_T r_{TG})$. Thus, there are 6 free parameters for the substitution rates. In addition, the equilibrium base frequencies π_A , π_C , π_G , and π_T are subject to the constraint: $\sum_{i \in \{A, C, G, T\}} \pi_i = 1$. This constraint means that the sum of the equilibrium frequencies must equal 1, which reduces the number of free frequency parameters to 3, as the fourth parameter is determined by the others.

This parameterization ensures that the rate matrix $\mathbf{Q}$ correctly models the substitution process with the desired properties of stationarity and reversibility. A model can achieve stationarity even without perfect reversibility, as long as the overall rate of change for each nucleotide state eventually becomes zero.

Assuming the rate matrix $\mathbf{Q}$ is diagonalizable, which is typically the case in phylogenetics, it can be decomposed spectrally as follows:

$$\mathbf{Q} = \mathbf{U} \text{diag}[\lambda_0, \lambda_1, \lambda_2, \lambda_3] \mathbf{U}^{-1}, \quad (2.6)$$

where $\mathbf{U}$ is the matrix of right eigenvectors $\mathbf{v}_k$ of $\mathbf{Q}$, and $\text{diag}[\lambda_0, \lambda_1, \lambda_2, \lambda_3]$ is a diagonal matrix containing the eigenvalues $\lambda_0, \lambda_1, \lambda_2, \lambda_3$ of $\mathbf{Q}$ (Strang, 2012). Each column of $\mathbf{U}$,

denoted as $\mathbf{v}_k$, represents an eigenvector of $\mathbf{Q}$.

Eigenvalues: These are scalar values that are solutions to the characteristic equation $\det(\mathbf{Q} - \lambda\mathbf{I}) = 0$, where $\mathbf{Q}$ is the rate matrix and $\mathbf{I}$ is the identity matrix.

In the case of a Markovian model, the rate matrix $\mathbf{Q}$ typically has four eigenvalues, denoted $\lambda_0, \lambda_1, \lambda_2, \lambda_3$. Each column of right eigenvectors $\mathbf{v}_k$ corresponds to an eigenvalue. Therefore we end up with 4 columns of eigenvectors.

An eigenvector $\mathbf{v}_k$ of $\mathbf{Q}$ satisfies the equation:

$$\mathbf{Q}\mathbf{v}_k = \lambda_k\mathbf{v}_k, \quad (2.7)$$

where λ_k is the corresponding eigenvalue associated with $\mathbf{v}_k$. Eigenvectors help us to understand the dynamics of the substitution process. Right eigenvectors are crucial in phylogenetics as they describe the directions of evolutionary change under the model represented by $\mathbf{Q}$. Each right eigenvector $\mathbf{v}_k$ indicates a distinct direction of change in the evolutionary process, while the associated eigenvalue λ_k quantifies the rate of change along that direction.

The matrix $\mathbf{U}^{-1}$ contains the left eigenvectors $\mathbf{h}_k$, which satisfy:

$$\mathbf{h}_k^T \mathbf{Q} = \lambda_k \mathbf{h}_k^T, \quad (2.8)$$

where $\mathbf{h}_k^T$ denotes the transpose of the left eigenvector $\mathbf{h}_k$, which is a row vector. $\mathbf{U}^{-1}$ is defined as:

$$\mathbf{U}^{-1} = [\mathbf{h}_1^T, \mathbf{h}_2^T, \dots, \mathbf{h}_k^T], \quad (2.9)$$

with each $\mathbf{h}_k^T$ being a row vector corresponding to the left eigenvector of $\mathbf{Q}$ associated with the eigenvalue λ_k .

Left eigenvectors are associated with the backward process defined by $\mathbf{Q}$ and are used in the context of the transition probability matrix to form outer products with right eigenvectors $\mathbf{v}_k$.

Eigenvectors of a matrix $\mathbf{Q}$ are generally orthogonal if the matrix is symmetric or if $\mathbf{Q}$ is reversible. For symmetric matrices, eigenvectors corresponding to distinct eigenvalues are orthogonal to each other in the standard Euclidean sense. This means that their dot product is zero:

$$\mathbf{v}_i^T \mathbf{v}_j = 0 \text{ for } i \neq j. \quad (2.10)$$

For reversible matrices, eigenvectors are orthogonal with respect to the stationary distribution $\boldsymbol{\pi}$ (Weber, 2017). This orthogonality arises due to the detailed balance condition inherent in reversible processes.

For eigenvectors $\mathbf{v}_i$ and $\mathbf{v}_j$, orthogonality with respect to $\boldsymbol{\pi}$ is defined as:

$$\mathbf{v}_i^T \text{Diag}(\boldsymbol{\pi}) \mathbf{v}_j = 0 \text{ for } i \neq j. \quad (2.11)$$

This means that the weighted inner product (where weights are the stationary distribution probabilities) of $\mathbf{v}_i$ and $\mathbf{v}_j$ is zero if i and j are distinct eigenvectors.

The detailed balance condition of Equation 2.5 can be written in matrix form as:

$$\text{Diag}(\boldsymbol{\pi}) \mathbf{Q} = \mathbf{Q}^T \text{Diag}(\boldsymbol{\pi}). \quad (2.12)$$

where $\text{Diag}(\boldsymbol{\pi})$ is a diagonal matrix with $\boldsymbol{\pi}$ (the stationary distribution) as its diagonal entries. For example, if $\boldsymbol{\pi} = [\pi_A, \pi_C, \pi_G, \pi_T]$, then $\text{Diag}(\boldsymbol{\pi})$ would look like:

$$\text{Diag}(\boldsymbol{\pi}) = \begin{bmatrix} \pi_A & 0 & 0 & 0 \\ 0 & \pi_C & 0 & 0 \\ 0 & 0 & \pi_G & 0 \\ 0 & 0 & 0 & \pi_T \end{bmatrix}.$$

The values $\pi_A, \pi_C, \pi_G, \pi_T$ are actual numbers representing the equilibrium frequencies of nucleotides A, C, G, and T, respectively. In a valid probability distribution, these values satisfy $\pi_A + \pi_C + \pi_G + \pi_T = 1$.

To elaborate on the relationship between left and right eigenvectors of $\mathbf{Q}$, we use the stationary distribution $\boldsymbol{\pi}$:

$$\mathbf{h}_k = \text{Diag}(\boldsymbol{\pi}) \mathbf{v}_k. \quad (2.13)$$

This equation establishes how the left eigenvectors $\mathbf{h}_k$ are related to the right eigenvectors $\mathbf{v}_k$ through the stationary distribution $\boldsymbol{\pi}$. Specifically, it shows that the matrix $\text{Diag}(\boldsymbol{\pi})$ scales each component of $\mathbf{v}_k$ by the corresponding equilibrium frequency π_i .

The condition of 2.12 implies that $\mathbf{Q}$ is similar to the symmetric matrix $\mathbf{S}$:

$$\mathbf{S} = \text{Diag}(\sqrt{\boldsymbol{\pi}}) \mathbf{Q} \text{Diag}(\sqrt{\boldsymbol{\pi}})^{-1}. \quad (2.14)$$

Since $\mathbf{S}$ is symmetric, the eigenvectors of $\mathbf{S}$ are orthogonal in the standard Euclidian sense.

The role of orthogonality is that it simplifies spectral decomposition for the reason, that when eigenvectors are orthogonal, the matrix $\mathbf{U}$ formed from these eigenvectors is orthogonal. For an orthogonal matrix $\mathbf{U}$, the inverse is simply the transpose:

$$\mathbf{U}^{-1} = \mathbf{U}^T \quad \text{and} \quad \mathbf{U}^T \mathbf{U} = \mathbf{I}. \quad (2.15)$$

This orthogonality ensures that Equation 2.6 can be written as:

$$\mathbf{Q} = \mathbf{U} \text{diag}[\lambda_0, \lambda_1, \lambda_2, \lambda_3] \mathbf{U}^T, \quad (2.16)$$

This simplifies computations because finding the inverse of an orthogonal matrix (the transpose) is computationally straightforward. Additionally, the diagonalization of $\mathbf{Q}$ reduces the problem of $\mathbf{P}(t)$ in Equation 2.4, which involves computing the powers of $\mathbf{Q}$, to computing the exponential of eigenvalues.

From Equation (2.16), we can express $\mathbf{P}(t)$ using the spectral decomposition of $\mathbf{Q}$:

$$\mathbf{P}(t) = e^{\mathbf{Q}t} = \mathbf{U} e^{\text{diag}[\lambda_0 t, \lambda_1 t, \lambda_2 t, \lambda_3 t]} \mathbf{U}^T, \quad (2.17)$$

where:

$$e^{\text{diag}[\lambda_0 t, \lambda_1 t, \lambda_2 t, \lambda_3 t]} = \text{diag}[e^{\lambda_0 t}, e^{\lambda_1 t}, e^{\lambda_2 t}, e^{\lambda_3 t}], \quad (2.18)$$

and where $\text{diag}[e^{\lambda_0 t}, e^{\lambda_1 t}, e^{\lambda_2 t}, e^{\lambda_3 t}]$ is the diagonal matrix of exponentiated eigenvalues.

We insert Equation 2.18 into Equation 2.17

$$\mathbf{P}(t) = \mathbf{U} \text{diag}[e^{\lambda_0 t}, e^{\lambda_1 t}, e^{\lambda_2 t}, e^{\lambda_3 t}] \mathbf{U}^T. \quad (2.19)$$

To provide a more detailed understanding, we represent $\mathbf{P}(t)$ as 4 sums using the outer products of eigenvectors:

$$\mathbf{P}(t) = \sum_{k=0}^3 \mathbf{v}_k e^{\lambda_k t} \mathbf{h}_k^T. \quad (2.20)$$

The constraints on $\mathbf{Q}$ imply that its eigenvalues satisfy $\lambda_0 = 0 > \lambda_1 \geq \lambda_2 \geq \lambda_3$ (Karlin, 2014).

The row-sum condition, $\sum_{j=1}^N q_{ij} = 0$, ensures that one of the eigenvalues of the matrix $\mathbf{Q}$ is zero. This λ_0 corresponds to the first left eigenvector $\mathbf{h}_0^T$, which is equal to the stationary distribution vector $\boldsymbol{\pi}^T$. The stationary distribution $\boldsymbol{\pi}$ satisfies $\boldsymbol{\pi} \mathbf{Q} = \mathbf{0}$, meaning it represents the equilibrium distribution of the states.

Since $\lambda_0 = 0$, the term involving $e^{\lambda_0 t}$ simplifies to the scalar value 1 in Equation ??). The right eigenvector $\mathbf{v}_0$ is represented by $\mathbf{1}$, a column vector of ones. The matrix $\mathbf{1} \boldsymbol{\pi}^T$ has each row equal to the stationary distribution vector $\boldsymbol{\pi}^T$, representing the steady-state probabilities.

Thus, the spectral decomposition of $\mathbf{P}(t)$ is given by:

$$\mathbf{P}(t) = \mathbf{1} \boldsymbol{\pi}^T + \sum_{k=1}^3 \mathbf{v}_k \mathbf{h}_k^T e^{\lambda_k t}. \quad (2.21)$$

Each term $\mathbf{v}_k \mathbf{h}_k^T e^{\lambda_k t}$ represents a component that decays over time based on $e^{\lambda_k t}$, illustrating dynamic system evolution.

$$\mathbf{P}(t) = \mathbf{1}\boldsymbol{\pi}^T + \mathbf{v}_1 \mathbf{h}_1^T e^{\lambda_1 t} + \mathbf{v}_2 \mathbf{h}_2^T e^{\lambda_2 t} + \mathbf{v}_3 \mathbf{h}_3^T e^{\lambda_3 t} \quad (2.22)$$

This matrix $\mathbf{1}\boldsymbol{\pi}^T$ describes how the system tends towards a uniform distribution across all states, weighted by the probabilities in $\boldsymbol{\pi}$.

This representation of the spectral decomposition reveals that the transition probability matrix $\mathbf{P}(t)$ consists of a stationary component $\mathbf{1}\boldsymbol{\pi}^T$ and a sum of exponentially decaying terms involving the eigenvalues and eigenvectors. Each term in the sum represents the contribution of one eigenvalue λ_k to the overall evolution of the system. The contributions of each eigenvalue are multiplied by their corresponding right eigenvector and the transpose of the left eigenvector.

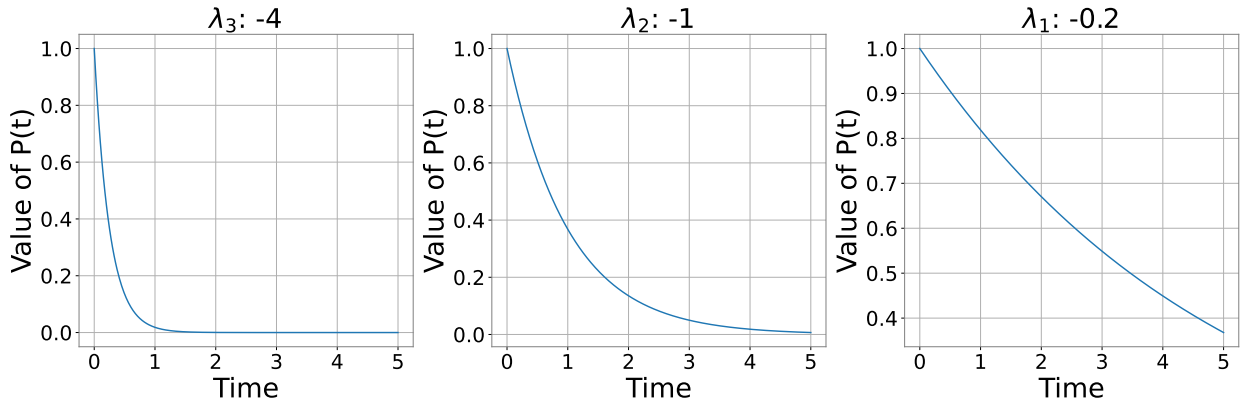


Figure 2.1: Above is the exponential behavior associated with various Eigenvalues λ_k depicted.

Figure 2.1 demonstrates that smaller eigenvalues lead to a faster decrease towards zero, indicating faster evolutionary change. In phylogenetic analysis, selecting the largest non-zero eigenvalue (like λ_1) along with its corresponding eigenvectors $\mathbf{v}_1$ and $\mathbf{h}_1$ is crucial for accurate model representation.

Multiplicity: Algebraic multiplicity is the number of times an eigenvalue appears as a root of the characteristic polynomial. Geometric multiplicity is the number of linearly independent eigenvectors associated with an eigenvalue (Tataru and Hobolth, 2011). The number of linearly independent eigenvectors can be less than or equal to its algebraic multiplicity (Knill, 2011). In the context of Equation 2.22, each eigenvalue λ_k has both an algebraic multiplicity and a geometric multiplicity of 1.

This characteristic is fundamental in eigenanalysis as it indicates distinct evolutionary dynamics represented by unique directions in the vector space.

Jukes-Cantor Model Model Overview

The Jukes-Cantor model (JC) is the simplest model for nucleotide substitution where all substitutions between nucleotides (A, C, G, T) occur at the same rate, denoted as μ . It assumes symmetry in substitution rates, i.e., the rate from nucleotide i to nucleotide j is the same as from j to i . The JC model is time-reversible.

Here is the Jukes-Cantor rate matrix $\mathbf{Q}$ (Jukes et al., 1969):

$$\mathbf{Q} = \begin{pmatrix} -3\mu & \mu & \mu & \mu \\ \mu & -3\mu & \mu & \mu \\ \mu & \mu & -3\mu & \mu \\ \mu & \mu & \mu & -3\mu \end{pmatrix}, \quad (2.23)$$

where μ is the average substitution rate per site.

To find the eigenvalues of $\mathbf{Q}$, we solve the characteristic equation:

$$\det(\mathbf{Q} - \lambda \mathbf{I}) = 0. \quad (2.24)$$

This gives us a polynomial whose roots are the eigenvalues of $\mathbf{Q}$.

Calculating the determinant gives:

$$\det(\mathbf{Q} - \lambda \mathbf{I}) = \begin{vmatrix} -3\mu - \lambda & \mu & \mu & \mu \\ \mu & -3\mu - \lambda & \mu & \mu \\ \mu & \mu & -3\mu - \lambda & \mu \\ \mu & \mu & \mu & -3\mu - \lambda \end{vmatrix}.$$

Expanding this determinant yields:

$$\lambda(\lambda + 4\mu)^3 = 0.$$

Thus, the eigenvalues are $\lambda_1 = -4\mu$ and $\lambda_2 = 0$, with the following algebraic multiplicities:

- $\lambda_1 = -4\mu$ (multiplicity of 3)
- $\lambda_2 = 0$ (multiplicity of 1)

Let's clarify: If $\mu = \frac{1}{3}$, then $\lambda_1 = -\frac{4}{3}$.

In the case of the Jukes-Cantor matrix, the eigenvalue λ_1 has a geometric multiplicity of 3, meaning there are three linearly independent eigenvectors associated with it.

Eigenvectors corresponding to $\lambda_1 = -4\mu$:

$$\mathbf{v}_1 = \begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{v}_2 = \begin{bmatrix} -1 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{v}_3 = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \quad (2.25)$$

These eigenvectors are orthogonal to $\mathbf{v}_0$.

The one eigenvector corresponding to $\lambda_2 = 0$ is:

$$\mathbf{v}_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \Rightarrow \boldsymbol{\pi} = \frac{1}{4} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{bmatrix}. \quad (2.26)$$

The eigenvector $\mathbf{v}_0$ associated with $\lambda = 0$ in the Jukes-Cantor model initially assigns equal probabilities (1) to each nucleotide state (A, C, G, T). Normalization of $\mathbf{v}_0$ adjusts these probabilities such that they sum to 1. This normalization is crucial because it ensures that $\mathbf{v}_0$ represents a valid probability distribution. In biological terms, the normalized $\mathbf{v}_0$, denoted as $\boldsymbol{\pi}$, signifies the stationary distribution of nucleotide frequencies under the model. In the context of the Jukes-Cantor model, the eigenvector $\mathbf{v}_0$ corresponding to the eigenvalue $\lambda = 0$ represents the stationary distribution $\boldsymbol{\pi}$ of the Markov process after normalization.

Due to the symmetry of the Jukes-Cantor rate matrix $\mathbf{Q}$, the left and right eigenvectors associated with the same eigenvalue λ_1 coincide.

Since $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ are orthogonal to $\mathbf{v}_0$, the transition probability matrix $\mathbf{P}_{JC}(t)$ over time t can be written as:

$$\mathbf{P}_{JC}(t) = \mathbf{v}_0 \mathbf{h}_0^\top e^{\lambda_2 t} + e^{\lambda_1 t} (\mathbf{v}_1 \mathbf{h}_1^\top + \mathbf{v}_2 \mathbf{h}_2^\top + \mathbf{v}_3 \mathbf{h}_3^\top). \quad (2.27)$$

Since $\lambda_2 = 0$, $e^{\lambda_2 t} = e^{0 \cdot t} = 1$, and $\mathbf{v}_k = \mathbf{h}_k$, so:

$$\mathbf{P}_{JC}(t) = \mathbf{v}_0 \mathbf{v}_0^\top + e^{-4\mu t} (\mathbf{v}_1 \mathbf{v}_1^\top + \mathbf{v}_2 \mathbf{v}_2^\top + \mathbf{v}_3 \mathbf{v}_3^\top). \quad (2.28)$$

Substitute the outer products and set $\mu = \frac{1}{3}$. To ensure that the transition matrix $\mathbf{P}_{JC}(t)$ is a valid probability matrix (with rows summing to 1), normalization is required.

$$\mathbf{P}_{JC}(t) = \frac{1}{4} \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} + e^{-\frac{4}{3}t} \begin{bmatrix} 3 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ -1 & -1 & -1 & 3 \end{bmatrix} \right) \quad (2.29)$$

This simplifies to:

$$\mathbf{P}_{JC}(t) = \begin{bmatrix} \frac{1}{4} + \frac{3}{4}e^{-\frac{4}{3}t} & \frac{1}{4} - \frac{1}{4}e^{-\frac{4}{3}t} & \frac{1}{4} - \frac{1}{4}e^{-\frac{4}{3}t} & \frac{1}{4} - \frac{1}{4}e^{-\frac{4}{3}t} \\ \frac{1}{4} - \frac{1}{4}e^{-\frac{4}{3}t} & \frac{1}{4} + \frac{3}{4}e^{-\frac{4}{3}t} & \frac{1}{4} - \frac{1}{4}e^{-\frac{4}{3}t} & \frac{1}{4} - \frac{1}{4}e^{-\frac{4}{3}t} \\ \frac{1}{4} - \frac{1}{4}e^{-\frac{4}{3}t} & \frac{1}{4} - \frac{1}{4}e^{-\frac{4}{3}t} & \frac{1}{4} + \frac{3}{4}e^{-\frac{4}{3}t} & \frac{1}{4} - \frac{1}{4}e^{-\frac{4}{3}t} \\ \frac{1}{4} - \frac{1}{4}e^{-\frac{4}{3}t} & \frac{1}{4} - \frac{1}{4}e^{-\frac{4}{3}t} & \frac{1}{4} - \frac{1}{4}e^{-\frac{4}{3}t} & \frac{1}{4} + \frac{3}{4}e^{-\frac{4}{3}t} \end{bmatrix} \quad (2.30)$$

For expressing transition probabilities in the simplified form, we have:

$$\mathbf{P}_{ij}(t) = \frac{1}{4} - \frac{1}{4}e^{-\frac{4}{3}t} \quad \text{for } i \neq j, \quad (2.31)$$

$$\mathbf{P}_{ii}(t) = \frac{1}{4} + \frac{3}{4}e^{-\frac{4}{3}t} \quad \text{for } i = j. \quad (2.32)$$

This demonstrates how the probability formula is derived using spectral decomposition with the specified eigenvalues.

GTR (General Time Reversible) Model Overview

The General Time Reversible (GTR) (Tavare, 1986) model is a widely used evolutionary model in molecular phylogenetics, representing the most general neutral, independent, finite-sites, time-reversible model possible for nucleotide substitution patterns.

The term 'General' in GTR signifies that the model does not impose specific constraints on the substitution rates among nucleotides. The GTR model includes six parameters for the relative rates of nucleotide substitutions: $\alpha, \beta, \gamma, \delta, \epsilon$, and ζ in the instantaneous rate matrix $\mathbf{Q}$. These parameters specify distinctive rates for all possible nucleotide exchanges (Gavin Huttley Group, 2024).

$$\mathbf{Q} = \begin{pmatrix} * & \pi_C\alpha & \pi_G\beta & \pi_T\gamma \\ \pi_A\alpha & * & \pi_G\delta & \pi_T\epsilon \\ \pi_A\beta & \pi_C\delta & * & \pi_T\zeta \\ \pi_A\gamma & \pi_C\epsilon & \pi_G\zeta & * \end{pmatrix} \quad (2.33)$$

Additionally, it has unequal base frequencies.

The total count of 9 free parameters in the model is often reduced to 8 parameters plus μ , where μ represents the overall number of substitutions per unit time. When time is measured in substitutions ($\mu = 1$), the model simplifies to 8 free parameters.

The formulation of the GTR model allows for the estimation of these substitution rates from observed sequence data using maximum likelihood methods. It provides a more realistic representation of evolutionary processes compared to simpler models, making it suitable for analyzing sequence evolution over longer timescales and across diverse taxa (Nei and Kumar, 2000).

3: A Test for Saturation

From a provided multiple-sequence alignment comprising n independent sites, a phylogenetic tree can be inferred, as depicted in Figure 3.1, where a branch with length t will be analyzed. Each column of this alignment represents a site pattern ∂ , comprising one nucleotide for each of the S taxa at the leaves of the tree $\mathbf{T}$ (Manuel et al., 2024). The likelihood of a tree, introduced by (Felsenstein, 1992) can be expressed as the probability of the observed sequence alignment data D given the parameters θ of the evolutionary model, such as substitution rates, branch lengths, base frequencies, and the tree topology (Ferrando, 2022). The pruning algorithm (Felsenstein, 1981) computes the likelihood of a given tree topology by recursively calculating the conditional likelihoods at each internal node of the tree, based on the likelihoods of its child nodes (Posada and Crandall, 2021).

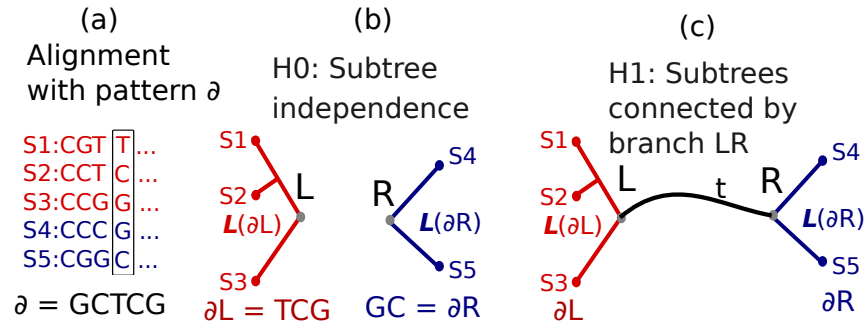


Figure 3.1: (a): Alignment of 5 sequences showing the pattern ∂ GCTCG at site 4. (b): The tree is split into the subtree T_L containing taxa S1, S2, and S3, and the subtree T_R containing taxa S4 and S5. The null hypothesis is that the alignment was generated independently by subtrees T_L and T_R . (c): The alternative hypothesis is that the alignment with pattern ∂ was generated by the subtrees T_L and T_R , which are connected by the branch LR with length t .

The left and right likelihood vectors $\mathcal{L}(\partial L)$ and $\mathcal{L}(\partial R)$ (see Figure 3.1(b)) result from the Pruning algorithm by removing the subpatterns ∂L and ∂R .

The likelihood vector $\mathcal{L}(\partial L)$ is defined as a column vector where each element represents the conditional probability of observing the subpattern ∂L given each nucleotide i at the parent node L :

$$\mathcal{L}(\partial L) := (Pr(\partial L \mid \text{Nucleotide } i \text{ at node } L))_i. \quad (3.1)$$

Illustration adapted from "When All is Lost: Measuring Phylogenetic Information with satute" by Manuel et al., 2024, Figure 3.1.

In Figure 3.1(c), LR represents the branch connecting the parent nodes L and R of the left and right clades, respectively, with a branch length t . The probability of seeing the full pattern ∂ given the branch time t is the probability of observing pattern ∂L and pattern ∂R at the time t , which results in the multiplication of the left partial likelihood vector, the diagonal matrix of the stationary distribution, the substitution probability matrix $P(t)$, and the right partial likelihood vector, which is written in matrix notation (Manuel et al., 2024).

$$Pr(\partial|t) = Pr(\partial L \text{ and } \partial R|t) = \mathcal{L}(\partial L)^T \text{Diag}(\boldsymbol{\pi}) P(t) \mathcal{L}(\partial R) \quad (3.2)$$

Using the eigen decomposition of $P(t)$ from Equation 2.21, we distribute $\text{Diag}(\boldsymbol{\pi})$:

$$\text{Diag}(\boldsymbol{\pi}) \mathbf{P}(t) = \text{Diag}(\boldsymbol{\pi}) \mathbf{1} \boldsymbol{\pi}^T + \sum_{k=1}^3 \text{Diag}(\boldsymbol{\pi}) \mathbf{v}_k \mathbf{h}_k^T e^{\lambda_k t}, \quad (3.3)$$

and simplify the first term $\text{Diag}(\boldsymbol{\pi}) \mathbf{1} \boldsymbol{\pi}^T$ with $\text{Diag}(\boldsymbol{\pi}) \mathbf{1} = \boldsymbol{\pi}$, to:

$$\text{Diag}(\boldsymbol{\pi}) \mathbf{1} \boldsymbol{\pi}^T = \boldsymbol{\pi} \boldsymbol{\pi}^T. \quad (3.4)$$

Recall Equation 2.13 $\mathbf{h}_k = \text{Diag}(\boldsymbol{\pi}) \mathbf{v}_k$, which simplifies second term to:

$$\sum_{k=1}^3 \mathbf{h}_k \mathbf{h}_k^T e^{\lambda_k t}. \quad (3.5)$$

Substituting both terms into Equation 3.2 gives (Manuel et al., 2024):

$$Pr(\partial|t) = \mathcal{L}(\partial L)^T \left(\boldsymbol{\pi} \boldsymbol{\pi}^T + \sum_{k=1}^3 \mathbf{h}_k \mathbf{h}_k^T e^{\lambda_k t} \right) \mathcal{L}(\partial R). \quad (3.6)$$

Now, distributing the likelihood vectors:

$$Pr(\partial|t) = \mathcal{L}(\partial L)^T \boldsymbol{\pi} \boldsymbol{\pi}^T \mathcal{L}(\partial R) + \sum_{k=1}^3 \mathcal{L}(\partial L)^T \mathbf{h}_k \mathbf{h}_k^T \mathcal{L}(\partial R) e^{\lambda_k t}. \quad (3.7)$$

Both terms can now be interpreted as inner products:

The first term:

$$\mathcal{L}(\partial L)^T \boldsymbol{\pi} \boldsymbol{\pi}^T \mathcal{L}(\partial R) = \langle \boldsymbol{\pi}, \mathcal{L}(\partial L) \rangle \langle \boldsymbol{\pi}, \mathcal{L}(\partial R) \rangle, \quad (3.8)$$

where $\langle \mathbf{a}, \mathbf{b} \rangle$ represents the scalar (or dot) product of vectors $\mathbf{a}$ and $\mathbf{b}$, resulting in a scalar.

The second term:

$$\sum_{k=1}^3 \mathcal{L}(\partial L)^T \mathbf{h}_k \mathbf{h}_k^T \mathcal{L}(\partial R) e^{\lambda_k t} = \sum_{k=1}^3 \langle \mathcal{L}(\partial L), \mathbf{h}_k \rangle \langle \mathcal{L}(\partial R), \mathbf{h}_k \rangle e^{\lambda_k t}. \quad (3.9)$$

Combining both terms, we have:

$$Pr(\partial | t) = \langle \boldsymbol{\pi}, \mathcal{L}(\partial L) \rangle \langle \boldsymbol{\pi}, \mathcal{L}(\partial R) \rangle + \sum_{k=1}^3 \langle \mathcal{L}(\partial L), \mathbf{h}_k \rangle \langle \mathcal{L}(\partial R), \mathbf{h}_k \rangle e^{\lambda_k t}. \quad (3.10)$$

The coefficient for pattern ∂ at branch LR is defined as:

$$C_k^\partial(L, R) := (\mathcal{L}(\partial L), \mathbf{h}_k)(\mathcal{L}(\partial R), \mathbf{h}_k). \quad (3.11)$$

This term with the product of two scalar products captures the relationship between the left partial likelihood vector $\mathcal{L}(\partial L)$ and the right partial likelihood vector $\mathcal{L}(\partial R)$, mediated by the left eigenvector $\mathbf{h}_k$.

Under the condition of stationarity, we have $\langle \boldsymbol{\pi}, \mathcal{L}(\partial L) \rangle = Pr(\partial L)$ and $\langle \boldsymbol{\pi}, \mathcal{L}(\partial R) \rangle = Pr(\partial R)$. Therefore, the probability of observing pattern ∂ given the branch time t is:

$$Pr(\partial | t) = Pr(\partial L) Pr(\partial R) + \sum_{k=1}^3 \langle \mathcal{L}(\partial L), \mathbf{h}_k \rangle \langle \mathcal{L}(\partial R), \mathbf{h}_k \rangle e^{\lambda_k t}. \quad (3.12)$$

The likelihood of observing a pattern ∂ given the branch time t comprises two components. The first component, represented by the product $Pr(\partial L)$ and $Pr(\partial R)$, assumes independence between the left and right subtrees. The second component involves the summation over the eigenvalues, which captures the contributions from the remaining eigenvectors of the tree over the branch length t (Manuel et al., 2024).

Saturation: Pattern probability under Independence

When sequences evolve independently and the branch length t approaches infinity, the contribution of the second term on the right-hand side of Equation 3.12 tends towards zero, indicating that the branch becomes saturated. To normalize the Equation, both terms are divided by $Pr(\partial L) \cdot Pr(\partial R)$. This normalization ensures that the sum of the first part equals 1.

$$\frac{Pr(\partial | t)}{Pr(\partial L) Pr(\partial R)} = 1 + \sum_{k=1}^3 \left\langle \frac{\mathcal{L}(\partial L)}{Pr(\partial L)}, \mathbf{h}_k \right\rangle \left\langle \frac{\mathcal{L}(\partial R)}{Pr(\partial R)}, \mathbf{h}_k \right\rangle e^{\lambda_k t} \quad (3.13)$$

We can rewrite the second term with the introduction of the Coefficient C_k^∂ of the eigenvector. The equation

$$\frac{Pr(\partial | t)}{Pr(\partial L) Pr(\partial R)} = 1 + \sum_{k=1}^3 C_k^\partial e^{\lambda_k t} \quad (3.14)$$

evaluates how likely pattern ∂ is at a specific branch and site within a phylogenetic tree, given the time t , normalized by the probabilities of observing the patterns on the left and right subtrees separately. It highlights the degree to which branch LR is connected, reflecting dependence between the left and right subtrees.

If:

$$\sum_{k=1}^3 C_k^\partial e^{\lambda_k t} \rightarrow 0 \quad \text{as } t \rightarrow \infty \quad (3.15)$$

Expectation and Variance Under Saturation

This means as well that the expectation of the sum of the coefficients equals zero:

$$\mathbb{E} \left[\sum_{k=1}^3 C_k^\partial \right] = 0. \quad (3.16)$$

Considering the second term of equation (3.13), we have:

$$\text{Var}[C_k^\partial] = \mathbb{E}[(C_k^\partial)^2] - 0^2 = \mathbb{E} \left[\left\langle \frac{\mathcal{L}(\partial L)}{Pr(\partial L)}, \mathbf{h}_k \right\rangle^2 \right] \mathbb{E} \left[\left\langle \frac{\mathcal{L}(\partial R)}{Pr(\partial R)}, \mathbf{h}_k \right\rangle^2 \right]. \quad (3.17)$$

To compute the overall variance, we sum over the patterns ∂ in the alignment, weighted by their relative frequency $\frac{n_\partial}{n}$:

$$\text{Var}[C_k^\partial] = \sum_{\partial} \frac{n_\partial}{n} \left[\left\langle \frac{\mathcal{L}(\partial L)}{Pr(\partial L)}, \mathbf{h}_k \right\rangle^2 \right] \left[\left\langle \frac{\mathcal{L}(\partial R)}{Pr(\partial R)}, \mathbf{h}_k \right\rangle^2 \right]. \quad (3.18)$$

This simplifies to:

$$\text{Var}[C_k^\partial] = \sigma_{L,k}^2 \sigma_{R,k}^2, \quad (3.19)$$

where $\sigma_{L,k}^2$ is computed as follows:

$$\hat{\sigma}_{L,k}^2 := \sum_{\partial} \frac{n_\partial}{n} \left\langle \frac{\mathcal{L}(\partial L)}{Pr(\partial L)}, \mathbf{h}_k \right\rangle^2. \quad (3.20)$$

Here, $\sigma_{L,k}^2$ and $\sigma_{R,k}^2$ are scalar quantities. In simple terms, they represent the variance of C_k^∂ associated with the left and right patterns, respectively. Each $\sigma_{L,k}^2$ and $\sigma_{R,k}^2$ is computed as a sum of squared projections of likelihood vectors (e.g., $\mathcal{L}(\partial L)$) onto eigenvectors ($\mathbf{h}_k$), averaged over all patterns ∂ in the dataset.

The standard deviation is given by:

$$\text{SD}[C_k^\partial] = \sqrt{\text{Var}[C_k^\partial]}. \quad (3.21)$$

The empirical standard deviation is an estimate of the standard deviation based on the sample data, which in this context are the alignment patterns ∂ observed across different sites in the phylogenetic tree. It is given by:

$$\frac{\hat{\sigma}_{L,k}\hat{\sigma}_{R,k}}{\sqrt{n}}. \quad (3.22)$$

Here, n is the sequence length, i.e., the number of sites in the alignment. The empirical standard deviation accounts for the variability of C_k^∂ due to sampling effects and reflects how much the estimated mean $\hat{E}[C_k^\partial]$ could vary across different samples of patterns ∂ in the alignment. We estimate the mean value $\hat{E}[C_k^\partial]$ as a weighted sum over all possible patterns ∂ in the alignment, where the weights are given by the relative frequencies $\frac{n_\partial}{n}$ of the patterns in the alignment (Manuel et al., 2024):

$$\hat{E}[C_k^\partial] := \sum_{\partial} \frac{n_\partial}{n} C_k^\partial = \sum_{\partial} \frac{n_\partial}{n} \langle \frac{\mathcal{L}(\partial L)}{Pr(\partial L)}, \mathbf{h}_k \rangle \langle \frac{\mathcal{L}(\partial R)}{Pr(\partial R)}, \mathbf{h}_k \rangle. \quad (3.23)$$

The estimated mean $\hat{E}[C_k^\partial]$ represents the average value of C_k^∂ across all possible patterns ∂ . Under certain conditions, such as a large sample size n and weak dependence between patterns, the sampling distribution of $\hat{E}[C_k^\partial]$ can be approximated by a normal distribution ($\hat{E}[C_k^\partial] \sim N(0, \text{Var}[\hat{E}[C_k^\partial]])$), centered at $\mathbb{E}[C_k^\partial] = 0$ (expected under the null hypothesis of saturation) with a variance of $\text{Var}[\hat{E}[C_k^\partial]]$ (Manuel et al., 2024).

Z-test for saturation

The null hypothesis posits that the alignment was generated by independent subtrees, resulting in branch saturation. Conversely, the alternative hypothesis proposes that the alignment was generated by the left and right subtree, connected by the Maximum Likelihood Estimate (MLE) branch LR (Manuel et al., 2024). We test the null hypothesis that the branch is saturated. For this, we use the estimated mean $\hat{E}[C_1^\partial]$ and the empirical standard deviation from equation 3.22.

The null hypothesis is rejected, indicating the branch is informative, if:

$$\hat{E}[C_1^\partial] > z_\alpha \frac{\hat{\sigma}_{L,1}\hat{\sigma}_{R,1}}{\sqrt{n}}. \quad (3.24)$$

If the null hypothesis is not rejected, the branch is considered saturated:

$$\hat{E}[C_1^\partial] < z_\alpha \frac{\hat{\sigma}_{L,1}\hat{\sigma}_{R,1}}{\sqrt{n}}. \quad (3.25)$$

$\hat{E}[C_1^\partial]$ represents the estimated mean of C_1^∂ values across all patterns ∂ in the alignment. To enhance readability, we will henceforth refer to $\hat{E}[C_1^\partial]$ as the 'saturation value' in the Results and Discussion (Chapter 5).

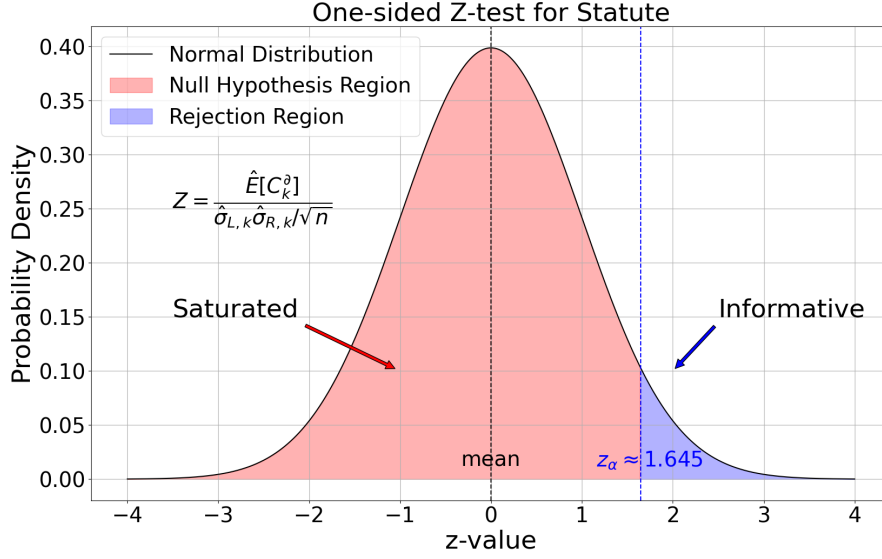


Figure 3.2: One-sided Z-test for Statute. The plot illustrates the normal distribution with shaded regions indicating the Null Hypothesis Region (red) and Rejection Region (blue). The critical value $z_\alpha \approx 1.645$ is marked, demonstrating the threshold for rejecting the null hypothesis of branch saturation in favor of informativeness.

Here, z_α , the α -percentile, denotes the critical value from the standard normal distribution corresponding to the significance level α_S of Statute. By default, $\alpha_S = 0.05$, for which the critical value $z_\alpha \approx 1.645$, corresponding to the 95th percentile (see Figure 3.2).

If the sample coherence $\hat{C}_1$ is negative, it indicates that the log-likelihood of the branch has a local maximum at negative infinity. This suggests that we cannot reject the null hypothesis that the branch is saturated using a two-sided test. However, a negative $\hat{C}_1$ does not necessarily indicate an informative branch, especially for very negative values far from zero.

Therefore, we employ a one-sided test that focuses on rejecting the null hypothesis only in the direction of positive values of $\hat{C}_1$. This approach is statistically more appropriate because a truly informative branch would have a log-likelihood approaching a local maximum at positive infinity, not negative infinity. A one-sided test helps us avoid mistakenly rejecting the null hypothesis due to very negative $\hat{C}_1$ values, which might not indicate an informative branch. Very negative $\hat{C}_1$ values, particularly for saturated branches, may arise from random fluctuations and do not necessarily indicate an informative branch.

If a branch LR is assigned as saturated, it may be attributed to too many mutations due to the branch length t being too large. It could also be due to incorrect alignment. Additionally, for a branch at the root, if this branch connects to saturated clusters on both sides in the alignment, saturation may occur. Another possibility is that the chosen tree topology, branch length, or the instantaneous rate matrix Q is incorrect (Ferrando, 2022).

Satute

Satute (Manuel et al., 2024) is the Python implementation of the saturation test in Equation 3.24.

Given a multiple sequence alignment, we use IQ-TREE (Minh et al., 2019) for phylogenetic reconstruction.

Satute requires an alignment as input and optionally accepts inputs for both the tree and the model. If the model and tree are not specified, IQ-TREE will automatically determine the best-fit model and generate the maximum likelihood (ML) tree. When varying sites are needed (e.g., (Yang, 1994)), the alignment is partitioned into distinct categories based on homogeneity. Each site is assigned to a rate category with the highest posterior probability. Every branch length is scaled according to its corresponding rate.

For each branch, Satute splits the alignment and the tree. IQ-TREE2 computes the statistic C_1^∂ using Equation 3.14 for each site.

Satute uses only the largest nonzero eigenvalue λ_1 with a multiplicity of 1, which determines the slowest vanishing phylogenetic signal. If λ_1 has a higher multiplicity, the corresponding C_k^∂ values will be summed. Equation 3.24 will then include covariances (Manuel et al., 2024). For an external branch $\hat{\sigma}_{L,1}$ is set to 1.

Satute will generate the following output files:

- **satute.log** - A log file containing setup information, such as the alpha value and the chosen model, and listing the number of sites per rate category.
- **satute.csv** - The main CSV file generated by Satute for each rate category.
- **satute.components.csv** - A CSV file containing information about the components of the test statistic for each site and the specific edge in the tree.
- **satute.nex** - A NEXUS file output by Satute.
- **satute.asr.csv** - A CSV file containing ancestral sequences (computed using the empirical Bayesian method) for all nodes of the tree.

The most basic input flag is '-msa' for the alignment, accepting DNA or protein alignments. The '-edge' flag provides the z-test on a single branch. The '-iqtree' flag is always necessary since all input data is provided by IQ-TREE. The '-additional' flag specifies a branch or edge name to focus the analysis on, which is useful when you want to check saturation on a specific branch. The .csv file includes saturation values with and without Bonferroni correction (Napierala, 2012).

Shorter branches in simulated trees might lead to less variation in the aligned sequences. This reduced variation can result in a smaller average $\hat{E}[C_1^\partial]$ value, suggesting the branches are more saturated (less informative) than might be the case in reality. This can lead to the saturation test falsely rejecting the null hypothesis (saturation) more often (Type I error). The Bonferroni correction is calculated as the product of the number of leaves on the left side (L) multiplied by the number of leaves on the right side (R) for each branch.

4: Bootstrap

Description of Standard Bootstrap

(Felsenstein, 1985) introduced the nonparametric bootstrapping by (Efron, 1977) into phylogeny for estimating the phylogenetic support of certain clades or splits in an inferred phylogenetic tree.

In phylogenetic bootstrapping, pseudoalignments are produced by drawing columns from the original alignment with replacement, matching the number of sites in the alignment. The purpose of bootstrapping is to assess the uncertainty in the phylogenetic inference caused by the limited size of the dataset. For each pseudoalignment, a maximum likelihood (ML) tree (Felsenstein, 1973) is constructed, and the splits are collected according to their frequency.

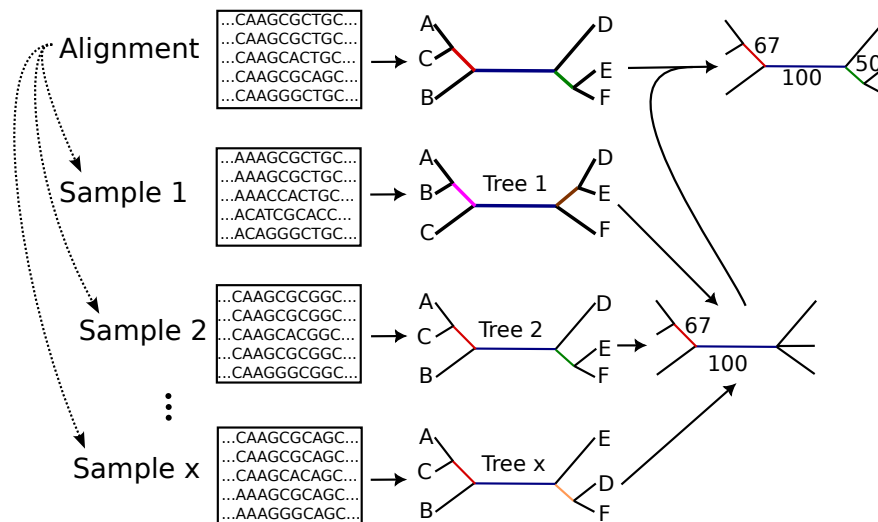


Figure 4.1: Standard Bootstrap (SBS)

These splits, expressed as percentages of their occurrence, are placed on the branches of the reconstructed ML tree inferred from the original alignment and are called bootstrap values.

In this example, with 3 bootstrap trees analyzed (see Figure 4.1), the grouping AC—BDFE appears in 2 out of 3 bootstrap replicates, representing a bootstrap value of 66.67%. This illustrates that in bootstrapping, the emphasis is on the splits or branch patterns observed

Illustration was adapted from Bioinformatics for Phylogeny and Evolutionary Tree Reconstruction, Schmidt, Figure 4.1

across ML trees generated from resampled alignments, rather than the exact topologies (branching order) of the ML trees themselves (Holmes, 2003). Additionally one can reconstruct a consensus tree from the bootstrap tree.

IQ-TREE employs a heuristic search algorithm combining random and deterministic elements for ML tree searches (Vinh and Von Haeseler, 2004; Vos, 2003). The stochastic NNI introduces random perturbations, while subsequent hill-climbing NNI refinements facilitate the discovery of high-scoring trees (Guindon and Gascuel, 2003). This approach facilitates the discovery of higher-scoring local optima (Nguyen et al., 2015).

Ultimately, the best scoring tree identified through these methods represents the final ML tree.

Bootstrap is generally conservative but biased, it tends to underestimate the true probabilities of a clade to be correct (Felsenstein and Kishino, 1993), (Hillis and Bull, 1993).

Good reliable bootstrap support values start at around 80, (Schmidt et al., 2023).

Runtime: The downside of standard phylogenetic bootstrapping is that it requires significantly more computational time. This is because, for each of the x alignment bootstrap samples, a new best ML tree must be recalculated from the resampled alignment. Consequently, the total runtime is the runtime to calculate the original ML tree multiplied by the number of alignment bootstrap samples.

UFBoot

An ultrafast bootstrap approximation method was introduced by (Minh et al., 2013), aimed at improving the accuracy of tree searches in biological research. In UFBoot, approximately 1000 bootstrap alignments are generated from the original alignment, with corresponding bootstrap trees initially initialized as placeholders (Minh et al., 2013). These alignments are created for subsequent tree inference and evaluation.

IQPNNI (Important Quartet Puzzling with NNI) samples local maxima and their neighborhoods in the tree space defined by NNI operations.

UFBoot separately generates approximately 100 starting trees using IQPNNI (Vinh and Von Haeseler, 2004; Minh et al., 2005) as part of the process to compute the ML tree from the original MSA. Only these trees are then evaluated using RELL bootstrapping (see Figure 4.2) if their log-likelihood is larger than l_{min} .

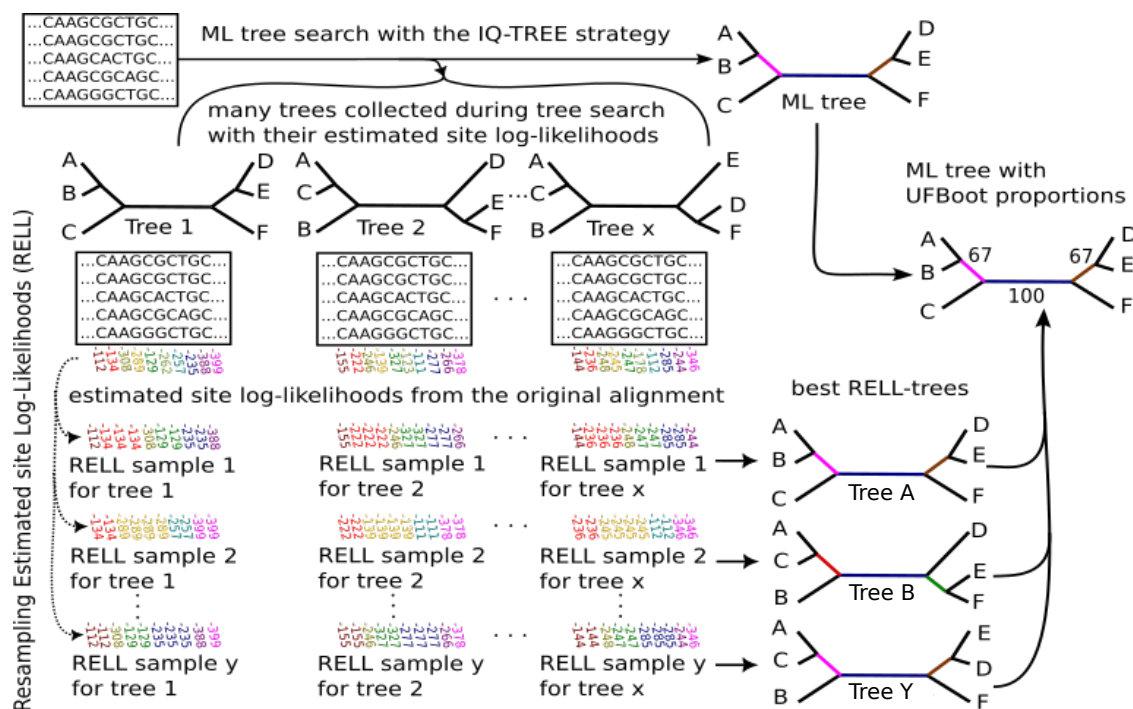


Figure 4.2: UFBoot: Each tree has in each column of its alignment a site-log-likelihood value. See that for sample 1 for every tree the same sites are getting sampled, each RELL sample 1 alignment gets of course different site Log-Likelihoods values though the positions are the same. RELL sample 1 for all trees samples the blue site log-likelihood on position 2 three times, RELL sample 1 for tree 1 gets the site values from the original alignment tree 1, RELL sample 1 for tree 2 gets the site Log-Likelihood values from the original tree 2 and so on.

Shimodaira and Hasegawa (Hasegawa and Kishino, 1994) developed a method to resample

Illustration was adapted from Bioinformatics for Phylogeny and Evolutionary Tree Reconstruction, Schmidt, Figure 4.2.

the estimated site log-likelihoods for each tree. This resampling is typically performed at least 1,000 times (see Figure 4.2). Instead of resampling sequence alignments directly, RELL resamples the site log-likelihoods. This means creating pseudo-replicates by drawing from the log-likelihoods of individual sites. For each bootstrap alignment, RELL generates 1,000 pseudo-replicates by resampling the estimated site log-likelihoods.

It computes the log-likelihood of the candidate tree against each distinct bootstrap alignment independently. For each RELL sample, a log-likelihood is computed by summing up the site log-likelihoods.

Whenever a new tree T whose log-likelihood exceeds l_{min} is found, UFBoot quickly computes the RELL score of T for each bootstrap alignment. If T has a higher RELL score than the current bootstrap tree, UFBoot updates the current bootstrap tree to T for the corresponding bootstrap alignment.

To escape local optima, the IQP step randomly deletes a fraction of the leaves of the tree and re-inserts the leaves using the quartet puzzling method.

To avoid over-optimistic bootstrap support values and therefore false positives, because of choosing only a single optimal bifurcating tree for each bootstrap MSA, UFBboot2 selects randomly a tree as the next best RELL tree, which has a RELL score lower than ϵ_{boot} which is default set to 0.5 (Hoang et al., 2018).

These best RELL trees are mapped onto the ML tree, and branch splits are generated to calculate UFBoot support values for each branch.

UFBoot stops collecting candidate trees when the correlation coefficient c_F of the split occurrence frequencies exceeds 0.99.

Finally, UFBoot computes a consensus tree from the set of bootstrap trees and also maps the split support values onto the ML tree.

UFBoot2 was evaluated for branch support accuracy following the methodology described in (Hillis and Bull, 1993). This assessment involved plotting the proportion of true splits with $x\%$ support against all splits with $y\%$ support, demonstrating nearly unbiased performance in simulations using MSAs sourced from the PANDIT database (Whelan et al., 2006). For example, a branch with a UFBoot value of 80% has a probability of 0.95 of being correct.

Throughout our study, the option '-bb 10000' (Hoang et al., 2018) was employed for efficiency reasons, where the number specifies the number of bootstrap replicates.

Runtime: The significant computational speed improvement of UFBoot arises from its resampling of site patterns from a smaller subset of trees. By utilizing the site log-likelihoods obtained from ML tree inference for the resampling process, UFBoot achieves remarkable efficiency compared to traditional bootstrap methods. This efficiency is a result of not needing to resample full ML trees from the entire dataset, as required in normal bootstrap.

5: Results and Discussion

5.1. Comparison between Saturation and UFBoot Support Values

Introduction

The purpose of this analysis is to comprehensively investigate the relationship between saturation values and UFBoot support values and to draw a conclusion about their correlation.

To establish this correlation, additional tree files with node names were generated for each Satute run, separate from the tree files containing UFBoot values. Each branch string was matched with the corresponding UFBoot values and node names, as well as saturation values extracted from the .csv files.

Furthermore, to visualize this relationship, a 2D scatter plot was created where UFBoot values were represented on the x-axis and saturation values on the y-axis. Each data point on the plot corresponds to a branch in the phylogenetic tree, with its position determined by its UFBoot (x-coordinate) and saturation (y-coordinate) values.

This coefficient is determined as the covariance between saturation values (see Equation 3.23 in Chapter 3) and UFBoot support values divided by the product of the standard deviations of UFBoot support and saturation values.

We assess the relationship between saturation and UFBoot support values by calculating Pearson’s correlation coefficient (Pearson, 1895). This coefficient is determined as the covariance between saturation (see Equation 3.23) and UFBoot support values divided by the product of the standard deviations of UFBoot support and saturation values.

The alpha parameter α_S of Satute for Equation 3.24 was set to 0.05 for all alignments.

For our upcoming correlation examinations involving the subsequent model tree (see Figure 5.1(a)), we simulated 100 alignments for each variation, differing only in branch lengths.

All Satute runs utilized these alignments as inputs, employing the -ufboot 10000 option, which specifies the number of UFBoot replicates (Hoang et al., 2017) for inferring ML trees.

64-leaf template tree: Our selected 64-leaf template tree, depicted in Figure 5.1(a) from Carmen’s thesis (Ferrando, 2022), consists of four branches, each connecting to a 16-taxa clade and a center branch. All branches within the four subtrees have a length of 0.01. The center branch connects Nodes t0 and t1.

For our example tree, the center branch had a length of 0.02, while the four branches connecting to the 16-taxa clades each had a length of 3.0. From this tree, 100 alignments were

simulated using AliSim (Ly-Trong et al., 2022). The Maximum Likelihood trees were inferred using JC+G4 (Jukes et al., 1969). The relative substitution rates according to IQ-TREE2 inferring for alignment Nr.: 1 are for category 1 = 0.1413, for category 2 = 0.4833, for category 3 = 1.004, and for category 4 = 2.372. The original branch lengths are multiplied by the substitution rate from their respective categories of the inferred tree.

All 100 alignments were used for subsequent evaluations (see Tables 5.1 to 5.5). Alignment number 1 was specifically utilized to display the UFBoot values on the inferred ML tree (see Figure 5.1(b)), as well as to display the correlation coefficient (CC) (see Figures 5.2(a) and 5.2(b)).

In Figure 5.1(b), the inferred ML tree from alignment Nr. 1 is depicted. Saturated branches are highlighted in red and labeled, as well as the neighboring branches adjacent to them, which are also labeled with their UFBoot values. The four branches connecting to clades have very high UFBoot values, with three branches labeled with values of 100 and one branch labeled with a value of 92. The saturation of the center branch is due to all branches diverging from the root connecting to saturated clusters in the alignment. Therefore information linking the four other saturated branches together is lost.

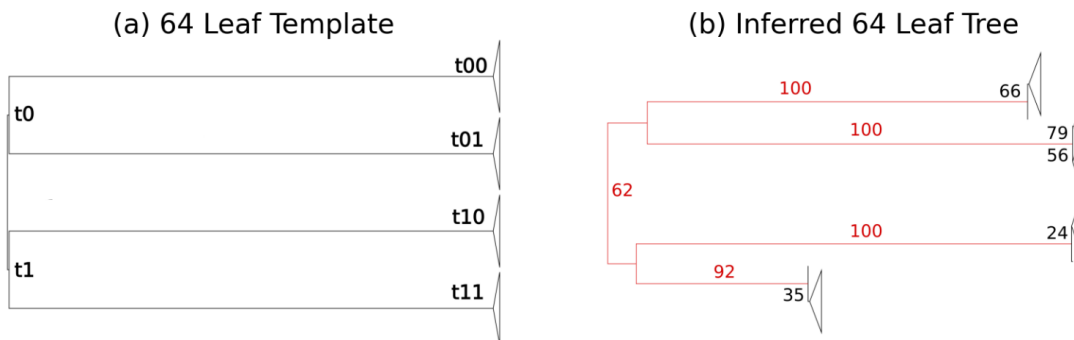


Figure 5.1: (a): A 64-leaf template tree comprising four clades is constructed, where each of the 16 taxa within every clade has a branch length of 0.01. The clades are named according to their parent inner nodes: clade t00, t01, t10, and t11, ensuring clarity and correspondence in the tree structure. For alignment 1: The center branch has a length of 0.2. Each of the 4 branches (t0-t00, t0-t01, t1-t10, and t1, t11) connecting to a clade, as well as to the center branch, has a length of 3.0. The 2 internal nodes of the center branches are called t0, t1 and the 4 internal nodes at the clades are called t00, t01, t10, and t11.

(b): Inferred tree with model Jukes Cantor+G4 (Jukes et al., 1969) from alignment Nr.: 1. The UFBoot values of the saturated branches are shown in red, and of the branches next to them are shown in black.

In Figure 5.2(a) the correlation coefficient of -0.04 for the first sample of the 64-leaf template is depicted. In the scatter plot, each data point represents a branch. It can be seen that there is no correlation. There are individual informative branches with low UFBoot support values and high saturation values and four saturated branches with high UFBoot values. The 5th saturated branch has a medium-high UFBoot value.

The p -values in Figure 5.2(b) are suboptimal for drawing the regression line because all informative branches have p -values near zero. However, larger p -values facilitate easier recognition of saturated branches at a glance.

To standardize the saturation value, it is divided by the square root of the variance adjusted by the number of sites. This standardized value indicates how many standard deviations the observed saturation value deviates from the mean, assuming the branch is saturated.

Once standardized, this value is compared against a standard normal distribution. The p -value is then calculated based on the probability of observing a value as extreme or more extreme than the standardized test statistic, under the assumption of the null hypothesis that the branch is not saturated (Evans et al., 2011).

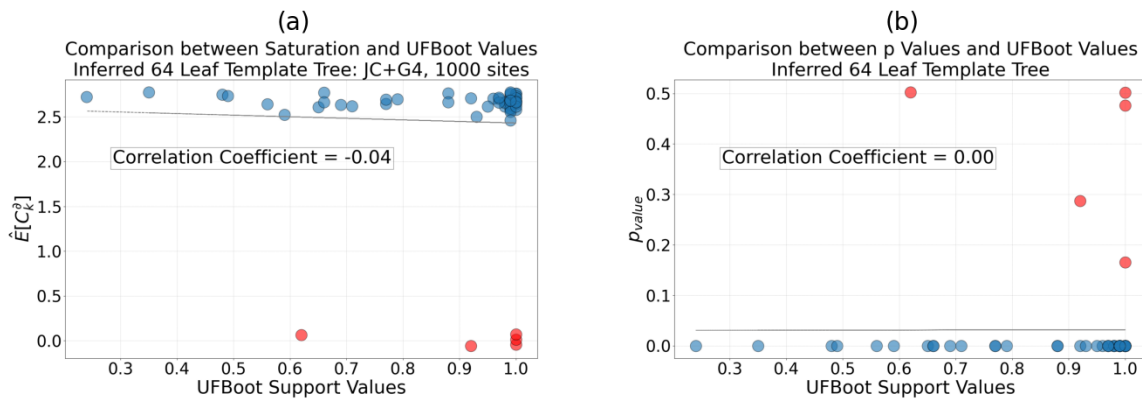


Figure 5.2: (a) Jittering in the y direction is introduced in the graph to distinguish the saturated branches which are plotted as red dots. The correlation coefficient for the first alignment between the saturation values and the UFBoot support values is here shown for the 4th and highest category rate. (b) Comparison of p -values and UFBoot values for the same inferred template tree using the 4th category rate. This depiction highlights the five saturated branches, marked with red dots.

In Table 5.1, the average correlation coefficients (CC) per rate category (RC) and the counts of saturated branches from 100 alignments are depicted. Across rate categories 2 to 4, the central branch in the inferred ML trees is consistently saturated. There was no correlation.

If we exclude the four main branches identified as saturated with low saturation values from the correlation coefficient calculations, the mean CC across the 100 inferred trees shows only an insignificant increase. Specifically, for rate category 4, the mean CC from over 100 alignments is 0.09 with a SD of 0.11. For rate category 3, the mean is 0.13 with an SD of 0.10, and for rate category 2, the mean is 0.14 with an SD of 0.10.

Excluding the values of all five saturated branches, the average correlation coefficients across the 100 inferred trees for the informative data points (minus the 5 branches) decrease slightly further compared to the original correlation coefficients presented in Table 5.1. For rate category 4, the mean correlation coefficient is -0.14 with a SD of 0.16. In rate category 3,

RC	CC	standard deviation (SD)	5 saturated branches (SB)	4 SB	3 SB
4	-0.077	0.054	79	17	4
3	-0.068	0.044	93	7	0
2	-0.065	0.046	97	3	0
1	-0.035	0.000	0	0	0

Table 5.1: Distribution of saturated branches and correlation coefficients by rate category across 100 inferred trees. Abbreviations: rate category (RC), correlation coefficient (CC) – representing the average of 100 alignments per rate category, SD of the correlation coefficient, SB. The columns indicate the amount of inferred trees with corresponding numbers of saturated branches.

the mean correlation coefficient is -0.07 with an SD of 0.20. For rate category 2, the mean correlation coefficient is 0.05 with an SD of 0.22. The values of the informative branches alone show no correlation; the informative data points are too scattered. Additionally, the data points of the saturated branches have no overall effect on the correlation coefficient, either negatively or positively for this model tree.

Next, the branch lengths of the four branches (t0-t00, t0-t01, t1-t10, and t1-t11) were systematically reduced to 0.5, 0.25, 0.1, and 0.01 substitutions per site. The aim was to assess whether shortening these branches from their original lengths, characterized by low saturation values and high UFBoot values, could enhance the correlation coefficients. For each variation in branch length, 100 alignments were simulated using the JC+G4 model (Jukes et al., 1969), and ML trees were inferred.

RC	CC	5 SB	4 SB	3 SB	2 SB	1 SB
4	0.021	89	11	0	0	0
3	0.001	0	3	4	1	3
2	0.046	0	0	0	0	0
1	0.234	0	0	0	0	0

Table 5.2: Mean correlation coefficients for four internal branches with lengths of 0.5, based on 100 alignments. If one or two branches were saturated, the longest one or two of the four longest branches (t0-t00, t0-t01, t1-t10, t1-t11) were considered saturated. If at least three branches were saturated, the center branch was also considered saturated.

RC	CC	5 SB	4 SB	3 SB	2 SB	1 SB
4	0.156	0	1	0	10	10
3	0.170	0	0	0	0	0
2	0.228	0	0	0	0	0
1	0.456	0	0	0	0	0

Table 5.3: Mean correlation coefficients for four internal branches with a length of 0.25, based on 100 alignments. In ML trees featuring a single saturated branch, the center branch was saturated in four instances, and the longest of the four branches, serving as the parent branch to one of the four subtrees, was saturated six times. Furthermore, whenever at least two branches were saturated, the center branch was always among them.

Tables 5.2 to 5.5 present the mean correlation coefficients across the 100 inferred trees for each

branch length setting. The observed trends show a positive shift in correlation coefficients from rate category 4 (highest rate) to rate category 1 (lowest rate) as internal branch lengths decrease.

Table 5.2 (Branch length 0.5): Correlation coefficients exhibit a moderate increase with decreasing branch lengths. Notably, saturation levels and UFBoot values in the center branch play a significant role in this improvement.

Table 5.3 (Branch length 0.25): The peak in correlation coefficients occurs at this branch length, indicating the most substantial positive shift across all variations. Lower UFBoot scores and saturation levels, especially in the center branch, contribute prominently to this peak.

Table 5.4 (Branch length 0.1): Correlation coefficients start to decline slightly at this shorter branch length. UFBoot values and saturation levels show a slight decrease across all branches within the clades.

Table 5.5 (Branch length 0.01): At this minimal branch length, CCs decrease further, and the data points become more scattered. The effects observed in previous tables diminish significantly, likely due to increased variability in UFBoot scores and saturation levels.

Analysis of the data indicates that for branch lengths up to 0.25 substitutions per site, the primary driver of the observed trend in CCs is the center branch. Specifically, the values associated with the center branch show a consistent pattern of decreasing UFBoot scores and saturation levels as the RCs decrease. This contrasts with the other branches, which generally exhibit higher UFBoot and saturation values across the different RCs.

However, as branch lengths decrease beyond this threshold, the benefits diminish, suggesting a limit to the improvement achievable through further reductions in branch length alone.

This emphasizes that the observed rise in CCs does not imply any correlation between UFBoot values and saturation values.

RC	CC	SB
4	0.077	0
3	0.076	0
2	0.077	0
1	0.295	0

Table 5.4: Results from 100 alignments for four internal branches with a length of 0.1.

RC	CC	SB
4	-0.254	0
3	-0.194	0
2	-	0
1	0.128	0

Table 5.5: Results from 100 alignments for four internal branches with a length of 0.01. No sites were assigned to rate category 2.

5.2. Statistical Analysis

Additionally, real alignments from the EvONAPS database (Reden et al., 2024) were used to calculate the correlation coefficient (CC) between the saturation values and the UFBoot values. We analyzed a total of 163 alignments sourced from three datasets: 5 alignments from OrthoMaM (Ranwez et al., 2007), 79 alignments from PANDIT (Whelan et al., 2003), and 79 alignments from TreeBASE (Piel et al., 2000). These datasets were chosen based on specific criteria, including a minimum of 50 sequences and at least 1000 sites per alignment. They have an average of 3321 sites. Additionally, each alignment was required to contain at least one internal branch with a length ranging between 1 and 2 units.

In Figure 5.3, boxplots depict the correlation coefficients per rate category. The statistical analyses were performed using Satute, which utilizes IQ-TREE for ML tree inferring with the best-fit model.

The boxplots reveal an absence of statistically significant correlations between UFBoot scores and saturation levels. This is further supported by the overall mean correlation coefficient of -0.05 (indicating a slight negative trend on average) across all rate categories, with a standard deviation of 0.15.

Table 5.6 complements this analysis by presenting the average CCs across all files within each rate category (RC).

After removing the data points of only the inner saturated branches with UFBoot values of 90 or above (5th column), a modest increase in CC is observed. Despite the marginal rise, these branches represent a significant proportion, approximately 67.26%, with a standard deviation of 6.29%, comprising roughly two-thirds of the inner saturated branches.

To investigate the influence of model selection on branch saturation detection using Satute, we compared the number of saturated branches identified under two models within the same rate category (categories 1, 2, 3, and 4): (1) the General Time Reversible (GTR) model and (2) the model identified as the best fit according to the EvONAPS database. For this analysis, we selected the GTR+F+G4 model for all alignments, where the empirical base frequencies +F are the default if the model has unequal base frequencies. The FreeRate model (Yang, 1995; Soubrier et al., 2012) is recommended for the analysis of large data sets (Schmidt et al., 2023). This option determines the base frequencies (A, C, G, T) for each alignment individually, rather than using pre-defined values. It adapts to the specific base composition of each alignment. This can be helpful when base composition varies significantly across alignments.

We utilized pre-generated trees retrieved from the EvONAPS database as input for the analysis using the -tree option in Satute.

This comparison will reveal whether the GTR model, with its eight free parameters, is generally suitable for detecting saturated branches using Satute compared to the EvONAPS-derived best-fit model.

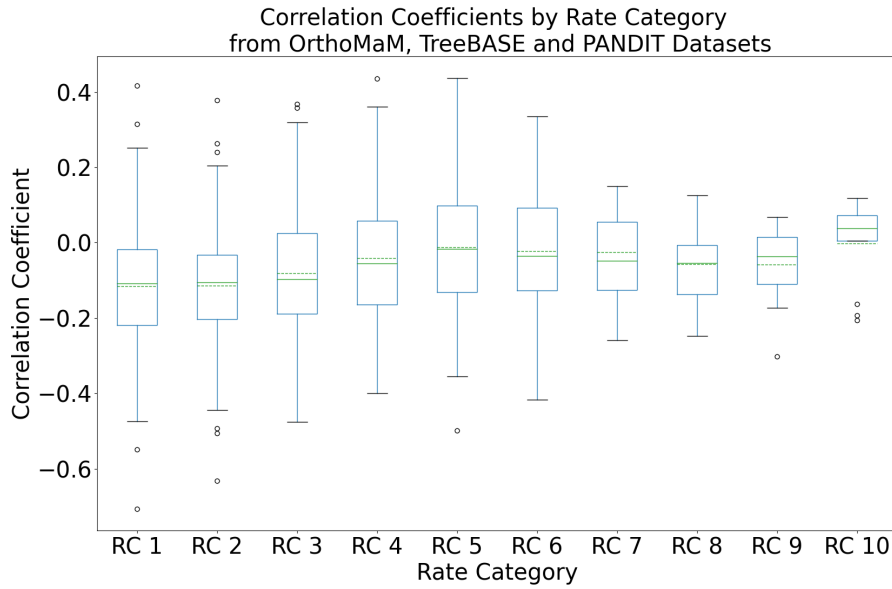


Figure 5.3: Boxplot of correlation coefficients per rate category (RC) from 163 inferred ML trees from real alignments with the best-fit model. The ML trees were inferred using the best-fit model, which led to the presence of 10 rate categories in some alignments (13 out of 163). Green line: Median, green dotted line: Mean. The height of the box indicates the spread of the middle 50% of the data. The upper adjacent value of the whiskers is the largest data point that is within 1.5 times the interquartile range (IQR) above the third quartile (Q3). The lower adjacent value is the smallest data point that is within 1.5 times the IQR below the first quartile (Q1). Circles represent outliers.

RC	Inf T	Avg Inner SB	Avg CC	$CC \neg (\text{Inner SB} \geq 90\%)$	Inner(SB $\geq 90\%$ / SB)
1	163	1.5	-0.116	-0.108	173/243
2	157	0.9	-0.115	-0.109	87/143
3	148	2	-0.082	-0.069	180/298
4	141	6	-0.041	-0.001	498/818
5	116	24	-0.013	0.096	1670/2814
6	71	32	-0.023	0.101	1481/2241
7	41	51	-0.026	0.110	1485/2096
8	24	92	-0.057	0.092	1714/2210
9	15	140	-0.058	0.148	1487/2103
10	13	236	-0.002	0.220	2290/3063

Table 5.6: Summary of inner saturated branches (SB) by Rate Category (RC). The second column, inferred trees (Inf T), represents the number of alignments for which ML trees were inferred at this rate category using the best-fit model. In the third column is the average number of inner saturated branches for one inferred ML tree of this rate category, and in the 4th their average correlation coefficients (CC). In the 5th column, the correlation coefficients for each rate category are listed after excluding data points (branch values) with saturated inner branches having UFBoot values $\geq 90\%$. The last column indicates the number of inner saturated branches labeled with UFBoot values $\geq 90\%$ relative to the absolute count of saturated inner branches.

We calculated the absolute mean difference in the percentage of saturated branches across all alignments of both phylogenetic models. Additionally, we performed a paired t-test to assess the statistical significance of the differences observed.

Null hypothesis: The null hypothesis posits that there is a systematic difference in the mean percentage of saturated branches between the two model settings (GTR and best-fit model). For the significance threshold, we adopt the commonly used standard of 0.05.

T-statistic: The T-statistic is calculated as the ratio of the mean difference in saturated branch percentages between two model settings to the standard error of this difference ($T = \frac{\text{Mean difference}}{\frac{\text{SD}}{\sqrt{n}}}$). It quantifies the size of the difference relative to the variability observed in the data.

P-value: The p-value is derived by comparing the T-statistic to the t-distribution, which provides the probability of obtaining a T-statistic as extreme or more extreme than the observed value under the null hypothesis. A lower p-value indicates against the null hypothesis, suggesting that the observed differences are only due to random chance or that there are no significant differences between both phylogenetic model settings.

To identify outliers, we calculated the z-scores for the percentage of saturated branches across all alignments. A threshold of 3 standard deviations was used to determine outliers, where any z-score greater than 3 was considered an outlier.

PANDIT

In our analysis of the 79 PANDIT alignments (protein and associated nucleotide domains with inferred trees; (Whelan et al., 2003)), we compared the percentage of saturated branches between two model settings. This comparison was conducted for each alignment across all rate categories (RC 1, 2, 3, and 4), examining each rate category individually.

The mean difference (MDIFF) across all rate categories (RC 1-4) in saturated branch percentages was 0.407% with a SD of 2.799. The overall T-statistic produced the lowest p-values. With outliers it yielded a t-value (w) of 6.328 with a p-value (w) of 8.35×10^{-10} . Without outliers, the overall T-statistic yielded a t-value (w/o) of 6.670 with a p-value (w/o) of 1.17×10^{-10} (see Table A.5 in the appendix).

For RC 4, which encompasses 79 alignments, the comparison between the GTR+F+G4 and best-fit+G4 model settings revealed a mean difference (MDIFF) of 1.111%, with an SD of 1.95%. The t-value (w) of 5.017 resulted in a low p-value (w) of 1.41×10^{-6} .

For RC 3, which includes 67 alignments, the mean difference (MDIFF) in saturated branch percentages between the GTR+F+G4 and best-fit+G4 model settings was 0.465%, with a standard deviation (SD) of 0.758%. The T-statistic (t-value (w)) of 4.989 resulted in a low p-value (w) of 1.87×10^{-6} as well. The three outliers in RC 4 and the 2 outliers in RC 3 did not influence the already low p-value in a significant way (see Table A.5).

RC 2 and RC 1 each had only 14 and 1 alignment(s), respectively, whose inferred trees had

saturated branches.

This states that the model selection (GTR+F+G4 versus best-fit+G4) for the same rate category has no impact on the detection of the number of saturated branches.

The individual saturated branches in percent for each alignment can be seen graphically in ascending order in Figures A.6 to A.9 in the appendix. These figures depict bar plots showing the percentage of saturated branches for both the PANDIT GTR+G4 and PANDIT best-fit+G4 models across rate categories 3 and 4.

On average, the 79 alignments from the PANDIT dataset exhibited high percentages of saturated branches, particularly in rate category 4 where the average percentages were 50.5074% (SD = 21.7248%) for the PANDIT GTR+F+G4 model and 50.1382% (SD = 21.5580%) for the PANDIT best-fit+G4 model (see Table A.6 in the appendix).

The standard deviations (SDs) reported here reflect the variability in the percentage of saturated branches within each model setting across the PANDIT dataset. These are different from the SDs associated with p-values, which reflect the variability in the statistical significance of the observed differences.

For RC 3, the average percentages of saturated branches were 4.1945% (SD = 4.4069%) for the PANDIT GTR+F+G4 model and 4.0045% (SD = 4.5031%) for the PANDIT best-fit+G4 model (see Table A.6 in the appendix).

These high average branch saturation values for RC 4 reflect the dataset's focus on the root, superkingdom, and kingdom levels.

TreeBASE

In our analysis of the initial 79 TreeBASE alignments, we obtained pre-generated trees for 61 alignments from the EvONAPS database.

We compared the percentage of saturated branches between the two model settings for each alignment across all rate categories (RC 1, 2, 3, and 4), as well as for each rate category individually.

The overall MDIFF in saturated branches percentage across all rate categories (RC 1, 2, 3, and 4) between both model settings was 0.407% with a SD of 2.799%. The t-value (w) of the overall T-statistic was 2.294, resulting in a p-value (w) of 0.023. The t-value (w/o) was 3.384, with a p-value (w/o) of 0.0008.

For RC 4, encompassing 53 alignments with at least one of the files of the alignment having saturated branches, the MDIFF was 0.798%, with a SD of 2.485%, both higher compared to PANDIT. The T-statistic t-value (w) of 2.524 resulted in a p-value (w) of 0.013. The t-value (w/o) without the outlier *M7224* was 2.874, with a p-value (w/o) of 0.005.

For RC 3, which comprises 32 alignments, the mean difference (MDIFF) in saturated branch

percentages was 0.796%, with a SD of 4.993%. The t-value (w) resulted in 1.250, yielding a p-value (w) of 0.216. Excluding the outlier *M39175* with the highest difference of 38.76 yielded a t-value (w/o) of 1.797, with a noticeably lower p-value (w/o) of 0.077, slightly above the significance threshold.

RC 2 and RC 1 each had higher p-values (w) above the significance threshold, with p-values (w) of 0.098 and 0.216, respectively. However, they only had 24 and 11 alignments with values, respectively, which do not impact the overall p-value negatively for all RCs. The overall p-value remains low because the larger number of alignments in RC 3 and RC 4, which have lower individual p-values, outweigh the higher p-values in RC 1 and RC 2.

Table A.7 in the appendix summarizes these comparisons.

The MDIFF and SD observed in saturated branch percentages were generally larger compared to those in the PANDIT dataset. Without the outlier in RC 3, the p-value (w/o) was just 0.027 above the threshold. We say that the systematic difference in RC 3 between the two model settings for RC 3 is weak. The overall p-value for all rate categories (RCs) and specifically for RC 4 was significantly lower than the significance threshold. Therefore, we can conclude that the choice of model setting within the same rate category has minimal impact.

See Figures A.10 and A.13 for the height of the bars in ascending order for the saturated branches in percent for each TreeBASE alignment.

On average, the 61 alignments from the TreeBASE dataset exhibited lower percentages of saturated branches compared to PANDIT. In rate category 4, the average percentages were 21.7914% (SD = 31.3075%) for the TreeBASE GTR+G4 model and 21.1999% (SD = 30.8615%) for the TreeBASE best-fit+G4 model (see Table A.8 in the appendix).

For rate category 3, the average percentages of saturated branches were 7.7118% (SD = 17.9092%) for the TreeBASE GTR+F+G4 model and 7.0079% (SD = 17.2109%) for the TreeBASE best-fit+G4 model (see Table A.8 in the appendix).

The SDs of the percentage of saturated branches within each TreeBASE model setting were notably higher than those observed in the PANDIT alignments. This higher variability indicates greater fluctuation in branch saturation percentages across the TreeBASE dataset, which encompasses a diverse range of organisms including aquatic, fungal, and arthropod species.

OrthoMaM

The OrthoMaM database includes files labeled with identifiers from the European Neuroblastoma Study Group (ENSG), denoted by names starting with ‘ENSG’, where each identifier corresponds to a specific gene (Shaw and Eden, 1992). Due to the limited number of available files (five in total), statistical comparisons such as T-tests or mean difference analyses

for saturated branches were not conducted.

In comparison, all five ENGS files demonstrated significantly fewer saturated branches. This finding is consistent with expectations based on the OrthoMaM dataset, which predominantly comprises mammalian sequences. For the true tree and the model GTR+F+G4, the alignment ENSG00000119574_ZBTB45 (average gap count 15%) showed a saturated branch between Node62* and Node61*. The sequence of *Sarcophilus harrisii* exhibited a gap count of 78.73%. Both *Phascolarctos cinereus* and *Sarcophilus harrisii* belong to Marsupialia. Despite the high gap count in the latter, the branch between *Sarcophilus harrisii* and Node62* remains informative (saturation value: 0.0647 with a critical value of 0.0542), as does the branch between *Phascolarctos cinereus* (gap count of 48.46%) and Node62* for the true tree topology. However, due to the high gap counts, the OrthoMaM dataset is not suitable for evaluating saturation.

The graphics depicting all 5 alignments, where the height of each bar represents the percentage of saturated branches in ascending order, can be found in Appendix A.10.

The comparison across PANDIT, TreeBASE, and OrthoMaM datasets consistently showed minimal overall differences in the number of saturated branches between the GTR model and the best-fit model within each rate category. Despite variations in standard deviation and p-values, the observed differences were generally not statistically significant, indicating both models perform comparably for branch saturation detection across these diverse datasets.

We showcase two alignments and their inferred ML trees to discuss why saturation values cannot be compared with UFBoot values.

OrthoMaM: *ENSG00000119574_ZBTB45*:

The inferred ML tree data from the Zinc Finger and BTB Domain gene alignment reveal two saturated branches (marked in red) and a broad dispersion of data points within a narrow UFBoot value range (90-100) across the y-axis (saturation values).

The best-fit model selected by IQ-TREE2 is TVM+F+I+G4, which is the transversional Model featuring 7 free parameters and unequal base frequencies. +I allows for a proportion of invariable sites (Schmidt, 2022). The input data comprised 91 taxa with 1566 nucleotide sites.

RC 4 exhibited the highest relative rate of 3.509, followed by RC 3 (1.308), RC 2 (0.550), and RC 1 (0.127).

The correlation coefficient (CC) for the fastest RC (Category 4) was 0.120, while the coefficients for RCs 3, 2, and 1 were -0.013, -0.050, and -0.043, respectively.

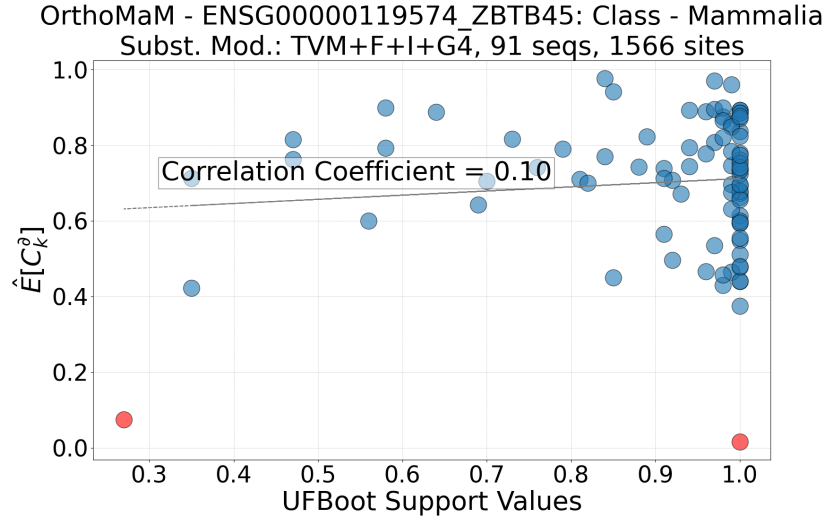


Figure 5.4: OrthoMaM - *ENSG00000119574.ZBTB45*: Class - Mammalia; Substitution model (Subst. Mod.): TVM+F+I+G4. Dataset: 91 sequences (seqs), 1566 nucleotides (nts) sites. The two saturated branches (data points) are marked in red.

In Figure 5.4, the first saturated branch has a UFBoot value of 100, while the second, shorter branch is marked with UFBoot 43 (both branches are marked in red). The branch is identified as saturated due to the sequence *Sarcophilus harrisii*, which exhibits a high gap count of 78.74%.

The pattern of broad dispersion of data points (branches) within a narrow UFBoot value range is not uncommon in alignments. The data points in Figure 5.4 for RC 4 are observed within moderate UFBoot ranges, expanding at higher UFBoot values. Specifically, at a UFBoot value of 100, 34 data points from informative branches are broadly distributed between saturation values of approximately 0.38 and 0.89. Although a single saturated branch with a UFBoot value of 100 is present, its influence on the overall CC is negligible. This is due to the wide array of saturation values observed among the 35 branches sharing the same UFBoot value, rendering correlation impractical. Lower rate categories also exhibit slightly lower CC despite the absence of saturated branches.

TreeBASE: M2942:

This alignment consists of a family of fungi (*Neofusicoccum*, *Dothiorella*, *Botryosphaeria*, *Cercospora*, *Diplodia*), known to be plant pathogens. The best-fit model is TNe+I+R2 (Tamura and Nei, 1993). The dataset comprises 59 sequences with 1009 nucleotide sites and 47 inner branches.

The relative rate for rate category 2 is 6.432, and for RC 1 it is 0.8384. The correlation coefficient for RC 2 is -0.35, and for RC 1 it is -0.21.

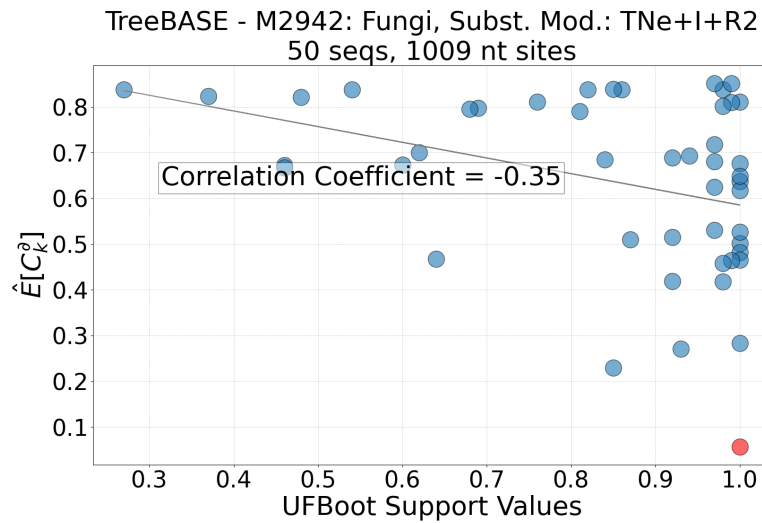


Figure 5.5: TreeBASE - M2942: Fungi, Subst. Mod.: TNe+I+R2. Dataset: 50 seqs, 1009 nts sites. The saturated branch is marked in red.

Figure 5.5 at RC 2 demonstrates that most branches (data points) have higher UFBoot values, while fewer branches are close together at lower UFBoot values. This distribution skews the correlation in one direction. Consequently, the correlation coefficient of -0.35 does not indicate a strong negative correlation but rather suggests no meaningful correlation between saturation values and UFBoot values. The single saturated branch (marked in red) does not substantially influence the overall correlation. This example illustrates that even though some alignments may show higher positive or negative correlation values, they do not necessarily indicate any significant relationship.

It is evident that there is no correlation between UFBoot values and saturation values, although a slight negative skew is present on average in the correlation coefficients. Bootstrapping, which assesses confidence in clade grouping, and Satute, which measures branch reliability, often result in a wide range of data points in the higher UFBoot range, complicating the comparison issue.

Given these observations, we want to shift our focus to understanding the properties of branches exhibiting high UFBoot values but low saturation values. This adjustment enables a more comprehensive exploration of their specific characteristics and implications.

5.3. Bootstrapping Behavior in the Presence of Long Saturated Branches

Introduction

Building upon these observations, we now direct our investigation towards exploring how variations in branch length, particularly focusing on saturated branches with high UFBoot values, influence the neighboring clades and their corresponding confidence values within the inferred ML tree.

To investigate the behavior of such branches with high UFBoot values further, we used the same phylogenetic tree depicted in Figure 5.1(a) and used a simple model with minimal parameters JC+G4.

100 alignments were simulated per different branch lengths of the model tree using AliSim (Ly-Trong et al., 2022).

For each multiple sequence alignment (MSA), we inferred 10,000 ultrafast bootstrap UFBoot trees and computed the ML trees along with their corresponding UFBoot values.

In this study, the term "Order" is introduced to denote the distance of branches from the center branch. The center branch, designated as Order 0, has its immediate neighbors classified as Order 1, branches that are two branch lengths away as Order 2, and so on, with each subsequent neighbor referred to as Order n .

These averaged UFBoot values for each Order were computed by averaging the UFBoot values within each Order separately in each individual tree. This analysis was performed independently for branches on the left side (clade t0) and the right side (clade t1) of the center branch. Then, across all 100 trees simulated for each different branch length, the averaged UFBoot values from the same Order were further averaged to derive the final average UFBoot value for that Order. Notably, this averaging process considered the Orders consistently across all trees, irrespective of the varying tree topologies inferred.

Both clades, while not perfectly symmetric due to the nature of ML tree inference, show averaged UFBoot values with similar results. Therefore, in this chapter, we present the values starting from the center branch and extending to a clade with 32 taxa. The values for the other clade, which start from the opposite side of the center branch, are provided in the appendix for Tables A.1-A.2.

We introduce the term tree variation (TV) (see Table 5.7) to represent the number of different topologies. Each unique TV corresponds to a distinct arrangement of external branches with the same Order.

TV specifically analyzes the distribution and frequency of taxa groups at different distances from the center branch. This includes assessing the occurrence and specific arrangements of pairwise and single external branches within clades of varying sizes. Pairwise Branches: External branches belonging to the same clade and are the same Order away from the center. The number of their external branches will be saved. Pairwise branches of the size n refer to $2n$ taxa (each pair consisting of 2 taxa) share the same parent node. This

configuration captures how taxa group together at various distances from the center branch. Single External Branches: Individual branches that do not form a clade but are classified by their Order. By examining the frequency of each taxa group size and their respective Orders from the center branch, TV provides insights into the structural variability of phylogenetic trees.

It is important to note that TV captures variations in how equidistant external branches assemble into specific clade sizes across different ML trees. While trees with the same TV generally exhibit similar overall structural patterns, precise branch placements, and taxon compositions within clades may differ between trees.

In summary, TV serves as a quantitative metric to explore the diversity of tree topologies resulting from phylogenetic inference, focusing on the distribution and arrangements of equidistant external branches forming specific clade sizes across multiple ML trees (see Figure A.3 with 2 examples).

The SD reflects the extent of variation in average UFBoot values among these ML tree topologies. The SD was calculated from the average UFBoot values per Order across each tree (see Table 5.8).

It must be clear that we have as input a MSA and infer various ML tree topologies from it.

We are interested in how we recognize branches where UFBoot tells us that the groups are very well separated given the MSA and at the same time Satute analyzes these distinct branches as not real or saturated. Given that saturated branches might exhibit high UFBoot values despite potentially unreliable phylogenetic signal, how does the presence of such saturated branches within an inferred ML tree influence the confidence levels (bootstrap support values) of neighboring clades? This question is particularly relevant because neighboring clades share sequence information with the saturated branch.

(a) Center branch t0-t1 with a constant length of 6.0:

For the trees, the central branch (t0-t1) consistently exhibited 6 substitutions per site. All branch lengths refer to the lengths in the true tree model from which alignments were simulated, not the branch lengths inferred in the ML trees. The lengths of the four parent branches (t0-t00, t0-t01, t1-t10, t1-t11) of the subtrees varied incrementally from 0.01 to 0.1, 0.2, 0.4, 0.6, 1.0, 1.5, 2.0, 2.5, and 3.0. For convenience, we define junction branch (JB) as both singular and plural junction branches (JBs) to refer to the four internal branches (t0-t00, t0-t01, t1-t10, t1-t11) that connect the center branch (t0-t1) of the tree to each of the four clades with 16 taxa. The lengths of the JBs always correspond to the branch lengths in the true tree. Similarly, when 'clade' is mentioned without specifying the parent clades (t0 and t1), it refers to one of the four smaller 16-taxa clades: t00, t01, t10, or t11.

Table 5.7: Summary of branches analyzed by Satute for rate category 4, indicating saturation by branch lengths for the four branches: t0-t00, t0-t01, t1-t10, and t1-t11. SB stands for saturated branches. An inferred ML tree with at least one SB implies saturation of the center branch. As the number of SBs increases, additional JB branches also become saturated. With the elongation of the JBs, these branches also became saturated. The number of saturated branches refers to the ML trees, while the branch lengths correspond to the JB lengths of the true tree. EXTB represents external single branches and their average number per tree. TV stands for tree variations.

JBs branch len	0 SB	1 SB	2 SB	3 SB	4 SB	5 SB	EXTB	TV
0.01	18	82	0	0	0	0	3.84	7
0.1	5	95	0	0	0	0	3.68	16
0.2	9	90	1	0	0	0	3.48	39
0.4	0	11	22	54	12	1	3.96	89
0.6	0	0	20	54	20	6	6.06	99
1.0	0	0	6	31	39	24	6.90	85
1.5	0	1	8	23	47	21	7.80	68
2.0	0	1	7	25	32	35	7.72	50
2.5	0	2	5	18	28	47	7.80	38
3.0	0	0	5	13	26	56	7.80	44

At the beginning, all branches have a substitution length of 0.01 per side, aside from the center branch with a length of 6 substitutions per site. For this reason, the ML tree variations are few and very similar, 91 alignments infer the same tree. All trees show the characteristic that branches of Order 1 have the lowest UFBoot values (see Figure 5.6).

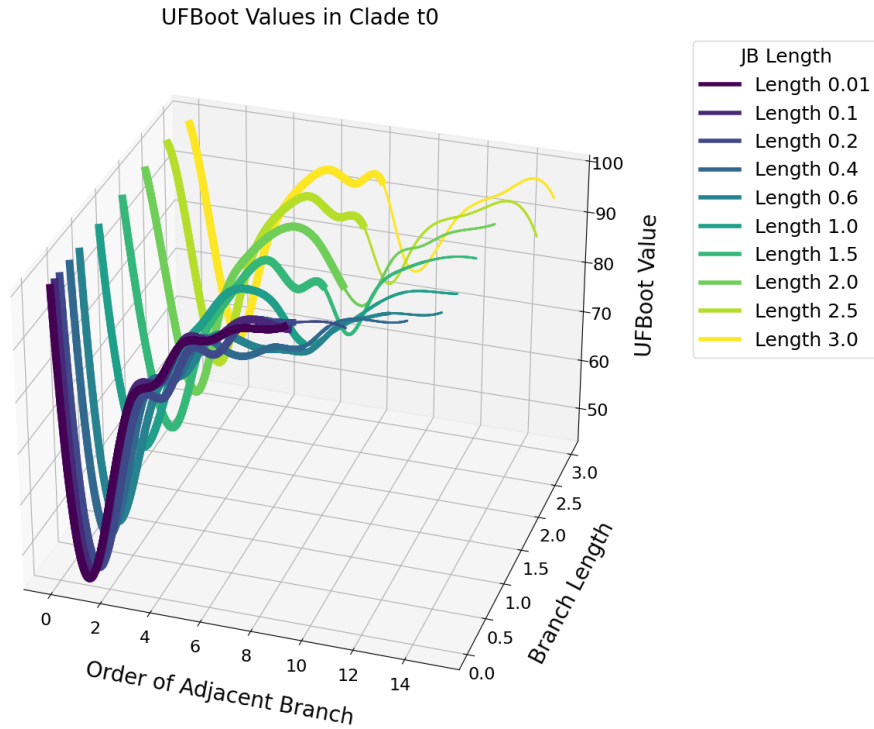


Figure 5.6: The representation of UFBoot values starts from the center branch at Order 0 and extends to directly adjacent branches labeled with their respective UFBoot values at Order 1 and further to the branches of the last Order. Subsequently, it includes branches at Order n , which denote the n th neighbor branch from the center. The Order of the furthest branches away from the center would be $n+1$ with taxa as the last nodes. Because we are not interested in external branches, UFBoot values labeled branches of Order n are our last plotted values. The number of trees that have branches at this Order is represented by the thickness of the colored curve.

The correct true tree topology does not appear as an ML tree. However, in each of the four clades t00, t01, t10, and t11, all taxa remain within their respective clades. The position at which each end of the center branch inserts seems not random. In the true topology, each side of the center branch connects to two 2 JB. In the inferred ML trees, the center branch consistently inserts into one external branch of the 16-taxa clade on each side, typically at the edge next to the first taxon, rather than centrally within the clade. This explains why, in 95 out of 100 inferred ML trees, we encounter 4 single external branches at Order 1, and 2 with a JB lengths of 0.01. The limited variation in tree topologies is evidenced by the fact that in 97 out of the 100 inferred trees, the remaining two 16-taxa clades are correctly rooted and maintain the correct true clade topology. This indicates that clades, that connect to the center branch via an external branch, have a JB correctly inserted into the middle of the subsequent other clade.

If the four JB reach a length of 0.2, and 0.4 the total maximal reached Order gets an increasing jump forward to 11, and respectively 13 (see Figure 5.6 or Table 5.8). This is where we get an increase in the number of tree variations. At a JB's length of 0.2, among the 100 alignments inferred, many still yield the same tree structure. Specifically, 38 alignments consistently produce one ML tree variation, while three groups of five alignments converge to different ML TVs. In the 38 alignments group, we still observe 2 clades per tree that are

correctly rooted and remain in a true clade topology. Have the JBs a length of 0.4, there are just 4 and 3 alignments which infer the same topologies, otherwise, all trees are unique. This is the first time that the root branch correctly connects to each of the four JBs in a few ML trees.

The number of tree variations reaches with the JBs length of 0.6 its maximum with 99. The rising number of external single branches occurs in combination with a rising number of single-paired taxa, this means 2 taxa side by side with the same order in their group. On average we meet 7.8 external branches with the occurrence of 4.09 single-paired taxa per tree. This means that nearly all JBs prefer to insert into an asymmetrical side of a clade. Hence, we observe an increase in the number of single external branches. These branches typically arise twice: first, near the root, and second, when a JB inserts asymmetrically between two clades. This asymmetry often results in single external branches appearing in the immediate vicinity on one side of the JB. It may happen that single external branches appear up to four times on the order 6, 7, 8, 9.

At JBs length of 1.5 the amount of TVs decreases because the JBs reach a length that leads to tree topologies that are closer to the true tree. The center branch connects already correctly to all 4 JBs in 30 inferred trees.

However, in all the other cases, where a JB inserts asymmetrically between 2 clades we observe through the increasing JB length a second time the effect of a stronger decrease of UFBoot values in the neighbor branches of the insertion, which reaches the deepest value on order 9 (see Figure 5.6).

The slight variations of the insertion points of the JBs lead in both UFBoot valleys to the highest standard deviations (see Table 5.8). At a JB length of 2.0, the center branch connects to the JBs correctly already in 52 cases. A decrease in single external branches and single-paired taxa at greater JB length would imply that at a certain JB length, the insertion point of a JB into one of the clades t00, t01, t10, t11 corresponds better to the positions of the true trees. This seems not to be the case. The number of single external branches and single-paired taxa stays high till the JB branch length of 3.0. The insertion position of the JBs into each clade does not improve; the preference remains to insert a JB between the last taxa of the first clade and the first taxa of the second clade.

At a JB length of 2.5, 57 inferred trees have a center branch correctly connected to JB branches. The center connects to JBs with length 3.0 in 64 trees correctly and the number of Order no longer increases.

Table 5.8: n stands for the number of inferred trees that have inner branches at this Order. UFBoot values and SD for Clade t0 are shown, with the UFBoot values for the center branch in Order 0. Clade t0 consists of the two 16-taxa clades t00 and t01 and is located opposite to Clade t1, with the center branch (Order 0) in between.

JB len	0.01			0.1			0.2			0.4		
Order	n	UFBoot	SD	n	UFBoot	SD	n	UFBoot	SD	n	UFBoot	SD
0	100	100.00	(± 0.00)	100	100.00	(± 0.00)	100	100.00	(± 0.00)	100	100.00	(± 0.00)
1	100	45.33	(± 22.00)	100	44.93	(± 18.86)	100	45.23	(± 19.32)	100	49.02	(± 18.51)
2	100	58.79	(± 23.81)	100	61.11	(± 19.94)	100	57.53	(± 21.64)	100	59.58	(± 20.04)
3	100	80.75	(± 11.71)	100	81.98	(± 9.50)	100	78.63	(± 10.04)	100	79.18	(± 10.61)
4	100	84.95	(± 11.16)	100	84.27	(± 9.17)	100	80.24	(± 11.58)	100	80.91	(± 12.31)
5	100	93.05	(± 5.19)	100	93.04	(± 3.86)	97	90.00	(± 5.07)	95	89.22	(± 7.04)
6	99	94.73	(± 4.82)	100	93.69	(± 3.99)	96	89.94	(± 5.48)	91	88.39	(± 7.08)
7	98	97.53	(± 2.11)	98	97.70	(± 1.85)	93	96.09	(± 3.25)	86	88.12	(± 16.68)
8	95	98.57	(± 1.64)	94	98.87	(± 1.22)	82	97.83	(± 4.18)	72	90.53	(± 12.16)
9	1	99.80	(± 0.00)	5	98.22	(± 2.54)	19	98.30	(± 2.42)	46	91.16	(± 11.35)
10							3	99.23	(± 0.65)	33	95.77	(± 5.31)
11							1	98.80	(± 0.00)	23	98.10	(± 2.67)
12										10	98.37	(± 2.37)
13										3	99.33	(± 1.15)

JB len	0.6			1.0			1.5			2.0		
Order	n	UFBoot	SD	n	UFBoot	SD	n	UFBoot	SD	n	UFBoot	SD
0	100	100.00	(± 0.00)	100	100.00	(± 0.00)	100	100.00	(± 0.00)	100	100.00	(± 0.00)
1	100	49.35	(± 20.25)	100	62.17	(± 23.06)	100	66.34	(± 23.21)	100	72.57	(± 20.85)
2	100	58.62	(± 20.23)	100	61.29	(± 17.59)	100	56.75	(± 16.76)	100	57.42	(± 16.01)
3	100	76.24	(± 12.03)	100	76.72	(± 14.60)	100	72.83	(± 13.39)	100	74.68	(± 12.85)
4	100	79.61	(± 11.85)	100	84.27	(± 9.64)	100	82.28	(± 10.62)	100	85.86	(± 9.03)
5	98	86.84	(± 6.83)	100	90.92	(± 6.17)	100	89.29	(± 8.38)	100	90.61	(± 7.07)
6	97	88.35	(± 9.36)	95	92.65	(± 5.70)	100	92.12	(± 7.43)	100	93.35	(± 7.56)
7	93	87.10	(± 15.04)	92	90.30	(± 14.90)	99	87.80	(± 19.04)	98	91.15	(± 15.68)
8	78	87.51	(± 13.78)	48	84.20	(± 16.76)	45	89.26	(± 11.50)	32	83.49	(± 19.41)
9	69	88.22	(± 13.45)	47	86.20	(± 12.88)	42	80.51	(± 19.10)	32	80.54	(± 18.74)
10	56	93.36	(± 8.13)	40	89.99	(± 9.61)	38	88.85	(± 14.65)	28	89.94	(± 9.14)
11	44	95.81	(± 4.71)	34	94.01	(± 5.50)	36	94.18	(± 7.49)	26	93.06	(± 7.9)
12	25	97.52	(± 3.08)	26	96.78	(± 3.73)	24	96.73	(± 4.93)	21	96.10	(± 3.78)
13	17	98.03	(± 2.47)	18	97.68	(± 2.89)	20	98.41	(± 2.84)	18	98.00	(± 2.74)
14	5	99.26	(± 0.70)	7	98.03	(± 2.73)	10	98.90	(± 0.81)	9	99.73	(± 0.28)

JB len	2.5			3.0		
Order	n	UFBoot	SD	n	UFBoot	SD
0	100	99.87	(± 0.51)	100	98.42	(± 4.07)
1	100	76.17	(± 19.52)	100	73.28	(± 24.14)
2	100	57.24	(± 15.21)	100	56.65	(± 17.60)
3	100	72.70	(± 13.89)	100	73.10	(± 13.29)
4	100	85.47	(± 7.47)	100	84.91	(± 8.76)
5	100	91.27	(± 6.08)	100	90.93	(± 6.17)
6	100	93.42	(± 9.56)	99	93.29	(± 9.52)
7	99	90.41	(± 17.73)	91	91.58	(± 16.67)
8	26	89.82	(± 13.20)	27	92.81	(± 10.55)
9	26	78.90	(± 20.25)	21	77.79	(± 22.76)
10	25	85.95	(± 14.75)	17	78.21	(± 22.19)
11	24	92.15	(± 9.05)	16	86.76	(± 14.68)
12	17	94.24	(± 5.57)	12	91.44	(± 7.05)
13	16	96.76	(± 4.55)	12	94.09	(± 4.69)
14	12	98.42	(± 2.09)	8	97.41	(± 2.51)
15	1	92.50	(± 0.00)	1	94.40	(± 0.00)

(b) All four JBs with constant length 3.0:

The center branch length for the true trees increases from 0.02 to 0.2, 0.4, 0.8, 1.2, 2.0, 3.0, 4.0, and 5.0 substitutions per site.

Every center branch in all inferred trees with a JB length of 0.02 is connected on each side with two JBs, but the specific clades these JBs connect appear random. The two original sister clades t00 and t01 share the same center node only in 40 of 100 inferred trees. The information regarding which clades a JB connects to the center branch is not preserved on either side of the center branch.

The same characteristic is observed up to a center branch length of 2.0. However, at this length, 98 out of the 100 inferred trees already show the correct sister clades on either side of the center branch. Already at a center branch length of 0.02, Satute validates this ML center branch as saturated (see Table 5.9) on all inferred trees.

The low UFBoot values were at Order 2 at the branch next to the JBs (see valley at Order 2 for curves in Figure 5.7).

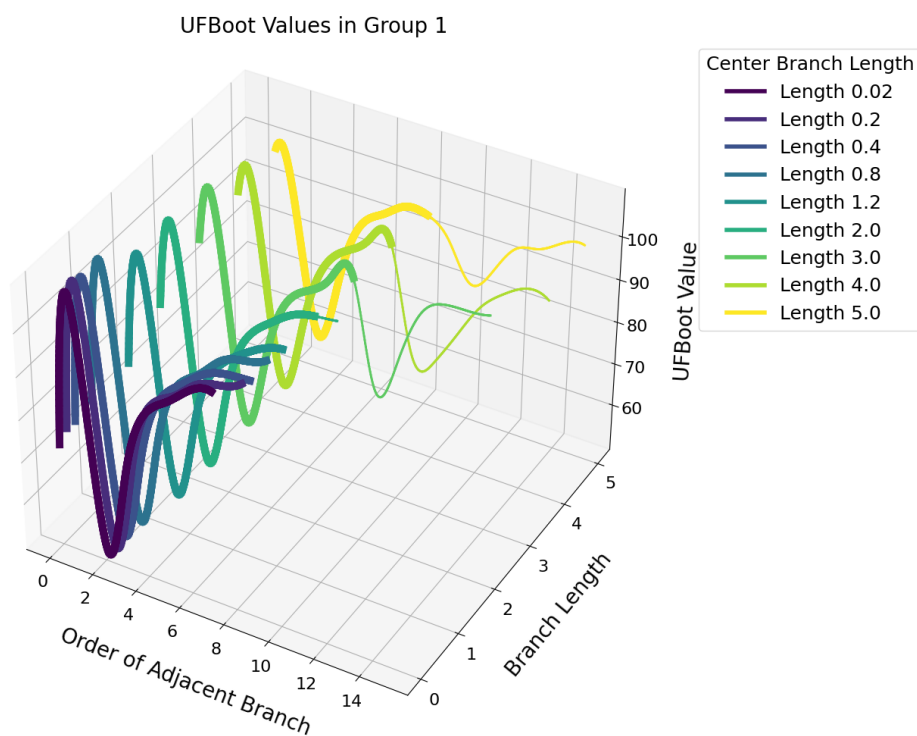


Figure 5.7: For each tree inferred from the 100 alignments, the mean UFBoot values of branches with the same order are calculated. These mean values are then averaged across all 100 trees for each branch order. The resulting average UFBoot values for each branch order are plotted against varying center branch lengths ranging from 0.02 to 5.0. Since clades t00 and t01 were not always sisters, the UFBoot values are plotted from the center branch to whichever clades formed as Clade Group 1. The thickness of the lines represents the number of inferred ML trees that have inner branches at this Order.

We see the same phenomenon that the longer saturated branches insert asymmetrically in the clades. The information on where to insert the longer saturated branch is lost.

At the center branch length of 3.0, the information is retained for all ML trees to have again the sister clades t00, t01, and respectively t10 and t11 grouped correctly together.

As the ML-JBs and the center branch receive approximately the same length, in two ML trees, a longer branch, instead of connecting with the center branch, starts interlinking both sister clades around Order 9 (see alignment Nr.: [74] in Figure A.3, where the TV with the higher Order of 15 is shown).

This can be observed throughout the entire section (a), where the center branch was fixed at a constant length of 6.0

As a consequence, we observed an Order increase from taxa from the second interlinked clade. As JB's have throughout section (b) the maximal length, the UFBoot value decline next to the insertion at around Order 9 was as severe as at the branch next to the center at its sister clade. For this, observe the sudden change in curves starting from a center branch length of 3.0 in Figure 5.7.

At a center branch length of 4.0, 94 ML trees maintain the connection of the center branch to all four JB's, resulting in no taxa being positioned more than 10th Order away from the center. However, at a center branch length of 5.0, the number of ML trees retaining the correct insertion of the center branch to all four JB's is reduced to 85.

Table 5.9: Summary of saturated branches by center branch length. SB stands for saturated branches, and CSC stands for correct sister clades (indicating the frequency with which the clades t00 and t01 formed clade t0, and the clades t10 and t11 formed clade t1). EXTB stands for external single branches, and TV stands for tree variation.

Center Branch len	0 SB	1 SB	2 SB	3 SB	4 SB	5 SB	CSC	EXTB	TV
0.02	0	0	0	7	17	76	40	7.02	13
0.2	0	0	0	5	15	80	49	7.04	11
0.4	0	0	1	7	13	79	54	7.04	15
0.8	0	0	1	8	11	80	72	6.76	18
1.2	0	0	2	5	18	75	89	7.08	12
2.0	0	0	0	9	11	80	98	7.11	16
3.0	0	0	1	7	19	73	100	6.92	20
4.0	0	0	4	15	21	60	100	7.34	27
5.0	0	1	1	12	26	60	100	7.68	32

Table 5.10: The number of inferred ML trees n , UFBoot values and standard deviation (SD) for Clade Group 1. From a center branch length of 0.02 to 2.0, the clade group may consist of any two of the four clades: t00, t01, t10, t11. However, with a center branch length ranging from 3.0 to 5.0, Clade Group 1 consistently comprises the clades t00 and t01. Details are available in the column CSC of Table 5.9. A noticeable shift in the order becomes apparent with a center branch length of 3.0.

Center Branch len	0.02			0.2			0.4			0.8		
Order	n	UFBoot	SD	n	UFBoot	SD	n	UFBoot	SD	n	UFBoot	SD
0	100	72.88	(± 11.24)	100	74.95	(± 11.75)	100	74.68	(± 12.11)	100	75.79	(± 12.23)
1	100	99.36	(± 0.87)	100	99.37	(± 0.83)	100	99.08	(± 1.44)	100	99.02	(± 1.20)
2	100	51.97	(± 12.10)	100	51.34	(± 12.68)	100	52.45	(± 11.84)	100	51.84	(± 11.69)
3	100	71.20	(± 13.35)	100	72.51	(± 14.07)	100	71.36	(± 12.62)	100	70.93	(± 14.85)
4	100	87.55	(± 6.27)	100	88.04	(± 6.03)	100	87.40	(± 5.90)	100	86.49	(± 7.36)
5	100	92.11	(± 5.15)	100	91.74	(± 6.61)	100	92.53	(± 5.88)	99	90.97	(± 6.34)
6	99	96.48	(± 2.29)	100	96.08	(± 2.90)	100	96.60	(± 2.93)	99	95.81	(± 3.18)
7	99	98.09	(± 1.89)	98	97.84	(± 2.23)	98	98.35	(± 1.89)	97	97.68	(± 2.43)
8				1	100.00	(± 0.00)	1	98.80	(± 0.00)	1	99.50	(± 0.00)

Center Branch len	1.2			2.0			3.0			4.0		
Order	n	UFBoot	SD	n	UFBoot	SD	n	UFBoot	SD	n	UFBoot	SD
0	100	82.51	(± 12.04)	100	88.83	(± 9.41)	100	94.93	(± 6.00)	100	97.36	(± 2.90)
1	100	98.08	(± 2.52)	100	96.72	(± 4.15)	100	93.72	(± 8.03)	100	88.37	(± 13.61)
2	100	56.05	(± 12.64)	100	55.59	(± 13.08)	100	55.69	(± 12.64)	100	54.95	(± 12.52)
3	100	74.25	(± 12.39)	100	73.63	(± 13.85)	100	72.03	(± 14.23)	100	71.20	(± 13.17)
4	100	88.38	(± 5.98)	100	88.24	(± 6.44)	100	87.09	(± 6.90)	100	86.59	(± 6.41)
5	100	93.19	(± 4.82)	100	93.23	(± 5.36)	100	91.58	(± 8.70)	100	91.85	(± 5.40)
6	100	96.55	(± 2.64)	100	96.82	(± 2.40)	100	96.26	(± 3.17)	100	96.04	(± 2.71)
7	99	98.27	(± 1.86)	98	98.35	(± 2.00)	99	98.04	(± 3.00)	100	97.07	(± 7.54)
8				1	99.00	(± 0.00)	2	73.50	(± 11.50)	6	72.63	(± 32.16)
9							2	81.75	(± 12.75)	6	72.08	(± 25.71)
10							2	95.00	(± 3.00)	5	80.44	(± 13.87)
11							2	98.75	(± 0.25)	4	88.75	(± 4.67)
12							2	99.10	(± 0.10)	4	93.67	(± 5.55)
13							1	99.80	(± 0.00)	4	96.75	(± 3.30)

Center Branch len	5.0		
Order	n	UFBoot	SD
0	100	98.97	(± 2.07)
1	100	82.92	(± 15.32)
2	100	57.07	(± 12.97)
3	100	73.50	(± 14.75)
4	100	87.51	(± 7.63)
5	100	91.65	(± 6.91)
6	100	95.05	(± 6.00)
7	99	94.79	(± 12.85)
8	15	90.56	(± 16.87)
9	15	81.98	(± 19.68)
10	10	86.37	(± 13.43)
11	10	93.52	(± 6.53)
12	8	95.16	(± 5.57)
13	7	98.16	(± 1.60)
14	5	99.44	(± 0.69)

These analyses illustrate that for such trees UFBoot would be the wrong choice.

It is inherent in the nature of resampling in bootstrapping and in UFBoot that the overall tree structure of the original alignment(s) - whether from the original alignment in bootstrapping or from the original starting alignments in UFBoot - and with that very long branches with neighboring short branches and branch length difference play a role.

Although the random resampling process in bootstrapping does not consider branch lengths directly, the main structure of the phylogenetic trees, particularly the placement of distinct branches, tends to remain consistent across bootstrap replicates, due to significant and consistent sequence differences. The consistent recovery of similar tree structures across bootstrap replicates results in high bootstrap support values for very long branches. This is due to a cumulative effect from the increasing number of pseudoalignments.

In UFBoot, biased starting trees have been criticized for providing inflated bootstrap values, and in normal bootstrapping, saving only a single optimal tree per pseudoalignment when there are multiple equally optimal trees can lead additionally to inflated bootstrap values (Simmons and Norton, 2014).

If resampling of sites does not significantly affect the overrepresentation of the very long branch, it becomes particularly crucial for the directly adjacent very short branches (compare the higher SD for Order 1 and 2 in Table 5.8). The support for very short branches, especially those on positions of the sites directly adjacent to long branches, can be more sensitive to resampling variability. This sensitivity arises because their support may depend on a few critical sites that can be randomly included or excluded during resampling. When these critical sites are randomly excluded, the perceived support for these short branches can be significantly reduced. In contrast, short branches further away from the long branch are less affected by resampling variability because they rely less on the immediate neighboring positions within these sites for their support. The information at these positions on these sites may be captured more consistently during resampling, potentially resulting in more accurate UFBoot values.

Selecting UFBoot as a phylogenetic tool for trees with significant branch length differences highlights the need to understand how resampling can impact the analysis, particularly when dealing with saturated branches. Without this understanding, visualized values may lead to incorrect data interpretations. Our analysis highlights that UFBoot values of 100 do not inherently indicate a shared phylogenetic history between two groups.

Satute, utilizing partial likelihood vectors approaching zero, accurately assesses phylogenetic relationships by validating saturation within alignments. In contrast, UFBoot primarily measures group separation rather than phylogenetic relationships. Our analysis underscores that UFBoot may not be optimal for alignments with highly divergent sequences.

Satute offers a dual analysis by integrating UFBoot. In cases where such branch length variation does not occur and most branches will not be saturated, UFBoot may provide additional beneficial information.

Another aspect is that UFBoot and Bootstrapping do not consider the exact tree structure. Although information about the overall tree structure is present—in Bootstrapping through

maximum likelihood tree calculations and in UFBoot through the maximum likelihood alignment—the statistical support of a branch in Bootstrapping is determined by the frequency of a clade across replicates. Thus, the exact branching order, and consequently the precise tree structure, is not taken into account. Despite its limitations, bootstrapping provides a reasonable approximation of the actual confidence levels of the observed clades (Efron et al., 1996; Horowitz, 2001).

To the long branches which have a preference to dock near external single branches. These phenomena exist because of a disposition to insert the node near the first branches of a clade. So it is not completely random. Nonetheless, the location where the long branch exactly inserts into the subtree is not reliably predictable. This unpredictable placement of long branches makes it difficult to reliably root the tree using only the sequence data itself. It shows that rooting with random sequences is more or less arbitrary. The decision of rooting is usually done by a method called outgroup rooting.

An outgroup (Kinene et al., 2016) is chosen based on its presumed phylogenetic relatedness to the group of interest (the ingroup). The outgroup is expected to have diverged from the ingroup before the ingroup taxa diverged from each other.

The outgroup method, (Wheeler, 1990; Tarrio et al., 2000; Hess and De Moraes Russo, 2007; Boykin et al., 2010), roots phylogenetic trees by assuming divergence of one or more taxa from the ingroup, establishing evolutionary direction (Brady et al., 2011; Williams, 2014). It aids in identifying unique features of ingroup sequences but relies on prior knowledge to select an appropriate outgroup, posing challenges in taxa lacking consensus outgroups (Rota-Stabelli and Telford, 2008).

The matter is just in our model tree; such a long, saturated central branch indicates that it is just there by chance, due to missing reliability.

In ML tree reconstruction, the presence of a long, saturated branch poses unique challenges. We already observed that where such saturated branch inserts in the clade may not reflect any true evolutionary relationship.

We might assume that the alignment sections on either side of a very long saturated branch are reconstructed independently. This independence suggests that the evolutionary histories of clades, distanced by the saturated branch, may be treated as distinct entities in the tree reconstruction process. Consequently, these distant clades should not exert interdependent influences on each other’s placement solely because of the presence of the long saturated branch. In essence, in the context of ML tree reconstruction, when clades are separated by a long saturated branch, their evolutionary relationships may be interpreted as if they are part of independent trees.

In the next section, we explore the challenge of verifying the assumption that clades on either side of a long saturated branch are inferred independently from each other during ML tree reconstruction, using a model of two simple clades separated by such a branch.

5.4. Influence of Clades Separated by a Saturated Branch on Tree Reconstruction

Introduction

This study investigates whether clades separated by a saturated branch can influence the overall ML tree reconstruction.

To address the question of whether a clade on one side of a saturated branch can affect the reconstruction on the opposite side, we first evaluate the true topology using UFBoot.

Given that the long saturated branch starts with high UFBoot support, we will closely examine how varying the branch length of the left clade affects the UFBoot values of both the left and right clades in the true tree topology, as well as whether it is observable if these clades influence each other.

A template Newick string was created with a saturated central branch of constant length 3.0 and two clades, each containing 16 leaves. The right clade had branches with a consistent length of 0.01 substitutions per site, denoted as 'r'. For the left clade, branches labeled 'l' were varied in length from 0.3 to 3.0 in increments of 0.3, resulting in 10 distinct Newick files, each with one of these branch lengths.

The label 'l0' represents the uppermost branch adjacent to the center within the left clade, and 'l1' represents the bottom branch next to the center within the left clade (see Figure 5.8(e) for the tree topology).

For each of these Newick files, 100 multiple sequence alignments were simulated using AliSim (Ly-Trong et al., 2022), once for 1,000 sites and once for 10,000 sites.

All alignments were inferred using the Jukes-Cantor model with four gamma-distributed rates. During inferring for each ML tree, 5000 UFBoot trees were collected in each run. These files with 5000 UFBoot trees were later used for each of the true trees to map them with UFBoot values. That way we mapped UFBoot values onto 1,000 true trees for 1,000 sites and onto 1,000 true trees for 10,000 sites.

Median and average UFBoot values were computed for the central branch, as well as for branches at the first, second, and third orders away from the center branch, across 100 true trees for each of 10 different lengths and for both site numbers, totaling 2,000 trees.

In Table 5.11 are all the number of saturated branches shown for each rate category, inferred with the JC+G4 model are the true template trees. Leaf and inner branches refer to the branches in the left clade.

In Figure 5.9(a), the superimposed UFBoot values on the first two branches next to the center from the right clade are shown for the true trees from alignments of 1,000 sites.

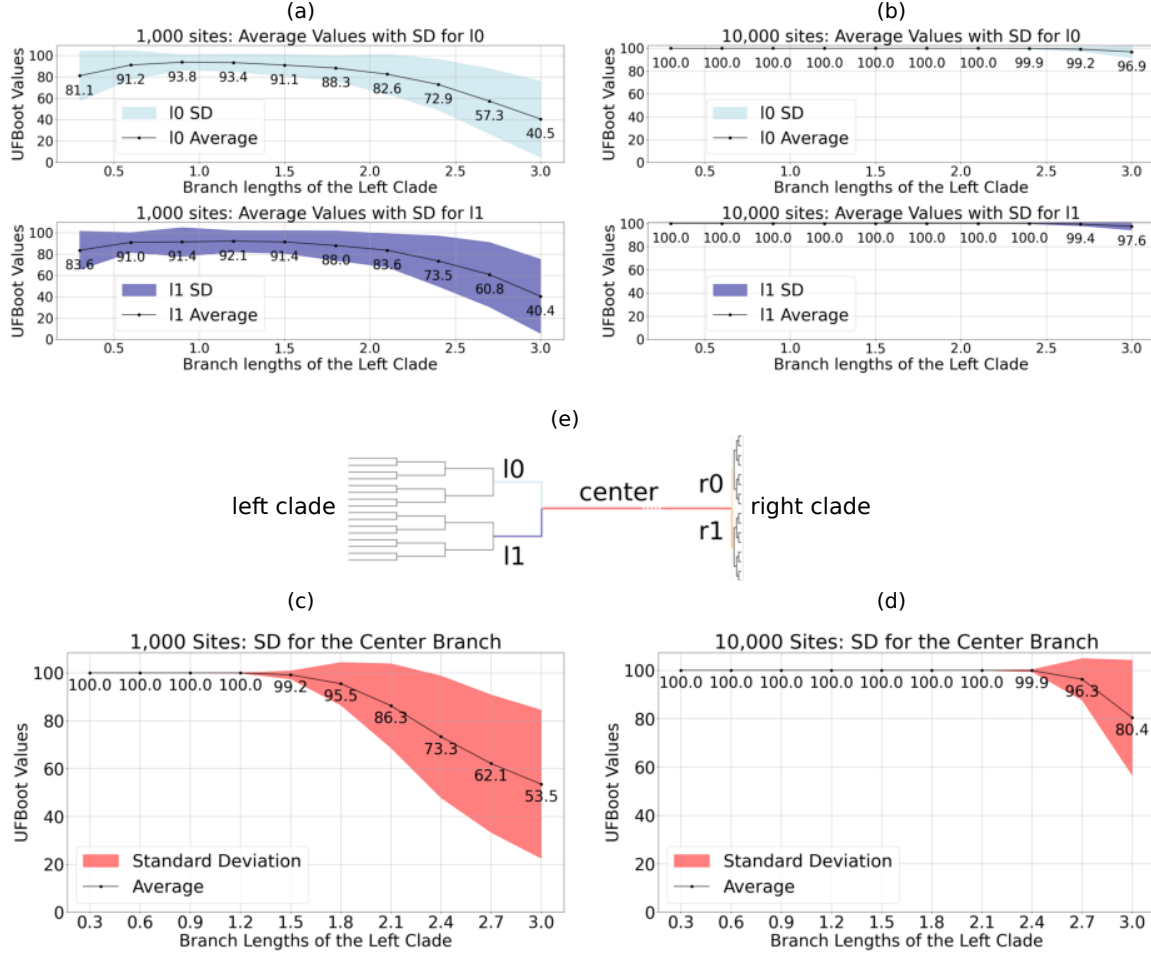


Figure 5.8: The UFBoot values for a specified branch on the true tree, averaged across 100 alignments for each of 10 different branch lengths in the left clade (ranging from 0.3 to 3.0 in steps of 0.3), are depicted. In (a) and (b), the UFBoot values for the direct neighboring branches of the center (I0 and I1) in the left clade are shown for 1,000 site alignments and 10,000 site alignments, respectively. In (c) and (d), the UFBoot values for the center branch across all branch lengths of the left clade are displayed for 1,000 sites and 10,000 sites, respectively. The tree in (e) illustrates the topology of the true tree. This figure depicts a 32-leaf tree template designed for 1,000 and 10,000 sites. A total of 10 different trees were generated, with the branches of the left clade elongated from 0.3 to 3.0. The center branch remains consistently at 3.0, while the branch lengths in the right clade remain fixed at 0.01.

UFBoot disparity

We have symmetric clades, therefore branches should show nearly identical labeled UFBoot values on opposing sides.

However, branches r0 and r1 exhibit a significant dissimilarity in their UFBoot values. This discrepancy is notable because the values are mapped on opposing sides of the same clade with identical branch lengths within the same true tree topology.

len/n RC	Center Branch	Inner Branches	Leaf Branches	Sites		len/n RC	Center Branch	Inner Branches	Leaf Branches	Sites
0.3/100						1.8/97				
1	0	0	0	243.80		1	0	0	0	267.32
2	0	0.09	0	253.63		2	1	14	0.20	321.14
3	1	1.86	0	246.11		3	1	14	16	189.19
4	1	14	2.04	256.47		4	1	14	14.31	222.35
0.6/100						2.1/96				
1	0	0	0	258.06		1	0	0	0	276.57
2	1	0.07	0	249.55		2	1	14	1.57	329.69
3	1	13.94	0	239.55		3	1	14	16	168.61
4	1	14	15.98	236.65		4	1	14	14.44	225.13
0.9/100						2.4/84				
1	0	0	0	253.17		1	0	0	0	284.64
2	1	5.35	0	270.63		2	1	14	3.5	310.14
3	1	14	10.61	254		3	1	14	16	178.52
4	1	14	16	234		4	1	4	14.79	226.69
1.2/100						2.7/75				
1	0	0	0	255.93		1	0	0.05	0	295.4
2	1	13.67	0	279.74		2	1	14	15.52	429.04
3	1	14	16	259.79		3	1	14	16	48.49
4	1	14	14.88	204.54		4	1	14	14.8	227.07
1.5/100						3.0/54				
1	0	0	0	258.22		1	0	0.96	0	302.51
2	1	13.92	0	275.97		2	1	14	15.89	419.70
3	1	14	16	250.41		3	1	14	16	51.12
4	1	14	14.68	215.40		4	1	14	14.70	226.66

Table 5.11: The average number of saturated branches for the topology of the true tree is displayed. The elongated branch length of the left clade (len) and sample size (n) are listed in the shaded cells.

The most significant difference is observed in tree structures with branch lengths of 0.3 to 1.2 for the left clade. The correlation coefficient was calculated to be 0.59 for 1,000 sites, see Figure 5.9(a), and 0.84 for 10,000 sites, see Figure 5.9(b), which are both too low.

For control purposes, we superimposed the curves of UFBoot values for the two branches of the left clade in the first order. The correlation coefficient of 1.0 aligns with the expected ideal value, reflecting the symmetry inherent in the left clade.

Due to the UFBoot disparity, we opted to permute all 32 sequences randomly for all samples before inferring the tree to introduce more randomness. Therefore we hoped to escape a possible systematic error. Surprisingly the correlation coefficient for 1,000 sites is even worse, with a correlation coefficient of 0.23, while for sequence lengths of 10,000 sites, it remains relatively stable, changing only slightly from 0.84 to 0.82. The problem of the UFBoot discrepancy remains.

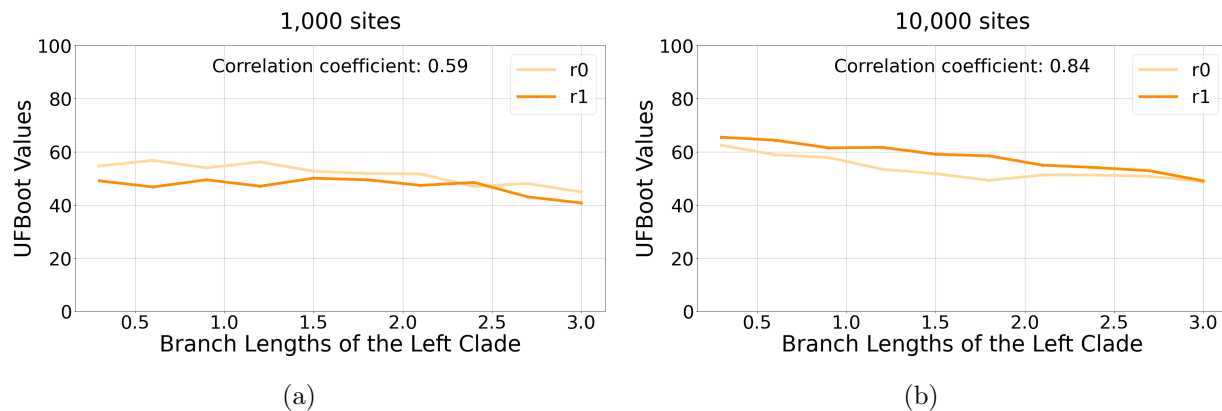


Figure 5.9: For each branch length of the left clade, the averages of the UFBoot values for r0 and r1, based on 100 samples, are depicted. Sequences with 10,000 sites, see Figure 5.9(b) exhibit slightly higher UFBoot values compared to those with 1,000 sites. This increase in UFBoot values is likely attributed to the larger amount of sequence data, which enhances the statistical robustness of the inferred tree topology. The correlation coefficient for both curves is 0.84, 25 points higher than the correlation coefficient of curves from 1,000 site alignments. It is still too low from the expected correlation coefficient near 1.0.

Following that, we performed sequence rotation within the alignments. In every sample, all sequences belonging to the lower right branch r1 were exchanged with sequences belonging to the upper right branch r0. The names for the sequences stayed in their place. We would anticipate an inverted pattern to the original inference, as depicted in Figure 5.9(a), up to an elongated branch length of 0.9 this would be the case. However, this trend is not consistently observed beyond this elongated branch length.

We investigated whether this problem arises solely with UFBoot. Therefore, we conducted an additional analysis on the same alignments using bootstrapping. The resulting correlation coefficient remained low at 0.64, reaffirming our initial findings.

To rule out the possibility that AliSim is responsible for this limitation, we simulated 100 alignments for each of two selected lengths using Seq-Gen (Rambaut and Grass, 1997). The differences in r0 and r1 values, as shown in Table 5.12, remained substantial, and this trend was consistent across both 1,000 and 10,000 sites.

UFBoot 1000	Median 0.3	Average 0.3	Median 2.4	Average 2.4
r1	60	54.3	58	51
r0	50	49.9	40.5	46
center	100	100	86	75.1
l0	92	85.9	79	72.8
l1	89.5	81.7	78	69.8

Table 5.12: Simulation with Seq-gen for alignments of 1,000 sites. Values for all branches were calculated. For a clearer overview, only values for branches r1, r0, center, l0, and l1 are shown.

The methodologies examined can not with certainty conclude that there is a potential association between the observed UFBoot discrepancy and the IQ-TREE2 software itself; definitive conclusions regarding causality or underlying factors remain elusive.

5.4.1. Analysis of Average UFBoot Values

In all graphs in Figure 5.8, the standard deviation of the average UFBoot values is relatively high throughout all different lengths for the left clade. It can be attributed to IQ-TREE2 generating 5000 trees in its UFBoot file for each inferring, which are later used for mapping UFBoot values on the branches of the true trees. This results in UFBoot values of zero on some branches.

Further, we see that the branch lengths influence the standard deviation. For the center branch, the longer the branches of the left clade are the higher the standard deviation, which illustrates the influence of the left clade (see Figure 5.8(c)). For convenience, unless otherwise specified, all references to "branch length" in this document pertain to the branches in the left clade.

For the branches l0, and l1 next to the center, from a branch length of 1.5 to 3.0 in the left clade, the standard deviation (SD) begins steadily to increase for the alignments of 1,000 sites. For alignments consisting of 10,000 sites (see Figure 5.8(c)), we observe a delayed and smaller increase in standard deviation after branch lengths of 2.4, illustrating a weaker influence of similar characteristics. Such effects happen in 10,000 site alignments due to a higher chance of finding enough informative sites to support the branch in the long branches, resulting in a delayed and smaller increase in SD and a weaker decline in UFBoot values.

UFBoot values for branches on the right clade (r0 and r1) for alignments consisting of 1,000 sites, exhibit a consistently high standard deviation of approximately ± 35 across all ten elongated branch lengths. The line for the average UFBoot values on branches for the right clade fluctuates steadily. For alignments of 10,000 sites, there is a slight decrease in the average UFBoot values across all branch lengths. For 1,000 sites, the slope of the curve is even less pronounced, indicating an almost horizontal trend with a minimal downward slope.

We can say that an influence of the left clade is insignificant or hardly observable with UFBoot for alignments of 1,000 and 10,000 sites to the right clade (see Figures 5.9(a) and 5.9(b)). Interpreting UFBoot values in this context requires caution.

The primary effect, besides that UFBoot support primary focus is within its clade itself, are the resampling effects, already discussed in the previous section 5.3. UFBoot's high SD highlights as well the inherent variability in UFBoot values due to resampling.

The standard deviations for branches l0 and l1 in the left clade show a distinctive U-shaped pattern. In Figure 5.8(a), we note higher standard deviations initially, peaking before branch lengths reach 0.6 and again beyond longer lengths exceeding 2.1. The lowest standard deviations for the branches in the left clade are observed at medium branch lengths ranging from 0.9 to 1.5. Very long branch lengths might lead to issues with model assumptions in UFBoot, resulting in increased variability and a high SD. It simply means UFBoot is consistently recovering these branches of medium length for the true tree topology across replicates.

The declining UFBoot values are observed for both the center branch and the branches l0, l1 are reflected through mutual data saturation in both Figures 5.8(a) and 5.8(c) as in the Table 5.11.

5.4.2. Robinson-Foulds Distances to Assess Left and Right Clade Independence

To address the limitations of UFBoot for very long branches and assess the potential independence of the left and right clades due to the saturated branch, we employed Robinson-Foulds (RF) distances.

The symmetric Robinson-Foulds (RF) distance is a measure of dissimilarity between two phylogenetic trees. It is defined as the sum ($A + B$) where A is the number of partitions (or splits) of data implied by the first tree but not the second tree, and B is the number of partitions of data implied by the second tree but not the first tree (Robinson and Foulds, 1981).

If a very long branch is saturated, it effectively contains no informative data. Consequently, the left and right clades on either side of this saturated branch may appear to behave as independent trees in the ML reconstruction process, similar to how subalignments might be inferred separately. To simulate an independent tree topology for the left clade, we inferred trees from subalignments of only 16 sequences from the left clade (see Figure 5.10(c)).

To measure RF distances, we used maximum likelihood trees from the full alignments, splitting the Newick trees (Archie et al., 1986) between the left and right clades to collect each left clade topology. Trees where branches of the left clade were intertwined with branches of the right clade, making splitting impossible, were discarded (see Figure 5.10(a)).

For the true trees, we collected all left clades with the true tree topology, which were mapped with UFBoot values (see Figure 5.10(b)).

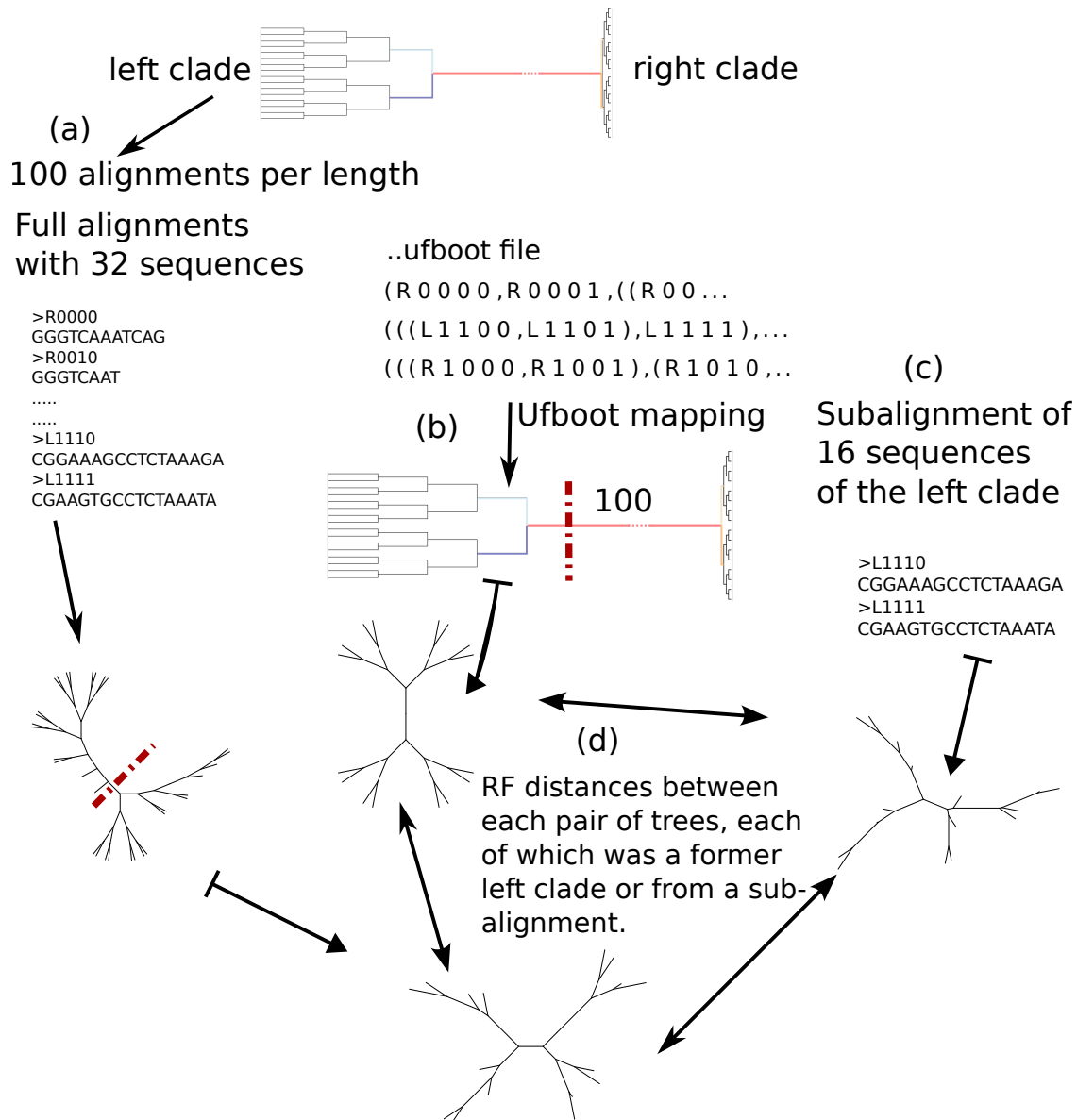


Figure 5.10: Illustration of the three methods for generating left clad trees: (a) Collection of left clad trees from ML analyses of the full alignments, highlighting the feasibility of separating at the central branch between both clades. (b) Generation of true left clad trees based on the same ML sample numbers. (c) Construction of left clad trees from a subalignment containing 16 sequences from the left clad.

True vs. Subalignment Tree RF Distance: The first RF distance measurement distance reflects the dissimilarity between the true left clad topology and trees inferred from the left clad subalignment.

True vs. Full Alignment Tree RF Distance: The second RF measurement assesses the dissimilarity between the true left clad topology and the left clades derived by splitting the ML trees from the full alignment.

Full vs. Subalignment Tree RF Distance: Thirdly, we additionally evaluated the dissimilarity between left clades inferred from full alignment trees versus subalignment trees (see Figure 5.10(d) for all three RF distances).

In Figure 5.11 are the relative RF distances of the first and second RF measurements from the true tree left clade topology depicted.

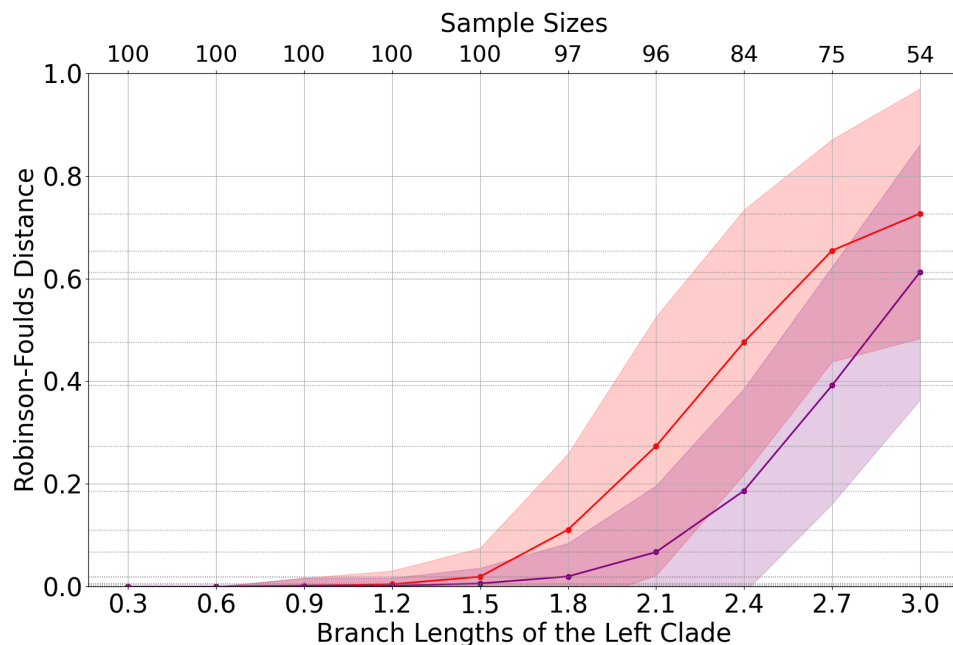


Figure 5.11: Relative Robinson-Foulds (RF) distances (division by the number of inner branches) between left clade trees derived from alignments of 1,000 sites. The secondary x-axis at the top shows the sample sizes, indicating the number of successful splits in the Maximum Likelihood ML trees. In the lower purple curve are the True vs. Full Alignment Tree RF Distance depicted. In the upper red curve the True vs. Subalignment Tree RF Distance. Standard deviation (SD) is included for both curves.

Notably, the True vs. Full Alignment Tree RF Distances (purple curve) are consistently lower compared to the True vs. Subalignment Tree RF Distances (red curve) (see Figure 5.11). We could show that the true tree topology is throughout all lengths closer to the resulting left clade split from the ML tree than to the topology of clades from individually aligned subalignments.

These results suggest that despite the presence of a saturated branch, the ML tree reconstruction effectively integrates information from the right clade into the reconstruction of the left clade topology. This indicates that informative sites within the flanking clades contribute positively to the overall tree topology. The successful splits of left clades from the ML tree of the full alignment were surprisingly high up to a branch length of 2.1 and 2.4, with sample sizes of 96 and 84, respectively.

The findings imply that the ML reconstruction process utilizes information from different regions of the alignment, even when challenged by a saturated branch.

Till the branch length of 2.1, the absolute RF distance to the true tree topology from the resulting subclade is just 0.87. The Full vs. Subalignment Tree RF Distance is very close at 3.26 to the True vs. Subalignment Tree RF Distance at 3.55. For all absolute RF measurements, between all three different RF distances see Table A.3 in the appendix. This similarity between the Full vs. Subalignment and True vs. Subalignment RF distances remains even at a branch length of 2.4. It demonstrates a similar level of dissimilarity between the Full vs. Subalignment Tree RF Distance and the True vs. Subalignment Tree RF Distance for these branch lengths. However, this similarity does not necessarily indicate that the ML tree reconstruction closely matches the true tree topology.

From a branch length of 1.8, the increase in RF distance is the steepest. The slower increase in the second RF measurement (True vs. Full Alignment Tree) results in the largest spacing at branch length 2.4 compared to the first RF distance measurement (True vs. Subalignment Tree RF Distance).

In alignments for 10,000 sites is already that much information in the sites, despite the saturated long branch that the ML tree topology equals the true tree topology. Simulated independent trees inferred from subalignments closely resemble the true tree topology, exhibiting minimal RF distances of 0.16 at a branch length of 2.7 and 1.12 at 3.0.

We analyzed additionally the average UFBoot values off all inner branches over all of all the different left clades. These results though less significant, support our RF analyses. Till the branch length of 2.1, the average UFBoot values of the true tree left clade are with around 88 very close to the UFBoot values of the split clades from the ML tree. The average UFBoot values of the clades from the subalignments are at these lengths around 10 values lower. Beyond a branch length of 2.4, the UFBoot values for the true tree clade decline rapidly (see Figure A.4 in the appendix).

We know already that the influence of the long saturated branch, however, doesn't necessarily guarantee an accurate placement of the flanking clades. The verification that both by a saturated section separated clusters in the alignment contribute positively to the overall tree reconstruction makes outgroup rooting a feasible phylogenetic approach.

While the saturated branch introduces some ambiguity, the inferred tree topology may allow us to tentatively designate the left clade as an outgroup relative to the right clade (ingroup), but it's important to acknowledge the limitations introduced by the saturated branch. In a "real" phylogenetic tree, the outgroup should represent the true evolutionary relationship, something the saturated branch might compromise.

6: Conclusion

In this study, we investigated the relationship between saturation values from Satute and UFBoot values in phylogenetic tree reconstruction.

We established that there is no correlation between saturation and UFBoot values. Saturation values, provided by Satute, and UFBoot values serve as complementary tools in phylogenetic analysis. While UFBoot values assess group separation in alignments, Satute evaluates branch reliability within phylogenetic trees.

We described that the high bootstrap support values observed for very long branches are a consequence of the consistent recovery of similar tree topologies across numerous bootstrap replicates. This phenomenon can be attributed to a cumulative effect arising from the number of pseudoalignments. While every resampling consistently supports the very long branch, the positional information of the sites for the immediate neighboring branches can lead to various neighboring branch configurations.

In such instances, Satute is the appropriate tool, as it accurately measures reliability through partial likelihood vectors that, if approaching zero, validate saturation. If all branches in a tree are reliable and such detriment branch length variation does not occur, UFBoot sampling can be an appropriate tool for assessing branch confidence.

The phenomenon that long branches have a preference to dock near external single branches exists because of a disposition to insert the node near the first branches of a clade. Rooting such long saturated branches is therefore unreliable in the ML tree reconstruction.

We could show that the resulting left clade split from the ML tree is consistently closer to the true tree topology across all branch lengths compared to individually aligned subalignment.

In our analyses, we verified that despite the presence of a saturated branch, the ML tree reconstruction effectively integrates information from one clade, thereby positively influencing the reconstruction of the other clade topology and consequently, the overall tree topology.

However, it is important to note that our conclusions are based on circumstantial evidence through our limited dataset based on a single model tree.

Bibliography

- H. Akaike. A new look at the statistical model identification. IEEE Transactions on Automatic Control, 19(6):716–723, 1974. doi: 10.1109/TAC.1974.1100705.
- James Archie, William H. E. Day, Joseph Felsenstein, Wayne Maddison, Christopher Meacham, F. James Rohlf, and David Swofford. The newick tree format. <https://phylipweb.github.io/phylip/newicktree.html>, 1986. Accessed: 2023-06-30.
- Magnus Bordewich and Charles Semple. On the computational complexity of the rooted subtree prune and regraft distance. Annals of combinatorics, 8:409–423, 2005.
- Laura M Boykin, Laura Salter Kubatko, and Timothy K Lowrey. Comparison of methods for rooting phylogenetic trees: A case study using *orcuttieae* (poaceae: Chloridoideae). Molecular phylogenetics and evolution, 54(3):687–700, 2010.
- Seán G Brady, Jessica R Litman, and Bryan N Danforth. Rooting phylogenies using gene duplications: an empirical example from the bees (apoidea). Molecular Phylogenetics and Evolution, 60(3):295–304, 2011.
- Lena Collienne and Alex Gavryushkin. Computing nearest neighbour interchange distances between ranked phylogenetic trees. Journal of Mathematical Biology, 82(1):8, 2021.
- David A Duchêne, Niklas Mather, Cara Van Der Wal, and Simon YW Ho. Excluding loci with substitution saturation improves inferences from phylogenomic data. Systematic Biology, 71(3):676–689, 2022.
- Bradley Efron. Bootstrap methods: another look at the jackknife. In Breakthroughs in statistics: Methodology and distribution, pages 569–593. Springer, 1977.
- Bradley Efron, Elizabeth Halloran, and Susan Holmes. Bootstrap confidence levels for phylogenetic trees. Proceedings of the National Academy of Sciences, 93(23):13429–13429, 1996.
- Merran Evans, Nicholas Hastings, and Brian Peacock. Statistical distributions. John Wiley & Sons, 2011.
- J Felsenstein. Evolutionary trees from dna sequences: a maximum likelihood approach. Journal of molecular evolution, 17(6):368–376, 1981. ISSN 0022-2844. doi: 10.1007/bf01734359. URL <https://doi.org/10.1007/BF01734359>.
- Joseph Felsenstein. Maximum-likelihood estimation of evolutionary trees from continuous characters. American journal of human genetics, 25(5):471, 1973.

- Joseph Felsenstein. Confidence limits on phylogenies: An approach using the bootstrap. <https://pubmed.ncbi.nlm.nih.gov/28561359/>, 1985.
- Joseph Felsenstein. Phylogenies from restriction sites: a maximum-likelihood approach. Evolution, 46(1):159–173, 1992.
- Joseph Felsenstein. Inferring phylogenies.—joseph felsenstein. 2003. sinauer associates, sun-derland, massachusetts., 2003.
- Joseph Felsenstein and Hirohisa Kishino. Is there something wrong with the bootstrap on phylogenies? a reply to hillis and bull. Systematic Biology, 42(2):193–200, 1993.
- Carmen Cinta Viñas Ferrando. Implementation of a Test for Branch Saturation in a Phylogenetic Tree. Universitat Politècnica de Catalunya, Facultat de Matemàtiques i Estadística, 2022.
- Gavin Huttley Group. Substitution models in molecular evolution. Website, 2024. URL https://gavinhuttley.github.io/tib/molevol/substitution_models.html. Accessed: July 4, 2024.
- Stéphane Guindon and Olivier Gascuel. A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. Systematic biology, 52(5):696–704, 2003.
- Masami Hasegawa and Hirohisa Kishino. Accuracies of the simple methods for estimating the bootstrap probability of a maximum-likelihood tree. 1994.
- Pablo N Hess and Claudia A De Moraes Russo. An empirical test of the midpoint rooting method. Biological Journal of the Linnean society, 92(4):669–674, 2007.
- David M Hillis and James J Bull. An empirical test of bootstrapping as a method for assessing confidence in phylogenetic analysis. Systematic biology, 42(2):182–192, 1993.
- Simon YW Ho and Lars S Jermini. Tracing the decay of the historical signal in biological sequence data. Systematic biology, 53(4):623–637, 2004.
- Diep Thi Hoang, Olga Chernomor, Arndt von Haeseler, Bui Quang Minh, and Le Sy Vinh. UFBoot2: Improving the Ultrafast Bootstrap Approximation. Molecular Biology and Evolution, 35(2):518–522, 10 2017. ISSN 0737-4038. doi: 10.1093/molbev/msx281. URL <https://doi.org/10.1093/molbev/msx281>.
- Diep Thi Hoang, Olga Chernomor, Arndt Von Haeseler, Bui Quang Minh, and Le Sy Vinh. Ufboot2: improving the ultrafast bootstrap approximation. Molecular biology and evolution, 35(2):518–522, 2018.
- Michael Hoff, Stefan Orf, Benedikt Riehm, Diego Darriba, and Alexandros Stamatakis. Does the choice of nucleotide substitution models matter topologically? BMC bioinformatics, 17:1–13, 2016.
- Susan Holmes. Bootstrapping phylogenetic trees: theory and methods. Statistical Science, 18(2):241–255, 2003.

- Joel L Horowitz. The bootstrap. In Handbook of econometrics, volume 5, pages 3159–3228. Elsevier, 2001.
- Jaime Huerta-Cepas. Tutorial: Environment for tree exploration. http://etetoolkit.org/docs/latest/_downloads/ETE.pdf, 2015.
- Lars S Jermini and Bernhard Misof. Measuring historical and compositional signals in phylogenetic data. bioRxiv, pages 2020–01, 2020.
- Lars S Jermini, Simon YW Ho, Faisal Ababneh, John Robinson, and Anthony WD Larkum. The biasing effect of compositional heterogeneity on phylogenetic estimates may be underestimated. Systematic biology, 53(4):638–643, 2004.
- Fangzhi Jia, Nathan Lo, and Simon YW Ho. The impact of modelling rate heterogeneity among sites on phylogenetic estimates of intraspecific evolutionary rates and timescales. PLoS One, 9(5):e95722, 2014.
- Thomas H Jukes, Charles R Cantor, et al. Evolution of protein molecules. Mammalian protein metabolism, 3:21–132, 1969.
- Samuel Karlin. A first course in stochastic processes. Academic press, 2014.
- Tonny Kinene, J Wainaina, Solomon Maina, and LM Boykin. Rooting trees, methods for. Encyclopedia of Evolutionary Biology, page 489, 2016.
- Oliver Knill. Linear algebra with probability. Harvard College Course Math 19b, 2011.
- Philippe Lemey, Marco Salemi, and Anne-Mieke Vandamme. The phylogenetic handbook: a practical approach to phylogenetic analysis and hypothesis testing. Cambridge University Press, 2009.
- David A Levin and Yuval Peres. Markov chains and mixing times, volume 107. American Mathematical Soc., 2017.
- Nhan Ly-Trong, Suha Naser-Khdour, Robert Lanfear, and Bui Quang Minh. Alisim: a fast and versatile phylogenetic sequence simulator for the genomic era. Molecular biology and evolution, 39(5):msac092, 2022.
- Cassius Manuel. New Measures of Phylogenetic Information Allow to Test for Saturation. Springer Nature, 2021.
- Cassius Manuel, Arndt von Haeseler, Carme Vinas, Enes Sakalli, and Christiane Elgert. When all is lost: Measuring phylogenetic information with satute. CIBIV, 2024.
- Ravi R. Mazumdar. Notes on markov chains and continuous-time markov chains, 2012. URL https://ece.uwaterloo.ca/~mazum/ECE605_2012/Notes/MarkovChains_CTMC.pdf. Chapter 6.
- Sean P Meyn and Richard L Tweedie. Markov chains and stochastic stability. Springer Science & Business Media, 2012.

- Bui Quang Minh, Le Sy Vinh, Arndt Von Haeseler, and Heiko A Schmidt. piqpnni: parallel reconstruction of large maximum likelihood phylogenies. Bioinformatics, 21(19):3794–3796, 2005.
- Bui Quang Minh, Minh Anh Thi Nguyen, and Arndt Von Haeseler. Ultrafast approximation for phylogenetic bootstrap. Molecular biology and evolution, 30(5):1188–1195, 2013.
- Bui Quang Minh, J Trifinopoulos, D Schrempf, HA Schmidt, and R Lanfear. Iqtree version 2.0: tutorials and manual phylogenomic software by maximum likelihood. <http://www.iqtree.org/doc/iqtree-doc.pdf>, 2019.
- Bui Quang Minh, Heiko A Schmidt, Olga Chernomor, Dominik Schrempf, Michael D Woodhams, Arndt Von Haeseler, and Robert Lanfear. Iq-tree 2: new models and efficient methods for phylogenetic inference in the genomic era. Molecular biology and evolution, 37(5):1530–1534, 2020.
- Matthew A Napierala. What is the bonferroni correction? Aaos Now, pages 40–41, 2012.
- Masatoshi Nei and Sudhir Kumar. Molecular Evolution and Phylogenetics. Oxford University Press, 2000.
- Lam-Tung Nguyen, Heiko A Schmidt, Arndt Von Haeseler, and Bui Quang Minh. Iq-tree: a fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. Molecular biology and evolution, 32(1):268–274, 2015.
- Lam-Tung Nguyen, Arndt von Haeseler, and Bui Quang Minh. Complex Models of Sequence Evolution Require Accurate Estimators as Exemplified with the Invariable Site Plus Gamma Model. Systematic Biology, 67(3):552–558, 11 2017. ISSN 1063-5157. doi: 10.1093/sysbio/syx092. URL <https://doi.org/10.1093/sysbio/syx092>.
- Karl Pearson. Correlation coefficient. In Royal Society Proceedings, volume 58, page 214, 1895.
- Hervé Philippe, Henner Brinkmann, Dennis V Lavrov, D Timothy J Littlewood, Michael Manuel, Gert Wörheide, and Denis Baurain. Resolving difficult phylogenetic questions: why more sequences are not enough. PLoS biology, 9(3):e1000602, 2011.
- WH Piel, MJ Donoghue, and MJ Sanderson. Treebase: a database of phylogenetic knowledge. To the interoperable “Catalog of Life” with partners Species, 2000:41–47, 2000.
- David Posada and Keith A Crandall. Felsenstein phylogenetic likelihood. Journal of molecular evolution, 89(3):134–145, 2021.
- Andrew Rambaut and Nicholas C Grass. Seq-gen: an application for the monte carlo simulation of dna sequence evolution along phylogenetic trees. Bioinformatics, 13(3):235–238, 1997.
- Vincent Ranwez, Frédéric Delsuc, Sylvie Ranwez, Khalid Belkhir, Marie-Ka Tilak, and Emmanuel JP Douzery. Orthomam: a database of orthologous genomic markers for placental mammal phylogenetics. BMC evolutionary biology, 7:1–12, 2007.

- Reden, Schmidt, and von Haeseler. Evonaps, 2024. URL <http://evonaps.cibiv.univie.ac.at/index.php>. University of Vienna, CIBIV.
- David F Robinson and Leslie R Foulds. Comparison of phylogenetic trees. Mathematical biosciences, 53(1-2):131–147, 1981.
- Omar Rota-Stabelli and Maximilian J Telford. A multi criterion approach for the selection of optimal outgroups in phylogeny: recovering some support for mandibulata over myriochelata using mitogenomics. Molecular Phylogenetics and Evolution, 48(1):103–111, 2008.
- Gerardo Rubino and Bruno Sericola. Sojourn times in finite markov processes. Journal of Applied Probability, 26(4):744–756, 1989.
- Heiko Schmidt. Bioinformatics for phylogeny and evolutionary tree reconstruction. University Lecture, 2022.
- Heiko Schmidt, Jana Trifinopoulos, Minh Bui, and Rob Lanfear. Iq-tree frequently asked questions: How do i interpret ultrafast bootstrap (ufboot) support values?, 2023. URL <http://www.iqtree.org/doc/Frequently-Asked-Questions>. Last update: Sep 6, 2023.
- Gideon Schwarz. Estimating the dimension of a model. The annals of statistics, pages 461–464, 1978.
- Peter J Shaw and Tim Eden. Neuroblastoma with intracranial involvement: an ensq study. Medical and pediatric oncology, 20(2):149–155, 1992.
- Mark P Simmons and Andrew P Norton. Divergent maximum-likelihood-branch-support values for polytomies. Molecular Phylogenetics and Evolution, 73:87–96, 2014.
- Julien Soubrier, Mike Steel, Michael SY Lee, Clio Der Sarkissian, Stéphane Guindon, Simon YW Ho, and Alan Cooper. The influence of rate heterogeneity among sites on the time dependence of molecular rates. Molecular biology and evolution, 29(11):3345–3358, 2012.
- Gilbert Strang. Linear algebra and its applications. 2012.
- Jeremy G Sumner, Peter D Jarvis, Jesús Fernández-Sánchez, Bodie T Kaine, Michael D Woodhams, and Barbara R Holland. Is the general time-reversible model bad for molecular phylogenetics? Systematic biology, 61(6):1069–1074, 2012.
- Koichiro Tamura and Masatoshi Nei. Estimation of the number of nucleotide substitutions in the control region of mitochondrial dna in humans and chimpanzees. Molecular biology and evolution, 10(3):512–526, 1993.
- Rosa Tarrio, Francisco Rodriguez-Trelles, and Francisco J Ayala. Tree rooting with outgroups when they differ in their nucleotide composition from the ingroup: the drosophila saltans and willistoni groups, a case study. Molecular phylogenetics and evolution, 16(3):344–349, 2000.

- Paula Tataru and Asger Hobolth. Comparison of methods for calculating conditional expectations of sufficient statistics for continuous time markov chains. BMC bioinformatics, 12: 1–11, 2011.
- Simon Tavare. Some probabilistic and statistical problems in the analysis of dna sequences. Lectures on Mathematics in the Life Sciences, 17(2):57–86, 1986.
- Le Sy Vinh and Arndt Von Haeseler. Iqpnni: moving fast through tree space and stopping in time. Molecular biology and evolution, 21(8):1565–1571, 2004.
- RA Vos. Accelerated likelihood surface exploration: the likelihood ratchet. Systematic biology, 52(3):368–373, 2003.
- Marcus Weber. Eigenvalues of non-reversible markov chains—a case study. Mathematics, 2017.
- Gunter Weiss and Arndt von Haeseler. Testing Substitution Models Within a Phylogenetic Tree. Molecular Biology and Evolution, 20(4):572–578, 04 2003. ISSN 0737-4038. doi: 10.1093/molbev/msg073. URL <https://doi.org/10.1093/molbev/msg073>.
- Jia F Weng, Doreen A Thomas, and Iven Mareels. Maximum parsimony, substitution model, and probability phylogenetic trees. Journal of Computational Biology, 18(1):67–80, 2011.
- Ward C Wheeler. Nucleic acid sequence phylogeny and random outgroups. Cladistics, 6(4): 363–367, 1990.
- Simon Whelan, Paul IW de Bakker, and Nick Goldman. Pandit: a database of protein and associated nucleotide domains with inferred trees. Bioinformatics, 19(12):1556–1563, 2003.
- Simon Whelan, Paul IW de Bakker, Emmanuel Quevillon, Nicolas Rodriguez, and Nick Goldman. Pandit: an evolution-centric database of protein and associated nucleotide domains with inferred trees. Nucleic Acids Research, 34(suppl_1):D327–D331, 2006.
- Tom A Williams. Evolution: rooting the eukaryotic tree of life. Current Biology, 24(4): R151–R152, 2014.
- Ziheng Yang. Maximum likelihood phylogenetic estimation from dna sequences with variable rates over sites: approximate methods. Journal of Molecular evolution, 39:306–314, 1994.
- Ziheng Yang. A space-time process model for the evolution of dna sequences. Genetics, 139 (2):993–1005, 1995.
- Yu Yutsumura. Stochastic matrix (markov matrix) and its eigenvalues and eigenvectors. Website, 2024. URL <https://yutsumura.com/stochastic-matrix-markov-matrix-and-its-eigenvalues-and-eigenvectors/>.

A: Appendix

A.1. Manuel:

A.1.1. correlation.py

Correlation diagram between saturation and UFBoot values

usage: correlation.py [-h] [-p] csv

Comparison overview between sample coherence and bootstrap values

This script reads data from a CSV file and generates plots with correlation analysis. It requires a CSV file path as an argument, it will search for the .node.treefile and .treefile automatically. It

needs all three files for creating the UFBoot-saturation diagram with a regression line.

positional arguments:

csv Path to the CSV file

options:

-h, --help show this help message and exit
-p, --print-p-value-output Print p-value output

A.1.2. Subtree Creator.py

A PyQt5 app written with the ete3 (Huerta-Cepas, 2015) library, for pruning trees.

It accepts trees in newick format and saves after cutting between Nodes, all following working subtrees and a remaining subtree. It allows turning the tree into an alignment.

Used main methods from ete3:

```
detach()  
delete(prevent_nondicotomic=True,  
        preserve_branch_length=False)  
search_nodes(name)  
get_children()
```

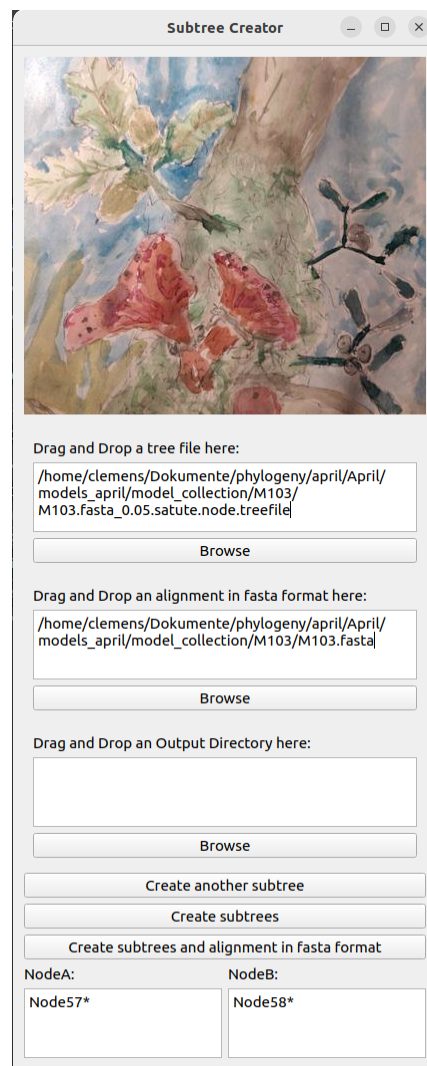


Figure A.1: GUI of the app Subtree creator

A.1.3. nodestyle.py

Creates images in .pdf, format, and detects automatically if .treefile contains bootstrap values. It is written with ete3 library. Allows implementation in satute.

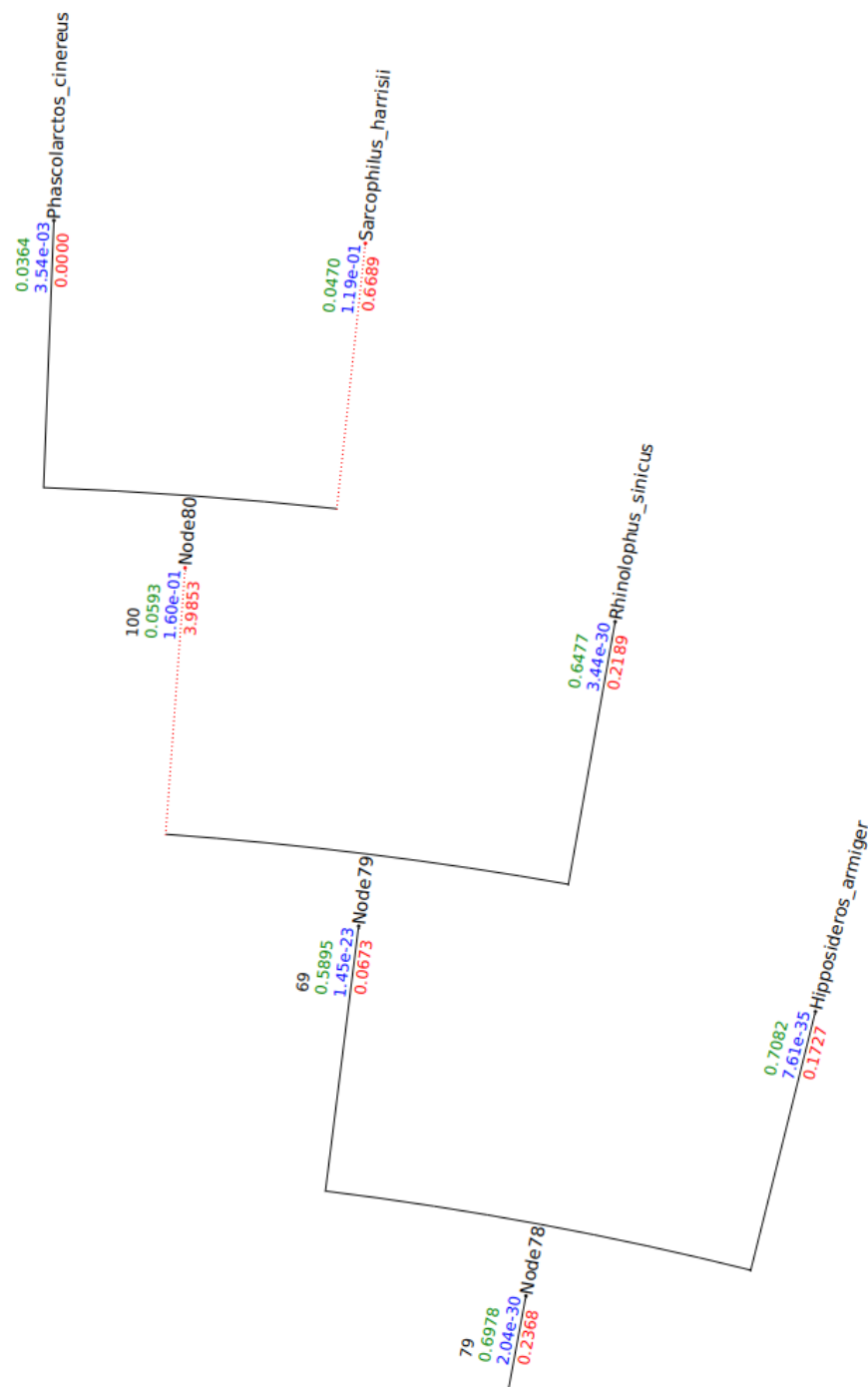


Figure A.2: With nodestyle.py created. red dotted line: saturated branch, red colored number: branch length, blue colored number: p-value, green colored number: delta, black: UF-Boot value -ufboot 10000 -model TVM+F+I+G4; file: ENSG00000119574.ZBTB45.fasta: **saturated** branches between Chinese rufous horseshoe bat and Tasmanian devil.

A.2. GitHub

Subtree Creator.py, nodestyle.py, correlation.py can be found on GitHub at:

https://github.com/clemensheiderer/for_my_thesis

Files can be found on the CIBIV server, fitch, under the directory: /project/satute.

A.3. Statistics

Table A.1: UFBoot values and SD for clade Group 2. When the center branch has a length ranging from 0.02 to 2.0, TVs show the intermingling of clades t00 and t11. However, with a center branch length of 3.0, Group 2 accurately comprises clades t10 and t11. Details can be found in the column CSC of Table 5.9. A noticeable shift in the Order becomes apparent with a center branch length of 3.0. n represents the number of inferred trees that have inner branches of that Order.

Center Branch len		0.02			0.2			0.4			0.8		
Order	n	UFBoot	SD	n	UFBoot	SD	n	UFBoot	SD	n	UFBoot	SD	
1	100	99.39	(±0.59)	100	99.39	(±0.97)	100	98.96	(±1.78)	100	98.94	(±1.26)	
2	100	57.30	(±12.29)	100	58.14	(±12.91)	100	54.91	(±11.57)	100	58.45	(±10.71)	
3	100	75.39	(±13.20)	100	74.72	(±13.77)	100	73.16	(±10.56)	100	75.70	(±13.02)	
4	100	88.80	(±6.34)	100	88.72	(±6.19)	100	88.10	(±5.00)	100	88.79	(±6.35)	
5	100	92.82	(±5.57)	100	93.14	(±5.49)	100	92.54	(±4.68)	100	92.60	(±5.63)	
6	100	96.62	(±2.54)	100	96.64	(±2.62)	100	96.52	(±2.46)	100	96.32	(±2.60)	
7	99	98.07	(±2.13)	100	98.13	(±1.97)	97	98.15	(±1.86)	98	98.02	(±2.21)	
8							1	100.00	(±0.00)				
Center Branch len		1.2			2.0			3.0			4.0		
Order	n	UFBoot	SD	n	UFBoot	SD	n	UFBoot	SD	n	UFBoot	SD	
1	100	97.62	(±6.44)	100	97.29	(±2.57)	100	92.77	(±8.25)	100	88.50	(±13.46)	
2	100	52.29	(±12.60)	100	50.30	(±12.25)	100	52.32	(±13.57)	100	50.84	(±13.89)	
3	100	73.19	(±13.07)	100	72.62	(±13.10)	100	70.76	(±13.77)	100	70.79	(±14.94)	
4	100	88.49	(±6.05)	100	87.82	(±6.26)	100	86.51	(±6.15)	100	86.16	(±7.72)	
5	100	92.61	(±5.20)	100	91.96	(±6.24)	100	91.46	(±5.37)	99	91.33	(±7.57)	
6	99	96.58	(±2.46)	100	96.21	(±2.79)	100	95.42	(±5.36)	98	95.70	(±3.71)	
7	98	98.10	(±2.50)	98	98.07	(±1.89)	97	97.28	(±5.67)	96	95.31	(±11.19)	
8	1	87.00	(±0.00)	1	98.00	(±0.00)	5	79.00	(±24.36)	9	90.33	(±11.70)	
9	1	96.80	(±0.00)				5	70.56	(±15.64)	7	69.39	(±24.25)	
10	1	99.50	(±0.00)				5	89.96	(±5.51)	6	81.17	(±11.14)	
11							4	90.38	(±4.08)	5	92.14	(±4.81)	
12							4	95.15	(±2.03)	4	94.62	(±4.57)	
13							4	97.72	(±1.00)	4	97.15	(±2.49)	
14							1	99.00	(±0.00)	3	97.40	(±2.41)	
Center Branch len		5.0											
Order	n	UFBoot	SD										
1	100	78.96	(±20.28)										
2	100	50.57	(±14.06)										
3	100	70.81	(±14.47)										
4	100	85.28	(±7.49)										
5	100	90.63	(±6.78)										
6	100	94.21	(±7.57)										
7	100	94.14	(±12.66)										
8	21	78.77	(±25.91)										
9	19	71.67	(±24.33)										
10	16	86.81	(±13.02)										
11	14	93.07	(±6.91)										
12	12	94.59	(±3.93)										
13	11	97.85	(±1.78)										
14	6	98.42	(±1.34)										

Table A.2: UFBoot values and SD for Clade t1 are shown. Clade t1 consists of the two 16-taxa clades t10 and t11 and is located opposite to clade t0, with the center branch in between. n stands for the number of inferred trees that have branches of this Order.

JB len	0.01			0.1			0.2			0.4		
Order	n	UFBoot	SD	n	UFBoot	SD	n	UFBoot	SD	n	UFBoot	SD
1	100	39.94	(±19.75)	100	42.23	(±18.61)	100	42.76	(±19.89)	100	48.22	(±18.53)
2	100	53.12	(±22.91)	100	58.98	(±22.96)	100	56.92	(±22.71)	100	58.78	(±20.77)
3	100	78.23	(±11.57)	100	79.72	(±11.06)	100	78.31	(±11.75)	100	78.58	(±10.81)
4	100	82.80	(±11.89)	100	82.21	(±11.47)	100	80.87	(±12.41)	100	78.01	(±11.67)
5	100	82.80	(±11.89)	99	92.11	(±4.89)	96	90.53	(±5.48)	97	88.88	(±6.40)
6	100	91.87	(±5.90)	98	92.82	(±4.65)	95	90.37	(±5.59)	95	87.98	(±6.73)
7	100	93.83	(±5.08)	97	97.49	(±2.22)	94	96.16	(±6.30)	85	92.21	(±11.28)
8	99	97.05	(±2.34)	89	98.55	(±1.47)	84	97.98	(±3.59)	76	92.38	(±8.81)
9	97	98.36	(±1.66)	1	98.80	(±0.00)	17	97.91	(±2.39)	49	96.19	(±6.28)
10							5	99.00	(±1.14)	33	98.43	(±2.37)
11							4	99.75	(±0.38)	16	99.17	(±0.87)
12										4	99.30	(±0.48)
13										2	99.80	(±0.00)

JB len	0.6			1.0			1.5			2.0		
Order	n	UFBoot	SD	n	UFBoot	SD	n	UFBoot	SD	n	UFBoot	SD
1	100	49.90	(±20.25)	100	63.90	(±19.76)	100	70.06	(±23.16)	100	74.56	(±20.46)
2	100	57.79	(±18.51)	100	59.87	(±16.81)	100	57.52	(±18.17)	100	56.46	(±17.23)
3	100	77.54	(±11.04)	100	75.67	(±14.05)	100	74.30	(±13.24)	100	74.95	(±12.75)
4	100	79.24	(±12.81)	100	82.17	(±11.49)	100	84.91	(±9.32)	100	85.35	(±9.93)
5	99	88.88	(±6.40)	100	89.85	(±6.75)	100	91.06	(±5.87)	100	91.65	(±5.87)
6	96	89.42	(±8.68)	97	91.68	(±7.66)	100	92.16	(±8.91)	100	93.56	(±6.82)
7	91	82.05	(±20.68)	91	85.99	(±20.24)	99	90.49	(±17.37)	96	92.46	(±15.02)
8	75	86.34	(±15.17)	50	82.63	(±21.17)	40	85.29	(±18.05)	28	80.77	(±22.63)
9	66	84.27	(±16.19)	47	78.71	(±19.00)	39	79.33	(±19.09)	27	73.64	(±26.40)
10	54	90.55	(±13.09)	45	88.34	(±11.20)	36	89.39	(±11.73)	24	82.21	(±21.30)
11	45	94.44	(±7.17)	39	92.57	(±7.28)	30	93.58	(±6.13)	24	89.72	(±11.50)
12	22	94.63	(±6.10)	28	94.05	(±4.78)	24	96.69	(±3.67)	15	89.91	(±9.92)
13	15	97.23	(±3.42)	22	97.06	(±2.86)	21	98.19	(±2.42)	13	94.89	(±5.57)
14	8	98.22	(±2.21)	10	98.24	(±2.18)	9	99.33	(±0.63)	5	97.90	(±2.81)

JB len	2.5			3.0		
Order	n	UFBoot	SD	n	UFBoot	SD
1	100	74.39	(±21.60)	100	77.69	(±17.81)
2	100	53.04	(±14.81)	100	55.97	(±17.51)
3	100	74.38	(±13.59)	100	71.86	(±15.09)
4	100	85.33	(±8.10)	100	84.28	(±10.58)
5	100	90.93	(±6.45)	100	90.30	(±9.06)
6	98	93.76	(±7.59)	100	93.10	(±10.84)
7	97	93.19	(±13.59)	99	93.96	(±13.51)
8	25	80.83	(±25.66)	30	87.27	(±19.57)
9	25	73.71	(±20.09)	21	69.21	(±25.14)
10	21	80.94	(±17.44)	18	79.22	(±18.20)
11	19	89.21	(±9.61)	18	89.64	(±10.92)
12	18	93.03	(±6.93)	15	93.00	(±8.64)
13	17	96.81	(±3.56)	14	97.18	(±4.11)
14	10	98.22	(±2.26)	8	96.64	(±5.61)
15				1	96.50	(±0.00)

1 file [74] has this combination:

- 1 pairwise branch in Group 1 or 2 with Order: 12
- 1 pairwise branch in Group 3 or 4 with Order: 4
- 1 pairwise branch in Group 3 or 4 with Order: 5
- 2 pairwise branches in Group 1 or 2 with Order: 11
- 2 pairwise branches in Group 1 or 2 with Order: 12
- 2 pairwise branches in Group 1 or 2 with Order: 14
- 2 pairwise branches in Group 3 or 4 with Order: 6
- 2 pairwise branches in Group 3 or 4 with Order: 7
- 2 pairwise branches in Group 3 or 4 with Order: 8
- 2 pairwise branches in Group 3 or 4 with Order: 9
- 4 pairwise branches in Group 1 or 2 with Order: 13
- 4 pairwise branches in Group 1 or 2 with Order: 15
- 4 pairwise branches in Group 3 or 4 with Order: 7
- Single external branch in Group 1 or 2 with Order: 10
- Single external branch in Group 1 or 2 with Order: 9
- Single external branch in Group 3 or 4 with Order: 1
- Single external branch in Group 3 or 4 with Order: 2
- Single external branch in Group 3 or 4 with Order: 2
- Single external branch in Group 3 or 4 with Order: 3

2 files [49, 79] have this combination:

- 1 pairwise branch in Group 1 or 2 with Order: 6
- 1 pairwise branch in Group 1 or 2 with Order: 6
- 1 pairwise branch in Group 3 or 4 with Order: 5
- 2 pairwise branches in Group 1 or 2 with Order: 8
- 2 pairwise branches in Group 1 or 2 with Order: 8
- 2 pairwise branches in Group 3 or 4 with Order: 7
- 4 pairwise branches in Group 1 or 2 with Order: 9
- 4 pairwise branches in Group 1 or 2 with Order: 9
- 4 pairwise branches in Group 3 or 4 with Order: 8
- 8 pairwise branches in Group 3 or 4 with Order: 5
- Single external branch in Group 1 or 2 with Order: 3
- Single external branch in Group 1 or 2 with Order: 3
- Single external branch in Group 1 or 2 with Order: 4
- Single external branch in Group 1 or 2 with Order: 4
- Single external branch in Group 3 or 4 with Order: 2
- Single external branch in Group 3 or 4 with Order: 3

Figure A.3: Tree variance details for three inferred trees with a JB length of 0.6 in the original tree model, derived from alignments [74] and [49, 79]. All three inferred ML trees (including two TVs) originate from the same true tree with a center branch length of 3.0. TV of alignment [74] has in Group 1 or 2 both 16-taxa clades interlinked with a long branch. In the second TV, the 4 JB's are correctly connected with the center branch.

Table A.3: RF Distances between the three different left clade topologies

Length	True vs. Subalignment RF	True vs. Split ML RF	Full vs. Subalignment RF	Numbers (Left Clade)
0.3	0	0	0	100
0.6	0	0	0	100
0.9	0.02	0.02	0.04	100
1.2	0.06	0.02	0.04	100
1.5	0.25	0.08	0.25	100
1.8	1.44	0.24	1.42	100
2.1	3.55	0.87	3.26	97
2.4	6.19	2.42	5.47	96
2.7	8.50	5.09	7.38	84
3.0	9.44	7.96	7.44	75

Table A.4: RF Distances between the three different left clade topologies for 10,000 sites

Length	True vs. Subalignment RF	True vs. Split ML RF	Full vs. Subalignment RF	Numbers (Left Clade)
0.3	0	0	0	100
0.6	0	0	0	100
0.9	0	0	0	100
1.2	0	0	0	100
1.5	0	0	0	100
1.8	0	0	0	100
2.1	0	0	0	100
2.4	0	0	0	100
2.7	1.12	0	1.12	100
3.0	0.16	0	0.16	100

Table A.5: Comparison of saturated branches percentage between model settings (GTR+F+G4 and best-fit+G4) for each individual alignment in the PANDIT dataset. Mean differences, standard deviations (SD), T-statistics, and p-values represent averages across all 79 alignments. RC 1 had only 1 alignment with DIFF = 0. There was insufficient data to calculate outliers. RC 2 had no outliers.

RC outlier		RC outlier		RC 1 2			
1 -		2 -		3 outlier PF01734 PF02811 DIFF 2.91 3.053			
		RC 1 2 3					
		4 outlier PF01074 PF00632 PF00291 DIFF 8.08 8.511 7.938					
RC	MDIFF (%)	SD (%)	t-value (w)	p-value (w)	t-value (w/o)	p-value (w/o)	n_a
1	0	0	-	-	-	-	1
2	0.141	0.228	2.220	0.035	-	-	14
3	0.465	0.758	4.989	1.87×10^{-6}	4.965	2.17×10^{-6}	67
4	1.111	1.95	5.017	1.41×10^{-6}	5.195	6.57×10^{-7}	79
All	0.407	2.799	6.328	8.35×10^{-10}	6.670	1.17×10^{-10}	

Table A.6: Average saturated branches as a percentage for PANDIT GTR+F+G4 and PANDIT best-fit+G4 models, 79 alignments, the model identified as the best fit according to EvONAPS, with pre-generated trees retrieved from the EvONAPS database.

Rate Category	PANDIT GTR+G4		PANDIT best-fit+G4	
	Mean SB (%)	SD (%)	Mean SB (%)	SD (%)
1	0.0064	0.0571	0.0064	0.0571
2	0.0999	0.2674	0.0999	0.2408
3	4.1945	4.4069	4.0045	4.5031
4	50.5074	21.7248	50.1382	21.5580
Overall	13.7020	24.0321	13.5623	23.8763

Table A.7: Comparison of saturated branches percentage between model settings (GTR+F+G4 and best-fit+G4) for each individual alignment in the TreeBASE dataset. Mean differences (MDIFF), standard deviations (SD), t-values, and p-values represent averages across all 61 alignments.

			RC	Outliers	n_a	DIFF (%)
			1	M5225	0.49	
			2	M39175	0.78	
			3	M39175	38.76	
			4	M7224	16.26	

RC	MDIFF (%)	SD (%)	t-value (w)	p-value (w)	t-value (w/o)	p-value (w/o)	n_a
1	0.058	0.144	1.279	0.216	1.000	0.331	11
2	0.025	0.121	1.690	0.098	1.426	0.161	24
3	0.796	4.993	1.250	0.216	1.797	0.077	32
4	0.798	2.485	2.524	0.013	2.874	0.005	53
All	0.407	2.799	2.294	0.023	3.384	0.0008	

Table A.8: Average percentage of saturated branches for TreeBASE GTR+F+G4 and TreeBASE best-fit+G4 models, 61 alignments, the model identified as the best fit according to EvONAPS, with 61 pre-generated trees retrieved from the EvONAPS database.

Rate Category	TreeBASE GTR+G4		TreeBASE best-fit+G4	
	Mean SB (%)	SD (%)	Mean SB (%)	SD (%)
1	0.3861	0.9655	0.3757	0.9442
2	1.5631	4.7074	1.5525	4.7097
3	7.7118	17.9092	7.0079	17.2109
4	21.7914	31.3075	21.1999	30.8615
Overall	7.8631	19.9903	7.5340	19.5653

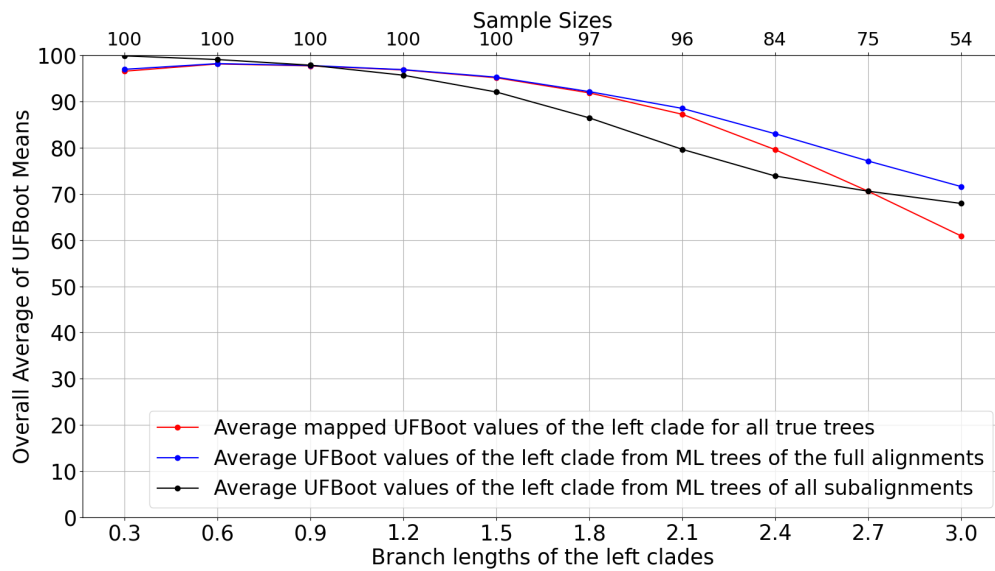


Figure A.4: The average UFBoot values of all branches from the left clade trees across all obtained samples of alignments of 1,000 sites.

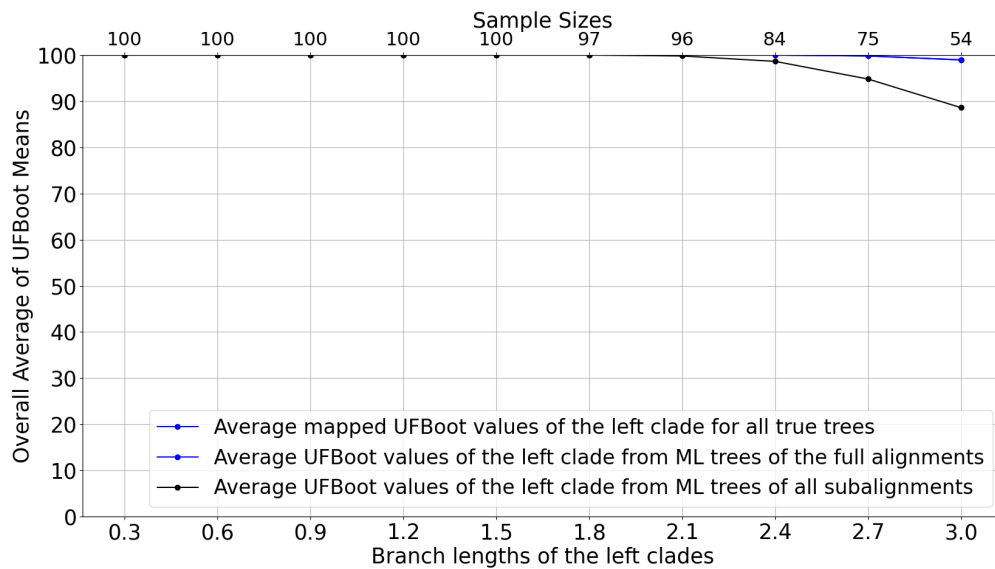


Figure A.5: The average UFBoot values of all branches from the left clade trees across all obtained samples of alignments of 10,000 sites.

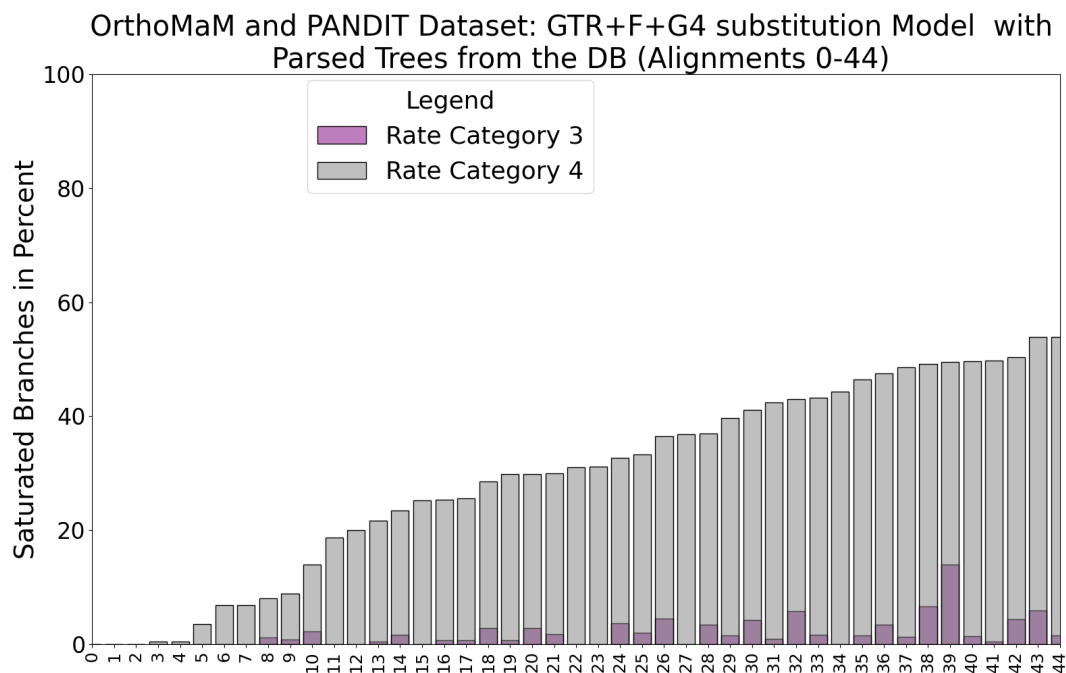


Figure A.6: Alignment number with naming and the percentages for Rate3 and Rate4 with the chosen model GTR+G4. OrthoMaM dataset are alignment numbers 0-4.

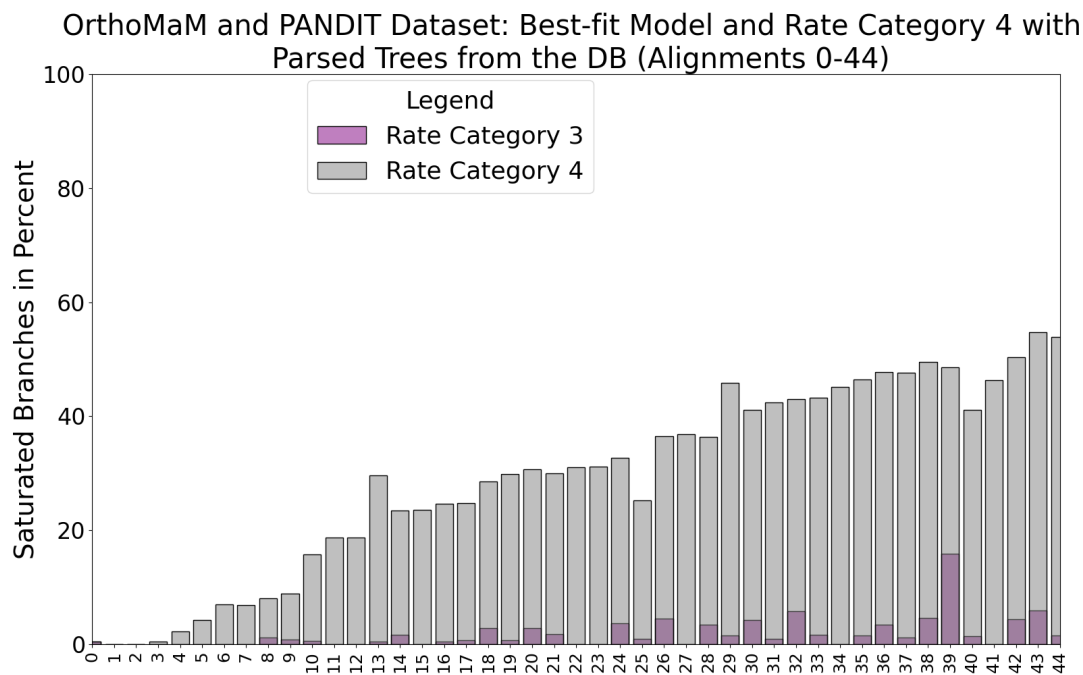


Figure A.7: Alignment number with naming and the percentages for Rate3 and Rate4 with the best-fit model. OrthoMaM dataset are alignment numbers 0-4.

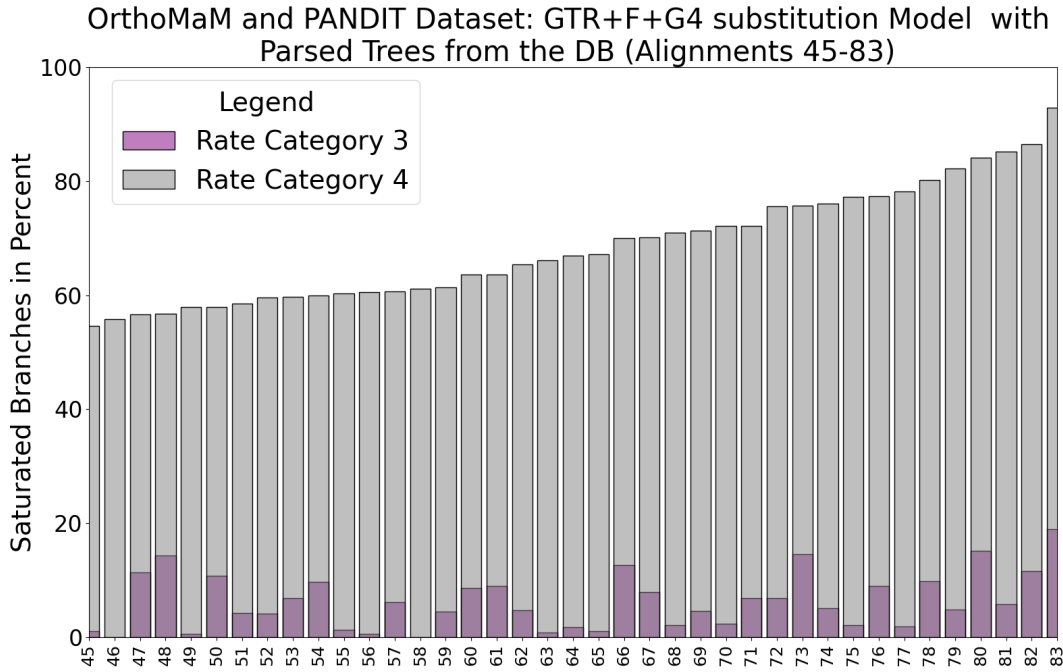


Figure A.8: Alignment number with naming and the percentages for Rate3 and Rate4 with the chosen model GTR+G4. OrthoMaM dataset are alignment numbers 0-4.

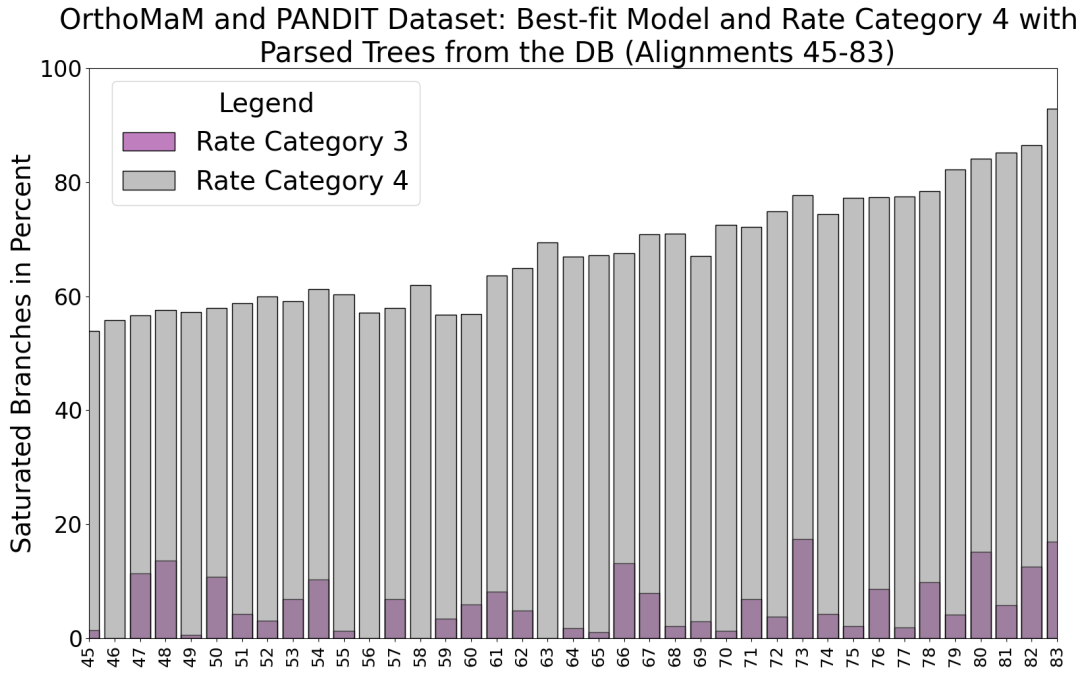


Figure A.9: Alignment number with naming and the percentages for Rate3 and Rate4 with the best-fit model. OrthoMaM dataset are alignment numbers 0-4.

Table A.9: OrthoMaM and PANDIT dataset with the best-fit model with parsed trees from the db

Alignment Number	Alignment	Category Rate	Best fit Model	Saturated Branches in Percent	Number of Saturated Branches	Number of Saturated Inner Branches	Number of Saturated Exterior Branches	Number of Branches
0	ENSG0000008-7460.GNAS	3	TIM2+F+I+G4	0.49	1	1	0	205
		4		0.49	1	1	0	
1	ENSG00000138-316.ADAMTS14	3	TIM2+F+G4	0.0	0	0	0	219
		4		0.0	0	0	0	
2	ENSG0000016-0714.UBE2Q1	3	TIM3e+G4	0.0	0	0	0	217
		4		0.0	0	0	0	
3	ENSG000001-04870.FCGRT	3	TPM2u+F+I+G4	0.0	0	0	0	205
		4		0.49	1	1	0	
4	ENSG000001-19574.ZBTB45	3	TVM+F+I+G4	0.0	0	0	0	179
		4		2.23	4	2	2	
5	PF00478	3	SYM+G4	0.00	0	0	0	165
		4		4.24	7	6	1	
6	PF00141	3	TVM+F+G4	0.00	0	0	0	583
		4		7.03	41	28	13	
7	PF07733	3	TVM+F+G4	0.00	0	0	0	173
		4		6.94	12	6	6	
8	PF00109	3	GTR+F+G4	1.20	4	3	1	333
		4		8.11	27	22	5	
9	PF01134	3	GTR+F+G4	0.84	2	2	0	237
		4		8.86	21	15	6	
10	PF03936	3	SYM+G4	0.58	1	1	0	171
		4		15.79	27	17	10	
11	PF00120	3	GTR+F+G4	0.00	0	0	0	203
		4		18.72	38	29	9	
12	PF01604	3	TIM3+F+G4	0.00	0	0	0	245
		4		18.78	46	35	11	
13	PF00291	3	TVMe+G4	0.53	1	1	0	189
		4		29.63	56	42	14	
14	PF00310	3	GTR+F+G4	1.74	2	2	0	115
		4		23.48	27	21	6	
15	PF00483	3	TVMe+G4	0.00	0	0	0	127
		4		23.62	30	21	9	
16	PF00009	3	TVMe+G4	0.48	2	1	1	413
		4		24.70	102	70	32	
17	PF00890	3	TVM+F+G4	0.80	1	1	0	125
		4		24.80	31	21	10	
18	PF00685	3	TVMe+G4	2.86	3	3	0	105
		4		28.57	30	20	10	
19	PF01094	3	TVM+F+G4	0.80	2	1	1	251
		4		29.88	75	61	14	
20	PF01501	3	TVM+F+G4	2.92	4	4	0	137
		4		30.66	42	37	5	
21	PF00704	3	GTR+F+G4	1.83	6	5	1	327
		4		29.97	98	70	28	
22	PF00100	3	GTR+F+G4	0.00	0	0	0	161
		4		31.06	50	36	14	
23	PF01055	3	SYM+G4	0.00	0	0	0	215
		4		31.16	67	54	13	
24	PF03435	3	GTR+F+G4	3.74	4	2	2	107
		4		32.71	35	27	8	
25	PF01074	3	TVMe+G4	1.01	1	1	0	99
		4		25.25	25	19	6	
26	PF00004	3	GTR+F+G4	4.59	22	19	3	479
		4		36.53	175	113	62	
27	PF00122	3	GTR+F+G4	0.00	0	0	0	201
		4		36.82	74	55	19	
28	PF03006	3	TVM+F+G4	3.47	6	3	3	173
		4		36.42	63	41	22	
29	PF05343	3	TVMe+G4	1.53	2	1	1	131
		4		45.80	60	44	16	
30	PF00136	3	GTR+F+G4	4.26	6	6	0	141
		4		41.13	58	40	18	
31	PF07394	3	GTR+F+G4	1.01	1	1	0	99
		4		42.42	42	28	14	
32	PF01968	3	TVM+F+G4	5.79	7	7	0	121
		4		42.98	52	33	19	
33	PF00215	3	GTR+F+G4	1.75	3	3	0	171
		4		43.27	74	49	25	
34	PF00413	3	SYM+G4	0.00	0	0	0	133
		4		45.11	60	40	20	

Table A.10: OrthoMaM and PANDIT dataset with the best-fit model with parsed trees from the db

Alignment Number	Alignment	Category Rate	Best fit Model	Saturated Branches in Percent	Number of Saturated Branches	Number of Saturated Inner Branches	Number of Saturated Exterior Branches	Number of Branches
35	PF00001	3	GTR+F+G4	1.60	2	1	1	125
		4		46.40	58	34	24	
36	PF01590	3	TVM+F+G4	3.45	31	19	12	899
		4		47.72	429	258	171	
37	PF04055	3	TVM+F+G4	1.25	16	8	8	1281
		4		47.62	610	363	247	
38	PF01547	3	TVM+F+G4	4.65	14	10	4	301
		4		49.50	149	100	49	
39	PF01697	3	TVM+F+G4	15.89	17	12	5	107
		4		48.60	52	31	21	
40	PF00632	3	SYM+G4	1.42	2	0	2	141
		4		41.13	58	38	20	
41	PF00696	3	SYM+G4	0.0	0	0	0	205
		4		46.34	95	71	24	
42	PF00288	3	GTR+F+G4	4.46	22	13	9	493
		4		50.30	248	165	83	
43	PF01757	3	TVM+F+G4	5.98	7	6	1	117
		4		54.70	64	41	23	
44	PF04043	3	TVM+F+G4	1.62	4	4	0	247
		4		53.85	133	96	37	
45	PF00246	3	TVM+G4	1.49	4	4	0	269
		4		53.90	145	93	52	
46	PF00999	3	GTR+F+G4	0.0	0	0	0	163
		4		55.83	91	57	34	
47	PF01593	3	GTR+F+G4	11.43	20	13	7	175
		4		56.57	99	70	29	
48	PF01979	3	TVM+G4	13.6	17	13	4	125
		4		57.6	72	46	26	
49	PF04389	3	TVM+F+G4	0.69	1	1	0	145
		4		57.24	83	51	32	
50	PF04427	3	TVM+F+G4	10.83	17	12	5	157
		4		57.96	91	56	35	
51	PF00270	3	TVM+F+G4	4.33	17	13	4	393
		4		58.78	231	142	89	
52	PF00078	3	TVM+F+G4	3.14	9	7	2	287
		4		59.93	172	119	53	
53	PF00350	3	TVM+F+G4	6.92	11	11	0	159
		4		59.12	94	60	34	
54	PF00149	3	TVM+G4	10.35	68	48	20	657
		4		61.19	402	264	138	
55	PF00102	3	GTR+F+G4	1.37	3	1	2	219
		4		60.27	132	88	44	
56	PF01656	3	TVM+F+G4	0.0	0	0	0	147
		4		57.14	84	51	33	
57	PF01103	3	SYM+G4	6.90	10	8	2	145
		4		57.93	84	55	29	
58	PF00587	3	TVM+G4	0.0	0	0	0	229
		4		62.01	142	99	43	
59	PF00328	3	TVM+G4	3.52	7	5	2	199
		4		56.78	113	76	37	
60	PF00520	3	SYM+G4	5.96	37	31	6	621
		4		56.84	353	222	131	
61	PF02378	3	TVM+F+G4	8.26	10	8	2	121
		4		63.64	77	52	25	
62	PF02518	3	TVM+F+G4	4.87	63	30	33	1293
		4		64.97	840	461	379	
63	PF00459	3	TVM+G4	0.0	0	0	0	121
		4		69.42	84	53	31	
64	PF04983	3	TVM+F+G4	1.86	4	3	1	215
		4		66.98	144	88	56	
65	PF00753	3	GTR+F+G4	1.08	7	2	5	649
		4		67.18	436	237	199	
66	PF00856	3	TVM+F+G4	13.20	26	18	8	197
		4		67.51	133	85	48	
67	PF00664	3	TVM+F+G4	8.03	11	7	4	137
		4		70.80	97	60	37	
68	PF00702	3	GTR+F+G4	2.15	10	7	3	465
		4		70.97	330	185	145	

Table A.11: OrthoMaM and PANDIT dataset with the best-fit model with parsed trees from the db

Alignment Number	Alignment	Category Rate	Best fit Model	Saturated Branches in Percent	Number of Saturated Branches	Number of Saturated Inner Branches	Number of Saturated Exterior Branches	Number of Branches
69	PF00656	3	TVMe+G4	2.95	7	6	1	237
		4		67.09	159	99	60	
70	PF07714	3	SYM+G4	1.37	4	4	0	291
		4		72.51	211	130	81	
71	PF03717	3	GTR+F+G4	6.96	8	7	1	115
		4		72.17	83	50	33	
72	PF02811	3	TVMe+G4	3.82	5	4	1	131
		4		74.81	98	60	38	
73	PF01734	3	TVMe+G4	17.48	18	13	5	103
		4		77.67	80	44	36	
74	PF01223	3	SYM+G4	4.27	5	4	1	117
		4		74.36	87	56	31	
75	PF00593	3	GTR+F+G4	2.16	29	18	11	1345
		4		77.25	1039	622	417	
76	PF07717	3	TVM+F+G4	8.64	21	9	12	243
		4		77.37	188	114	74	
77	PF00294	3	TVMe+G4	1.99	3	3	0	151
		4		77.48	117	69	48	
78	PF04851	3	TPM3u+F+G4	9.91	11	7	4	111
		4		78.38	87	52	35	
79	PF03372	3	TVMe+G4	4.23	14	10	4	331
		4		82.18	272	156	116	
80	PF07596	3	GTR+F+G4	15.17	22	15	7	145
		4		84.14	122	68	54	
81	PF00443	3	GTR+F+G4	5.79	7	3	4	121
		4		85.12	103	56	47	
82	PF01040	3	TPM3u+F+G4	12.62	13	10	3	103
		4		86.41	89	50	39	
83	PF03787	3	TVM+F+G4	16.99	26	18	8	153
		4		92.81	142	75	67	

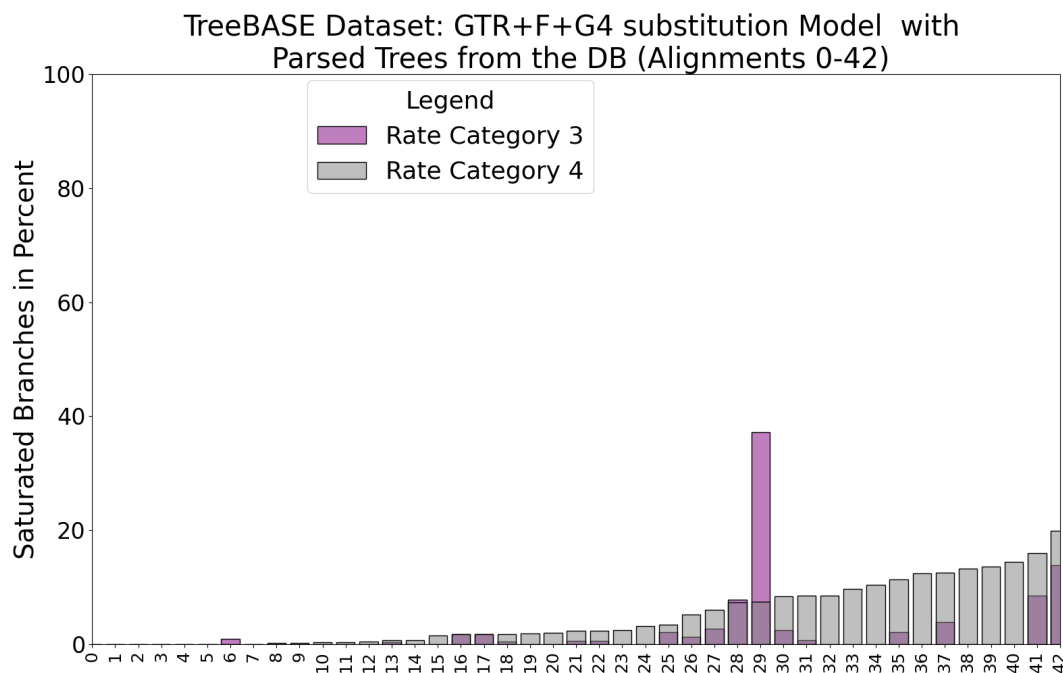


Figure A.10: Folder number with naming and the percentages for Rate3 and Rate4 with the chosen model GTR+G4. Treebase dataset

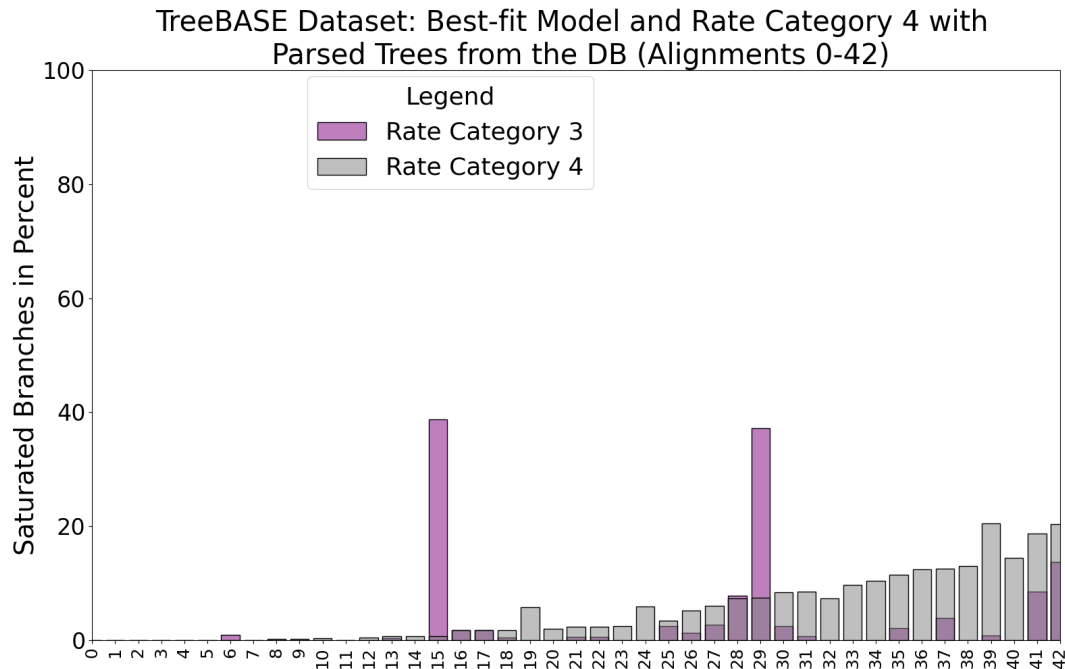


Figure A.11: Folder number with naming and the percentages for Rate3 and Rate4 with the best-fit model. Treebase dataset

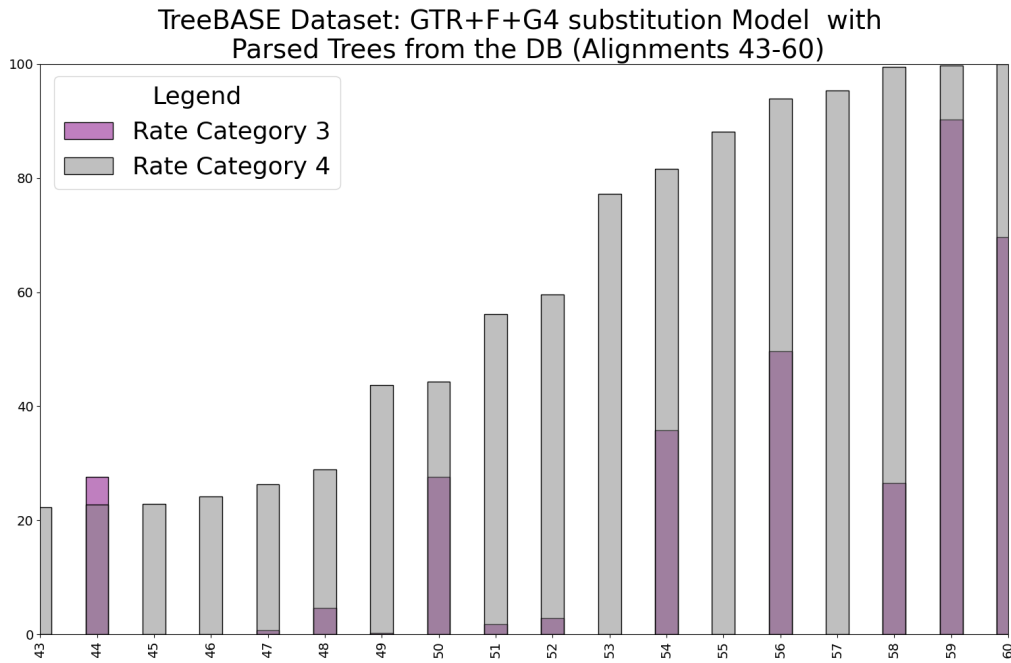


Figure A.12: Folder number with naming and the percentages for Rate3 and Rate4 with the chosen model GTR+G4 for the Treebase dataset.

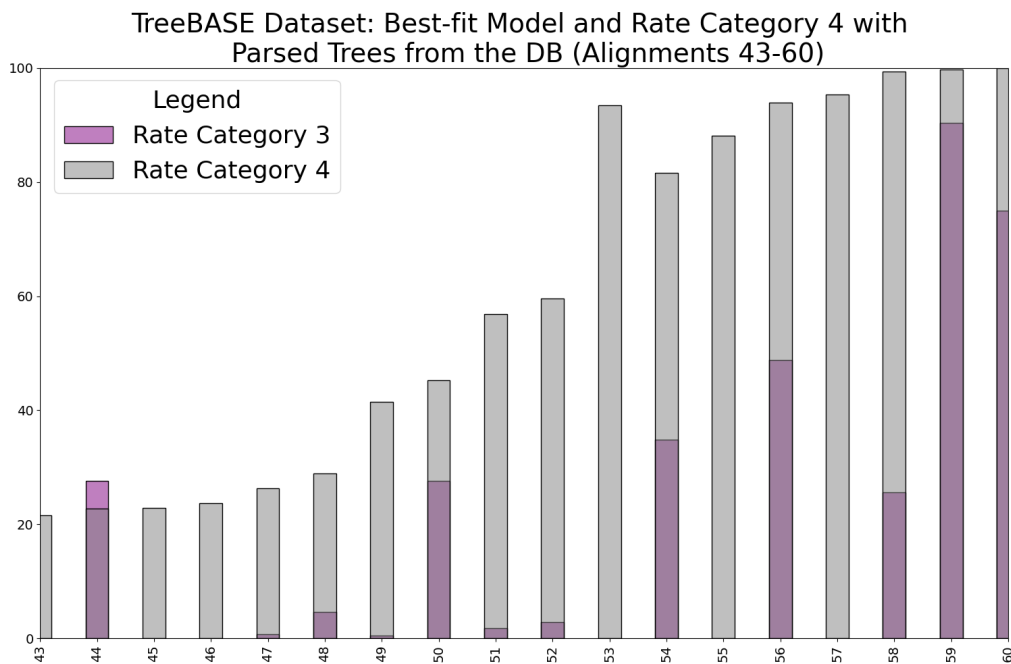


Figure A.13: Folder number with naming and the percentages for Rate3 and Rate4 with the best-fit model for the Treebase dataset.

Table A.12: Treebase dataset with the best-fit model with parsed trees from the db

Alignment Number	Alignment	Category	Rate	Best-fit Model	Saturated Branches in Percent	Number of Saturated Branches	Number of Saturated Inner Branches	Number of Saturated Exterior Branches	Number of Branches
0	M42533	3		GTR+F+G4	0.00	0	0	0	247
		4			0.00	0	0	0	
1	M4151	3		GTR+F+G4	0.00	0	0	0	1471
		4			0.00	0	0	0	
2	M24980	3		TIM2+F+G4	0.00	0	0	0	131
		4			0.00	0	0	0	
3	M10947	3		TVM+F+G4	0.00	0	0	0	305
		4			0.00	0	0	0	
4	M15742	3		GTR+F+I+G4	0.00	0	0	0	217
		4			0.00	0	0	0	
5	M48838	3		TIM3+F+G4	0.00	0	0	0	409
		4			0.00	0	0	0	
6	M2942	3		HKY+F+I+G4	1.03	1	1	0	97
		4			0.00	0	0	0	
7	M20140	3		GTR+F+G4	0.00	0	0	0	183
		4			0.00	0	0	0	
8	M17943	3		TIM2+F+G4	0.00	0	0	0	887
		4			0.23	2	1	1	
9	M10950	3		TVM+F+G4	0.00	0	0	0	305
		4			0.33	1	1	0	
10	M3208	3		TIM2+F+G4	0.00	0	0	0	249
		4			0.40	1	1	0	
11	M48832	3		TN+F+G4	0.00	0	0	0	233
		4			0.00	0	0	0	
12	M103	3		SYM+G4	0.00	0	0	0	219
		4			0.46	1	1	0	
13	M394	3		TN+F+G4	0.35	1	1	0	285
		4			0.70	2	2	0	
14	M1117	3		TIM2+F+G4	0.00	0	0	0	275
		4			0.73	2	1	1	
15	M39175	3		TN+F+I+G4	38.76	50	25	25	129
		4			0.78	1	1	0	
16	M44848	3		TIM3+F+G4	1.80	3	1	2	167
		4			1.80	3	1	2	
17	M44847	3		TIM2+F+G4	1.82	3	1	2	165
		4			1.82	3	1	2	
18	M19159	3		TVM+F+G4	0.46	1	1	0	219
		4			1.83	4	4	0	
19	M24163	3		SYM+I+G4	0.00	0	0	0	153
		4			5.88	9	6	3	
20	M13052	3		TN+F+I+G4	0.00	0	0	0	147
		4			2.04	3	3	0	
21	M30163	3		GTR+F+G4	0.61	1	1	0	165
		4			2.42	4	2	2	
22	M24505	3		TVM+G4	0.61	1	0	1	165
		4			2.42	4	3	1	
23	M2471	3		TIM2+F+G4	0.00	0	0	0	121
		4			2.48	3	2	1	
24	M3858	3		TVM+F+I+G4	0.00	0	0	0	183
		4			6.01	11	7	4	
25	M24201	3		TIM2+F+G4	2.58	6	2	4	233
		4			3.43	8	3	5	
26	M24362	3		GTR+F+G4	1.37	6	1	5	439
		4			5.24	23	9	14	
27	M18340	3		TIM2+F+G4	2.72	4	0	4	147
		4			6.12	9	2	7	
28	M25311	3		TIM2+F+I+G4	0.00	0	0	0	131
		4			14.50	19	14	5	
29	M2095	3		GTR+F+G4	7.91	14	11	3	177
		4			7.34	13	11	2	
30	M15517	3		TIM3+F+G4	37.27	60	29	31	161
		4			7.45	12	7	5	

Table A.13: Treebase dataset with the best-fit model with parsed trees from the db

Alignment Number	Alignment	Category	Rate	Best fit Model	Saturated Branches in Percent	Number of Saturated Branches	Number of Saturated Inner Branches	Number of Saturated Exterior Branches	Number of Branches
31	M24363	3		GTR+F+G4	2.58	12	6	6	465
		4			8.39	39	17	22	
32	M1758	3		GTR+F+G4	0.78	1	1	0	129
		4			8.53	11	10	1	
33	M1470	3		TN+F+G4	0.00	0	0	0	163
		4			7.36	12	6	6	
34	M32842	3		GTR+F+G4	0.00	0	0	0	123
		4			9.76	12	12	0	
35	M19895	3		GTR+F+G4	0.00	0	0	0	115
		4			10.43	12	11	1	
36	M15509	3		TIM+F+G4	2.20	15	9	6	683
		4			11.57	79	43	36	
37	M4763	3		GTR+F+G4	0.00	0	0	0	217
		4			12.44	27	15	12	
38	M37476	3		GTR+F+G4	4.00	7	4	3	175
		4			12.57	22	17	5	
39	M39363	3		TIM3e+G4	0.00	0	0	0	307
		4			13.03	40	8	32	
40	M4510	3		TIM2+F+I+G4	0.85	1	0	1	117
		4			20.51	24	15	9	
41	M15511	3		TVM+F+G4	8.56	16	11	5	187
		4			18.72	35	21	14	
42	M49218	3		TIM+F+G4	13.72	86	48	38	627
		4			20.41	128	81	47	
43	M18310	3		TIM3+F+G4	0.00	0	0	0	139
		4			21.58	30	16	14	
44	M15514	3		GTR+F+G4	27.64	34	19	15	123
		4			22.76	28	17	11	
45	M20488	3		TIM3+F+G4	0.00	0	0	0	105
		4			22.86	24	16	8	
46	M1224	3		TIM3+F+G4	0.00	0	0	0	417
		4			23.74	99	27	72	
47	M31836	3		GTR+F+G4	0.72	8	6	2	1115
		4			26.28	293	230	63	
48	M44232	3		TVM+F+G4	4.67	5	2	3	107
		4			28.97	31	19	12	
49	M4794	3		TIM+F+G4	0.49	2	2	0	407
		4			41.52	169	116	53	
50	M5225	3		TIM3+F+G4	27.59	56	31	25	203
		4			45.32	92	48	44	
51	M18331	3		TIM2+F+G4	1.86	8	3	5	431
		4			56.84	245	103	142	
52	M18309	3		TIM2+F+G4	2.88	19	0	19	659
		4			59.64	393	186	207	
53	M7224	3		GTR+F+I+G4	0.00	0	0	0	123
		4			93.50	115	59	56	
54	M39181	3		TIM3+F+G4	34.86	38	31	7	109
		4			81.65	89	50	39	
55	M6446	3		GTR+F+G4	0.00	0	0	0	101
		4			88.12	89	46	43	
56	M16938	3		TPM3u+F+G4	48.85	64	41	23	131
		4			93.89	123	63	60	
57	M1639	3		GTR+F+I+G4	0.00	0	0	0	341
		4			95.31	325	162	163	
58	M3956	3		TNe+G4	25.57	212	120	92	829
		4			99.40	824	413	411	
59	M3938	3		TIM3e+G4	90.41	735	357	378	813
		4			99.75	811	404	407	
60	M3030	3		TIM3e+G4	74.93	281	142	139	375
		4			100.00	375	186	189	